

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INTERNETOVÝ PORTÁL HUDEBNÍ SKUPINY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

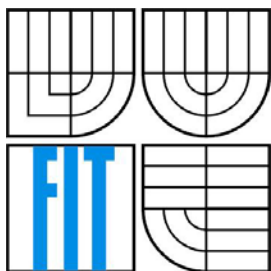
AUTOR PRÁCE
AUTHOR

Zdeněk Štěpaník

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

INTERNETOVÝ PORTÁL HUDEBNÍ SKUPINY

THE OFFICIAL WEBSITE OF MUSIC GROUP

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

Zdeněk Štěpaník

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Jiří Jaroš

BRNO 2007

Zadání

Zadání bakalářské práce/5530/2006/xstepa31

Vysoké učení technické v Brně - Fakulta informačních technologií
Ústav počítačových systémů

Akademický rok 2006/2007

Zadání bakalářské práce

Řešitel: **Štěpaník Zdeněk**
Obor: Informační technologie
Téma: **Internetový portál hudební skupiny**
Kategorie: Web

Pokyny:

1. Seznamte se s principy tvorby dynamických www stránek.
2. Seznamte se detailně s programovacími technikami PHP a databází MySQL.
3. Seznamte se důkladně s požadavky kladenými na internetový portál hudební skupiny. Podle výsledků analýzy navrhnete koncepci portálu.
4. Navrženou koncepci realizujte.
5. Zhodnoťte dosažené výsledky a diskutujte možné pokračování v projektu.

Literatura:

- <http://cz.php.net/>
- <http://www.mysql.com/>

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1-3.

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

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

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

Vedoucí: **Jaroš Jiří, Ing.**, UPSY FIT VUT
Datum zadání: 1. listopadu 2006
Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
602 00 Brno, Štefánikova 2



doc. Ing. Zdeněk Kotásek, CSc.
vedoucí ústavu

Licenční smlouva

Licenční smlouva je uložena v archivu Fakulty informačních technologií Vysokého učení technického v Brně.

Abstrakt

Tato bakalářská práce se zabývá implementací internetového portálu hudební skupiny Marshal. Jedná se o portál vytvořený programovacími technikami jazyka PHP a založený na databázi MySQL, jenž zahrnuje registraci fanoušků, fotogalerii s možností bodování a komentování jednotlivých fotek, diskuzi umožňující reagovat na jednotlivá vlákna, články, novinky, ankety atd. Klíčovou součástí pro administraci je systém CMS, který jsem vyvinul za účelem správy obsahu portálu.

Klíčová slova

hudební skupina, Marshal, portál, IS, CMS, PHP, MySQL, Flash, fotogalerie, diskuze, anketa

Abstract

This bachelor thesis deals with an implementation of the official website of music group Marshal. The website has been created using programming language PHP and is based on MySQL database. It includes fans registration, photogallery that supports rating and commenting photos, discussions with an option to reply on a particular thread, articles, news, enquiries, etc. The key component for administration is CMS system which was developed to manage contents of the website as a part of this thesis.

Keywords

music group, Marshal, website, IS, CMS, PHP, MySQL, Flash, photogallery, discussion, enquiry

Citace

Zdeněk Štěpaník: Internetový portál hudební skupiny, bakalářská práce, Brno, FIT VUT v Brně, 2007

Internetový portál hudební skupiny

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jiřího Jaroše.

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

.....
Jméno Příjmení
Datum

Poděkování

Tímto chci poděkovat Ing. Jiřímu Jarošovi, jakožto vedoucímu, za konzultace a materiály poskytnuté k vytvoření této práce.

© Zdeněk Štěpaník, 2007.

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

Obsah

Obsah	1
Úvod	2
1 Technologický základ	3
1.1 Historie WWW	3
1.2 PHP	7
1.3 JavaScript.....	8
1.4 MySQL	8
1.5 Flash.....	9
2 Struktura portálu	10
2.1 Instalační balíček	10
2.2 Zprovoznění na webovém serveru	11
2.2.1 Instalace serverových aplikací	12
2.2.2 Instalace volitelných komponent	13
2.2.3 Integrovaní do webového serveru.....	15
3 Implementační řešení portálu.....	15
3.1 Rozhraní CMS	15
3.2 Uživatelé	19
3.3 Články.....	20
3.4 Novinky	22
3.5 Ankety	22
3.6 Diskuze	24
3.7 Fotogalerie	26
3.8 Registrace fanoušků.....	29
4 Závěr	31
Literatura	32
Seznam použitých zkratk a symbolů.....	33
Seznam příloh	34

Úvod

Internet je bezpochyby celosvětovým fenoménem konce 20. století. Není již větší firmy, která by k němu nebyla připojena a neměla vlastní webovou stránku. K Internetu je již připojeno mnoho domácností, menších firem, organizací, o školách a univerzitách nemluvě. Na Internetu se dnes prezentují nejen komerční subjekty, ale i jednotlivci a seskupení s nekomerčními záměry. Takovými seskupeními mohou být i hudební skupiny, které na svých stránkách či portálech nabízejí svým fanouškům zajímavé informace a zábavu. A právě vytvořením internetového portálu hudební skupiny se zabývá tato bakalářská práce.

Tato práce podává technickou zprávu charakterizující všechny důležité aspekty vedoucí k vytvoření a zprovoznění tohoto typu portálu, cílem práce je osvětlit všechna zákoutí, jež tvorba takového projektu přináší. Jelikož s pojmem internetový portál je úzce spojen pojem Internet, pod nímž si běžný uživatel velice pravděpodobně představí službu nejrozšířenější, a sice World Wide Web, budu se v první kapitole věnovat právě této službě a technologickému základu Internetu, a to i z historického hlediska. Rozeberu technologie, které jsem při vývoji portálu uplatnil, jednotlivě, ale i v širším měřítku. Druhou kapitolu věnuji struktuře portálu, nastíním roli jednotlivých komponent a rozeberu postup, jímž lze portál efektivně zprovoznit na webovém serveru, a tím tedy umožnit fanouškům jej navštěvovat. Cílem třetí kapitoly je programová analýza portálu, včetně rozboru jednotlivých komponent a popisu, jakým způsobem jsou dílčí součásti integrovány pro vzájemnou spolupráci.

1 Technologický základ

Při tvorbě internetového portálu hudební skupiny Marshal (dále jen portál) jsem uplatnil řadu technologií, jež jsou podrobně rozebrány v této kapitole. Klíčovými technologiemi z hlediska vývoje portálu jsou skriptovací jazyky PHP¹, JavaScript a databázový server MySQL². Vezmeme-li v potaz interaktivitu, jelikož se jedná o portál s dynamickým a případně multimediálním obsahem, uplatnil jsem také masivně se rozvíjející technologii Flash³. Jelikož je vyvinutý portál webovou aplikací, podívejme se nejprve na vývoj webových technologií a zasadíme výše uvedené technologie do širšího kontextu.

1.1 Historie WWW

V roce 1990, kdy byla služba World Wide Web poprvé spuštěna na půdě výzkumného centra CERN, jsme si vystačili s pouhými třemi technologiemi. První z nich byl jazyk HTML (*HyperText Markup Language*), který slouží k zápisu webových stránek. HTML je dodnes ústřední technologií Webu, od níž se vše odvíjí. Dnes již sice existuje jazyk HTML ve verzi 4.01, nicméně je stále zpětně kompatibilní s původní jednoduchou verzí HTML.

Druhou nezbytnou technologií je přenosový protokol HTTP (*HyperText Transfer Protocol*), který zajišťuje přenos HTML stránek z WWW serveru do prohlížeče na straně klienta. Původní verze HTTP 0.9 byla velmi jednoduchá. V důsledku zvýšených požadavků na možnosti kontroly přenosu dokumentů a samotného zrychlení přenosu vznikly nové verze HTTP 1.0 a 1.1. HTTP 1.1 se stal standardem, který podporují všechny nejvýznamnější WWW servery a prohlížeče.

Třetí technologií nezbytnou pro implementování služby WWW jsou URL (*Uniform Resource Locator*). Každý objekt přístupný na Webu má svoji jedinečnou URL adresu, která slouží k vytváření odkazů na daný objekt.

Z dnešního pohledu spojení těchto tří technologií nenabízí mnoho – umožňuje pouze prohlížení elektronických dokumentů, které jsou provázány systémem odkazů. Jak se tedy ubíral vývoj dál k dnešní podobě Webu, který je interaktivní a reaguje na požadavky uživatele ?

První inovací byla možnost automatického generování stránek, které obsahují informace proměnlivé v čase. HTML stránka je soubor uložený na disku WWW serveru, který má své URL. Nic však nebrání tomu, aby URL ukazovalo na nějaký spustitelný soubor (program), který vygeneruje

¹ <http://www.php.net>

² <http://www.mysql.com>

³ <http://www.adobe.com/products/flash>

požadovanou HTML stránku. Tato stránka pak může obsahovat aktuální informace. Jelikož je spustitelný soubor vyvoláván WWW serverem, bylo zapotřebí vyvinout rozhraní, které by definovalo způsob spuštění programu a předávání dat mezi WWW serverem a programem. Rozhraní bylo pojmenováno jako CGI (*Common Gateway Interface*). Programům, které generují HTML stránky, se proto často říká CGI skripty.

Další vývoj přirozeně směřoval k tomu, aby uživatel mohl ovlivnit chování CGI skriptu. V HTML 2.0 se tedy objevily elementy, které umožňovaly na stránce definovat formulář. Údaje vyplněné uživatelem do formuláře odeslal prohlížeč serveru a ten je pomocí rozhraní CGI předal CGI skriptu k dalšímu zpracování. Tímto způsobem funguje na Internetu dodnes mnoho služeb. Typickým příkladem jsou různé vyhledávací portály. Uživatel zadá do vstupního formuláře klíčová slova, ta se odešlou vyhledávacímu serveru, kde CGI skript prohledá indexy (svoji databázi ...). Výsledkem běhu CGI skriptu je stránka v HTML, jež obsahuje odkazy na stránky vyhovující dotazu.

Vše vypadá jednoduše, ale skutečnost bývá složitější. Psaní CGI skriptů nebylo úplně snadné. Pro jejich psaní se používaly nejčastěji různé interpretované jazyky, jako Perl nebo příkazové shelly Unixu. Nebyl však problém použít v podstatě libovolný programovací jazyk, a tak existuje mnoho CGI skriptů napsaných v jazycích C a C++. Rovněž jazyk Java lze pro psaní CGI skriptů použít – takovým programům se pak říká servlety. Velké databázové systémy, jako např. Oracle, umožňují psaní CGI skriptů přímo ve vlastním jazyce (např. PL/SQL).

Pro vytvoření CGI skriptu tedy byla nutná znalost nějakého programovacího jazyka. Kromě toho musel člověk ovládat rozhraní CGI, které nepředávalo parametry zrovna v šikovném formátu. Většina CGI skriptů se proto z větší části skládala z kódu, který převáděl získané parametry do použitelné podoby.

V interpretovaných CGI skriptech navíc stálo spoustu práce zabezpečení skriptu. Šikovný hacker totiž mohl odesláním speciálního textu v polích formuláře získat přístup k systému, na kterém běžel WWW server.

CGI skripty generují svůj výstup v jazyce HTML. Jazyk HTML však nejde přímo kombinovat s jinými jazyky, a proto bylo generování HTML kódu v CGI skriptu nepříjemná záležitost, kde se každý řádek HTML kódu zadával jako parametr příkazu print či echo podle použitého jazyka. Správa větších aplikací je rovněž náročná, protože aplikace je roztroušena v mnoha samostatných souborech s HTML stránkami a CGI skripty.

Šikovný programátor však pomocí CGI skriptů dokázal vytvořit přímo kouzla. I technologie CGI má však své meze. Vidíme, že CGI skripty se provádějí na WWW serveru. Uživatelská odezva je tedy velmi pomalá – uživatel si stáhne stránku, poté vyplní a odešle formulář zpět na server, server spustí CGI skript a od něj získaný výstup zašle zpět do uživatelova prohlížeče.

Zhruba ve stejné době jako CGI skripty se poměrně rozšířila i další technologie SSI (*Server Side Includes*). SSI byly jednoduché příkazy, které se zadávaly do HTML stránky jako komentář. Stránky však byly uloženy v souborech se speciální příponou .shtml a tak WWW server věděl, že

před odesláním stránky v ní má provést všechny SSI příkazy. SSI umožnily provádění jednoduchých úkonů, jako vložení jiného souboru do stránky, nebo vypsání data poslední modifikace dokumentu. Své uplatnění našly především na rozsáhlých serverech, které chtěly mít ve všech stránkách standardizované záhlaví a patičku – optimální úkol pro nasazení SSI.

Řešení pomalé odezvy CGI skriptů spočívalo v přesunutí prováděných programů na stranu klienta - do prohlížeče. Zhruba ve stejné době – během roku 1996 – byly představeny dvě různé technologie, které řeší daný problém.

První technologií byl nový jazyk Java představený firmou Sun Microsystems. Tento jazyk umožňoval psaní Java appletů, což byly krátké programy, které byly začleněny přímo do HTML stránky. Ve stránce měly vyhrazeny prostor, který byl zcela pod jejich kontrolou. Možnosti Javy jsou opravdu široké – od jednoduchých animací oživujících stránku až po zábavné hry či aplikace.

Velkou výhodou Javy je její nezávislost na platformě – programy se po síti přenášely ve formě tzv. byte-code (bajtového kódu), který je spustitelný v libovolném operačním systému, pokud pro něj existuje interpret Javy JVM (*Java Virtual Machine*). Dnes můžeme s odstupem času říci, že Java je výborná a perspektivní technologie.

Druhou novou technologií roku 1996 byl JavaScript. JavaScript je jednoduchý skriptovací jazyk se syntaxí vycházející z jazyka Java. S JavaScriptem přišla firma Netscape a zabudovala jej do svého prohlížeče Netscape Navigator. JavaScript se zapisuje přímo do HTML kódu stránky a umí posloužit v mnoha situacích. Jeho nejčastější použití bylo ve spojení s formuláři. Krátké skripty v JavaScriptu tak mohly kontrolovat správnost údajů v polích formuláře ještě před odesláním na server. Uživatel tak získal nesrovnatelně rychlejší odezvu v porovnání s klasickým způsobem využívajícím pouze CGI skripty. Druhou oblastí použití JavaScriptu byla drobná vylepšení interaktivnosti stránek.

Dnes Javu i JavaScript podporují všechny nejrozšířenější prohlížeče – Microsoft Internet Explorer (MSIE), Mozilla Firefox i Netscape Navigator. Pravdou však je, že implementace JavaScriptu se může v různých prohlížečích lišit, což může vést k neúplné kompatibilitě kódu. To nutí autory stránek k vytváření složitějších skriptů, které se umějí přizpůsobit vlastnostem jednotlivých prohlížečů.

Možnosti JavaScriptu na straně klienta byly v původní implementaci od firmy Netscape nevyužity, a proto přišel v roce 1997 Microsoft s novou technologií dynamického HTML (DHTML). DHTML vychází ze stejné myšlenky jako JavaScript, ale mnohem více ji dotahuje. V DHTML je přístupný každý objekt webové stránky a s každým objektem můžeme kdykoliv manipulovat - měnit jeho obsah, způsob zobrazení či polohu. Jako objekty jsou přístupné i styly připojené ke stránce a jejich změnou lze velice jednoduchým způsobem dosáhnout zajímavých vizuálních efektů. Tato myšlenka později ovlivnila i JavaScript.

Přesuňme se však zpět na stranu serveru. Úspěch JavaScriptu byl tak obrovský, že se firma Netscape rozhodla pro využití JavaScriptu na straně serveru. Na serverech Netscape bylo možné do

HTML stránek psát skripty, které se provedou přímo na serveru. Uvnitř stránky byly skripty uzavřeny mezi párovými tagy <SERVER> a </SERVER> a server tak snadno rozpoznal, které části stránky má interpretovat. Výsledkem skriptů musel být HTML kód, který se doplnil do zbytku stránky a prohlížeči se již zasílala obyčejná HTML stránka. Řešení se dříve šířilo pod názvem LiveWire, dnes je jméno výstižnější – SSJS (*Server Side JavaScript*).

V SSJS je k dispozici mnoho objektů, které umožňují snadno pracovat s daty z formulářů, s databázemi atd. Výsledný efekt aplikací napsaných v SSJS je tedy stejný jako u CGI skriptů, s tím rozdílem, že psaní SSJS je mnohem jednodušší.

Aby Microsoft nezůstal pozadu, uvedl na trh ASP (*Active Server Pages*). ASP je obdobou SSJS. Jako programovací jazyk je možno využít VBScript nebo JScript, což je implementace JavaScriptu od Microsoftu. Systémy samozřejmě nejsou kompatibilní – ASP používá jiné značky k oddělení skriptu od stránky a hierarchie objektů, které zpřístupňují všechny důležité údaje, je rovněž rozdílná. Kromě VBScriptu a JScriptu je možno v ASP používat další jazyky, které dodávají třetí firmy – Perl, REXX, Python.

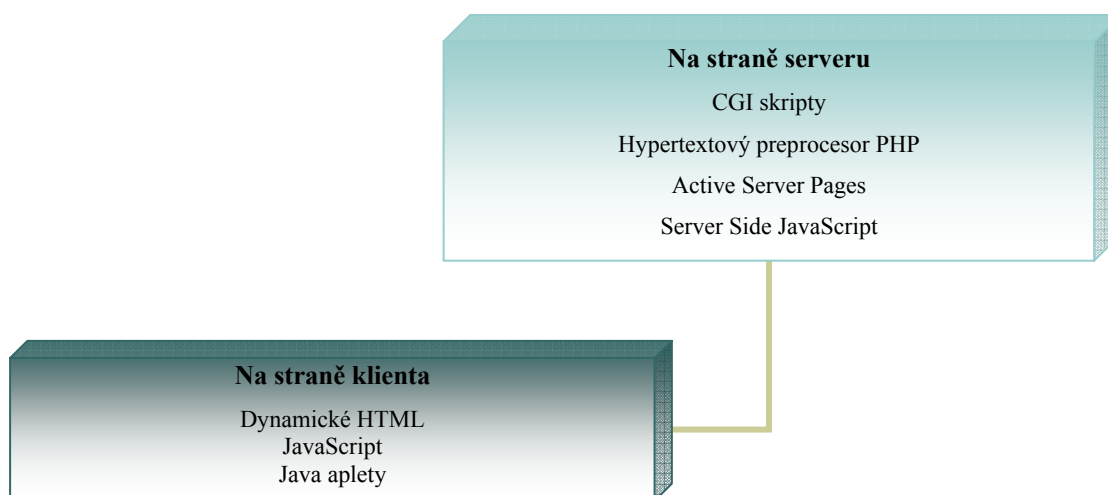
Na scénu webových technologií však brzy vstupuje systém PHP. SSJS a ASP mají jednu velkou společnou nevýhodu – jsou to komerční produkty, které nejsou nikterak levné a jejich použití je navíc svázáno s použitím WWW serveru dané firmy.

Všechny tyto nedostatky a mnohé další odstraňuje systém PHP. Princip použití PHP je obdobný jako u SSJS a ASP. Narozdíl od nich je však celý produkt šířen jako open-source. Analogicky bychom mohli položit otázku, jaký je nejpoužívanější webový server. Správná odpověď zní Apache⁴. Apache je používán na více než polovině všech webových serverů a vznikl na stejné bázi jako PHP. Mezi další osvědčené a volně šiřitelné systémy patří například i operační systém Linux a typografický systém TEX – opět špičky ve svém oboru.

Mezi další výhody PHP patří jeho interplatformová nezávislost. V současné době jsou k dispozici verze PHP pro většinu dostupných operačních systémů. Tak je tomu i v závislosti na webovém serveru, na němž PHP běží. Nejlépe si však dnes „rozumí“ se serverem Apache. PHP bylo vytvořeno pro psaní skriptů začleňovaných do HTML stránek, jeho použití je jasné, intuitivní a přímočaré. Díky těmto skutečnostem se PHP masivně rozšířilo, což dokazuje i počet serverů, na kterých je PHP používáno (již v roce 2001 bylo PHP používáno na více než 5,5 milionech domén).

Všechny zmíněné technologie lze rozdělit do dvou skupin podle toho, zda jsou prováděny na straně serveru nebo na straně klienta (viz. Obr. 1-1). Nic však nebrání jejich vzájemné kombinaci.

⁴ <http://httpd.apache.org>



Obr. 1-1: Technologie běžící na straně klienta resp. na straně serveru

1.2 PHP

PHP je jedním z nejpoužívanějších serverových skriptovacích jazyků pro HTML stránky. PHP, původně známé jako *Personal Home Pages*, představil na podzim roku 1994 Rasmus Lerdorf. Napsal ho jako prostředek pro záznamy o čtenářích svého online životopisu. První verze byla uvolněna počátkem roku 1995, kdy Rasmus zjistil, že pokud projekt uvolní jako open-source, lidé opraví chyby, které „nadělal“. První verze byla velmi prostá a měla jednoduchý parser, který rozeznal několik speciálních maker a poskytoval několik tehdy běžně užívaných funkcí.

V polovině roku 1995 byl parser přepsán a přejmenován na PHP/FI verze 2. FI znamená *Form Interpreter* (interpret formulářů), který Rasmus přidal, aby se PHP vypořádalo s narůstajícími požadavky webových stránek. Tehdy byla také přidána podpora mSQL. PHP/FI prošlo mohutným rozvojem a začali k němu přispívat další lidé.

V půlce roku 1997 přepsali Zeev Suraski a Andi Gutmans hlavní parser a PHP se změnilo na projekt celé skupiny. Tím byly položeny základy PHP3, pojmenovaného *PHP: Hypertext Preprocessor*.

Následující verze, PHP4, je další kompletní přepis od Suraskiho a Gutmanse a je postavena na Zend Engine. Poslední verzí je PHP5, která již běží na vylepšeném Zend Engine II, jenž přinesl nový objektový model, řadu rozšíření, efektivnější správu paměti a vyšší výkon. PHP má nyní přes dvě stovky pravidelných přispěvatelů, kteří pracují na různých částech projektu. Obsahuje velké množství rozšiřujících modulů, podporu všech oblíbených www serverů a standardní podporu MySQL a ODBC.

1.3 JavaScript

JavaScript je interpretovaný, multiplatformní programovací jazyk se základními objektově orientovanými schopnostmi. JavaScript se zrodil roku 1996, kdy jej firma Netscape vložila do svého prohlížeče Netscape Navigator. Následně bylo jádro jazyka integrováno i do dalších webových prohlížečů a rozšířeno přidáním objektů reprezentující okno prohlížeče a jeho obsah. Tato klientská verze JavaScriptu umožňuje vložit do webových stránek proveditelný obsah. Stránky se tak stávají dynamické – mohou obsahovat nejrůznější programy, které komunikují s uživatelem, řídí prohlížeč, či dynamicky vytváří obsah HTML. Při práci skriptu není třeba kontaktovat server, veškerou práci skriptu zajišťuje sám prohlížeč.

Jádro jazyka syntakticky připomíná C++ a Javu. Syntaxí však podobnost končí. JavaScript je jazyk bez typové kontroly, což znamená, že proměnné nemusí mít specifikovaný datový typ. Navíc je JavaScript čistě interpretovaný jazyk, na rozdíl například od kompilovaných C a C++ a na rozdíl od Javy, která je před interpretací kompilována do bajtového kódu.

Dle bezpečnostních zásad neumožňuje JavaScript na straně klienta souborové operace a přístup k systémovým objektům, výjimkou je však čtení a zápis cookies. Rovněž nepodporuje práci se sítí, avšak umožňuje donutit webový prohlížeč k načtení libovolného URL.

1.4 MySQL

MySQL (*My Structured Query Language*) je multivláknový, multiuživatelský relační databázový systém typu DBMS (*Database Management System*). Původně vznikl na půdě švédské firmy TcX v roce 1996, v současnosti se však jeho vývojem zabývá firma MySQL AB. Je považován za úspěšného průkopníka dvojího licencování – je k dispozici jak pod bezplatnou licenci GPL, tak pod komerční placenou licenci.

MySQL, jenž je založen na standardu relačních databázových systémů - jazyku SQL (*Structured Query Language*), díky své spolehlivosti a rychlosti směle konkuruje - zejména na poli internetového publikování - komerčním databázovým serverům (např. Microsoft SQL Server), jejichž cena (řádově desítky tisíce korun) je činí v podstatě nedostupnými pro řadu institucí, o soukromých osobách ani nemluvě. Kromě toho, že neklade na uživatele finanční nároky, je další nespornou výhodou MySQL jeho nezávislost na operačním systému. Je velice oblíben a nasazován v kombinaci MySQL, PHP a Apache jako základní softwarové vybavení webového serveru. Díky těmto skutečnostem je MySQL v současné době velice rozšířen (dle firmy MySQL AB má celosvětově více než 10 milionů instalací) a má vysoký podíl mezi používanými databázemi.

1.5 Flash

Flash se stal standardem pro internetové prezentace. Jedná se o moderní technologii, která účinně kombinuje vektorovou a bitmapovou animaci, zvukové efekty a skriptovací jazyk ActionScript za účelem vytváření interaktivního webového obsahu. Flash poprvé představila společnost Macromedia roku 1996, čímž vznikl Macromedia Flash 1.0. Tato verze sice ještě neobsahovala skriptování, ale nastínila nadějný směr vývoje webových animací.

Flash prošel rychlým vývojem, reflektoval trendy webových technologií a v současnosti již existuje ve verzi 8, jež se pyšní vytvářením multiplatformních animací, pokročilým skriptovacím jazykem ActionScript 2.0, obousměrným streamovaným přenosem zvuku a videa pomocí kamer a mikrofonů, prací s videem včetně alpha kanálu, blend filtry, možností cacheovat vektory jako rastry, ClearType antialiasingem písma, podporou mobilních zařízení a spoustou dalších pokročilých možností.

Flash animace se vytvářejí ve Flash editoru - zde se nakreslí (vloží) obrázky, umístí se do určitých vrstev, nadefinují se jejich pohyby a transformace v časové ose, mohou se přidat zvuky a skripty a nakonec se celá animace exportuje do formátu SWF, který je možno přehrát. Přehrávání je pak možné buď v prohlížeči (musí mít nainstalovaný Flash plugin), nebo ve zvláštním přehrávači (Flash player).

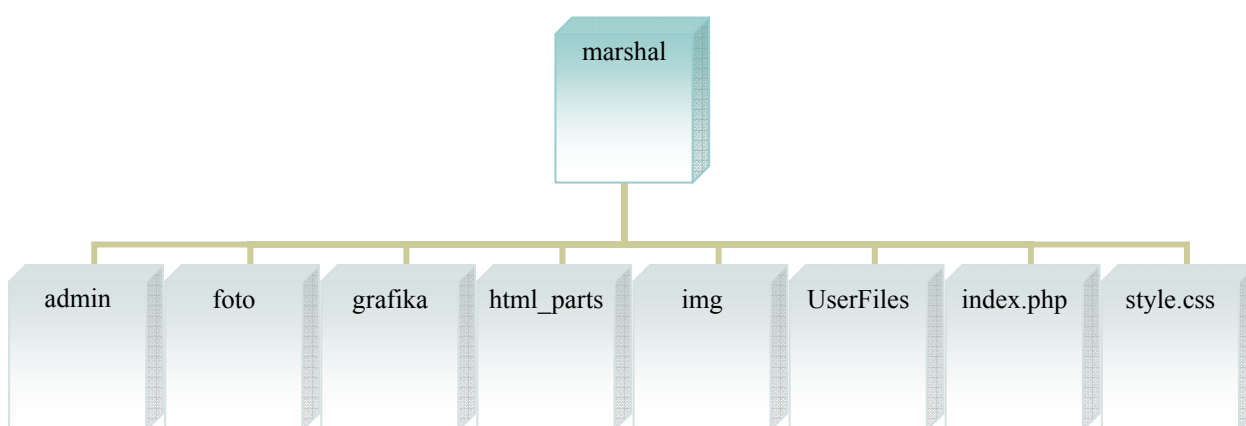
V současnosti lze ve Flashi vytvořit cokoliv od reklamních bannerů, animovaných menu, kreslených scén až po prezentace, nápadité formuláře, hry, pokročilé webové aplikace i celé stránky. Nejen díky těmto skutečnostem v současnosti Flash udává směr vývoje interaktivního webového obsahu.

2 Struktura portálu

Cílem této kapitoly je nastínit strukturu portálu a rozebrat, jakým způsobem lze portál zprovoznit na webovém serveru.

2.1 Instalační balíček

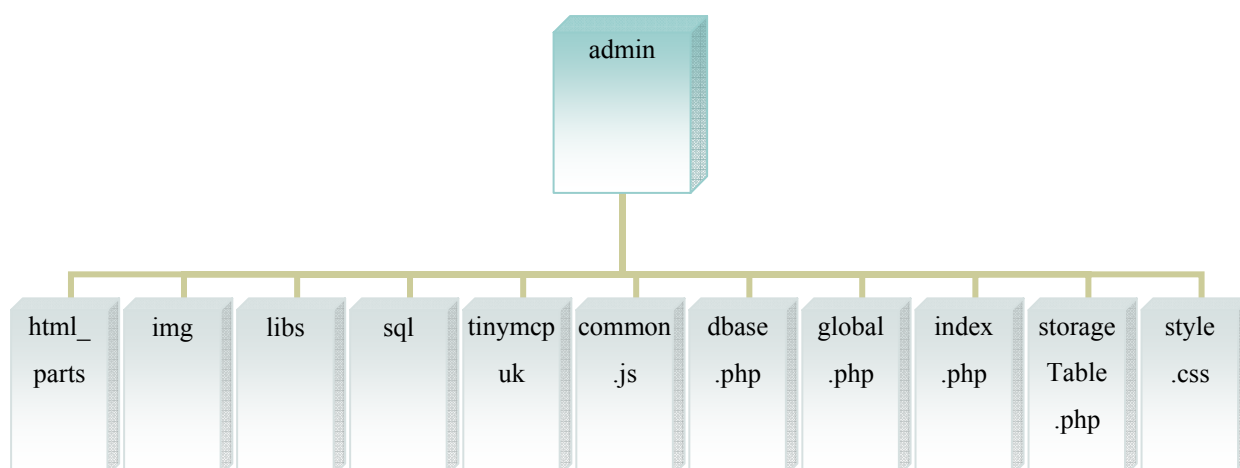
Instalační balíček portálu umístěný na CD-ROMu v přílohách obsahuje následující adresářovou strukturu (viz. Obr. 2-1):



Obr. 2-1: Adresářová struktura instalačního balíčku portálu

- admin – klíčová komponenta portálu – CMS (*Content Management System*) admin – administrační rozhraní pro správu obsahu portálu
- foto – fotogalerie se snímky, jež byly vloženy administrátorem pomocí CMS do portálu
- grafika – grafické prvky portálu
- html_parts – šablony s HTML kódem používané při generování stránek
- img – pomocné grafické prvky systémového rozhraní
- UserFiles – uživatelské soubory administrátorů, např. obrázky používané ve článcích, soubory ke stažení atd.
- index.php – hlavní skript portálu
- style.css – soubor s CSS (*Cascade Style Sheets*) styly

Následuje struktura adresáře admin, jenž zahrnuje rozhraní CMS (Obr. 2-2):



Obr. 2-2: Adresářová struktura rozhraní CMS

- html_parts – šablony s HTML kódem používané rozhraním CMS
- img – grafické prvky rozhraní
- libs – knihovny se skripty v jazyce PHP
- sql – sql skripty pro vytvoření potřebných databázových struktur v databázi MySQL
- tinymceuk – pokročilý subsystém WYSIVIG (*What You See Is What You Get*) HTML editoru, který administrátoři využijí například při kompozici a vkládání novinek, článků atd.
- common.js – pomocné skripty v jazyce JavaScript
- dbase.php – databázový konektor
- global.php – inicializační a konfigurační soubor CMS rozhraní
- index.php – hlavní skript CMS rozhraní
- storageTable.php – univerzální skript pro zobrazení MySQL schématu v intuitivní tabulce
- style.css – styly CSS, které používá CMS subsystém

2.2 Zprovoznění na webovém serveru

Jelikož je portál komplexní aplikací provozovanou na webovém serveru, je nutno specifikovat postup, jímž lze celý systém zprovoznit. Ve spektru internetových technologií existuje řada platform a webových serverů, na nichž je možno portál provozovat, avšak já se budu věnovat pouze popisu

instalace na platformu Windows a webový server Apache, neboť jsem portál na těchto technologiích vyvíjel. Tento popis slouží zároveň jako manuál pro instalaci.

Potřebnými prerekvizitami jsou Apache HTTP Server 2.0 (a vyšší), MySQL Essential 5.0 (a vyšší), PHP 5.0 (a vyšší), balíček zajišťující kompatibilitu Apache2 a PHP5 a runtime knihovny Visual C++ 2005. Dále volitelně Zend Studio Server pro optimalizaci a vysoký výkon, Zend Studio Client pro efektivní vývoj a možnost ladění a v neposlední řadě PHPMyAdmin pro snadnou administraci. Všechny tyto prerekvizity jsou umístěny na přiloženém CD-ROMu v adresáři *install_resources*. Důvodem, proč se budu věnovat podrobnému popisu vytvoření serverové platformy, je skutečnost, že správně nainstalovat a konfigurovat všechny klíčové serverové aplikace pro vzájemnou spolupráci, je mnohdy „oříšek“ i pro profesionálního administrátora.

2.2.1 Instalace serverových aplikací

Nejprve si na pevném disku připravíme potřebné adresáře pro instalaci serverových aplikací. Na některém z oddílů disku (nejlépe jiný oddíl, než na kterém je nainstalován operační systém) vytvoříme adresář pro serverové aplikace (např. D:_webservers; dále se budu v textu odkazovat na tento adresář symbolem \$WEBSERVER_DIR) a v něm vytvoříme podsložky *apache* (dále jen \$APACHE_DIR), *php* (dále jen \$PHP_DIR) a *mysql* (dále jen \$MYSQL_DIR). Začneme s instalací webového serveru Apache. Spustíme instalátor *apache_2.2.4-win32-x86-no_ssl.msi*, odsouhlasíme licenční podmínky a po přečtení předinstalačního readme pokračujeme k vyplnění informací o serveru. Zde zadáme síťovou doménu (např. kapela-marshal.cz), jméno serveru (např. www.kapela-marshal.cz) a email administrátora serveru (např. admin@kapela-marshal.cz), tyto informace nejsou příliš relevantní, avšak je dobrým zvykem je korektně vyplnit, dále zvolíme instalaci pro všechny uživatele, port 80 a instalovat jako službu. Vybereme vlastní instalaci a jako adresář, do něž bude Apache nainstalován, zvolíme \$APACHE_DIR, následně potvrdíme instalaci. Tímto jsme nainstalovali webový server Apache.

Nyní budeme pokračovat instalací PHP. Obsah archivu *php-5.1.6-Win32.zip* rozbalíme do \$PHP_DIR a soubor *php.ini-recommended* přejmenujeme na *php.ini*, který posléze otevřeme pro editaci. Nalezneme direktivu *extension_dir* a její hodnotu změním na \$PHP_DIR/ext, čímž jsme nastavili cestu pro rozšiřující moduly. Nyní je třeba povolit specifická rozšíření. Přesuneme se tedy do sekce dynamických rozšíření (Dynamic Extensions) a povolíme rozšíření (odkomentováním příslušného řádku) *mbstring*, *gd2* a *mysql*. Soubor *php.ini* uložíme a zavřeme.

V následujícím kroku pozměníme konfigurační soubor Apache tak, aby byl schopný pracovat s PHP. Otevřeme soubor \$APACHE_DIR/conf/httpd.conf pro editaci, nalezneme v něm sekci podpory dynamických sdílených objektů (Dynamic Shared Object (DSO) Support) a vložíme direktivu `LoadModule php5_module "$PHP_DIR/php5apache2.dll";` poté se přesuneme do sekce modulu *mime_module* (`<IfModule mime_module>`), kde vložíme direktivu

AddType application/x-httpd-php .php. Následně nalezneme sekci modulu `dir_module` (`<IfModule dir_module>`) a v ní přidáme do obsahu direktivy `DirectoryIndex` řetězec "`index.php`". Ještě zbývá na konec souboru vložit řádek s nastavením cesty pro konfigurační soubor PHP, tedy `PHPIniDir "$PHP_DIR"`, a soubor můžeme uložit a zavřít.

Abychom zajistili bezproblémovou spolupráci Apache2 a PHP5, je nutné provést kroky, které provedou potřebnou aktualizaci a zajistí plnou kompatibilitu. Otevřeme tedy archiv `php5apache2.dll-php5.1.x.zip`, v němž se nacházejí potřebné soubory. Soubor `httpd.exe.manifest` rozbalíme do `$APACHE_DIR/bin` a soubor `php5apache2.dll` do `$PHP_DIR`. Poté spustíme `vcredist_x86.exe` a potvrdíme instalaci runtime knihoven Visual C++ 2005; zajištění kompatibility je tímto dokončeno.

Nyní je čas nainstalovat databázový systém MySQL. Spustíme tedy instalátor `mysql-essential-5.0.27-win32.msi` a provedeme následující kroky. Vybereme vlastní instalaci, cílový adresář zvolíme `$MYSQL_DIR` a potvrdíme. Po dokončení instalace přeskočíme registraci na MySQL.com a možnost konfigurovat MySQL server necháme aktivní. Zvolíme následující možnosti: detailní konfigurace, serverová stanice, multifunkční databáze, úložný prostor vybereme nejlépe na některém z datových oddílů pevného disku, dále Decision Support (DSS)/OLAP, povolíme TCP/IP síťování, port 3306, striktní mód, nejlepší podpora pro multijazyčnost, instalovat jako službu, nastavíme heslo supervisory root a potvrdíme zápis konfigurace. Nyní ještě zkopírujeme knihovnu `libmysql.dll` z adresáře `$PHP_DIR` do instalačního adresáře operačního systému Windows (standardně `C:\windows`). Tímto jsme úspěšně dokončili instalaci serverových aplikací Apache, PHP a MySQL, provedli nezbytnou konfiguraci a zajistili jejich vzájemnou spolupráci.

2.2.2 Instalace volitelných komponent

Jak jsem se již zmínil výše, lze volitelně instalovat i další serverové komponenty, jež zaručí snadnou administraci, možnost ladění a optimalizaci pro vysoký výkon webového serveru Apache i jádra PHP. Kvalitním řešením je kombinace Zend Studio Server a Zend Studio Client. První ze zmíněných linkuje k jádru PHP kompletní engine pro ladění (debug) a tzv. Zend Optimizer, což je víceprůchodový analyzátor PHP skriptů zajišťující jejich optimalizaci pro nejvyšší výkon; druhý je jedním z nejkvalitnějších klientských prostředí pro vývoj a ladění aplikací napsaných v jazyce PHP. Další užitečnou komponentou je PHPMyAdmin, který ocení především administrátoři při správě databáze MySQL. Jedná se o velice rozšířený a oblíbený produkt začleňovaný na webových serverech.

Na začátek si opět připravíme adresáře pro instalaci produktů. V adresáři `$WEBSERVER_DIR` vytvoříme podadresáře `zendstudio_server` (dále jen `$ZENDSERVER_DIR`) a `zendstudio_client` (dále jen `$ZENDCLIENT_DIR`), poté se přesuneme do adresáře `$APACHE_DIR/htdocs` a zde vytvoříme podadresář `phpmyadmin` (dále jen `$PHPMYADM_DIR`).

Začneme s instalací Zend Studio Server. Spustíme instalátor *ZendStudioServer-5.1.0a-Windows-i386.exe*, odsouhlasíme podmínky licenční smlouvy a jako adresář pro instalaci zvolíme \$ZENDSERVER_DIR. Vybereme možnost instalovat na Apache 2.x a ponechat existující PHP. Následně budeme vyzváni k výběru umístění souboru *php.ini*, jenž nalezneme v \$PHP_DIR. Analogicky provedeme výběr kořenového adresáře webového serveru - \$APACHE_DIR a document rootu - \$APACHE_DIR/htdocs. V poli povolených hostů ponecháme IP adresu localhost (127.0.0.1), vložíme mezeru a zadáme lokální IP adresu stanice. Poté je nutno zadat heslo, které bude sloužit pro přístup k administrační konzoli, pro ověření heslo zopakujeme a potvrdíme instalaci. Před provedením instalace potvrdíme zastavení webového serveru, po jejím dokončení bude server opětovně spuštěn. Na konci ponecháme aktivní volbu otevření konzole, přihlásíme se heslem, které jsme zadali výše a přejdeme do kategorie PHP Info. Zde uvidíme informace o prostředí webového serveru a samotného PHP, posuneme-li se do sekce rozšíření PHP, měli bychom vidět nově nainstalovaná rozšíření Zend Debugger a Zend Optimizer. Tímto jsme úspěšně nainstalovali Zend Studio Server.

Nyní přejdeme k instalaci Zend Studio Client. Spustíme instalátor *ZendStudioClient-5_2_0.exe*, odsouhlasíme podmínky licenční smlouvy, zvolíme typickou instalační sadu, povolíme pomocné objekty prohlížeče (Browser Helper Objects (BHOs)) a jako cílový adresář pro instalaci vybereme \$ZENDCLIENT_DIR. Vytvoření zástupců a asociací přípon ponecháme implicitní, zvolíme PHP verzi 5.x a potvrdíme instalaci. Po dokončení zamítneme nainstalování Zend Guard a ponecháme aktivní volbu spustit vývojové prostředí Zend. Ve vývojovém prostředí zvolíme Tools -> Preferences -> Debug, v konfiguraci vybereme jako Debug Mode možnost Server, do pole Debug Server URL vložíme <http://localhost/> a provedené změny potvrdíme. Přejdeme opět do Tools, ale tentokrát zvolíme možnost Check Debug Server Connection. Měli bychom získat hlášení o úspěšném připojení k ladicímu serveru. Instalace Zend Studio Client je tímto dokončena.

Posledním produktem, jenž stojí za integraci do webového serveru, je PHPMyAdmin. Jeho instalaci provedeme jednoduše rozbalením archivu *phpMyAdmin-2.9.1-all-languages.zip* do adresáře \$PHPMYADM_DIR, poté je ještě nutné provést nezbytnou konfiguraci. Otevřeme tedy webový prohlížeč a zadáme URL <http://localhost/phpmyadmin>, získáme webovou stránku s chybovým hlášením o odepření přístupu, což je normální situace, o jejíž nápravu se v zápětí postaráme. Přejdeme na odkaz nastavovací skript, následně v sekci Servers zvolíme možnost Add a nastavíme následující parametry: Server hostname – localhost, Server port – 3306, Connection type – tcp, PHP extension to use – mysql, Authentication type – config, User for config auth – root, Password for config auth – vepíšeme heslo supervisory root, jež jsme zadali při konfiguraci MySQL serveru. Poté klikneme na tlačítko Add a zvolíme možnost Configuration -> Download. Server nám nabídne konfigurační soubor ke stažení a ten uložíme pod jménem *config.inc.php* do \$PHPMYADM_DIR. Nyní, když v prohlížeči přejdeme opět na URL <http://localhost/phpmyadmin>, zobrazí se nám již plně funkční administrační rozhraní PHPMyAdmin pro správu databáze MySQL.

2.2.3 Integrovaní do webového serveru

Abychom portál rozeběhli na webovém serveru, potřebujeme funkční a technologicky odpovídající serverovou platformu, jejíž příklad vytvoření je dokumentován výše. Nyní je potřeba rozbalit instalační balíček portálu *package.zip*, umístěný na příloženém CD-ROMu, do dokumentu rootu webového serveru (`$APACHE_DIR/htdocs`) a vytvořit potřebné databázové struktury, s nimiž bude portál provázán. K tomuto slouží skript *main.sql* v adresáři `sql` rozhraní CMS (viz. Obr. 2-2).

Nejjednodušším způsobem, jak tento skript provést, je importovat jej pomocí aplikace PHPMyAdmin, nebo jej předat interpretu MySQL pomocí příkazového řádku. Tento skript zajistí nejen vytvoření potřebných databázových schémat, ale také zaručí přístup ke správě portálu nejprivilegovanějšímu (a zprvu jedinému) superuživateli s uživatelským jménem `supervisor`. Prvotní heslo superuživatele je nastaveno stejně jako uživatelské jméno, tedy `supervisor`, to je možno (a zároveň silně doporučeno) změnit po prvním přihlášení do portálu.

3 Implementační řešení portálu

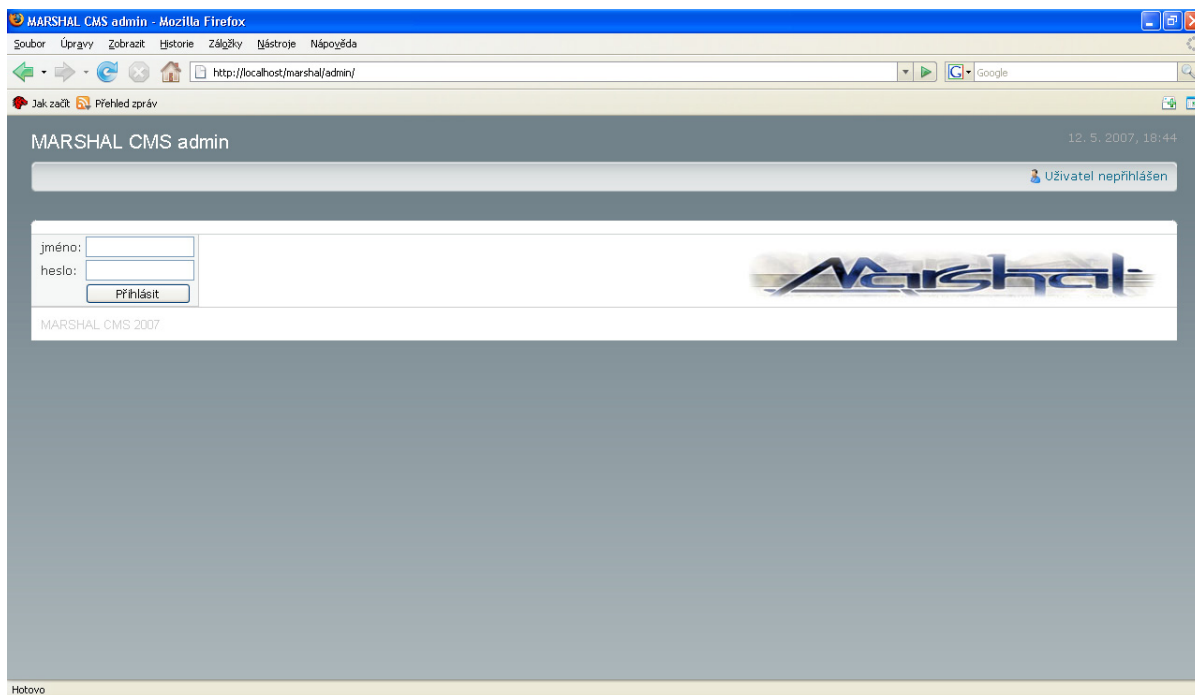
V implementačním řešení portálu se zaměřím na jednotlivé komponenty systému, rozeberu jejich role a nastíním, jakým způsobem mezi sebou dílčí součásti spolupracují. Každé komponentě se budu věnovat podrobně, nebudu však zabíhat do zbytečných detailů na úrovni zdrojového kódu, pro názornou dokumentaci použiji snímky (screenshoty) z běžícího systému.

Mezi součásti, které vytvářejí obsahovou náplň portálu, patří: fotogalerie s možností hodnocení fotografií, diskuze umožňující reagovat na jednotlivá vlákna, ankety s neomezeným počtem možných odpovědí, články a novinky vytvořené pomocí integrovaného WYSIWYG editoru a v neposlední řadě příspěvky k jednotlivým diskuzím, článkům či fotografiím. Jednotlivé součásti a jejich vzájemné vztahy zobrazuje model systému na příloženém ER (*Entity Relationship*) diagramu.

Nad těmito komponentami leží vrstva rozhraní CMS, která zajišťuje jejich administraci v přehledném, intuitivním prostředí. Toto rozhraní poskytuje administrátorům správu dynamického obsahu portálu, jenž se pak zobrazuje návštěvníkům (fanouškům) na webové stránce.

3.1 Rozhraní CMS

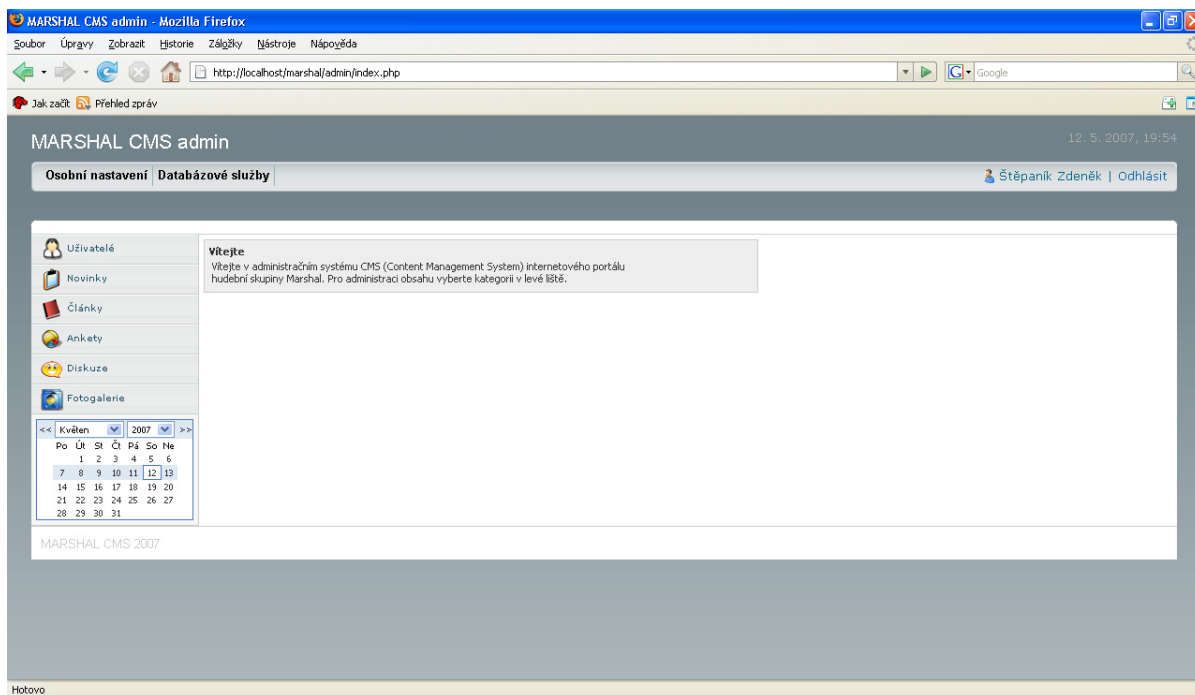
Rozhraní CMS se nachází v podadresáři `admin` kořenového adresáře portálu (viz. Obr. 2-1). Ve své podstatě se jedná o integrovaný informační systém, jejímž cílem je zajistit administrátorům kontrolu nad celým portálem a umožnit jim spravovat dynamická data, která se zobrazují návštěvníkům na `www` stránce. Přihlášení do rozhraní CMS ilustruje následující snímek (Obr. 3-1):



Obr. 3-1: Přihlášení do rozhraní CMS

Přihlášení probíhá následujícím způsobem: uživatel zadá do zobrazeného formuláře svoje přihlašovací jméno (login) a heslo, formulář je poté odeslán na server, který se pokusí nalézt v databázové tabulce uživatelů (viz. Příloha 2 – Model systému) odpovídající login. Pokud jej nalezne, zašifruje heslo hašovací funkcí MD5 (*Message-Digest Algorithm 5*) a porovná ho se záznamem v databázi. Jestliže si hesla odpovídají, inicializují se příslušné struktury (session proměnné PHP a cookies) a uživatel je přihlášen. Z uvedeného mechanismu však vyplývá bezpečnostní riziko, jelikož heslo přichází na server v nezašifrované podobě. Zašifruje se až na serveru, tzn., že by mohlo dojít k odposlechu na síti a k zjištění hesla. Toto je však problém, který se dá jednoduše odstranit v konfiguraci webového serveru Apache, jenž umožňuje nastavit komunikaci na zabezpečeném kanálu HTTPS. Na tomto kanálu jsou přenášena data šifrována pomocí SSL (*Secure Sockets Layer*), což zabraňuje odposlouchávání a podvržení dat.

Po úspěšném přihlášení se uživateli zobrazí hlavní menu systému CMS, které ilustruje následující snímek (Obr. 3-2):



Obr. 3-2: Hlavní menu systému CMS

Toto menu tvoří levý příkazový panel, pravý obsahový panel a vrchní lišta tvořící pop-up menu s doplňujícími funkcemi. V levém panelu jsou zobrazeny všechny dostupné kategorie, jejichž obsah může administrátor upravovat (*Uživatelé*, *Novinky*, *Články*, *Ankety*, *Diskuze*, *Fotogalerie*) a kalendář zobrazující aktuální datum s možností listování. Pravý panel zobrazuje obsah aktuální kategorie, kterou administrátor vybral. Vrchní lišta obsahuje příkazy *Osobní nastavení* resp. *Databázové služby*, které po přejetí myši zobrazí „vyskakovací“ menu *Změnit heslo* resp. *Exportovat databázi* a *Importovat databázi*. Příkazem *Změnit heslo* si aktuálně přihlášený uživatel může změnit stávající heslo, což je velice doporučeno superuživateli po prvním přihlášení do portálu. Funkce *Exportovat databázi* slouží k exportu dat z databáze do formátu XML (*Extensible Markup Language*), tato funkce je užitečná pro zálohování databáze přímo z rozhraní CMS. Analogicky funkce *Importovat databázi* umožňuje importovat uložená data z XML. Důležitým faktem je, že funkce pro export zálohuje pouze data, nikoliv strukturu databáze (informace o primárních a cizích klíčích, restrikce pro specifické tabulky atd.); pro kompletní strukturální i obsahovou zálohu je třeba použít sofistikovanější funkce pro export databáze (do formátu SQL), kterou obsahuje například PHPMyAdmin.

Grafický layout (rozvržení) CMS systému byl vytvořen s ohledem na přehlednost a přímočaré ovládání. Panely jsou rozmístěny intuitivně, cílem byla jednoduchá orientace pro uživatele. Každou kategorii symbolizuje specifická ikona s názvem sekce, po jejímž rozkliknutí se zobrazí obsah vybrané sekce v pravém panelu.

Když se zamyslíme nad obecnou podstatou systémů CMS, dospějeme k výsledku, že mezi jejich hlavní možnosti musí patřit zobrazování obsahu databázových tabulek a vkládání, úprava a odstraňování dat z databáze. Mojí snahou při tvorbě tohoto portálu bylo zahrnout všechny tyto funkce

do systému CMS pokud možno co nejuniverzálněji, což vedlo k vytvoření modulů *storageTable* a *main* (viz. Obr. 2-2), které tyto funkce implementují.

Modul *storageTable* slouží k zobrazení obsahu databázové tabulky. Využívá speciálního schématu databáze MySQL pojmenovaného *information_schema*, které obsahuje důležité informace o vlastnostech databázové tabulky. Nalezneme zde jednotlivé atributy, datové typy, definice primárních, cizích i unikátních klíčů, restriktce atd. Díky těmto údajům je modul schopen univerzálně zobrazit obsah jakékoliv tabulky. Výstupem modulu (viz. Obr. 3-3) je intuitivní tabulka v HTML kódu, která zobrazuje záznamy dané tabulky databáze. Mezi zajímavé funkce této tabulky patří podpora řazení dle určitého sloupce a schopnost fixace záhlaví při skrolování obsahu. Výstup může být ovlivněn generátorem filtrů, jejímž úkolem je filtrovat jen určité záznamy.

Uživatelé

Filtr

Login: Příjmení: Kategorie: [vše]

Akce	Login	Práva	Příjmení	Jméno	Email	Poslední přihlášení	Vytvořil/Upravitel	Poslední úprava
	krejci	fan	Krejčí	Michal	m.krejci@email.cz		Štěpaník Zdeněk	2007-05-12 22:41:18
	mervis	fan	Merva	Milan	m.merva@post.cz		Štěpaník Zdeněk	2007-05-13 00:12:24
	michalkova	fan	Michalíková	Lucie	l.michalkova@atlas.cz		Štěpaník Zdeněk	2007-05-13 00:13:45
	musil	fan	Musil	Petr	petr.musil@post.cz		Štěpaník Zdeněk	2007-05-12 22:40:18
	nemec	fan	Němec	Marek	m.nemec@atlas.cz		Štěpaník Zdeněk	2007-05-12 22:43:08
	novak	fan	Novák	Petr	petr.novak@seznam.cz		Štěpaník Zdeněk	2007-05-12 22:38:32
	sobek	fan	Sobek	Milan	m.sobek@seznam.cz		Štěpaník Zdeněk	2007-05-13 00:10:09
	supervisor	administrátor	Štěpaník	Zdeněk	xsteba31@stud.fit.vutbr.cz	2007-05-13 00:13:45		
	svobodova	fan	Svobodová	Jana	svobodova@gmail.cz		Štěpaník Zdeněk	2007-05-13 00:11:07
	urbanek	fan	Urbaněk	Karel	karel.urbanek@seznam.cz		Štěpaník Zdeněk	2007-05-12 22:42:31

Obr. 3-3: Ukázka výstupu modulu *storageTable*

Modul *main* obsahuje funkce pro vložení, úpravu a zrušení záznamů databázové tabulky. Tyto funkce jsou stěžejní pro celý systém, neboť zajišťují většinu databázových operací. K získání informací a správnému provedení těchto operací modul rovněž používá *information_schema*. Výstupem modulu (viz. Obr. 3-4) je formulář v HTML, jenž obsahuje prvky uživatelského rozhraní korespondující s prvky odpovídající databázové tabulky. Po odeslání formuláře na server se provedou patřičné změny v databázi. V modulu *main* je také integrován pokročilý HTML WYSIWYG editor založený na open-source projektu TinyMCE⁵. Je však rozšířen o moduly souborového manažeru a editoru obrázků. Administrátoři jej využijí například při kompozici článků, novinek atd.

⁵ <http://tinymce.moxiecode.com>

Novinky

Primární klíč

Titulek *

Text *

Path:

Článek

Vytvořil/Upravil

Poslední úprava

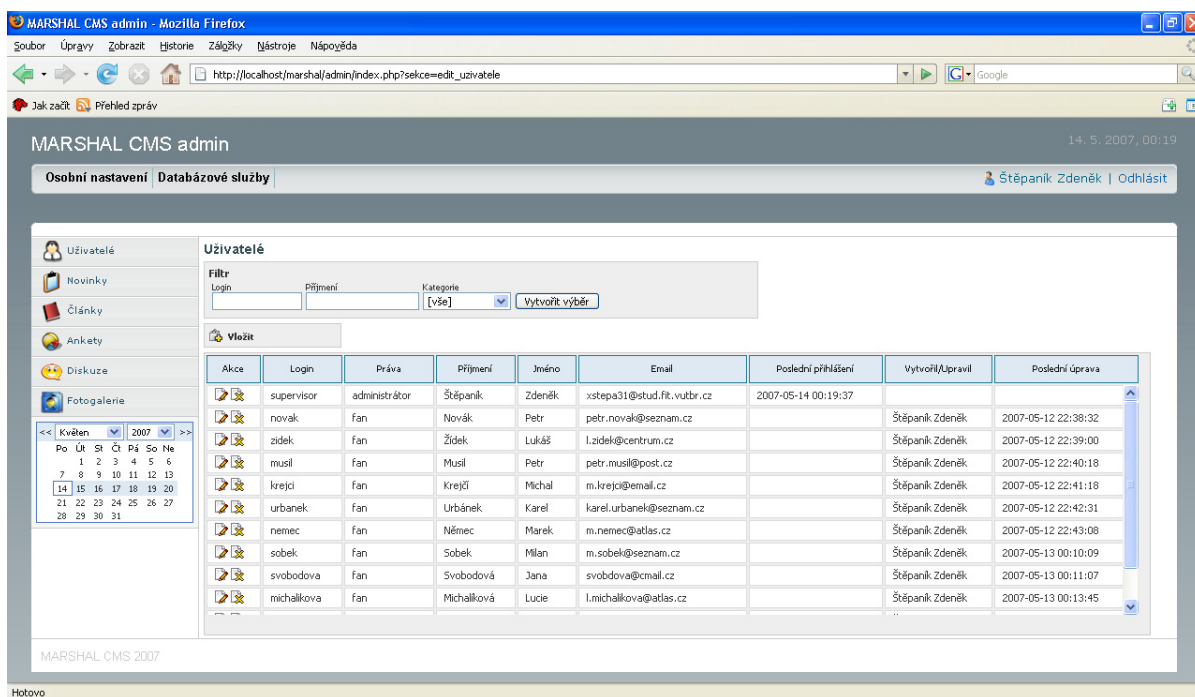
* povinné

Obr. 3-4: Ukázka výstupu modulu main

3.2 Uživatelé

V systému jsou definovány dvě kategorie uživatelů. První z nich je fanoušek, který může procházet webové stránky portálu, smí hodnotit fotografie, zanechávat příspěvky atd. Fanoušek však nemá žádná práva v systému CMS, je mu umožněna pouze změna hesla. Druhou kategorií je administrátor, jenž má plnou kontrolu nad portálem. Administrátor může provádět stejné úkony jako fanoušek, navíc má však všechna privilegia v systému CMS. Má přístup ke všem kategoriím portálu a může libovolně měnit jejich obsah. Administrátor může vytvářet nové uživatele, může také upravovat jejich údaje nebo je smazat. Kategorie uživatelů a jejich práva jsou nadefinována v konfiguračním souboru *global.php* (viz. Obr. 2-2). Pokud by tedy bylo zapotřebí vytvořit novou kategorii, například uživatele, jenž by pouze publikoval a upravoval články, stačilo by jednoduchým způsobem pozměnit tento konfigurační soubor.

Po nainstalování portálu na webový server, existuje v systému pouze jediný superuživatel a tím je supervisor. Tento uživatel má administrátorská práva a pokud existuje pouze jediný administrátor portálu, stačí v systému pouze tento jediný administrátorský účet. Pokud je však více osob, které budou portál spravovat, je nutné, aby superuživatel vytvořil nové administrátorské účty v systému. Práci s uživatelskými účty v rozhraní CMS dokumentuje následující snímek (Obr. 3-5):



Obr. 3-5: Správa uživatelů v rozhraní CMS

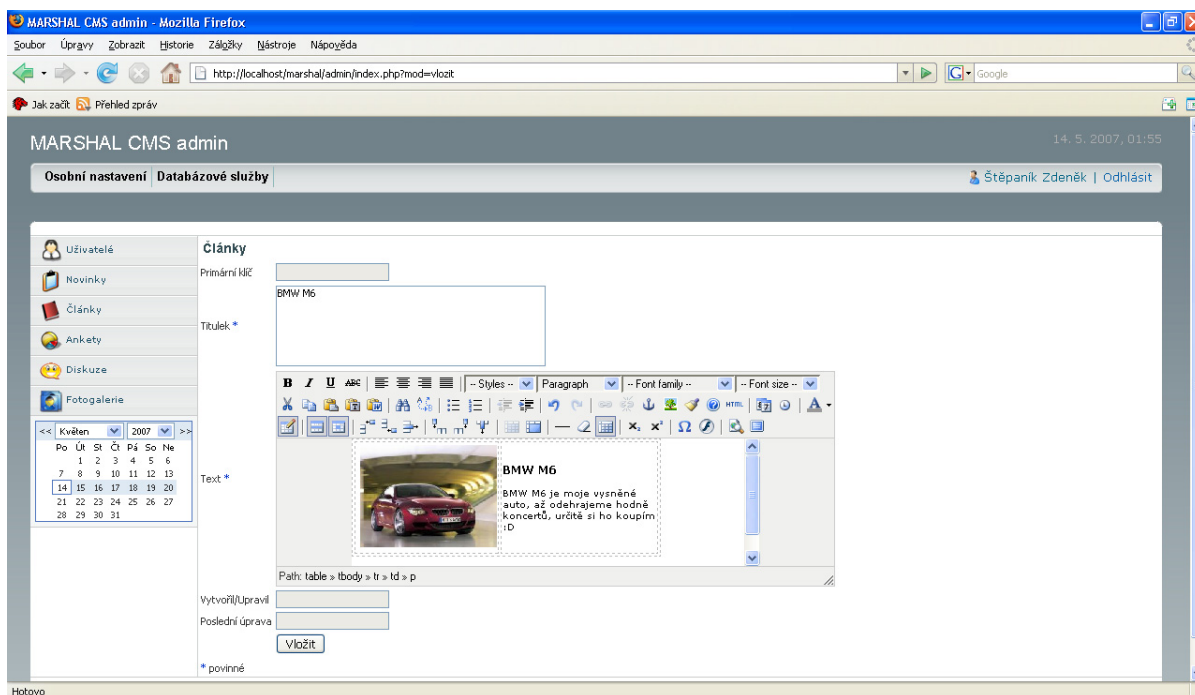
3.3 Články

Pro publikování článků je v systému účelně integrován modul WYSIVYG editoru, jenž umožňuje grafickou kompozici HTML. Nejedná se tedy o vkládání článků na úrovni kódu HTML, avšak na úrovni kompoziční práce s jednotlivými prvky, např. s tabulkami, obrázky, odstavci, odkazy atd. Tento modul je založen na velice oblíbeném open-source editoru TinyMCE, je však rozšířen o spoustu vylepšení, jakými jsou například souborový manažer, jenž umožňuje nahrávání a správu souborů, a obrázkový editor, pomocí něhož si může administrátor převzorkovat, oříznout či otočit nahrané snímky přímo na serveru, bez použití jiného grafického softwaru.

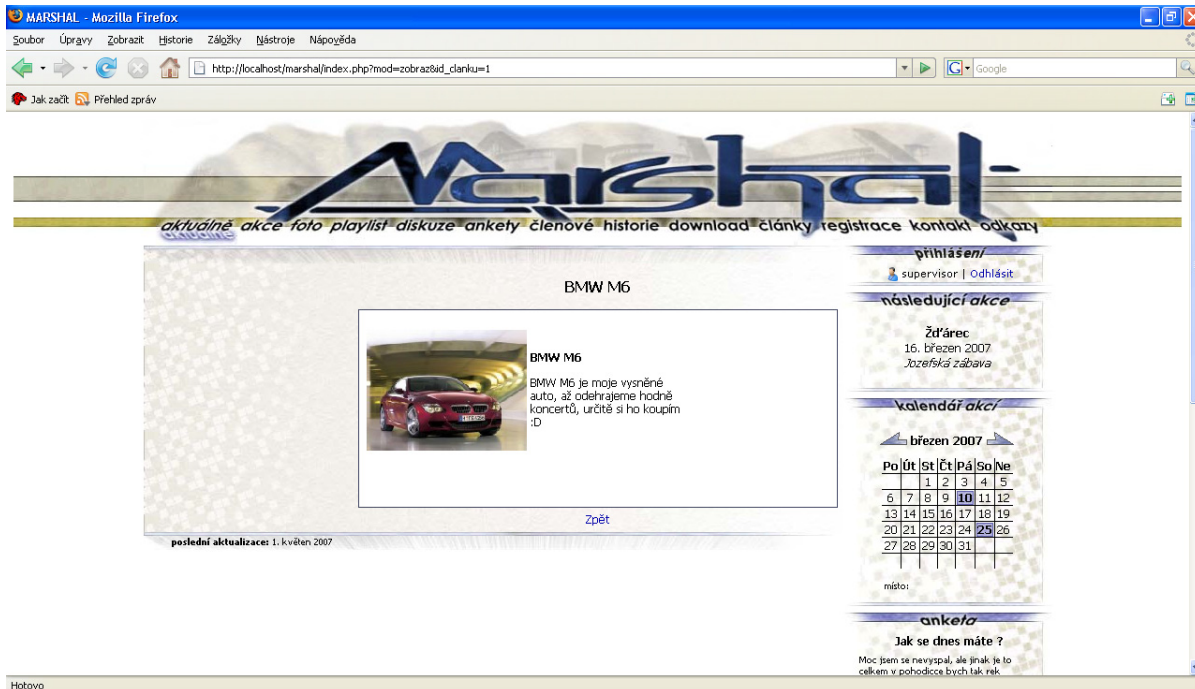
Tento modul je napsán v jazyce JavaScript a dokáže na www stránce elegantně změnit formulářové prvky typu `<TEXTAREA>` `</TEXTAREA>` z „obyčejného“ textového pole na HTML editor. Používá konektor napsaný v jazyce PHP, jenž zpřístupňuje výše uvedené souborové a grafické operace na straně serveru. Možnosti tohoto modulu jsou opravdu široké, umožňuje od jednoduchých operací typu změny stylu písma či vložení obrázku až po pokročilé funkce, jakými jsou vložení/editace tabulky, vložení multimediálního obsahu (Flash, RealMedia, ...), kontrola pravopisu, editace v celoobrazovkovém módu atd. Umí také zpracovat text vytvořený v aplikaci Microsoft Word, stačí jej zkopírovat do schránky a pak vložit pomocí funkce *Paste from Word*.

Články vytvořené v rozhraní CMS jsou ihned přístupné fanouškům na www stránce, zobrazí se jim přesně v takové podobě, v jaké byly vytvořeny pomocí WYSIVYG editoru. Přihlášení uživatelé mohou k těmto článkům zanechávat příspěvky, které se pak zobrazují všem návštěvníkům portálu.

Analogie mezi obsahem vytvořeným v CMS a obsahem, jenž se zobrazuje na webové stránce návštěvníkům, je zřejmá a logická, vysvětluje podstatu a účel rozhraní CMS. Tuto vazbu v kategoriích článků zobrazují následující snímky (Obr. 3-6 resp. 3-7):



Obr. 3-6: Kompozice článku v rozhraní CMS



Obr. 3-7: Zobrazení článku návštěvníkovi na www stránce

3.4 Novinky

Novinky jsou kategorií, která se zobrazuje návštěvníkovi při otevření webové stránky portálu. Je to kategorie, která má návštěvníka zaujmout, musí tedy obsahovat zajímavé prvky. Koncepce novinek je v systému obdobná jako u článků. Administrátor je vytváří v rozhraní CMS pomocí WYSIVYG editoru, kde může vkládat nejen statický obsah, ale i dynamický, jako např. Flash či jiné interaktivní prvky.

Novinky obsahují většinou krátký text, v systému jsou však úzce spjaty s články, na které se mohou odkazovat. Implementační stránka novinek je téměř stejná jako u článků, nemá tedy smysl se zabývat jejich detailnějším rozbořením. Rozdíl mezi novinkami a články je pouze v zobrazení na webové stránce portálu (viz. Obr. 3-8).

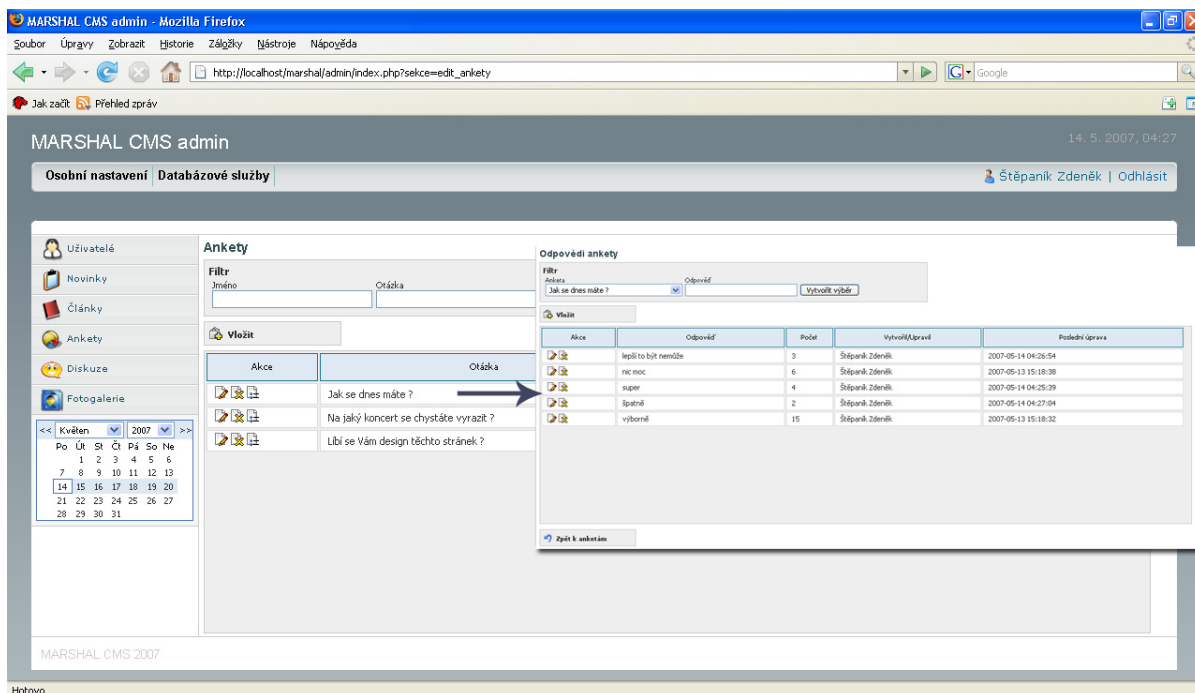


Obr. 3-8: Zobrazení novinek na webové stránce

3.5 Ankety

Ankety jsou v systému implementovány pomocí dvou vázaných tabulek v databázi. První tabulka uchovává otázku ankety a stavovou informaci, zdali je anketa aktivní; druhá tabulka obsahuje odpovědi na anketu (viz. Příloha 2 – Model systému). Při zadávání ankety v rozhraní CMS se nejprve vloží záznam do první tabulky, tedy vyplní se otázka ankety a pomocí zaškrtnávacího políčka se nastaví, zdali je anketa aktivní. Poté se zpřístupní volba nastavení odpovědí na anketu. Obě tabulky jsou mezi sebou provázány pomocí odkazů přes tzv. cizí klíč, který umožňuje tuto vzájemnou vazbu.

Tímto způsobem je možné vkládat neomezené množství možných odpovědí a vytvářet tak univerzální ankety. Obr. 3-9 demonstruje zadávání anket v rozhraní CMS.



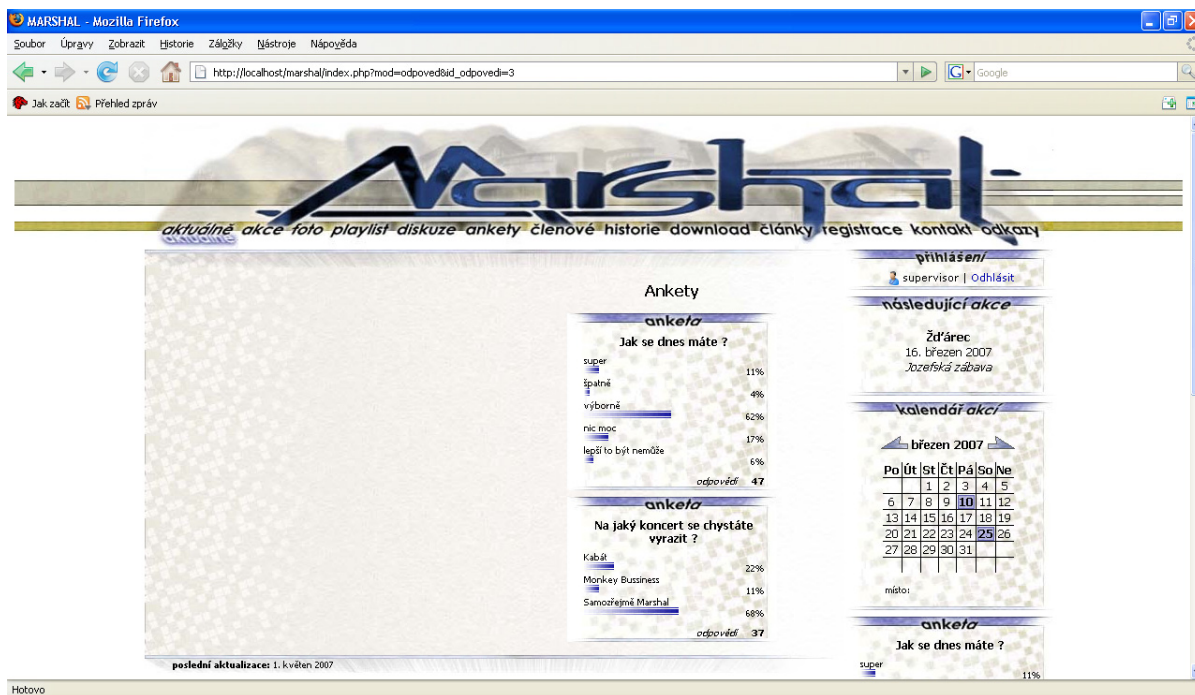
Obr. 3-9: Zadávání anket v rozhraní CMS

Ankety se zobrazují návštěvníkům na webové stránce jednak v kategorii *Ankety*, jednak v pravém panelu, který slouží k přihlašování uživatelů a zobrazování aktuálních informací. Cílem ankety je shromáždit názory fanoušků a zobrazit statistiku jednotlivých odpovědí. Fanoušek může na anketu odpovědět jednoduše kliknutím na jednu z možných odpovědí, server ji pak zaznamená a aktualizuje statistiku.

Zaznamenávání odpovědí je vyřešeno nerestriktivním způsobem, server si tedy neukládá IP adresu respondenta. Tento způsob s sebou sice přináší několik úskalí, jako např. možnost neobjektivní statistiky, avšak nedochází k omezování klientů, kteří vstupují do portálu z neveřejné IP adresy. Pokud by byly ankety implementovány restriktivním způsobem, server by nepovolil odpovědět vícekrát na jednu anketu uživateli, který do portálu vstupuje ze stejné IP adresy. To by vedlo k naprosto objektivním anketám, avšak mohlo by se stát, že pasivní uživatelé (bez veřejné IP adresy) by byli ochuzeni o možnost odpovědět, jelikož jeden pasivní uživatel z lokální sítě by zamezil možnost odpovědět všem ostatním uživatelům stejné lokální sítě.

Důležitým údajem pro systém při generování anket je stavová informace, která určuje, zdali je daná anketa aktivní, nebo ne. Na webové stránce portálu se zobrazí pouze ankety, jejichž stav je aktivní, ostatní generátor anket nebere v potaz.

Zobrazení anket na webové stránce portálu ilustruje následující snímek (Obr. 3-10):



Obr. 3-10: Zobrazení anket na www stránce

3.6 Diskuze

Diskuze umožňuje přihlášeným uživatelům konfrontovat svoje názory a přispívat na témata, která zadal administrátor v rozhraní CMS (viz. Obr. 3-11). Nepřihlášení uživatelé mají možnost jednotlivé příspěvky pouze číst, vkládání reakcí je jim umožněno až po přihlášení. K tomuto účelu slouží formulář v pravém panelu webové stránky. Zde fanoušek zadá své přihlašovací jméno a heslo, pomocí nichž se přihlásí do portálu, pokud je ovšem zaregistrován a má aktivovaný uživatelský účet (viz. Kapitola 3-8 – Registrace fanoušků).

Uživatelé mohou do svých příspěvků vkládat emotikony, jednoduše řečeno „smajlíky“. Tito smajlíci se nacházejí ve složce grafických prvků portálu. Fanouškům se zobrazují v tabulce pod textovým polem pro odpověď, kde stačí kliknout a specifický smajlík se vloží do reakce respondenta.

Diskuze je v systému navržena jako vláknová, což znamená, že uživatel může reagovat na kterýkoliv příspěvek jiného uživatele. To s sebou přináší několik nepříjemností v její implementaci, neboť je třeba rekurzivně procházet databázi. Algoritmů pro implementaci diskuze je spousta, jednodušší z nich bývají většinou málo efektivní, složitější jsou rychlé, avšak jsou komplikované například pro smazání určitého vlákna.

Při realizaci diskuze je potřeba pouvažovat nad tím, jestli bude častou operací vložení příspěvku. Pokud ano, vyplatí se zvolit klasický rekurzivní algoritmus, jelikož ten provádí vložení příspěvku velice rychle. Jestliže se však mnohem častěji provádí výpis diskuze, je lepší použít pokročilý algoritmus, jako např. *Traverzování kolem stromu*, jenž provádí velice rychlý výpis

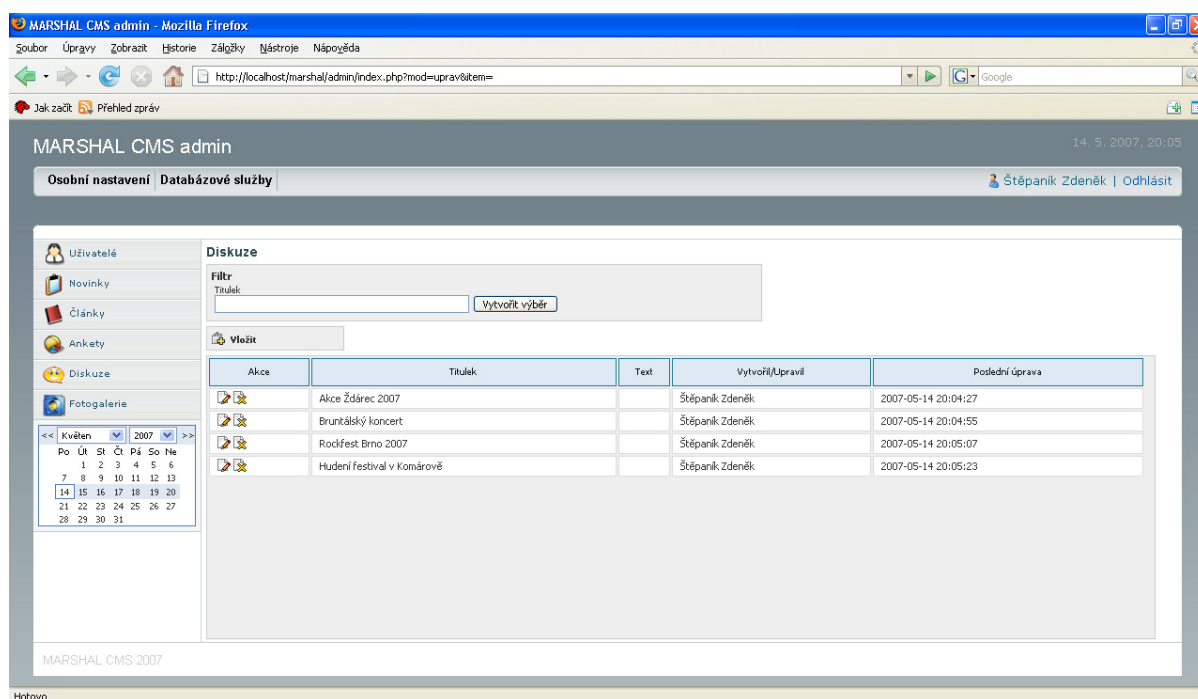
diskuze, je však pomalejší při vkládání a rušení příspěvku. Pokročilý algoritmus vyžaduje také sofistikovanější organizaci databáze, neboť se při něm data ukládají ve stromové struktuře.

Implementaci jsem nejprve prováděl pokročilým algoritmem, avšak nepodařilo se mi jej dotáhnout do zdárného konce. Přešel jsem tedy na rekurzivní algoritmus, který se jeví funkční, avšak může být pomalý při rozsáhlejších diskuzích.

Tento algoritmus vychází z kořenové úrovně diskuze, zobrazí obsah v této úrovni a poté prohledává podúrovně a vypisuje nalezené příspěvky. Do podúrovni se rekurzivně zanořuje tak dlouho, dokud nenalezne prázdné vlákno. Uživatel může reagovat na kterékoliv vlákno, svou reakci vloží do textového pole, které se odešle na server ke zpracování, posléze je vytvořeno nové vlákno s reakcí uživatele.

Vláknové diskuze jsou dnes součástí většiny webových publikačních systémů. Staly se nepsaným pravidlem u webových magazínů. Tyto diskuze však mají jeden společný problém - kvůli nechronologickému řazení může být při opakované návštěvě problém rozpoznat nové příspěvky. Proto některé servery nabízí zobrazení pouze nových příspěvků. Tato funkce ale nebývá dvakrát přátelská – nové příspěvky se zobrazí bez kontextu, takže příspěvky jako „To je nesmysl, já si myslím pravý opak.“ jsou nečitelné.

To může vést k faktu, že některé portály stále používají diskuse „ploché“, bez znázornění vláken, všechny příspěvky pěkně pod sebou. Reakce na příspěvky jsou obvykle znázorněny odkazem na původní příspěvek. Tyto diskuze se vyznačují chronologickým řazením, a tak jsou na první pohled patrné nové příspěvky. Je otázkou programátora portálu, jaké diskuze při implementaci zvolí, já jsem zvolil diskuze vláknové; při případných negativních reakcích lze přejít v následující verzi k „ploché“ variantě. Ukázkovou diskuzi na www stránce portálu zobrazuje Obr. 3-12.



Obr. 3-11: Správa diskuzí v rozhraní CMS



Obr. 3-12: Diskuze na webové stránce portálu

3.7 Fotogalerie

Fotogalerie je součástí systému, jíž jsem věnoval největší pozornost. Je to implementačně nejobsáhlejší komponenta systému, neboť spojuje více technologií dohromady. Je implementována podobně jako ankety pomocí dvou vázaných tabulek v databázi. První tabulka definuje název fotogalerie, rok, v kterém byla vytvořena a počet shlédnutí. Druhá tabulka reprezentuje jednotlivé fotografie (viz. Příloha 2 – Model systému).

Rozhraní CMS obsahuje pro správu fotogalerie tyto moduly: modul pro nahrávání fotografií na server, modul pro převzorkování a vytvoření náhledů a v neposlední řadě modul pro rozvržení fotogalerie.

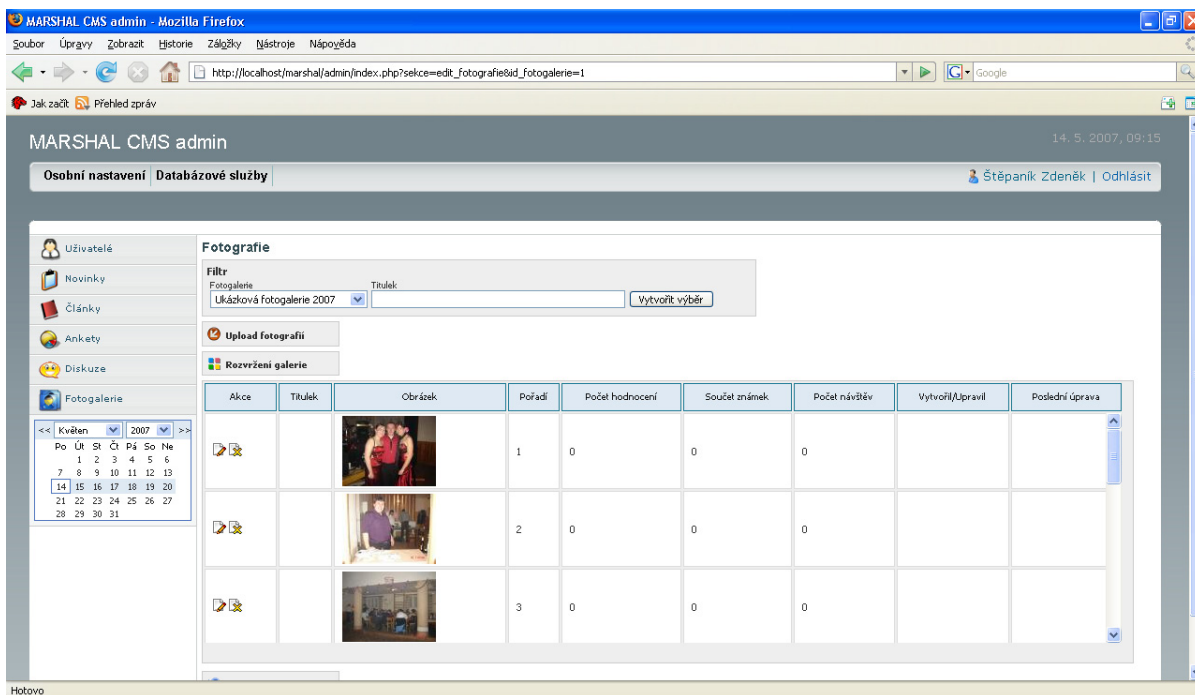
Nahrávání fotografií na server je umožněno pomocí standardního formuláře pro odeslání souboru, administrátor může v jednom kroku odeslat až 10 fotografií. Jakmile přijdou snímky na server, odstraní se z jejich názvů diakritika a jsou umístěny do adresáře fotogalerie, jíž se týkají. Diakritika je odstraněna z důvodu, že některé prohlížeče nedokáží zobrazit snímek, jehož název obsahuje diakritiku nebo nestandardní znaky. Na serveru je toto ošetřeno jednoduchým pravidlem - jestliže jméno souboru obsahuje nestandardní znak, je tento znak nahrazen podtržítkem. Tímto je jednoduše zabráněno chybné interpretaci snímků.

Pro převzorkování fotografií používá systém CMS funkce rozšíření gd2 jazyka PHP. Nejprve je načtena originální fotografie do paměti a poté je převzorkována na cílovou velikost, čímž vznikne

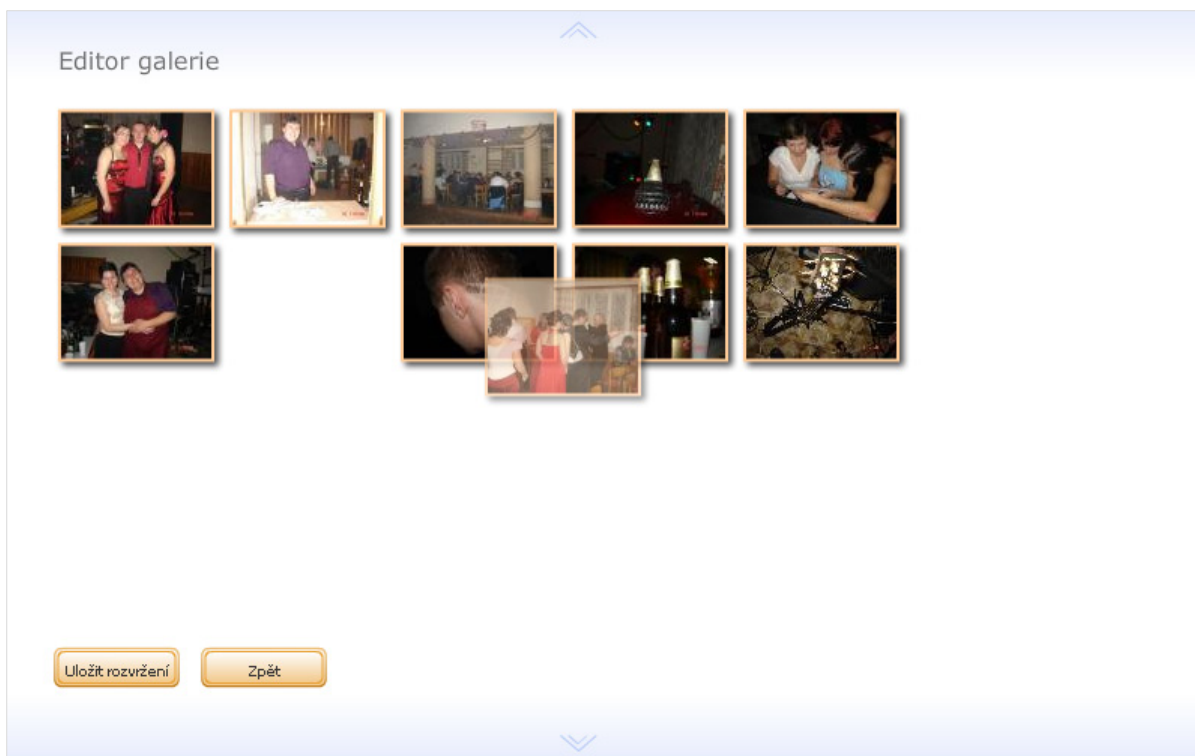
zmenšenina (thumbnail) původní fotografie. Zmenšenina se uloží do podadresáře fotogalerie a poslouží jako rychlý, datově neobjemný náhled originálního snímku.

Modul pro rozvržení fotogalerie slouží administrátorům k organizaci snímků ve fotogalerii pomocí metody drap&drop. Jedná se o jediný modul v systému, jenž používá progresivní technologii Flash. Samotná Flash aplikace, která umožňuje organizaci fotogalerie, se nachází v souboru *galleryEditor.swf* v adresáři CMS rozhraní portálu. Jedná se o zkompileovanou verzi, zdrojový soubor se jmenuje *galleryEditor fla*. Systém s Flash aplikací komunikuje pomocí konektoru napsaném v jazyce PHP, který jí předává informace o snímcích.

Aplikaci řídí skript v jazyce ActionScript, který načte data předané konektorem, zobrazí snímky, rozmístí je dle pořadového indexu a poté očekává, že uživatel přesune zobrazené snímky pomocí drag&drop tak, aby odpovídaly požadovanému rozmístění. Jakmile administrátor dokončí rozmístění snímků, potvrdí akci kliknutím na tlačítko *Uložit rozvržení*. Flash aplikace vygeneruje data o rozvržení, která pošle pomocí metody POST ke zpracování do CMS systému. Ten aktualizuje databázová data, čímž se rozvržení fotogalerie uloží. Stejným způsobem, jakým byly snímky ve Flash aplikaci rozmístěny a následně uloženy, se zobrazí i na webové stránce fanouškům. Ti zde mají možnost vkládat své komentáře a hodnotit jednotlivé snímky stupnicí 1-5. Pod každým snímkem se zobrazuje grafický pruh, jenž znázorňuje hodnocení daného snímku. Následující snímky dokumentují práci s fotogalerií v rozhraní CMS (Obr. 3-13), použití Flash modulu pro rozvržení (Obr. 3-14) a zobrazení galerie na webové stránce portálu (Obr. 3-15):



Obr. 3-13: Práce s fotogalerií v rozhraní CMS



Obr. 3-14: Editor rozvržení fotogalerie



Obr. 3-15: Zobrazení fotogalerie na webové stránce portálu

3.8 Registrace fanoušků

Registrovat se do portálu je fanouškům umožněno v sekci *Registrace* na webové stránce portálu. Modul registrace používá k vygenerování formuláře pro registraci fanouška komponenty rozhraní CMS, o nichž jsem se zmínil v kapitole 3-1. Konkrétně se jedná o modul *main*, jehož funkce jsou použity pro generování formulářových prvků korespondujících s prvky databázové tabulky registrace.

Tato tabulka obsahuje následující atributy: *Login*, *Heslo*, *Jméno*, *Příjmení*, *Email* a *Aktivační klíč*. Uživatel se při vstupu do kategorie *Registrace* zobrazí všechny atributy, kromě posledního, aktivačního klíče, navíc je přidán atribut pro ověření hesla. Tyto atributy jsou povinné a uživatel je musí všechny vyplnit. Po vyplnění je formulář odeslán na server, kde jsou jednotlivé prvky zkontrolovány, ověří se email, shodnost hesla a jeho ověření a pokud je vše v pořádku, server vygeneruje unikátní aktivační klíč. Klíč je vložen k ostatním údajům a server vytvoří záznam o registraci v databázi.

Tímto však uživatel není registrován resp. jeho účet není aktivován. Server po vytvoření záznamu v databázi vygeneruje emailovou zprávu obsahující aktivační klíč a odešle ji na email, který uživatel uvedl při registraci. Jestliže fanoušek zadal korektní email, může si přečíst novou zprávu, v níž je odkaz s aktivačním klíčem, který dokončí aktivaci nového uživatelského účtu.

Tento ověřovací mechanismus je v internetových aplikacích téměř vždy použit, jelikož jím server předchází zahlcení databáze nesmyslnými údaji. K vygenerování aktivačního klíče používá systém funkci *uniq_id*, která vrací unikátní hodnotu. Tato hodnota je následně zašifrována hašovací funkcí MD5, a tak vznikne unikátní klíč se 128-bitovou šifrou.

Samotná aktivace účtu probíhá tím způsobem, že je vytvořen nový záznam v tabulce uživatelů, vloží se zde nový uživatel kategorie *fan* (fanoušek) a poté je odstraněn záznam v tabulce registrace. Při aktivaci je uživatel v portálu automaticky přihlášen. Získává tím nové možnosti, jako např. možnost psát příspěvky do diskuze a komentovat fotografie.

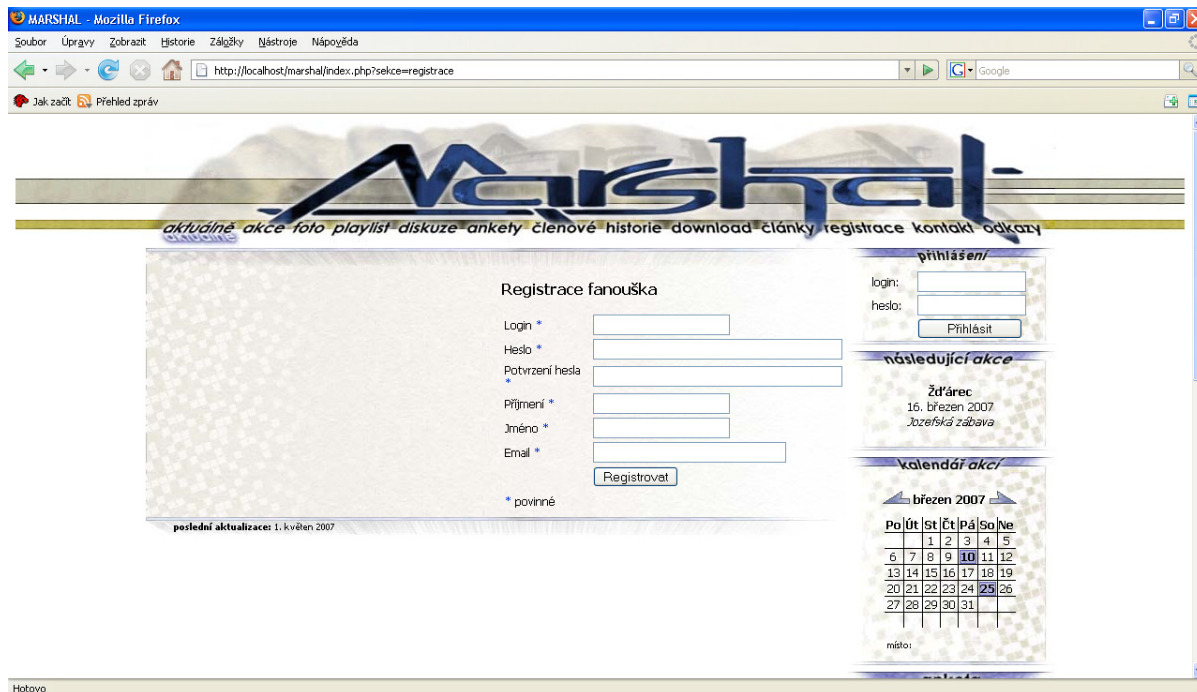
Přihlašování a odhlašování je po stránce implementační vyřešeno použitím *session* proměnných jazyka PHP. Jelikož je protokol HTTP bezstavový, tzn., že neumožňuje mezi jednotlivými přenosy uchovávat stav, vznikly v PHP právě *session* proměnné, což jsou struktury vázané na jednoznačný identifikátor prohlížeče, které umožňují přenášet stavovou informaci.

Stavové informace systém využívá při konstruování reakce na další vstup od uživatele, je to dáno tím, že portál může používat více uživatelů najednou. Je tudíž třeba jednak rozlišovat jednotlivé uživatele aplikace a jednak pro každého uživatele zvlášť udržovat potřebné stavové informace.

Uživatelův prohlížeč používá pro uchování stavové informace tzv. *cookies*. Mechanismus cookies je rozšířením protokolu HTTP, které bylo navrženo k uchování stavových informací na straně klienta.

Jediný problém může nastat, pokud si uživatel ve svém prohlížeči zakáže používání cookies, pak systém není schopen uživatele identifikovat a neumožní mu přihlášení do portálu, ani když je řádně zaregistrován a má aktivovaný účet.

Následující snímek zobrazuje formulář registrace na www stránce portálu (Obr. 3-16):



Obr. 3-16: Formulář registrace fanouška

4 Závěr

Předložená práce zmapovala technické a strukturální aspekty vyvinutého internetového portálu hudební skupiny Marshal, nastínila jeho principy, osvětlila možnosti, jakými lze využít, podala praktickou ukázkou vytvoření stabilní platformy pro jeho provoz a integraci a zasadila jej po stránce technologické do širšího kontextu.

Svou práci jsem již od počátku vývoje soustředil tím směrem, aby bylo možné obsah portálu jednoduchým, uživatelsky příjemným, ale zároveň velice efektivním způsobem spravovat a aktualizovat, což vedlo k vytvoření univerzálního systému CMS, jenž se programově vymyká náplni této bakalářské práce a na jehož vývoji jsem strávil nejvíce času. Tento vývoj mne však obohatil o nové zkušenosti a dokonce předčil má očekávání, jelikož o rozšířenou verzi tohoto systému projevil zájem brněnské společnosti IpiSoft Computers s.r.o. a Infram s.r.o., aby byl integrován do jejich informačního systému a intranetové podnikové agendy.

Portál jsem se snažil navrhnout s ohledem na flexibilitu, o tom svědčí například skutečnost, že používá vlastní mechanismus pro přístup k databázi s podporou ODBC, pokud by se tedy administrátor rozhodl v budoucnu přenést portál na jinou databázovou platformu, jednalo by se o záležitost s minimálním zásahem do zdrojového kódu. V širším časovém měřítku by se dala na portálu provést řada vylepšení, jako např. streamované audio a videoukázky, interaktivní nástěnka či chat s využitím technologie Flash; to už je však nad rámec této práce, tyto nápady uplatním v případném pokračování tohoto projektu – diplomové práci.

Literatura

- [1] Kosek, J. *PHP – tvorba interaktivních webových aplikací*. Praha, Grada Publishing 1996.
- [2] Kolektiv autorů. *PHP5, MySQL, Apache*. Praha, Computer Press 2006.
- [3] Gutmans, A. *Mistrovství v PHP5*. Praha, Computer Press 2005.
- [4] Williams, H. E., Lane, D. *PHP a MySQL – vytváříme webové databázové aplikace*. Praha, Computer Press 2002.
- [5] DuBois, P. *MySQL profesionálně*. Praha, Computer Press 2003.
- [6] Wikimedia Foundation, Inc. *Wikipedia, the free encyclopedia*.
<http://en.wikipedia.org>

Seznam použitých zkratek a symbolů

IS	<i>Information System</i>
CMS	<i>Content Management System</i>
PHP	původně <i>Personal Home Pages</i> , nyní <i>Hypertext Preprocessor</i>
SQL	<i>Structured Query Language</i>
MySQL	<i>My Structured Query Language</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
WWW	<i>World Wide Web</i>
URL	<i>Uniform Resource Locator</i>
CGI	<i>Common Gateway Interface</i>
SSI	<i>Server Side Includes</i>
JVM	<i>Java Virtual Machine</i>
MSIE	<i>Microsoft Internet Explorer</i>
DHTML	<i>Dynamic HTML</i>
SSJS	<i>Server Side JavaScript</i>
ASP	<i>Active Server Pages</i>
FI	<i>Form Interpreter</i>
ODBC	<i>Open DataBase Connectivity</i>
DBMS	<i>Database Management System</i>
CSS	<i>Cascade Style Sheets</i>
WYSIWIG	<i>What You See Is What You Get</i>
ER	<i>Entity Relationship</i>
MD5	<i>Message-Digest Algorithm 5</i>
SSL	<i>Secure Sockets Layer</i>
XML	<i>Extensible Markup Language</i>

Seznam příloh

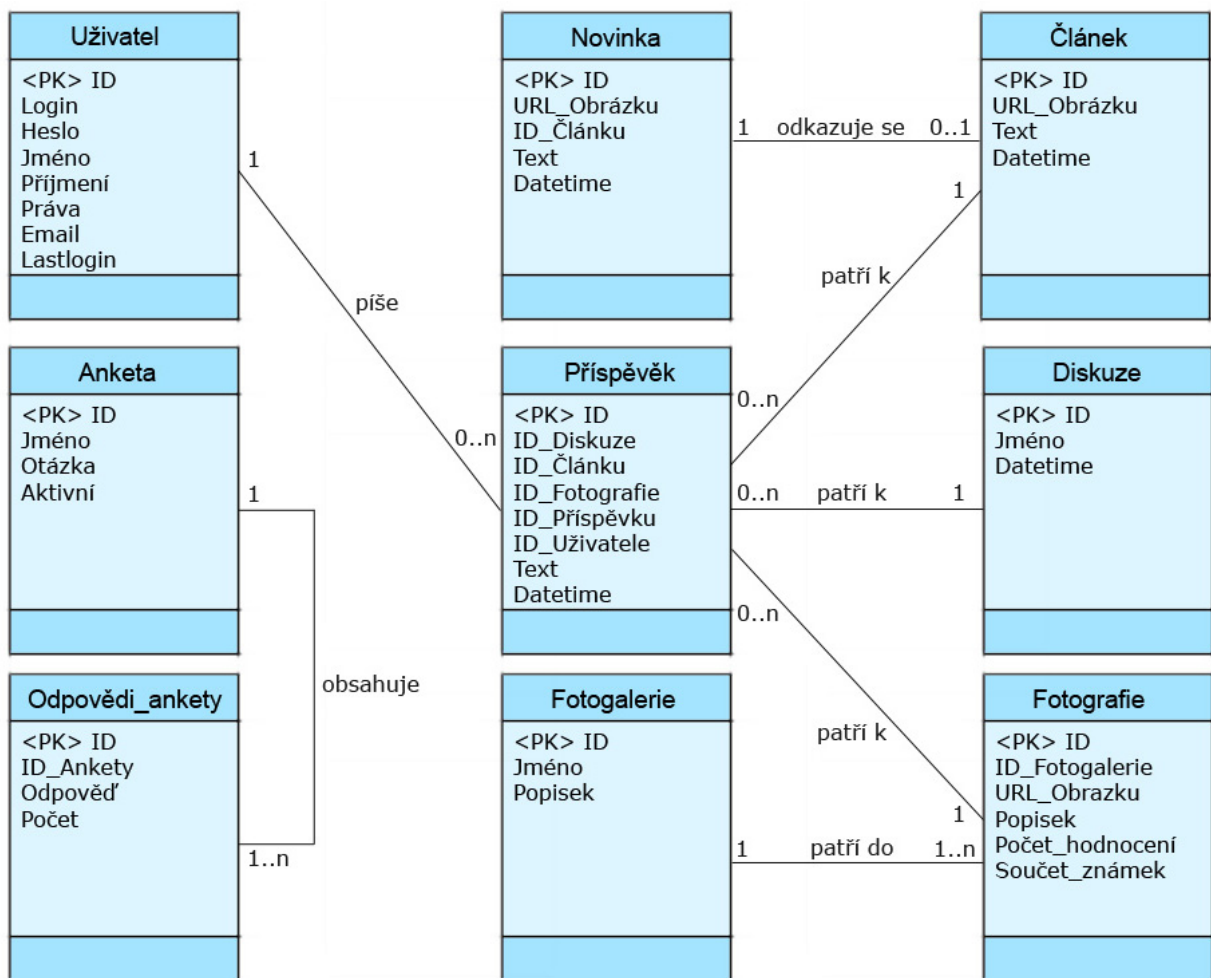
Příloha 1. CD-ROM

Příloha 2. Model systému – ER Diagram

Příloha 3. Model použití – Use Case Diagram

Příloha 2

Model systému – ER Diagram



Příloha 3

Model použití – Use Case Diagram

