

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

REDAKČNÍ A PUBLIKAČNÍ SYSTÉM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

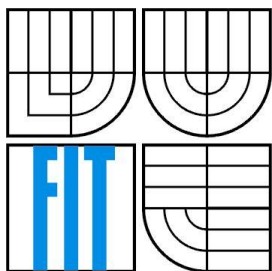
AUTOR PRÁCE
AUTHOR

JAN MIKULÁŠEK

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

REDAKČNÍ A PUBLIKAČNÍ SYSTÉM

EDITORIAL AND PUBLICATION SYSTEM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN MIKULÁŠEK

VEDOUCÍ PRÁCE
SUPERVISOR

ING. ŠÁRKA KVĚTOŇOVÁ

BRNO 2007

Abstrakt

Cílem této práce je vytvořit uživatelsky přívětivý a částečně automatizovaný redakční a publikační systém s ohledem na možnost exportu dat. Systém využívá nejnovější technologie a zároveň se snaží vyhnout nestandardním metodám. Avšak pro zabezpečení jsem se rozhodl nepoužít populární jednoduché hasovací funkce, protože některé jsou již prolomené.

Klíčová slova

Informační systém, databáze, PHP, MySQL, HTTP, SQL, AJAX, export

Abstract

The main aim of this thesis is to develop user friendly and partially automated editorial and publication system. System works on the newest technologies however it avoids not standard methods. However for security reasons I have chosen not to use popular and simple hash functions, because some of them are already cracked.

Keywords

Information system, database, PHP, MySQL, HTTP, SQL, AJAX, export

Citace

Zdenk Letko: Informační systém Studentské unie FIT VUT, bakalářská práce, Brno, FIT VUT v Brně

REDAKČNÍ A PUBLIKAČNÍ SYSTÉM

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Šárky Květoňové. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jan Mikulášek
2007-04-16

Poděkování

Především bych chtěl poděkovat Ing. Šárce Květoňové za její trpělivost a odpovědi na mé všetečné otázky. Dále bych chtěl poděkovat N. Dietzovi za poskytnutí grafického konceptu ve formátu PSD - tedy za poskytnutí podoby informačního systému s e-sportovou tematikou.

© Jan Mikulášek, 2007.

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

Obsah

Úvod	3
1 Cíle práce	4
1.1 Požadavky	4
1.2 Současný stav problematiky	4
2 Technologie	5
2.1 Databáze	5
2.1.1 Modely	6
2.1.2 SQL	7
2.2 HTTP	8
2.2.1 PHP	9
2.2.2 Ostatní hypertextové preprocesory	9
2.2.3 HTML a CSS	9
2.3 AJAX	10
2.3.1 JavaScript	10
2.3.2 Spojení technologií	11
2.4 Export dat	11
2.4.1 RSS a XML	11
2.4.2 CSV	12
2.4.3 PDF	12
3 Redakční a publikační systém	13
3.1 Analýza	13
3.1.1 Použití modulů	13
3.1.2 Zabezpečení	13
3.1.3 Možnost expanze	14
3.2 Návrh databáze	15
3.2.1 Tvorba databáze	15
3.2.2 Normálové formy	16
3.2.3 Příklad použití	16
3.2.4 ER diagramy	17
4 Implementace	20
4.1 Inspirace	21
4.2 Interface	21
4.3 Kde a jak systém funguje	23
4.3.1 Galerie	23

4.3.2	Editor WYSIWYG.....	24
4.3.3	Grafy.....	25
4.3.4	Další moduly.....	25
4.3.5	Testování.....	25
4.4	Slabiny.....	25
4.5	System interpretu.....	26
4.6	Možná rozšíření.....	27
4.6.1	Kalendář.....	27
4.6.2	Automatický export a import XML.....	27
5	Závěr.....	28
	Literatura.....	29

Úvod

Informační systémy (dále jen IS) jsou součástí každodenního života. Kolikrát si dnešní člověk ani neuvědomí, že nějaký systém v danou chvíli zrovna používá. Rozvoj IS se stále rozrůstá - zvedá se jak jejich počet, tak i jejich velikost a složitost. Různé IS mezi sebou komunikují, vyměňují si data pomocí různých a komplikovaných technologií, nebo jsou naopak primitivní a samostatné.

IS jsou zde nejen proto, aby usnadnily lidem práci, ale také proto, aby přiblížily svět Informačních technologií (dále jen IT) běžnému smrtelníkovi, který má před sebou monitor a klávesnici poprvé v životě. Těžko by asi šlo vysvětlovat kominíkovi, jak funguje databáze a SMTP protokol. Avšak naučit ho odesílat email za pomoci grafického rozhraní ve webovém prohlížeči je podstatně snazší. Ostatně i takovýto proces odeslání emailu v sobě zahrnuje hned několik IS, včetně těch automatizovaných a stále funkčních, a také pravděpodobně již několik zapomenutých a zaprášených programů. To vše je ve stínu pojmu Internet, který dodal vytváření IS největší boom.

IS mají většinou i nějaký účel, ostatně jako každý program. Fungují jako distribuce firemních šablon na Intranetu, jako přihlašování a registrování předmětů ve škole či jako rozesílání reklamních emailů zákazníkům, kteří si o ně požádali. Já mám za úkol vytvořit redakční a publikační systém, který bude bezpečný a bude ho moci používat jak IT odborník, tak i člověk, který neumí programovat. Musí být jednoduchý, přehledný a měl by z větší části vyhovovat vyhlášce Ministerstva Informatiky o přístupnosti webu. Zároveň však musí být uživatelsky přívětivý, dokázat třídit data a hlavně využívat již zažitých metod - například autentizace uživatelů.

Rozhodl jsem se využít i několika externích nástrojů. Vzhledem k jejich historii a referencím se jedná o kvalitní freewareové nástroje, které k použití v redakčním systému přímo vybízejí. Také jsem si dal za úkol vytvořit intuitivní prostředí, které nebude mást nutnou reklamou a na první pohled bude pochopitelné, kde se požadované informace nachází. Tento systém by měl být také snadno rozšiřitelný a modifikovatelný. Pokud potřebujeme vyměnit databázový server, jednoduše zároveň vyměníme také modul pro práci s databází.

Některé části této práce již několik let fungují v několika IS, některé jsou inovované. Za některé její moduly vděčím jejich autorům.

1 Cíle práce

1.1 Požadavky

Nejdříve je nutné zjistit, jak takový IS vypadá. Na internetu jich je poměrně velké množství, a tak nebyl zase tak veliký problém zjistit, jakým směrem se vydávají největší a nejčtenější české informační portály. V zadání nebylo přesně stanoveno, jak má IS vypadat. Dá se však předpokládat, že pod pojmem *Redakční a publikační systém* se neschovává webová prezentace v *Macromedia Flash*. Cílem je vytvořit jednoduchou a hlavně uživatelsky přívětivou webovou aplikaci, kterou bude moci ovládat jak zkušený informatik, tak i člověk, který nemá žádné zkušenosti s HTML či jiným sázením textu. Systém bude muset umět automaticky rozesílat emaily, autorizovat uživatele a být dostatečně bezpečný. Zároveň však nesmí uživatele obtěžovat neustálým zadáváním hesel. Systém také nesmí být náročný na systémové prostředky, avšak v jistých chvílích je nutné si vyžádat delší dobu spuštění skriptu.

1.2 Současný stav problematiky

Důležité je také vytvořit dobrou grafickou podobu webu, s dostatečně vysokým kontrastem písma a pozadí. Dalším důležitým parametrem je také layout stránky. Spousta dnešních webů je velmi nepřehledných, a to včetně profesionálních a komerčních stránek. Tyto IS si přiděly problémy především špatnou mapou webu, špatným přidáváním další funkčnosti a vkládáním dalších modulů na nesprávná místa. Většina z nich musí dnes své stránky vizuálně předělat, což většina z nich také dělá. Naštěstí je zde technologie CSS – kaskádové styly, která jim velice usnadní práci.

Spousta tvůrců si dnes ani nepřipouští spoustu neviditelných chyb a neexistující propojení IS. To opět nahrává do karet spoustě hackerů, kteří tak mohou poměrně pohodlně získávat informace, na které by jinak neměli mít přístup. Této otázce se budeme věnovat v kapitole 3.1.2 Zabezpečení.

Důležité před vlastní implementací je tedy také rozmyslet, zda se bude IS i dále rozvíjet, a případně zkusit odhadnout, jakým směrem rozvoj půjde. V našem případě se jedná o publikační systém, tedy většinou o přenos informace směrem od redakce ke čtenáři. Úkolem by tedy mělo být čtenáři poskytnout informace hned několika možnými způsoby a také doplnit přehledy. Nový a populární je tzv. Informační panel, kde si uživatel bez nutnosti znovu načítat stránku vyhledává zrovna to, co potřebuje. Avšak zcela nejlepší možností je, když je chtěná informace dopravena ke čtenáři automaticky a zároveň když čtenáři nejsou na obtíž a nikde nepřekáží jiné, pro návštěvníka méně důležité, informace.

V další kapitole se podíváme, jaké možnosti nám současná situace ve světě IT nabízí.

2 Technologie

Pro tuto BP jsou nejdůležitější dvě otázky. První otázkou je, kam se všemi daty, která chceme skladovat. Druhou neméně důležitou otázkou je, jak tato data poskytnout. Proto jsou první dvě části této kapitoly věnovány postupně především těmto stěžejním otázkám. Zbytek kapitoly pak pojednává o aktuálním trendu uživatelského prostředí a o exportu dat.

2.1 Databáze

Databáze je definovaná jako strukturální kolekce záznamů. Jedním z jejích účelů je poskytnout, měnit a uložit data. Dalším velmi důležitým prvkem je rychlost odpovědi. Dotazy, kterými se na databázi dotazujeme, mají také různé složitosti. Je proto důležité tabulky v databázi nejen dobře navrhnout, ale zároveň také věnovat i dostatek času implementaci jednotlivých dotazů.

Dozajista je potřeba sdílení několika dat pro několik aplikací současně. Proto se databáze, jako každý jiný softwarový produkt, podrobila inovaci. Jedním s podstatných milníků historie databází je dosažení nezávislosti dat. S tím souvisí základní problémy:

- Redundance – opakují se stejná data
- Nekonzistence – data se v různých místech liší

Tyto problémy s použitím aktuálních databází víceméně končí. Velký vliv na odstranění tohoto problému má technologie DBMS. Ta vzala uživateli databáze přímý přístup k datům, to zajišťuje bezpečnost uložených dat.

Představte si, že jste si otevřeli textový dokument a během půl hodiny jste v něm změnili několik řádků a nyní jej chcete uložit. Váš kolega ale potřeboval změnit pouze jeden a své změny již uložil. Ty se ale ve vašem souboru nezměnily a vy byste jeho změny přepsali. Proto, pokud vytváříme složitý nebo zabezpečený systém - jako je například internetové bankovníctví, bude také potřeba vytvořit *transakční databázi* a podívat se na *lock*.

Data, o která se zajímáme, bývají obvykle *persistentní*. Jinak řečeno svou dobou života dalece přežívají běh scriptu. Dalším pojmem je databázová integrita. Jedná se o takový stav, při němž záznamy v celé databázi vyhovují soustavě určitých definovaných pravidel. Tato pravidla obvykle odpovídají vybraným pravidlům z té části světa, pro kterou databáze slouží.

2.1.1 Modely

Je také několik typů neboli modelů databáze. Data totiž mohou být v databázi uložena hned několika způsoby a není proto od věci si zvolit ten model, který našemu problému nabídne nejlepší řešení. Takový proces se nazývá *modelování dat*. Mezi nejznámější modely patří:

- Hierarchický
- Síťový
- Relační
- Objektově orientovaný
- Objektově relační

2.1.1.1 Síťový

Tento model vychází ze zdokonalení hierarchické koncepce, na kterou přímo navazuje. Koncepce se používala především v 80. letech minulého století, a to v komerčních aplikacích. Tato databáze, resp. její struktura se modeluje pomocí tzv. *Bachmanových diagramů*.

2.1.1.2 Relační







Relační databáze je takovým systémem uspořádání, který je založen na relačním modelu a relační algebře. Data v takovéto databázi jsou seskupena do jednotlivých tabulek, nad kterými jsou prováděny jednotlivé operace. Z názvu je poměrně jasné, že jednotlivé tabulky mezi sebou mohou navazovat vztahy. S relační databází souvisí systém RDMBS.

Celé kouzlo relačního modelu je v tom, že jej lze poměrně snadno aplikovat na realitu. Každá tabulka má jisté řádky a sloupce. Pod řádky si lze představit knížky ve vaší knihovně. Pod sloupci typ knihy, jejího autora nebo například váhu. Jednotlivé typy lze nahradit klíčem, aby jsme odstranili redundanci (Obr. 1). Například tak můžeme mít tabulku autorů a pomocí klíče zjistit, který autor napsal vybranou knihu.

			id	jmeno_knihy	id_autor	cena
<input type="checkbox"/>			1	Pán Prstenů: Dvě Věže	1	300
<input type="checkbox"/>			2	Windows C#	2	1000
<input type="checkbox"/>			3	Hobit	1	245

Obr. 1 – cizí klíč

Ne vždy nám však tento typ relace stačí. Někdy potřebujete rozdělit zákazníky do několika skupin, ale každý zákazník může být ve více skupinách. Máte tedy tabulku pro zákazníky a tabulku pro skupiny. Bude proto nutné ještě vytvořit pomocnou spojovací tabulku (Obr. 2) - tzv. m_n tabulku, ve které obě předchozí propojíte.

←T→			autor	skupina
<input type="checkbox"/>			Tolkien	Pohadky
<input type="checkbox"/>			Tolkien	Naucne
<input type="checkbox"/>			SAMS	Naucne

Obr. 2 – propojovací n m tabulka

Tento model databáze je v aktuální chvíli nejpopulárnější. Mezi nejoblíbenější relační databáze patří *MySQL*, která poskytuje svou licenci pro nekomerční použití zdarma. Další oblíbenou alternativou je databázový systém *Interbase* nebo *Firebird*, který je zcela zdarma.

Tato bakalářská práce je navržena také na tomto modelu databáze. Může se ovšem stát, že tento model bude časem vytlačen objektovým modelem.

2.1.1.3 Objektový

Objektový model se nechal inspirovat v objektově orientovaném programování (OOP). Namísto tabulek jsou v objektových databázích použity objekty. Výhodou tohoto modelu je možnost přímo programovat na databázi. Některé objektové databáze jsou přímo navrženy pro spolupráci s masově používanými jazyky, jako například *Java*, *C#*, *Visual Basic .NET*, *Smalltalk* či *C++*. Právě tento složitý proces zatím omezuje jejich masové používání. Avšak jejich potenciál se skrývá za prvé v síle zpracování, které potřebují zároveň výkon a práci s velmi složitými daty, a za druhé - programátor přímo ovlivňuje funkčnost. Do budoucna se proto čeká značný rozmach tohoto modelu. Nejlepší vyhlídky jsou tedy připisovány právě objektově relačnímu modelu.

2.1.2 SQL

Pro komunikaci s relační databází byl vytvořen jazyk SQL (*Structured Query Language*). SQL se obecně používá pro vytváření (CREATE), vybírání (SELECT), vkládání (INSERT), inovaci (UPDATE), smazání (DELETE) a několik dalších příkazů – jako operace na databázi. SQL je také standardizováno jak v ANSI tak v ISO.

Bohužel jednotlivé databázové systémy používají jednotlivé jazyky. Naštěstí se tyto jazyky velmi podobají, a proto je lze snadno modifikovat pro správnou funkčnost.

```
CREATE TABLE `knihy` (  
  `id` INT( 6 ) UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY ,  
  `jmeno_knihy` VARCHAR( 50 ) CHARACTER SET utf8 COLLATE utf8_unicode_ci  
  NOT NULL ,  
  `id_autor` INT( 6 ) UNSIGNED NOT NULL ,  
  `cena` DOUBLE UNSIGNED NOT NULL ,  
  INDEX ( `id_autor` )  
  ) ENGINE = MYISAM ;
```

Obr. 3 – Jazyk SQL, vytváření tabulky

2.2 HTTP

HTTP je protokol určený původně pro výměnu hypertextových dokumentů ve formátu HTML. Používá obvykle port TCP/80, verze 1.1 protokolu je definována v [RFC 2616](#). Tento protokol je spolu s elektronickou poštou tím nejvíce používaným. Zasloužil se také o obrovský rozmach internetu v posledních letech. Pomocí rozšíření MIME (*Multipurpose Internet Mail Extensions*) umí přenášet jakýkoli soubor, používá se společně s formátem XML a *JavaScriptem* pro webové služby (spouštění vzdálených aplikací) a pomocí aplikačních bran zpřístupňuje i další protokoly, jako jsou např. FTP nebo SMTP.

Protokol funguje způsobem dotaz - odpověď. Klient se táže serveru vždy formou čistého textu, server poté odpoví pomocí několika řádků čistého textu popisujících výsledek dotazu (zda se dokument podařilo najít, jakého typu dokument je atd.), za kterými následují data samotného požadovaného dokumentu. První odpověď je považována za hlavičku.

Dotazovací část protokolu obsahuje tyto metody: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, CONNECT.

```
POST /item HTTP/1.1  
Host: 189.123.345.239  
Content-Type: text/plain  
Content-Length: 200
```

Obr. 4 - klient

```
200 OK  
Content-Type: text/plain  
Content-Length: 200
```

Obr. 5 – odpověď serveru

```
400 Bad Request  
Content-Length: 0
```

Obr. 6 – server nerozuměl

Příklad komunikace – Obr. 4 je okno klienta, Obr. 5 je server v případě, že zprávě rozuměl. Naopak pokud server nemohl dekodovat zprávu, výsledek bude podobný Obr. 6.

2.2.1 PHP

Zkratka z angl. *Personal Home Page*, nyní zamaskovaná titulkem *PHP: Hypertextový preprocesor*, to je nejpobulárnější multiplatformní skriptovací jazyk nejen pro generování HTML stránek. Jeho základy spadají k jazyku *Perl* a *C*, od kterých si vzal i svou jednoduchost a syntax. Myšlenkou jeho tvůrců bylo co nejvíce přiblížit programování dynamického obsahu veřejnosti, zároveň se ale neomezovat rychlost exekuce skriptu. Postupem času přišla nevyhnutelná objektová podpora. Tento interpret je také populární pro jeho velmi výstižné chybové hlášení a pro velmi rozsáhlou, lidsky psanou dokumentaci. V neposlední řadě samozřejmě také pro jeho možnosti rozšíření. Ve většině případů PHP běží jako modul multiplatformního softwareového serveru *Apache 2*.

2.2.2 Ostatní hypertextové preprocesory

Jako konkurence PHP se přímo nabízí programovací jazyk firmy *Microsoft* – ASP (*Active Server Pages*). Tento jazyk má svou velkou výhodu v začlenění do rodiny .NET a také v e svých modulech. Bohužel je komerční, a tak se netěší takové oblibě jako PHP.

Jako dalšího člena rodiny můžeme označit JSP (*Java Server Pages*), které teprve začínají svůj život. Jejich velkou výhodou je poměrně nízká náročnost na server, díky níž je lze použít pro velmi vytížené IS.

Otázkou je, jak je to s podporou právě velmi navštěvovaných webů. Tzv. *Round robin* u služby DNS totiž není vždy tím pravým řešením a multiprocessorové stroje na databázi také vždy nestačí. Příkladem může být špička registrace předmětů na naší fakultě. Naštěstí se již daří servery ochránit alespoň před pádem, avšak otázka DDoS (*Denial of Service*) je stále na stole.

2.2.3 HTML a CSS

HTML je zkratkou pro *HyperText Markup Language* a většina lidí si ho spojuje s formátováním textu. To však není výstižné. HTML toho dokáže mnohem více. Nejedná se zde pouze o konkrétní jazyk, který má několik verzí, ale především a hlavně o prohlížeč. HTML by byl bez jeho dynamického generování pouhou prezentací. Ostatně v polovině devadesátých let minulého století tomu tak bylo.

Právě prohlížeče v dnešní době obecně rozhodují o WWW. Přiznejme si, že dnes již *textový prohlížeč* použije pouze IT odborník v největší nouzi. Masový rozmach HTML se dá připsat společnosti *Microsoft*, která do standardní instalace OS *Windows* přidala aplikaci *Internet Explorer*. Jako konkurenty IE lze jmenovat prohlížeč *FireFox* a *Opera*.

Vývoj prohlížečů se dnes soustředí na správné zobrazování kaskádových stylů. Zbylé problémy, které přinášelo kódování zdrojových kódů, interpretace serverem a lokálním kódováním OS, na kterém běžel prohlížeč, jsou ty tam. CSS obecně pomáhá ke správnému zobrazování stránek a většina webmasterů si začíná uvědomovat, že pohodlné psaní těžko zvítězí nad standardy. Ostatně se o to snaží i společnosti vyvíjející prohlížeče, aby zdrojový text vypadal tak, jak má, nikoliv tak, jak by bylo dobré, aby vypadal.

Fakulta informacních technologií

- [Studium](#)
- [Věda a výzkum](#)
- [Události](#)
- [Aktuality](#)
- [Úřední deska](#)

- [Fakulta](#)
- [Součásti fakulty](#)
- [Osoby](#)
- [Centrum VT](#)
- [Knihovna](#)

Božetěchova 2
602 00 Brno

Tel.: +420 54114-1145
GSM: +420 72681-1145, 60407-1145
Fax: +420 54114-1270
E-mail: info@fit.vutbr.cz



Obr. 7 – HTML bez aplikace CSS

Obr. 8 – stejný HTML kód po aplikaci CSS

Na Obr. 7 je vidět HTML kód bez aplikace kaskádových stylů. Obr. 8 ukazuje stejný HTML kód, na který byl aplikován kaskádový styl.

2.3 AJAX

Jedná se o staronovou technologii založenou na jazyku Java. Termín *AJAX* se poprvé veřejně objevil v dubnu 2005. Myšlenky, na kterých je AJAX založen, jsou však výrazně starší. V roce 1998 představil *Microsoft* novou technologii nazvanou *Remote Scripting*, ve které v klientském prohlížeči běžel *Java applet* komunikující se serverem, přičemž tento applet poskytoval služby *JavaScriptovým funkcím*. Avšak velká popularita a rozšíření AJAXu začala několika službami společnosti *Google* (nejdříve *Gmail*, posléze *Google Maps* a další).

2.3.1 JavaScript

Javascript je multiplatformní, objektově orientovaný skriptovací jazyk, který běží na straně klienta – nezatěžuje tedy server. Nyní se zpravidla používá jako interpretovaný programovací jazyk pro WWW stránky, vkládaný přímo do HTML kódu stránky. Jsou jím obvykle ovládány různé interaktivní prvky GUI (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků.

JavaScript může uživateli velmi usnadnit práci. Před odesláním formuláře tak může například pomocí regulárního výrazu chtěná pole zkontrolovat a pokud něco nesedí, může upozornit uživatele na chybu a formulář neodeslat. Musí se však brát ohled i na to, že ne vždy musí mít uživatel *JavaScript* povolen a ne vždy umí prohlížeč s *JavaScriptem* vůbec pracovat. Proto je nutné stejnou ochranu aplikovat i na straně serveru.

JavaScript je možné použít i na straně serveru. První implementací *JavaScriptu* na straně serveru byl *LiveWire* firmy *Netscape* vypuštěný roku 1996.

2.3.2 Spojení technologií

AJAX je spojením *JavaScriptu* a XML. Je určen pro pohodlnost a nepochybně také pro ulehčení práce serveru. Již od návrhu HTML bylo vždy nutné načíst znovu celou stránku. To negativně působí na vytížení serveru, který se dokola a dokola musí zaobírat stejnými požadavky. Jelikož není potřeba při každém požadavku znovu a znovu sestavit celý HTML dokument, ale pouze provedené změny, je množství vyměňovaných dat výrazně nižší a teoreticky to může mít příznivý vliv i na zátěž databázových serverů či dalších backendových systémů. AJAX však naopak může zvýšit počet vyměňovaných HTTP požadavků. Třebaže přenášejí nižší množství dat, při nevhodné implementaci zátěž neklesne. Dříve se používal HTML element – *IFRAME*, který dnes díky AJAXu a CSS již ztrácí smysl. A co víc, jeho použití neodpovídá SEO (Search Engine Optimization). A to tedy rozhodně není náš záměr.

XML zde figuruje jako šablona, do které se mohou zapisovat data. Pro příklad RSS čtečka na vašich webových stránkách není nadlidský úkol, nýbrž program o velikosti necelých 20 řádků.

2.4 Export dat

Export dat je poměrně široký pojem. Musíme vědět, co, jak a kam exportovat. Protože IS má usnadnit přístup k datům, není naším cílem exportovat ve formátu, který není pro člověka čitelný.

2.4.1 RSS a XML

RSS vychází s formátu XML. Od svého vzniku byla již několikrát inovována. RSS slouží k tzv. syndikaci obsahu webu – tedy předávání obsahu. Původně tento formát sloužil pouze k předávání aktuálních novinek mezi jednotlivými servery, které takto velmi jednoduše mohly odkazovat na aktuální články na jiných serverech

Pod zkratkou RSS se schovává *Really Simple Syndication* nebo také *Rich Site Summary*. Nejnovější verze RSS má číslo 3.0, ale nejmasivněji používaná je stále verze 2.x, která je spravována na <http://blogs.law.harvard.edu/tech/rss>.

Software určený k práci s RSS se označuje jako *RSS čtečka*. Může se jednat o samostatný specializovaný program, o plugin do jiného programu nebo o AJAX script.

2.4.2 CSV

Comma separated values – tedy hodnoty oddělené čárkami. To je jednoduchý formát pro výměnu tabulkových dat. Soubor ve formátu CSV se skládá z řádků, ve kterých jsou jednotlivé položky odděleny znakem čárka (.). Hodnoty položek mohou být uzavřeny do uvozovek ("), což umožňuje, aby text položky obsahoval čárku. Pokud text položky obsahuje uvozovky, jsou uvozovky zdvojeny.

Tyto oddělovače se samozřejmě můžou lišit, podobně jako oddělovací řádky. Pro tento formát neexistuje specifikace, avšak jeho funkčnost je popsána v [RFC 4180](#).

2.4.3 PDF

Portable Document Format je souborový formát pro ukládání dokumentů nezávisle na softwaru i hardwaru, na kterém byly pořízeny. Tento formát zajišťuje, že se libovolný dokument na všech zařízeních zobrazí stejně. Nejznámějším prohlížečem je domovský a oficiální prohlížeč *Adobe Acrobat Reader*.

PDF vychází z *PostScriptu*, takže jejich schopnosti jsou do značné míry podobné. Nejviditelnější rozdíl je ve velikosti souborů. Jelikož soubory PDF automaticky používají kompresi, jsou typicky výrazně menší než odpovídající dokumenty ve formátu *PostScript*.

Specifikace toho formátu je dostupná v knižní podobě [ISBN 0201758393](#).

V následující kapitole se podíváme na aplikaci těchto technologií do praxe. Zaměřím se především na analýzu problému a jeho následné řešení. Již se nebudeme stále pohybovat v teoretické části, ale ukážeme si i přímé příklady řešení jednotlivých problémů, na které narazíme.

3 Redakční a publikační systém

Dobrym plánováním můžeme předejít spoustě problémů, které by pak mohly zapříčinit rozsáhlé předělávání celého projektu. Je tedy důležité si nejdříve systém na základě požadavků dobře navrhnout. K tomu nám může pomoci hned několik nástrojů. Mezi nejznámější pomocníky patří jazyk UML (*Unified Modeling Language*). Proto se v této kapitole budeme věnovat obecnému návrhu a dotkneme se i otázky bezpečnosti a expanze.

3.1 Analýza

Systém je potřeba navrhnout tak, aby se jeho části daly případně vyměnit či aby se daly snadno spravovat. Protože je PHP objektově orientovaný jazyk, je snadné tento problém řešit. Pro každý specifický problém, jako je například připojení k databázi, je vytvořen modul, který obsahuje třídu.

3.1.1 Použití modulů

Dále bude nutné vytvořit konfigurační soubor, kde se budou soustředit jednotlivá nastavení, která budou výrazně ovlivňovat chod systému. Mezi nejdůležitější nastavení patří konfigurace připojení k databázi, emailová adresa robota, který rozesílá emaily, titulek systému atp.

Dalšími moduly jsou aplikace třetích stran. Jsou obvykle vyvíjené již dlouhou dobu a také odzkoušené. Protože se jedná převážně o moduly, které se týkají interface, tato práce o nich pojednává převážně v kapitole 4.3 Kde a jak systém funguje.

3.1.2 Zabezpečení

Protože nechceme, aby v našem systému panovala anarchie, je nutné IS jistou formou zabezpečit. Je mnoho cest, jak takový IS zabezpečit. Některé jsou více paranoidní, některé méně. Důležitým faktorem samotné bezpečnosti je neobtěžovat uživatele všetečným dotazováním se na hesla. Důležité je také promyslet, kde systém může mít slabiny, a zkusit tyto slabiny minimalizovat.

Pokud půjdeme do důsledku, tak je také nutné popřemýšlet, zda nevyužít nestandardních metod. Jejich použití totiž může případnému útočníkovi dost znepříjemnit práci. Například je poměrně výhodné nepoužívat MD5 hash ([RFC 1321](#)), protože je již nějakou dobu prolomený. Ze stejného důvodu bych se vyhnul i SHA-1 ([RFC 3174](#), [RFC 4634](#)) [9]. Pokud se totiž útočník dostane do databáze, která obsahuje seznam uživatelů s hashy hesel, je pro něj poměrně snadné taková hesla prolomit v několika minutách. Na internetu navíc již existují databáze, které obsahují jednotlivé hashe – např. <http://passcracking.ru/> [12]. Překlad hashe na heslo pak trvá opravdu jen okamžik.

Získání databáze uživatelů je pak velmi nepříjemnou záležitostí, protože hesla si mohou odpovídat i mezi jednotlivými IS, které nevlastníte. Útočník pak může zkoušet přeskakovat mezi

jednotlivými IS a snadno získávat informace, které mu nenáleží. Problém lze vyřešit poměrně snadno, a to kombinací jednotlivých metod (Obr. 9).

Tato BP si ukládá tzv. autologin do cookies prohlížeče, tedy pokud je má uživatel povolené. Systém si do session ukládá data nezbytná pro kontrolu přihlášení a práv uživatele. Standardně je doba života cookie 30 minut. To by mohlo komplikovat život při psaní dlouhých článků. K tomu nám dopomohla cookie, díky které se uživatel před odesláním nového článku ještě stihne přihlásit. Pro zvýšení bezpečnosti je v cookie společně s heslem zahashovaná i IP adresa, která je následně kontrolována při přihlášení se záznamem v databázi. Pokud si uživatel přeje cookie smazat, stačí, když se odhlásí. Cookie je pak automaticky odstraněna – nahrazena novou cookie s prázdnými daty a expirací. To by mělo preventovat případné *undelete* metody.

Samozřejmě, pokud uživatel nepracuje doma, není zrovna bezpečné používat tento systém zabezpečení. Myslím si, že vše záleží na zvážení administrátora IS, který je schopen web spustit na šifrovaném spojení a ověřování cookie vypnout.

```
function crypto($source) {
    $salt[0] = "CrYpToFuNcYeYXy1"; // Limit this to 16 characters
    $salt[1] = "RtPc1FoR"; // Limit this to 8 characters
    $crypt[0] = crc32($source);
    $crypt[1] = crypt($source, $salt[0]);
    $crypt[2] = md5($source);
    $crypt = implode($salt[1], $crypt);
    return sha1($source.$crypt);
}
```

Obr. 9 – Vylepšená hashovací funkce

3.1.3 Možnost expanze

Je poměrně dobré systém připravovat na rozšíření už v jeho počátku. Je poměrně složité si představovat, jak lze systém rozšiřovat, především, když nemáme představu, jak takové rozšíření může vypadat. Avšak už při návrhu si můžeme usnadnit dost práce s případnou expanzí.

Obvykle se jedná o částečný nebo úplný jazykový mirror nebo o prosté rozšíření funkčnosti webu, přidání nebo update *RSS čtečky*, automatický export dat ve formátu XML pro zákazníka, který si od nás předobjednal data na půl roku atp.

Takovým typickým problémem je kódování. Pokud svůj systém navrhnete v kódování *latin2* a získáváte data z XML, které je kódované v unicode, musíte počítat s jistou ztrátou znaků, které nejsou ve středoevropských jazycích. Dalším problémem může být import dat z CSV, kdy je kódování obvykle (ale ne vždy) *Windowsovské CP1250*.

Za zvážení stojí také migrace na jiný operační systém. Při použití ASP by byl přechod na linux poměrně problematický. Naopak použití PHP jako modulu do *Windowského ISS* tolik práce nedá. Je dobré si uvědomit, že databázový server nemusí být vždy dostupný, a proto můžeme určit, jak se systém má chovat, zda zkusit záložní databázový server či vypsát chybovou hlášku.

S tím souvisí i další doplňky a pluginy, kterými budeme náš systém rozšiřovat. Pokud víme, že nějaká platforma podporuje více rozšíření, je vhodné ji zvolit. Pokud je to ale jen trochu možné, je lepší se takovýmto rozšířením raději vyhnout. Například tvorba dokumentů PDF v PHP lze udělat i bez použití modulů třetích stran.

Za zvážení stojí také design systému a očekávání klientských platforem. Dnes je poměrně snadné web pomocí CSS upravit pro zařízení s malým rozlišením - jako například mobilní telefon nebo PDA. Čtení informací je potom rozhodně pohodlnější.

3.2 Návrh databáze

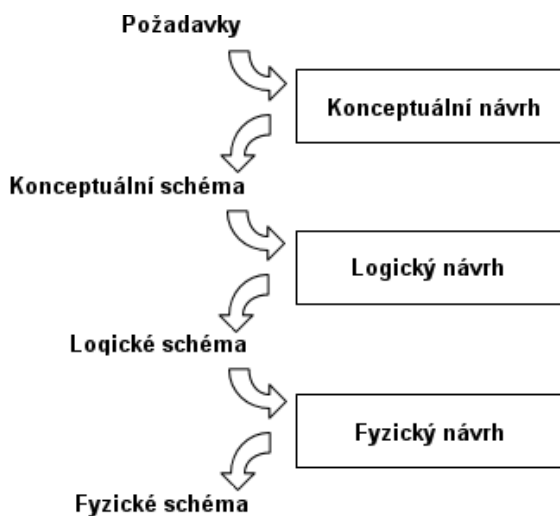
Při výběru databáze je nutné zjistit, jaký model databáze budeme potřebovat. Dále bude nutné zjistit, zda bude potřeba transakcí. Nezbytně je také nutná znalost předpokládané náročnosti na server. V případě velikých nároků je dobré zkusit zvolit jiný model. [8]

K návrhu nám pomůže ER diagram, na kterém můžeme vyčíst vztahy jednotlivých entit. K sestavení takového modelu je dobré použít i model použití (z angl. *Use case*).

3.2.1 Tvorba databáze

Dříve než začneme s pracovat s databází, je vhodné si vybrat dobré pracovní prostředí. Většina lidí zastává názor, že by se problémy neměli řešit přes konzoli. Velmi dobrým pomocníkem je webový interface – *PHPMysqlAdmin*. Je velice populární díky svému intuitivnímu vzhledu a nenáročnosti. Jeho přední vlastností je snadné nastavení. A bude fungovat vždy, protože bude pravděpodobně běžet na serveru, který má povolenou komunikaci nebo je sám databázovým serverem. Pokud však máme povolen přístup na databázi, lepším pomocníkem je komerční *EMS MySQL Manager*.

Dále je vhodné databázi normalizovat. Nic se však nesmí přehánět, a tak musíme brát ohled na to, jaká data budeme potřebovat a seskupovat. Normalizace má několik forem [7] [16].



Obr. 10 - Návrh databáze

3.2.2 Normálové formy

- 1.NF – První normální forma
- 2.NF – Druhá normální forma
- 3.NF – Třetí normální forma
- BCNF – Boyce Coddova normální forma
- 4.NF – Čtvrtá normální forma
- 5.NF – Pátá normální forma

1.NF – Relace (tabulka) je v první normální formě, pokud každý její atribut (sloupec) obsahuje jen atomické hodnoty. Tedy hodnoty z pohledu databáze již dále nedělitelné. Tato podmínka není splněna například u tabulky, kde je jméno a příjmení v jednom sloupci a aplikace přitom pracuje s těmito položkami jako samostatnými.

2.NF – Relace se nachází v druhé normální formě, pokud splňuje podmínky první normální formy a každý neklíčový atribut je plně závislý na primárním klíči, a to na celém klíči a nejen na nějaké jeho podmnožině.

3.NF – V této formě se nachází tabulka, opět splňuje-li předchází formy a všechny její neklíčové atributy, jsou vzájemně nezávislé.

3.2.3 Případ použití

Cílem psaní případů užití je nalezení hranic systému. Jsou psány z pohledu zákazníka a podávají první představu o rozsahu projektu. Nezabýváme se technologickými problémy a snažíme se navrhnout funkční podobu systému co nejsrozumitelnější pro zákazníka. Zjišťujeme, které procesy má systém podporovat a jací uživatelé ho budou používat.

V této BP v podstatě vystupují tři uživatelé:

- Neregistrovaný návštěvník
- Registrovaný uživatel
- Redaktor

Každý má svá jistá oprávnění. Neregistrovaný návštěvník může číst, ale již nesmí přispívat v diskuzích. Má přístup do všech veřejných sekcí informačního systému, ale není oprávněn se aktivně podílet na obsahu.

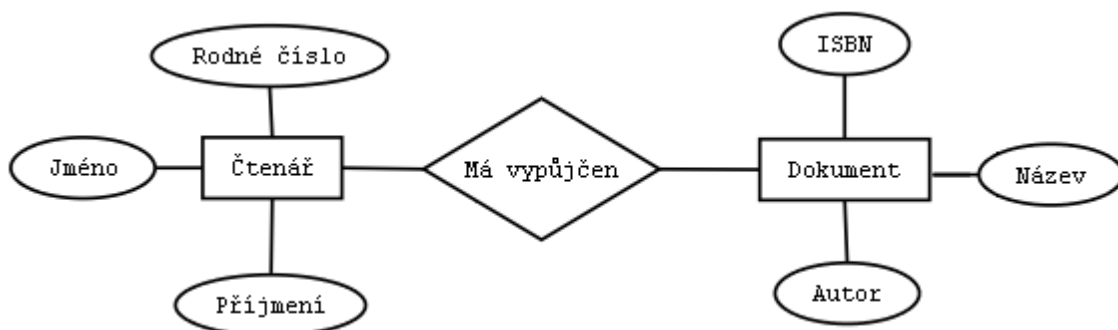
Registrovaným uživatelem je ten, kdo řádně prošel registrací. To zahrnuje úspěšné vyplnění registračního formuláře. Dále je nutné, aby informační systém ověřil emailovou adresu, kterou uživatel zadal. To se děje pomocí autorizačního - neboli aktivačního emailu, který systém automaticky odesílá. Registrace je dokončena, jakmile uživatel navštíví aktivační URL. V případě

nouze či jakýchkoliv nesnázi je možné si nechat znovu zaslat aktivační mail. Registrovaný uživatel se může aktivně podílet na obsahu webu.

Rozdíl mezi registrovaným uživatelem a redaktorem je pouze v přidělených právech – přístupu do jednotlivých sekcí a odvětví IS. Administrátor je tedy v podstatě redaktor s oprávněním přidělovat práva.

3.2.4 ER diagramy

V tomto diagramu jsou znázorněny jednotlivé entity, jejich vlastnosti a vazby mezi těmito entitami. Pro realizaci databázové struktury se ER diagram transformuje na tabulky. Každý vztah má svou kardinalitu a je typu 1:1, 1:N, M:N.



Obr. 11 –Příklad ER diagramu v Chenově notaci

ER Diagram nám může pomoci pochopit a vyladit program a odstranit tak jisté chyby. Takové chyby, pokud si je pouze představujeme, či slovně sdělujeme, nemusí být odhaleny.

Následuje popis ER Diagramu, který se týká této BP.

Uživatel (ash_member) – jak je patrné z názvu, tato tabulka slouží k ukládání informací o jednotlivých uživateli. Jedná se o nejpodstatnější tabulku, na které závisí většina obsahu webu. Její nejdůležitější sloupec je uživatelské jméno. Neméně důležité je i heslo, které je ve formě hashe. Další podstatnou informací jsou uživatelská práva. Zbytek v podstatě popisuje uživatele a obsahuje údaje, které uživatel zadal při registraci.

Novinky (ash_news, ash_forum_threads) – tabulka obsahující jednotlivé novinky. Dle sloupce *id_sections* se rozděluje do jednotlivých sekcí, u diskusních fór se pak rozděluje dle *id_forums*. Dle příznaku *status IS* u novinky určuje, zda je viditelná, nebo zda se na ní stále pracuje. Novinka samozřejmě pojednává i o autorovi (*id_members*) a o datu publikace.

Komentáře (ash_news_comments, ash_forums_comments) – zde se hromadí komentáře napsané ke všem novinkám. Každý komentář má svého autora a text - podobně jako samotná novinka. Tato tabulka je téměř identická s tabulkou pro ukládání komentářů v diskusních fórech.

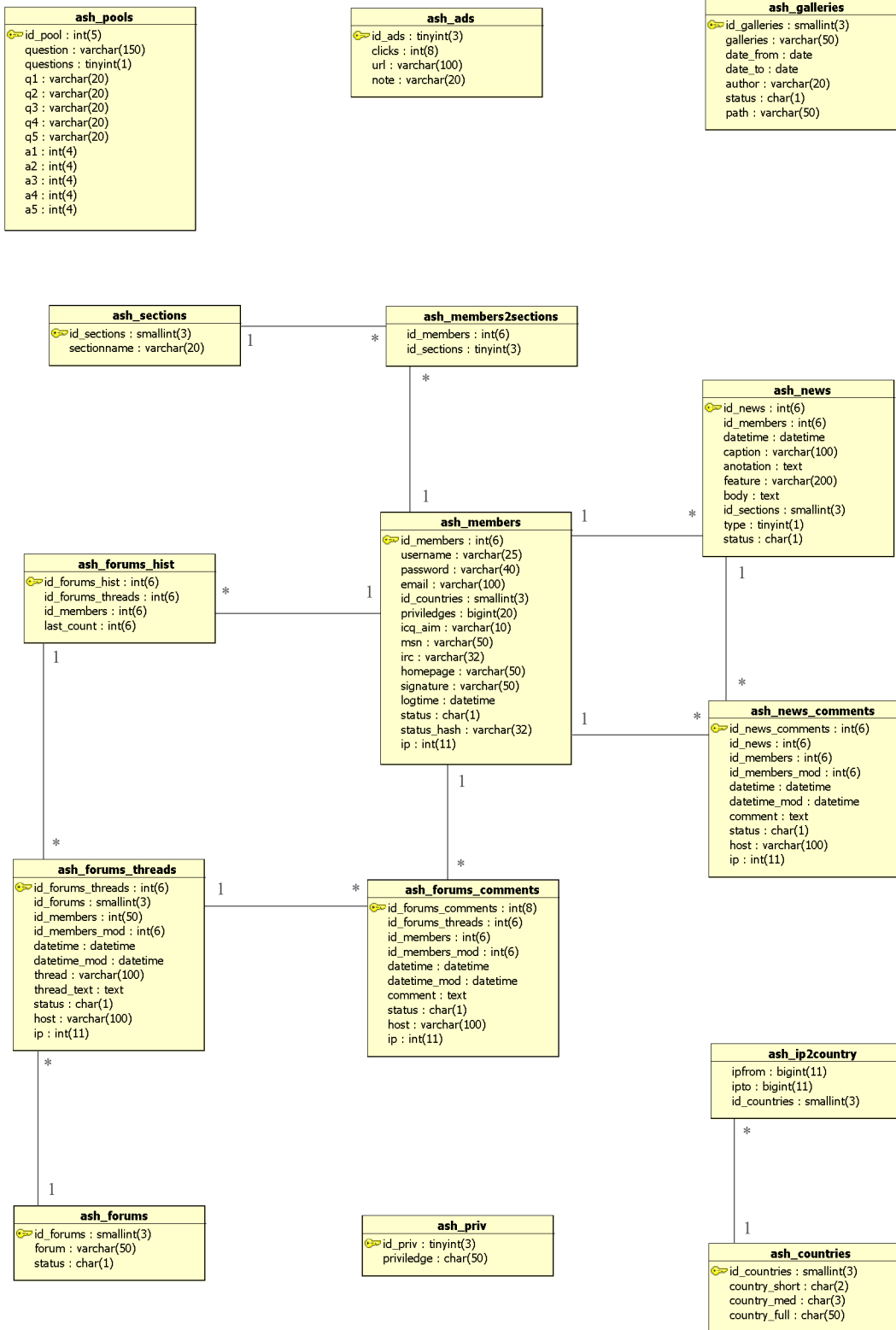
Diskusní fóra (ash_forums) – obsahuje seznam diskusních fór. *Status* v tomto případě obsahuje nejen informaci o zobrazování, ale také informaci, zda je daná diskuze veřejná, či nikoliv.

IP Adresy (ash_ip2country) – tato tabulka obsahuje rozsahy IP adres z projektu GeoIP. Systém pak může snadno získávat informaci, ze které země návštěvník pochází. Země jsou uloženy v tabulce ash_countries. Tabulky jsou provázány sloupcem *id_countries*.

Sekce (ash_sections) – tabulka, která rozděluje celý novinkový systém. Uživatel si tak bude moci vybrat, zda chce zobrazit novinky ze všech sekcí, nebo jen z jedné konkrétní sekce. Na sekce by se mohly například vázat i diskuze. Sekce obsahuje také seznam jednotlivých redaktorů. A redaktor také může být ve více sekcích.

Další tabulky se již nikam nevztahují, proto je nebudu uvádět v textovém souhrnu.

V další kapitole se budu věnovat převážně samotné implementaci. Rozebereme si pohled na grafickou stránku, a dotkneme se i otázky bezpečnosti.



Obr. 12 – ER Diagram

4 Implementace

V této kapitole je popsána implementace IS. Jedná se poměrně o netradiční a nekonvenční systém, který má svoje plus i mínus. Je zde také vysvětleno, z jakého důvodu jsem se rozhodl použít právě konkrétní způsob a proč jsem nezvolil ten standardní.

Systém byl navržen pro jednoduchou správu. To znamená, že i přes jisté programátorské obstrukce existuje vysvětlení, proč je použita dané konkrétní metoda. Podíváte-li se například na systém oprávnění, zjistíte, že každý uživatel má v entitu *priviledges* (bigint 20). To nám dává možnost použít až 64 práv, která se mohou dynamicky rozrůstat. Pokud administrátor potřebuje přidat oprávnění, postačí, když přidá dané právo na konec pole v souboru *config.inc.php*.

```
$site_privs = array(
    "admin" => 0,
    "write news" => 1,
    "enable news" => 2,
    "nuke comments" => 3,
    "edit members" => 4,
    "upload news images" => 5,
    "browse news images" => 6,
    "set priviledges" => 7,
    "export csv" => 8
);
```

Obr. 13 – Seznám práv

```
privilege("set priviledges", true);
..
$priviledges = strrev(decbin($fetcharray["priviledges"]));
foreach ($site_privs as $privsarray => $privsarg)
{
    if ( $priviledges[$privsarg] == 1 )
    ..
```

Obr. 14 – Skript pro nastavení oprávnění

Na základě dotazu oprávnění s textovou hodnotou a případným redirectem je poměrně snadné implementovat systém práv. Restrikce omezení do souborů je pak snadnou záležitostí. Jak je vidět na obrázku 13, tento script bude spuštěn pouze lidem s oprávněním *set priviledges*. Ostatní budou díky příznaku *true* přeměrováni na stránku, která je uvedena v konfiguračním souboru.

Dalším specifickým prvkem je autogenerování. Dejme tomu, že redaktor má z nějaké události seznam fotek, které jsou svými rozměry příliš velké pro umístění na stránky. Zároveň je fotek příliš mnoho na to, aby je redaktor manuálně zpracovával. Pro tento problém postačí adresář fotek vložit do adresáře galerie, kterou si redaktor přeje vytvořit. Posléze nakonfiguruje vybraný adresář a spustí skript na gerenování galerie. Script automaticky projde daný adresář a upraví fotky pro zobrazení na webových stránkách. Zároveň s tím pokládá také vodotisk, aby fotka nemohla být tak snadno použita konkurencí. Originály následně přesune do jiné složky pro případné budoucí použití.

4.1 Inspirace

Inspirace k tvorbě redakčního a publikačního systému jsem čerpal převážně s e-sportových webů. Jsou totiž ze všech redakčních systému na nejvyšší úrovni co se technického zpracování týče. Samozřejmě jsem se také inspiroval na nejpopulárnějších zpravodajských portálech - jak českých, tak také světových.

Celá teorie je postavená na ovládní informačního systému. Uživatel nikdy nesmí váhat, kde co na IS najít. V jednoduchosti je síla, systém musí být dostatečně intuitivní. Pokud dokážete, že uživatel se jedním kliknutím dostane tam, kam potřebuje, máte z půlky vyhráno. To samozřejmě neznamená, že okno webu bude zaplněné všemožnými odkazy.

Další důležitou roli hraje design stránek. Přehnaný vzhled, flash a vyskakující reklamy, které překrývají obsah, dokáží návštěvníky značně rozčarovat. Další velikou chybou je zobrazování desítek odkazů seskupených do jednotlivých sekcí. Určitě není naším cílem uživatele obtěžovat nepotřebnými informacemi. Naopak se mu budeme snažit „vnutit“ maximálně pouze dvě informace, které si nepřál vidět a upozornit ho tak na náš zájem neagresivně a slušně.

V neposlední řadě je důležité i programátorské pohodlí. Pokud se musí programátor zaobírat složitými zdrojovými texty, je velká pravděpodobnost, že svůj čas stráví raději předěláním již udělané práce, která není kvalitní. Proto je BP rozdělena na jednotlivé moduly, které obsahují důležité funkce. Seřazeny jsou podle abecedy či důležitosti.

4.2 Interface

Myslím si, že podobné nároky, které jsem uvedl v podkapitole 4.1 Inspirace, nesplňuje mnoho webů. Jedním z dobrých příkladů, jak by mohl takový e-sportovně založený redakční a publikační systém vypadat, je web <http://www.clancds.cz>. Naopak za názornou ukázkou, jak by takový systém vypadat neměl, může posloužit IS www.meetyourmakers.com/en/. Náhledy webů jsou k dispozici v příloze 2.

Jak jsem již zmínil, primární zaměření na interface je jednoduchost. Uživatel musí rozumět tomu, co se na obrazovce děje. Musíme také počítat s tím, že uživatel nemá či nechce používat technologie založené na javě.

Systém je tedy rozvržen do trojsloupcového vzhledu, kde bočnice slouží pro umístění reklam. Prostřední sloupec má navíc ve své horní části souhrnné informace, které se dají pomocí AJAXu měnit. Po přihlášení se objeví záložky, ze kterých lze ovládat další panely. Podobně fungují také záložky v nejnovějších prohlížečích.



Obr. 15 – Záložky

Další velikou výhodou je okamžité upozornění uživatele o informaci, kterou by se jinde dozvěděl až na další straně. Při registraci tak přímo můžeme kontrolovat shodnost hesel, obsazení uživatelského jména atp. Systém díky GeoIP při registraci odhaduje i zemi původu.

User Information

Required:

Username:

Password: Bad password

Repeat password:

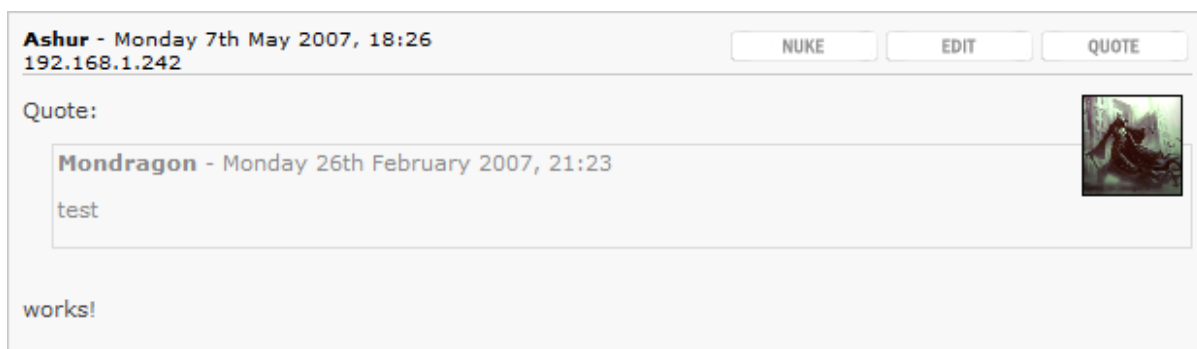
Email address:

Nationality:

Obr. 16 – Registrace s technologií AJAX

Grafický vzhled je kompletně psán v CSS. Je tedy snadné nějakou část změnit či předělat do jiné zobrazené formy. Díky CSS jsou navrženy i diskusní fóra do podoby, která snadno vypoví sdělení, jaké si autor přál. Redaktoři webu mají navíc obrázek, který definuje jejich postavení.

Obrázek současně překrývá jisté části, nikdy však nepřekryje text, který by se pod ním mohl potenciálně vyskytnout.

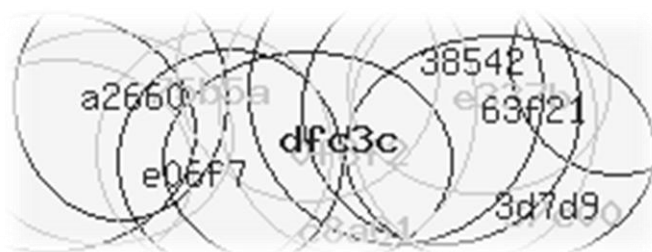


Obr. 17 – Diskusní fórum

4.3 Kde a jak systém funguje

Tento IS bude fungovat jako redakční systém pro profesionální tým a jeho redaktory. I to je důvod, proč je zdrojový kód psán anglicky. Je dosti pravděpodobné, že se o jeho rozšíření pokusí i někdo jiný. A musím říci, že psaní kódu v jiném než anglickém jazyce je zkrátka neprofesionální. Je nutné myslet na to, že svět nezačíná ani nekončí na hranicích a že čeština není světový jazyk.

Jelikož je systém určen pro již zajatý portál, bylo nutné jej připravit na případné problémy, které budou při přechodu ze starší verze na novější následovat. Některá data bude nutno importovat ze staré databáze, musí se inovovat systém proti spamovacím robotům a několik dalších vylepšení jistě ještě čeká na své odhalení.



Obr. 18 – Obrázek proti spamovacím robotům

4.3.1 Galerie

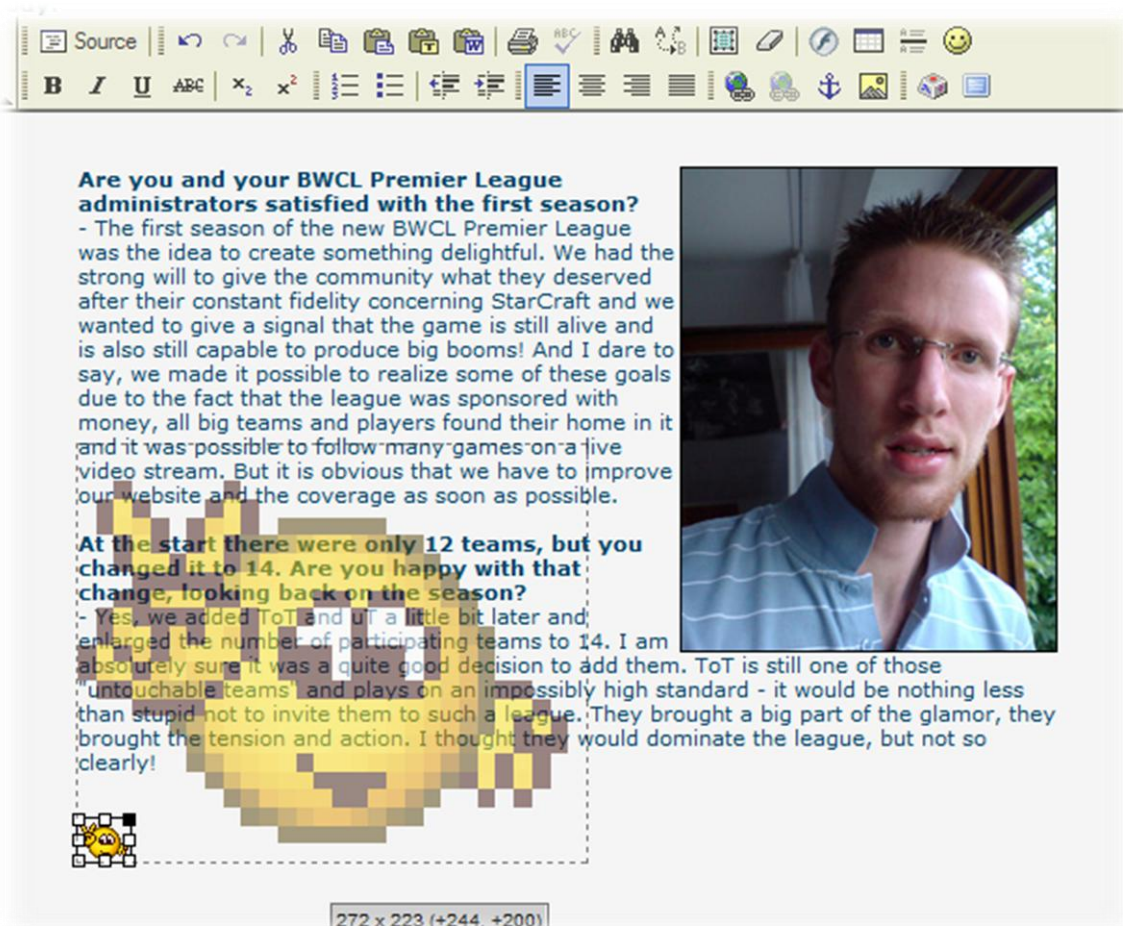
Systém obsahuje kromě základních požadavků jako je automatické generování fotogalerie z adresáře i různé doplňky. To zahrnuje změnu velikosti obrázku podle požadavků redaktora nebo konkrétní

umístění vodotisku s logem webu. V případě zapnuté javy se aktivuje prohlížení fotografií pomocí modulu Lightbox 2. Pokud uživatel javu nechce používat, obrázek se otevře v novém okně jako u jiného běžného IS.

4.3.2 Editor WYSIWYG

Pro pohodlnost byl implementován modul rozhraní pro práci s textem. Jedná se o velmi rozšířený FCKEditor, který je také založen na Javě. Znovu jsem dbal i na to, aby IS neztratil funkčnost, pokud uživatel Java povolenou nebo nainstalovanou nemá, případně pokud používá prohlížeč, který s javou neumí pracovat. V tomto editoru můžete uživatelům povolit jednotlivé nástroje, které používat mohou. Kategoricky se tedy liší psaní článků a komentářů.

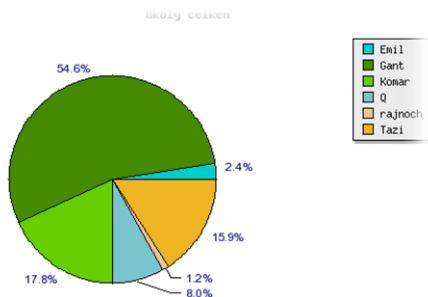
Pro pohodlí je implementováno také prostředí pro tvorbu článku přímo v jazyku XHTML, kde si můžete dovolit přesnější vyjádření toho, co opravdu potřebujete. Náhled takového článku je pak vidět v okně hned nad psaným kódem. Validita je ponechána přímo na redaktorovi.



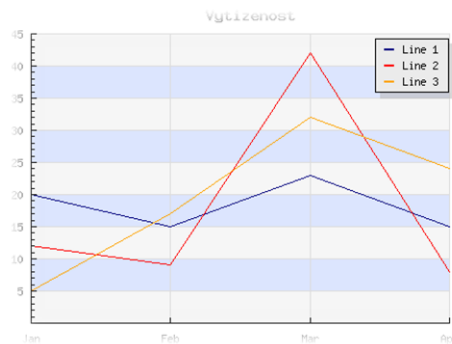
Obr. 19 – Náhled FKCEditoru, WYSIWYG modulu

4.3.3 Grafy

Pro generování grafů jsem použil modul JpGraph. Jedná se o objektově orientovanou knihovnu s dobře dokumentovaným API. Jeho možnosti využití jsou opravdu obrovské. Produktem této knihovny je obrázek o předem definovaných rozměrech, který obsahuje graf. Součástí



Obr. 20 – Koláčový graf



Obr. 21 – Lineární graf

4.3.4 Další moduly

Dalším doplňkem je plugin do modulu PHP, který přímo parsuje binární zazipovaná data z dané hry a exportuje je v podobě různých grafů a tabulek. Na vývoji tohoto pluginu jsem také částečně spolupracoval. Na jeho hlubší popis bohužel není v rozsahu BP místo, proto jsem to do ni nakonec nezařadil.

Naopak jsem se pro BP snažil takovýchto externích pluginů do PHP vyhnout. Nakonec jsem byl nucen použít pouze knihovnu GL2 pro generování antispamového obrázku a pro práci s fotografiemi. Další pluginy pro práci s fonty a pro generování PDF jsem obešel a použil jsem jinou implementaci. Náhled vodotisku a Lightbox 2 modulu naleznete v příloze.

4.3.5 Testování

Tento IS, resp. jeho části, již několik let fungují v několika webech. Dovolím si tedy říci, že většina IS je dobře odladěna. Pomocí příznaku můžeme ovlivnit vypisování chybových hlášek, a tak si v případě chyby dosti pomoci. Naopak naostro web nebude vypisovat chybovou hlášku – například SQL příkaz na server, který není dostupný.

4.4 Slabiny

Jednoznačnou slabinou systému je jeho zabezpečení. Ovšem z přesvědčení, že systém nebude podléhat cíleným útokům, a také z uživatelské pohodlnosti si myslím, že tak, jak je navržen,

postačuje. Je mi jasné, že použití HTML Editoru pro běžného návštěvníka je dost velkou dírou. Útočník pak může použít známé metody XSS (Cross-site scripting) [7] a krást tak neznalým uživatelům cookie či sessionid. Dozajista je dobré použití tzv. BBCode. Jeho implementace je pro případ nouze ponechána ve funkčním modulu *func.inc.php*. Na regulární výraz se můžete podívat v obrázku 17.

```

$patterns = array(
    '\[b\](.+?)\[\/b\]`is',
    '\[i\](.+?)\[\/i\]`is',
    '\[u\](.+?)\[\/u\]`is',
    '\[strike\](.+?)\[\/strike\]`is',
    '\[color=#([0-9]{6})\](.+?)\[\/color\]`is',
    '\[email\](.+?)\[\/email\]`is',
    '\[img\](.+?)\[\/img\]`is',
    '\[url=(([a-z0-9]+://) ([\w-]+\.[\w-]+\.) *[\w]+(:[0-9])?(/[^\n\r\t<]*?)?)\](.??)\[\/url\]`si',
    '\[url\]([a-z0-9]+?://){1}([\w-]+\.[\w-]+\.) *[\w]+(:[0-9])?(/[^\n\r\t<]*?)\][\/url\]`si',
    '\[url\]((www|ftp)\.[\w-]+\.) *[\w]+(:[0-9])?(/[^\n\r\t<]*?)\][\/url\]`si',
    '\[flash=([0-9]+),([0-9]+)\](.+?)\[\/flash\]`is',
    '\[indent\](.+?)\[\/indent\]`is',
    '\[size=([1-6]+)\](.+?)\[\/size\]`is'
);

```

Obr. 22 – Regulární výraz pro šablonu BBCode

Další slabinou je poměrně vysoké vytížení serveru. AJAX se totiž neustále dotazuje a tím může způsobit velmi vysoké vytížení. Určitě by bylo užitečné zjistit, o jak moc velké vytížení se jedná. Avšak s informací, které se mi dostaly do rukou, by ani stovka návštěvníků, která současně používá tyto skripty, neměla dělat serveru veliké potíže. Už několikrát se mi však stalo, že server pod náporom dotazů spadl. Bohužel nevím, kolik uživatelů v tu dobu systém používalo.

4.5 Systém interpretu

Protože spousta informačních systémů má občas problémy s běžným využíváním webové aplikace (jako například znovunačtení dokumentu), bylo třeba vyhnout se nejznámějším z nich. Nejjednodušší způsob, jak to dokázat, je ponechat si možnost přidat HTTP hlavičku kdykoliv v průběhu zpracování interpretu. Pokud totiž již něco vypíšete na výstup, nemůžete již hlavičku použít. A to je dosti limitující.

IS si tedy střádá výsledný HTML kód do několika proměnných, které vypíše až úplně na konci. To znamená, že klient potencionálně ztrácí čas, protože v běhu skriptu nedostává žádný výstup. Na druhou stranu IS dokáže lépe zpracovat jeho požadavky. Po přidání komentáře metodou post a následném znovunačtení stránky nehrozí tzv. doublepost. S tímto mají problémy i současné komerční aplikace.

4.6 Možná rozšíření

4.6.1 Kalendář

Jak již bylo řečeno v kapitole 3.2.3, systém musí být připraven na možnost rozšíření. V budoucnu se plánuje přidat kalendářní úkolově diskusní systém, který pomůže administrátorům lépe rozdělovat práci a poukazovat na termíny, které se blíží aktuálnímu datu, případně měsíci či jinému zvolenému intervalu.

BWCL vs xLo - 19:00	Days left: 0
WGTCL vs inFerno - 21:00	Days left: 0
Turnaj pro členy narodního A ymu	Days left: 5
WCG kvalifikace 2	Days left: 28
DREAMHACK SUMMER 2007	Days left: 41
LanCraft Fall 2k7	Days left: 124

Obr. 23 – Seznam položek v kalendáři



Obr. 24 – Grafická podoba kalendáře

Jako inspirace mi určitě poslouží kalendář z již existujícího IS, který je v podstatě předchůdcem aktuální verze. Náhled tohoto kalendáře je k dispozici u obrázků 17 a 18.

4.6.2 Automatický export a import XML

Následně bych se určitě rád soustředil na spolupráci s dalšími stránkami, které se věnují podobné tématice. Vyměňování, odkup nebo prodej sdělených informací by mohl být další cestou rozvoje tohoto IS. Proto bude důležité udělat modul, který bude periodicky spouštěn buď službou *cron*, nebo *plánovačem úloh*. Jeho úkolem bude stahovat či distribuovat data ve formě XML.

Jako inspiraci bych zde nejspíše použil komerční aplikaci firmy 100mega.cz, která distribuuje data potřebná pro chod eshopu, který je zaměřen na prodej spotřební elektrotechniky. Firma distribuuje jak struktury dat, tak i jednotlivé položky, ke kterým automaticky přidáváte marži.

5 Závěr

Tato práce vychází z již existujících zdrojových kódů, které už několik let spolehlivě fungují v několika informačních systémech. Na druhou stranu byl systém inovován, bylo použito nových technologií. Právě kvůli nim je doporučeno návštěvníkům používat nejnovější verze prohlížečů. Pokud má návštěvník starší verzi, samozřejmě si stále bude moci prohlížet obsah webu. Nebude však moci využívat všech vlastností IS.

Web bude sloužit jako IS pro skupinu redaktorů, kteří se věnují e-sportům. Plánovaná návštěvnost se by se dle aktuálních čísel měla pohybovat někde kolem 1000 lidí denně.

Při tvorbě BP jsem si osvojil práci s Javou a AJAXem, který je podle mě budoucností v tvorbě webových aplikací. Neustálé načítání celého dokumentu a v podstatě i kontrola formulářů je velmi neohrabaná a podobná konzolovým aplikacím. Tyto zkušenosti zajisté budu používat i ve své další tvůrčí činnosti a pravděpodobně i v případných updatech a upgradech této práce.

Literatura

- [1] Williams H. E., Lane D.: PHP a MySQL - Vytváříme webové databázové aplikace. 1. vyd. Praha: Computer Press, 2002, 530 s. ISBN 80-7226-760-4
- [2] Maciaszek L.A., Liong B.L.: Practical Software Engineering Addison Wesley. 2. vyd. Harlow England: Addison-Wesley, 2005, 864 s. ISBN 0-321-20465-4
- [3] Kosek, J.: HTML, tvorba dokonalých www stránek. Praha: Grada Publishing, 1998, 291 s. ISBN 80-7169-608-0
- [4] Budd, A. Moll, C. Collison, S.: CSS filtry, hacky a pokročilé postupy. Brno: Zoner Press, 2007, ISBN 978-80-86815-54-1
- [5] DuBois, Paul: MySQL profesionálně: komplexní průvodce použitím, programováním a správou MySQL. Brno: Mobil Media, 2003. 1071 s. ISBN 80-8659-341-X
- [6] Zábajník, O.: Vytváření dokumentů PDF v PHP. Dokument dostupný online na URL: <http://interval.cz/clanky/vytvoreni-dokumentu-pdf-v-php/>, 21.03.2007
- [7] Kozelková, H.: Editor ER diagramů s podporou převodu do relačního modelu. Dokument dostupný online na URL: <http://urtax.ms.mff.cuni.cz/~skopal/projekty/Kozelkova.pdf>, 15.02.2007
- [8] Mannila H., Raiha K.J.: The design of relational databases, Addison Wesley, 1994.
- [9] SHA hash functions. Dokument dostupný online na URL:<http://en.wikipedia.org/wiki/Sha-1>, 13.04.2007
- [10] MD5. Dokument dostupný online na URL:<http://cs.wikipedia.org/wiki/MD5>, 13.04.2007
- [11] Hausar, D.: Tvorba databází v MySQL - I. Dokument dostupný online na URL:<http://www.abclinuxu.cz/clanky/show/19642>, 22.04.2007
- [12] Cross-site scripting. Dokument dostupný online na URL: http://en.wikipedia.org/wiki/Cross_site_scripting, 15. 11. 2006
- [13] Dhakar, L.: Lightbox 2. Dokument dostupný online na URL:<http://www.huddletogogether.com/projects/lightbox2/>, 20. 12. 2006
- [14] Knabben, F.: FCKeditor - The text editor for Internet. Dokument dostupný online na URL:<http://www.fckeditor.net/>, 10. 12. 2006
- [15] JpGraph - PHP Graph Creating Library. Dokument dostupný online na URL:<http://www.aditus.nu/jpgraph/>, 14. 11. 2006
- [16] Zendulka, J., Rudolfová, I.: Databázové systémy IDS. Studijní opora. Dokument dostupný online na URL: https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/IDS-IT/texts/IDS_predn.pdf, 13.3.2007
- [17] Root.cz - linux, open source a free software. Dokument dostupný online na URL: <http://www.root.cz/>, 5.10.2006

Seznam příloh

Příloha 1. Copyright

Příloha 2. Důležité zdrojové texty

Příloha 3. Náhledy informačních systémů se stejnou tematikou

Příloha 4. CD/DVD

Příloha 1 - Copyright

Majitelem ochranné známky Macromedia Flash je společnost Adobe.

Majitelem ochranné známky Windows, Internet Explorer a .NET je společnost Microsoft.

Majitelem ochranné známky Firefox je společnost Mozilla.

Další ochranné známky jsou vlastnictvím příslušných společností.

Příloha 2 – Důležité zdrojové texty

```
function imageComposeAlpha( &$src, &$ovr, $ovr_x, $ovr_y, $ovr_w = false, $ovr_h
= false)
{
    if( $ovr_w && $ovr_h )
        $ovr = imageResizeAlpha( $ovr, $ovr_w, $ovr_h );

    imagecopy($src, $ovr, $ovr_x, $ovr_y, 0, 0, imagesx($ovr), imagesy($ovr) );
}

function imageResizeAlpha(&$src, $w, $h)
{
    $temp = imagecreatetruecolor($w, $h);
    $background = imagecolorallocate($temp, 0, 0, 0);
    ImageColorTransparent($temp, $background);
    imagealphablending($temp, false); // turn off the alpha blending to keep the
alpha channel

    imagecopyresized($temp, $src, 0, 0, 0, 0, $w, $h, imagesx($src),
imagesy($src));
    return $temp;
}
```

Obr 2.1, Funkce pro generování vodotisku

```
public function query()
{
    $numargs = func_num_args();
    if ($numargs == 0)
    {
        if ( DEBUG )
            $this->error("Calling query without arguments.");
        exit;
    }
    else
    if ($numargs == 1)
    {
        $arg_list = func_get_args();
        $query = $arg_list[0];
        $this->result = MySQL_query($query, $this->link_id);
    }
    else
    {
        $arg_list = func_get_args();
        $query = $arg_list[0];
        unset($arg_list[0]);
        for ($i = 0; $i < $numargs; $i++)
        {
            if ( !is_numeric($arg_list[$i]) )
                $arg_list[$i] = $this->quote_smart($arg_list[$i]);
        }
        $this->result = MySQL_query(vsprintf($query, $arg_list), $this->link_id);
    }

    if ( $this->result )
        return $this->result;
    else
    {
        if ($numargs == 1)
            echo $this->error("Invalid SQL query:<br /><b><span
style='color:#FF0000'>". $query."</span></b>");
        else
            echo $this->error("Invalid SQL query:<br /><b><span
style='color:#FF0000'>".vsprintf($query, $arg_list)."</span></b>");
        return false;
    }
}
```

Obr 2.2, Funkce query z třídy pro práci s databází

```

function toggle_item_ash(object, url, params)
{
    var obj = document.getElementById(object);
    var opt = {
        method: 'post',
        postBody: params,
        onSuccess: function(t)
        {
            obj.innerHTML = t.responseText;
            if(obj.style.display=='none')
                new Effect.Appear(obj);
        },
        onFailure: function(t)
        {
            alert('Error ' + t.status + ' -- ' + t.statusText);
            new Effect.Fade(obj);
        }
    }
    new Ajax.Request(url, opt);
}

```

Obr 2.3, Javascriptový URL dotaz

```

function browse_pics($dir = "./pictures/")
{
    if($checkDir = opendir($dir))
    {
        $cDir = 0;
        $cFile = 0;
        // check all files in $dir, add to array listDir or listFile
        while($file = readdir($checkDir)){
            if($file != "." && $file != ".."){
                if(is_dir($dir . "/" . $file)){
                    $listDir[$cDir] = $file;
                    $cDir++;
                }
                else{
                    $listFile[$cFile][0] = $file;
                    $listFile[$cFile][1] = date("Y-m-d",
filetime($dir . "/" . $file));
                    $listFile[$cFile][2] = filesize($dir . "/" . $file);
                    $listFile[$cFile][3] =
sprintf("%o", fileperms($dir . "/" . $file), -4);
                    $cFile++;
                }
            }
        }

        if(count($listFile) > 0)
        {
            foreach($listFile as $res)
                $sortAux[] = $res[1];
            array_multisort($sortAux, SORT_DESC, $listFile);

            for($k = 0; $k < count($listFile); $k++)
            {
                makepicture($dir, $listFile[$k][0]);
                $spacer = "";
                for($l = 0; $l < $i; $l++) $spacer .= "&nbsp;";
                $uploadindex .= "
<tr><td>".$listFile[$k][1]. " &rarr; <a
href='file.php?file=".urlencode($listFile[$k][0])."' target='new'><img
src='./save.gif' border='0'
alt='Download' /></a></td><td>[".number_format(($listFile[$k][2]/1024), 0, 3, "
")." KB]</td><td>".$listFile[$k][0]. "</td></tr>";
            }
        }
        closedir($checkDir);
    }
    return($uploadindex);
}

```

Obr 2.4, Procházení a rovnání adresářové struktury

```

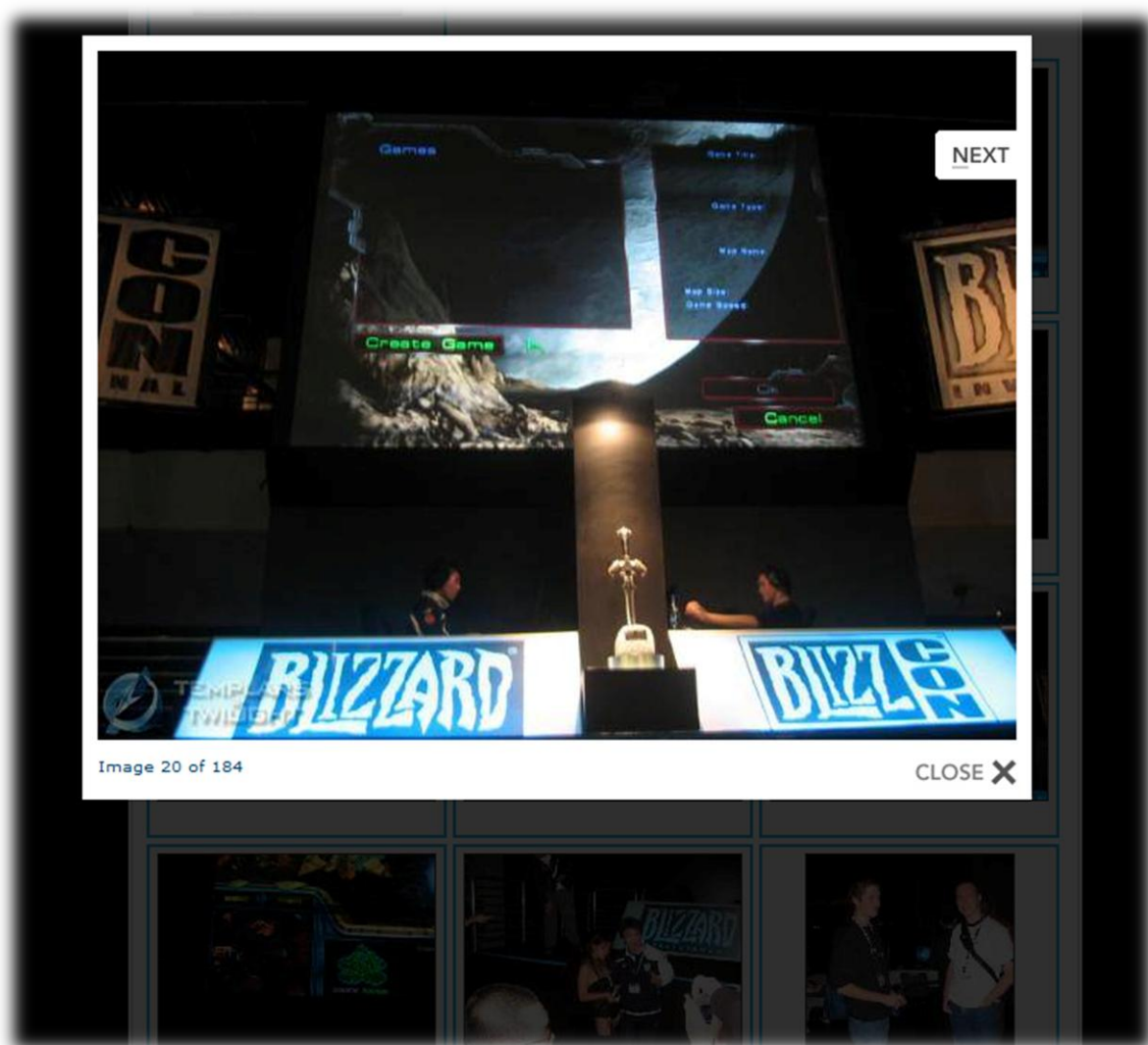
$error_msg = "";
if (strlen($_POST["username"]) < 5) $error_msg .= "Username must have at least
5 characters.<br />";
    if (!ereg('^[a-zA-Z]+[a-zA-Z0-9_\-]+[a-zA-Z0-9]+$', $_POST["username"]))
        $error_msg .= "Username contains disallowed characters.<br />";
if ( !isset($_SESSION["WEB_ToT_login_id"]) )
    if ((strlen($_POST["password2"]) < 4)|| (strlen($_POST["password2"]) <
4))
        $error_msg .= "Password must have at least 4 characters.<br />";
    if ($_POST["password"]!= $_POST["password2"]) $error_msg .= "Entered
passwords are not equal.<br />";
    if (strlen($_POST["email"]) < 5) $error_msg .= "Email address must have at
least 5 characters.<br />";
    if (!ereg('^[a-zA-Z0-9_\.\-]+@[a-zA-Z0-9_\-]+\.[a-zA-Z]{2,4}$',
$_POST["email"]))
        $error_msg .= "Email address is not valid.<br/>";

$result = $db->query($SQL_members_userexist, $_POST["username"],
$_POST["email"]);
if ( $db->num_rows($result) > 0 )
{
...

```

Obr 2.5, Kontrola registračního formuláře na straně serveru

Příloha 3 – Náhledy informačních systémů



Obr 2.6, Ukázka modulu Lightbox 2, nagenovaný obrázek s vodotiskem



Obr 2.7, <http://www.meetyourmakers.com>



Obr 2.8, <http://www.clan cds.cz>