

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INFORMAČNÍ SYSTÉM PRO SPRÁVU OBSAHU WEBOVÝCH PREZENTACÍ

BAKALÁŘSKÁ PRÁCE

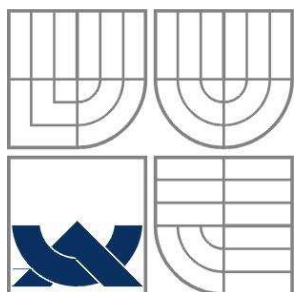
BACHELOR'S THESIS

AUTOR PRÁCE

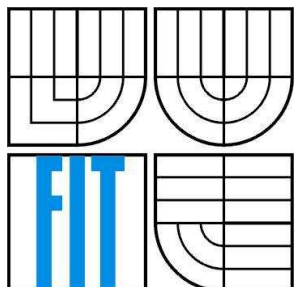
AUTHOR

Zbyněk Šulc

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

INFORMAČNÍ SYSTÉM PRO SPRÁVU OBSAHU WEBOVÝCH PREZENTACÍ

WEB PRESENTATIONS CONTENT MAINTENANCE INFORMATION SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Zbyněk Šulc

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. Tomáš Hruška, CSc.

BRNO 2007

Abstrakt

Předmětem práce je návrh a realizace systému pro správu obsahu webové prezentace. Tento systém má umožnit vytvoření a správu webové prezentace osobám, které nemají zkušenosti s technologiemi pro tvorbu webové prezentace. Systém dále umožňuje snadné přidávání komponent rozšiřující počet implementovaných funkcí. Tyto funkce pak mohou přidělovány na základě uživatelských skupin.

Klíčová slova

webová prezentace, informační systém, internet, HTML, správa obsahu

Abstract

The object of this elaborate is a concept for realization of system for web site administration. This system needs to be easy to use for people with no experience in web page program coding. Another side of this software is a possibility to supplement other extensions. The function of this system is assigned to users sorted by groups these users belong to.

Keywords

web presentation, information system, internet, HTML, content maintenance

Citace

Zbyněk Šulc: Informační systém pro správu obsahu webových prezentací, bakalářská práce, Brno, FIT VUT v Brně, 2007

Informační systém pro správu obsahu webových prezentací

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením prof. Ing. Tomáše Hrušky, CSc.

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

.....
Zbyněk Šulc
11.5.2007

Poděkování

Děkuji vedoucímu bakalářské práce prof. Ing. Tomáši Hruškovi, CSc., za odborné vedení, ochotu při řešení problémů, cenné rady a připomínky při zpracování a řešení bakalářské práce.

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.....	3
1.1 Stav problematiky.....	3
1.1.1 Web, HTML, CSS.....	3
1.1.2 Serverové skriptovací jazyky.....	4
1.1.3 Klientské skriptovací jazyky.....	4
1.1.4 AJAX.....	5
1.1.5 SEO.....	5
1.1.6 Znakové sady.....	6
1.1.7 Zabezpečení.....	6
1.1.8 Databáze.....	7
1.1.9 Webové prezentace.....	7
1.2 Uživatelské rozhraní.....	8
1.3 Zhodnocení.....	9
2 Cíl práce.....	10
3 Řešení.....	11
3.1 Výběr softwaru.....	11
3.1.1 Operační systém.....	11
3.1.2 Webový server.....	11
3.1.3 Skriptovací jazyk.....	12
3.1.4 Databázový systém.....	13
3.1.5 Hosting a doména.....	14
3.2 Model systému.....	14
3.2.1 Komponenta.....	15
3.2.2 Třída databáze.....	16
3.2.3 Počítadlo přístupů.....	17
3.2.4 Uživatelé a uživatelské skupiny.....	18
3.2.5 Login.....	19
3.2.6 Menu.....	20
3.2.7 Šablony.....	21
3.2.8 Zobrazení zpráv.....	22
3.2.9 Zpracování formulářů.....	23
3.3 Implementované funkce.....	25
3.3.1 Správa uživatelů.....	25

3.3.2	Nahrávání souborů	25
3.3.3	Správa galerie.....	26
3.3.4	Správa obsahu	26
3.3.5	Ankety a jejich správa	27
3.3.6	Správa odkazů	28
3.3.7	Vzkazy.....	28
3.3.8	Zaslání zprávy	29
	Závěr.....	31
	Literatura	32
	Seznam příloh	33

Úvod

Webové prezentace umožňují prezentaci firmám, organizacím nebo jednotlivým osobám na internetu. V síti Internet je pomocí vyhledávacích strojů možné velice efektivně vyhledávat ve velkém množství dokumentů. Dokumenty se ve výsledku řadí podle kvality, takže velice záleží na způsobu jejich provedení. Ne však ale každá firma si může dovolit zaměstnávat zaměstnance, který se bude starat pouze o prezentaci firmy na internetu. Cílem této práce je vytvořit systém, který bude sloužit ke správě webové prezentace a bude ho schopný používat i poučený laik.

Uživatelské prostředí správy systému by mělo být co nejjednodušší a intuitivní na ovládání. Součástí správy systému by měl být textový editor pro jednoduchou editaci obsahu prezentace.

1.1 Stav problematiky

Webové prezentace jsou spolu s e-mailem bezesporu nejvíce využívanými službami na internetu od jeho zrození. Za vynálezce webu je považován *Sir Timothy John Berners-Lee*, který v roce 1990 sloučil protokol TCP/IP, službu DNS a technologii Hypertext.

1.1.1 Web, HTML, CSS

World Wide Web, neboli zkráceně web je označení pro celosvětově propojené hypertextové dokumenty. Jednotlivé dokumenty se odkazují na svoje vlastní části a nebo na dokumenty umístěné na úplně jiném serveru. První web umožňoval na jednom počítači prohlížet dokumenty, které byly uloženy na počítači jiném. Tyto dokumenty byly mezi sebou provázané pomocí technologie hypertextu, tzv. odkazy. Dokumenty se mezi počítači vyměňovaly pomocí aplikačního protokolu HTTP, který fungoval nad transportním protokolem TCP/IP. Protokol TCP/IP slouží jako transportní protokol k přenosu dat mezi klienty a serverem. Aby mohli uživatelé jednoduše určit, se kterým serverem se chtějí spojit, umožňuje jim služba DNS zadat adresu serveru v podobě textu namísto IP adresy složené z čísel.

K tvorbě dokumentů pro web byl vytvořen jazyk HTML, který patří do podmnožiny jazyka SGML (*Standard Generalized Markup Language*). Jazyk je složen z množství značek, tzv. tagů, které vymezují části obsahu dokumentu, nebo ovlivňují formátování prvků dokumentu. Jazyk HTML měl postupem času několik verzí. Každá verze přidávala do jazyka nějaké nové prvky, jako například grafiku nebo formuláře. O něco později se objevilo několik dalších technologií pro vylepšení dosavadního způsobu vytváření HTML dokumentů, viz. [2].

Jelikož základní HTML obsahuje jak obsah dokumentu, tak i jeho formátování, vznikl nový jazyk CSS (*Cascading Style Sheets*) [4], který umožňuje na HTML značky navazovat jejich formát. Tím se dosáhlo oddělení formátování obsahu dokumentu od vlastního textu dokumentu. Formátování

dokumentu je potom uloženo ve speciálním souboru, na který se potom z těla dokumentu odkážeme. Tento soubor obsahuje tzv. selektory, které umožňují navázat na jednotlivé prvky v dokumentu jejich formát. Propojení formátu a prvku je zajištěno pomocí identifikátoru *id*, který je pro každý prvek unikátní, podle parametru *class*, který řadí prvek do skupiny prvků se stejným formátem, a nebo přímo podle jména HTML značky. Výhodou tohoto řešení je možnost nahrazení jednoho souboru, který obsahuje popis vzhledu stránky, a tím změna vzhledu celé stránky.

V dnešní době se pomalu přechází z normy HTML na novější normu XHTML, která vypouští některé značky a nutí programátora k tomu, aby popis formátování obsahu vyčlenil mimo vlastní text dokumentu.

1.1.2 Serverové skriptovací jazyky

Samotný jazyk HTML však umožňuje vytváření pouze statického obsahu. Aby bylo možné do dokumentu vkládat i obsah, který není přímo součástí HTML dokumentu, vytvořil *Rasmus Lerdorf* skriptovací jazyk PHP.

Jazyk PHP (*Hypertext Preprocessor*) [3] umožňuje dynamicky, na základě interakce s návštěvníkem generovat obsah stránky. Jazyk PHP je vyvíjen jako Open Source, z toho plyne, že si můžete stáhnout jeho zdrojové kódy a upravit si je podle vlastních představ. PHP programátorovi umožňuje na straně serveru provádět úkony, které by na straně klienta nebylo možné provést a nebo by nebyly bezpečné. Klasickým příkladem je odeslání oznámení o přijetí objednávky e-mailem.

Podobnou technologií jako PHP je komerční produkt od firmy Microsoft s názvem ASP (*Active Server Pages*). Oba jazyky dělají v podstatě to stejné, s tím rozdílem, že ASP je určeno pro servery na platformě operačních systémů Microsoft, kdežto PHP je určeno spíše pro operační systémy typu Unix (v dnešní době Linux, nebo BSD). Jak PHP, tak i ASP je možné provozovat na operačních systémech Windows a Linux. Nejčastěji používanými modely je ale spojení skriptovacího jazyka PHP, databázového serveru MySQL a webového serveru Apache na operačním systému Linux a skriptovací jazyk ASP, databázový systém MS SQL a IIS webový server na operačním systému Windows.

1.1.3 Klientské skriptovací jazyky

Zatímco PHP a ASP jsou jazyky, které se zpracovávají na straně serveru, jazyk JavaScript je určený pro zpracování na straně klienta. Může tak převzít určitou funkcionalitu na sebe a ulehčit tak zatížení serveru. JavaScript je určen spíše pro interakci s uživatelem, různé grafické projevy, reakce na vstup uživatele, apod. Jeho funkce jsou hodně omezené kvůli bezpečnosti. JavaScript například nemá možnost přístupu k souborovému systému klientského počítače. Jedinou možností, jak si uchovat nějaká data u klienta, poskytují takzvaná cookies, což jsou soubory uložené v určité složce. Tyto

soubory mají omezenou velikost - přibližně 4kB, v závislosti na prohlížeči. Další informace je možné najít na stránkách *jak psát web* viz. [1].

Název JavaScript byl spíš reklamní trik, JavaScript totiž nemá s programovacím jazykem Java vůbec nic společného. JavaScript se používá kromě kontroly formulářů také k asynchronnímu načítání a odesílání dat na server. Této technologii se říká AJAX.

Podobný, ale v praxi méně užívaný jazyk je VBS (*Visual Basic Script*) vyvinutý firmou Microsoft.

1.1.4 AJAX

Technologie AJAX je odpovědí na některé, ale bohužel ne všechny nedostatky protokolu http. V současné době, se po webových aplikacích požaduje stejná funkcionalita jako po desktopových aplikacích, ovšem k tomuhle nebyla nikdy tato dvojice určena. Zkratka AJAX se překládá jako asynchronní JavaScript a XML. V podstatě to znamená, že jakoby za zády prohlížeče provádíme další volání. Vytvářet další požadavky umožňuje objekt *XMLHttpRequest* a používá se k získání nebo odeslání nějakých dat, která potom zobrazíme u klienta, nebo zpracujeme na serveru. Tento objekt umožňuje komunikovat se serverem, aniž by o tom uživatel musel vědět. Výhoda tohoto řešení je v tom, že nemusíte, kvůli relativně malé změně v těle dokumentu, obnovovat celou stránku. Tím se šetří jak čas, tak i množství dat, která se musí mezi klientem a serverem přenášet.

V praxi se tato technologie používá hlavně k vylepšení kvality uživatelského rozhraní. Umožňuje programátorovi doplnit uživatelské rozhraní o prvky, které by bez použití technologie AJAX vyžadovaly odeslání údajů na server při každé změně. To by bylo nákladné jak na čas, tak i na přenos dat. Více o technologii AJAX se můžete dočíst v knížce o AJAXu a PHP [5].

1.1.5 SEO

Search Engine Optimization, optimalizace pro vyhledávače. V současné době hodně diskutovaná záležitost. Jedná se o způsob návrhu struktury stránky tak, aby byla co nejjednodušeji srozumitelná vyhledávači. Vyhledávače používají pro indexování stránek na internetu tzv. roboty. Tito roboti procházejí internetové stránky a hodnotí jejich obsah, aby je později mohli seřadit podle nějakého parametru. Tímto parametrem je právě kvalita obsahu stránky. Aby mohl robot určit kvalitu obsahu stránky, musí ji projít a získat z ní pro něj významná místa.

Většina robotů vyhledává informace uvedené v *title*, *h1*, *hlavičce*, odkazech a prvních pár řádků v *body*. Každý vyhledávač má potom nějaký svůj vzorec, do kterého dosadí získané informace, a z něj pak vyjádří hodnotu obsahu ve stránce. Největší význam mají údaje uvedené v *title*, potom v hlavičce *description* a v nadpise první úrovně *h1*. Zbytek dokumentu má vedlejší význam závislý na kvalitě obsahu stránky. Kvalitou obsahu stránky se rozumí počet výskytu klíčových slov v textu stránky, počet odkazů na jiné zdroje a počet stránek, které se odkazují na naši stránku.

Pokud provedeme dobrou optimalizaci pro vyhledávání, získáme tak velkou šanci umístit se na předních místech ve vyhledávacích a z toho plynoucí šanci, že naši stránku navštíví více návštěvníků, kteří hledají přesně to, co jim naše stránka nabízí.

1.1.6 Znakové sady

Další věc, která hraje při vývoji webové prezentace velkou roli, je znaková sada, ve které je dokument vytvořen. Při vzniku prvních počítačů bylo potřeba nějak zakódovat znaky abecedy. První počítače pracovaly pouze v anglickém jazyce a využívaly kódování ASCII na sedmi bitech. Tato tabulka měla 128 znaků a obsahovala pouze anglickou abecedu. Po rozšíření počítačů bylo potřeba implementovat kromě anglické abecedy také abecedy ostatních zemí, jako například Češskou, Ruskou, Polskou, atd. Bohužel každá velká firma, působící ve výpočetní technice, si vytvořila vlastní kódovací tabulku a vzniklo několik způsobů, jak kódovat jednotlivé abecedy. Pro český jazyk je to například tabulka *windows-1250* od firmy Microsoft nebo norma *ISO-8895-2* (Latin2).

V poslední době se usiluje o sjednocení všech abeced do jednoho kódování, nazvaného *Unicode*. Tento kód je šestnácti bitový a obsahuje veškeré světové abecedy včetně největších asijských abeced.

Základní otázkou při vytváření webové prezentace je, pro koho danou prezentaci vytváříme. Pokud se jedná o malou prezentaci pro malý okruh uživatelů, můžeme zvolit znakovou sadu typickou pro daný okruh uživatelů, např. pro webovou prezentaci ruské autodílny bychom nejspíše použily normu *ISO-8895-1* (Latin1). Pokud ale počítáme s tím, že prezentaci budou navštěvovat uživatelé z různých států, měli bychom zvolit znakovou sadu *Unicode*.

Může se stát, že v hlavičce dokumentu zapomeneme uvést, v jaké znakové sadě je dokument napsán, to potom může vést ke špatnému zobrazení návštěvníkovy stránky. Záleží to na znakové sadě, kterou má návštěvník nastavenou ve svém prohlížeči jako výchozí. Pokud nebude shodná s tou, v jaké je dokument napsán, tak tato chyba znemožní přečtení textu a návštěvník nám ze stránek odejde, i když možná obsahovaly přesně to, co hledal.

1.1.7 Zabezpečení

Jelikož informační systém většinou obsahuje citlivá data, potřebujeme zajistit, aby k určitým informacím přistupovali pouze ověřeni uživatelé. Tento problém se dá řešit několika způsoby, z nichž každý má své výhody, nevýhody a každý poskytuje jinou úroveň zabezpečení. V kapitole 3.1 *výběr softwaru* jsou stručně popsány nejpoužívanější webové servery. Každý s těchto serverů řeší problematiku přihlašování trochu odlišným způsobem. Zatímco u serveru Apache existuje soubor uchovávací přístupová práva uživatelů a jejich hesla, u serveru IIS se povolení přístupu rozhoduje na základě uživatelského účtu přímo v operačním systému Windows.

Dalším způsobem je řízení přístupů uživatelů do systému pomocí vlastních pravidel. Abychom mohli kontrolovat přístup uživatelů do určitých částí systému, je nutné, aby každý uživatel měl vlastní heslo a přihlašovací jméno. Pohle složitosti systému je možné uživatele organizovat do uživatelských skupin a jim pak přidělovat práva k částem systému.

Důležitou věcí je také způsob uložení hesel. V některých systémech je používáno uložení hesel přímo v textové podobě, což je asi ten nejhorší možný způsob. Lepší úroveň nabízí uložení hesla v zakódované podobě. Asi nejpoužívanější algoritmus pro jednosměrné zakódování je algoritmus MD5. Tento pak vrací třicetidvouznakový otisk zadaného řetězce.

1.1.8 Databáze

Pro uložení dat, ze kterých později budeme sestavovat obsah prezentace, se většinou používají databáze. Je možné použít ukládání dat do souborů, ale to se již využívá pouze tam, kde není možné využít nějaký databázový systém. Použití databázového systému pro uložení dat prezentace je výhodné především kvůli snadným operacím nad takto uloženými daty. Databázový systém umožňuje pomocí jazyka SQL (*Structured Query Language*) jednoduchou práci s uloženými daty. Jednoduché je tu obzvláště vyhledávání, řazení, vkládání a mazání. Implementace těchto operací nad soubory bývá většinou poměrně náročná.

1.1.9 Webové prezentace

Webovou prezentací rozumíme soubor dokumentů, přístupných pomocí internetového prohlížeče. Prezentace bývají vytvořeny většinou pomocí jazyka HTML, toto ale nemusí být vždy pravda. Objevují se i prezentace vytvořené v programu Flash nebo napsané v jazyku Java.

Společné pro všechny prezentace je to, že by měly obsahovat nějaký navigační prvek. V klasických prezentacích se jedná o menu, které umožňuje návštěvníkovi pohyb mezi jednotlivými dokumenty. Moderní prezentace umožňují designérům použít nepřeberné množství druhů fontů písma, grafiky a animací. Ovšem grafický design by neměl svojí kvalitou a provedením odpoutávat od samotného obsahu prezentace. Informace uvedené v prezentaci by měly být vždy považovány za hlavní.

Pokud chceme omezit přístup návštěvníků do určitých částí prezentace, musíme na nějakém místě umožnit návštěvníkovi jeho ověření. Nejčastějším řešením je formulář s dvojicí políček *jméno* a *heslo*. Po zadání správných přihlašovacích údajů je umožněn návštěvníkovy přístup do chráněné části prezentace. V chráněné části prezentace je pak přihlášenému uživateli umožněno např. editovat obsah prezentace.

Mezi další prvky, které je možné najít ve většině současných prezentací, patří například reklama, anonymní ankety, RSS a ATOM novinky, mapy prezentací a políčko pro vyhledávání v dokumentech zveřejněných v prezentaci.

1.2 Uživatelské rozhraní

Návrh uživatelského rozhraní prezentace je jednou z nejdůležitějších částí realizace systému. Pokud nebude uživatelské rozhraní jednoduché a uživatelsky přívětivé, většina návštěvníků dá přednost prezentaci, se kterou se jim bude lépe pracovat.

Při profesionálním návrhu celé prezentace se jeden člověk stará o návrh grafického uživatelského rozhraní a o návrh programu se zase stará někdo jiný. Ze zkušenosti víme, že jeden člověk není schopen obsáhnout jak kvalitní zpracování grafiky, tak i vlastního programu.

Menu by mělo umožňovat jednoduchou a logickou navigaci pro prezentaci. Položky menu by měly co nejpřesněji, ale také co nejstručněji vyjadřovat, co se pod nimi skrývá. U rozsáhlých prezentací se využívají menu, která mají několik úrovní. Počet úrovní by však neměl přesáhnout 3 úrovně, potom se menu stává nepřehledné a návštěvník se v něm lehce ztratí. Menu je jednou z oblastí, kde je hojně uplatňována optimalizace pro vyhledávání. Mezi hlavní položky menu je dobré umisťovat nejvíce vyhledávané části prezentace.

K pohybu mezi jednotlivými dokumenty se vedle menu používají hypertextové odkazy. Tyto odkazy by měly být odlišeny od okolního textu, aby je návštěvník mohl co nejjednodušeji najít a použít. Nejčastější způsob, jak se v prezentacích označují odkazy je podtržení nebo zvýraznění jinou barvou, než má okolní text.

Pro zadávání dat uživatelem se používají formuláře, které se odesílají ke zpracování na server. Formuláře obsahují políčka, do kterých je možné zadat jakýkoliv text, seznamy, ze kterých je možno vybrat některou z předvolených hodnot, nebo zaškrtačací políčka. Protože ale uživatel může do formuláře zadat v podstatě cokoliv, je důležité hodnoty zkontrolovat. Kontrolu zadaných hodnot je možné provést v prohlížeči pomocí JavaScriptu, což umožňuje návštěvníkovi okamžitou opravu chyby, nebo provést kontrolu na straně serveru. Kontrola na straně serveru by se měla provádět pokaždé, protože kontrola pomocí JavaScriptu nemusí fungovat, pokud je v prohlížeči vykonání JavaScriptu vypnuté, nebo podpora JavaScriptu přímo chybí.

Pokud uživatel zadá nesprávnou hodnotu, měl by na to být upozorněn. Nemělo by se stát, že kvůli jedné špatně zadané hodnotě bude muset návštěvník vyplnit celý formulář znovu. Tuto funkci řeší tzv. strnulé formuláře, které uchovávají zadané hodnoty do doby, než návštěvník zadá všechny hodnoty ve správném tvaru a server je zpracuje.

Poslední důležitou vlastností je dobře zvolený formát textu. Zápis vlastního textu by měl vycházet z typografických zásad. Text by měl být dobře čitelný, tzn. měl by mít dostatečnou velikost, aby ho měli možnost přečíst i návštěvníci s horším zrakem nebo přímo osoby s lehčí vadou zraku. A nakonec zvolené barvy podkladu a textu by měly být v dobrém kontrastu.

1.3 Zhodnocení

Při návrhu informačního systému je důležitá analýza požadavků zákazníka. Při analýze požadavků je nejdůležitější porozumět tomu, co zákazník od systému požaduje. Návrh informačních systémů zahrnuje spoustu oblastí z oboru informačních technologií. Programátor informačního systému musí zvládnout:

- Návrh databáze pro uložení dat
- Skriptovací jazyk na straně serveru
- Skriptovací jazyk na straně klienta
- Jazyk HTML pro popis dokumentu
- Jazyk CSS pro popis formátování výsledných dokumentů
- Jazyk XML pro popis dat, pro komunikaci technologií AJAX, nebo realizaci RSS novinek
- Znalost vyhledávačů kvůli optimalizaci SEO
- Konfigurace webového a databázového serveru

2 Cíl práce

Základním cílem práce je vytvořit informační systém portálového typu, který se bude skládat z komponent, které by bylo možné do systému jednoduše vkládat na základě požadavků zákazníka. Tyto komponenty by systém doplňovaly o další funkce, jako je například e-shop, katalog výrobků apod. Vytváření komponent by mělo být pro programátora co nejjednodušší, aby mu umožnilo provádět změny v co nejkratším čase.

Dalším požadavkem je možnost jednoduchého vytváření obsahu webové prezentace pro uživatele, kteří nemají zkušenost se značkovacími jazyky HTML a jsou zvyklí na tvorbu textu v prostředí MS Word. Umožní jim vkládání obrázků do textu, připojování souborů k obsahu prezentace a organizaci velkého počtu fotek do galerií.

Informace o uživateli musí být uloženy na bezpečném místě a jejich hesla v zakódované podobě.

3 Řešení

3.1 Výběr softwaru

Pro realizaci informačního systému je v neposlední řadě zapotřebí také nějaký software. Potřebujeme vybrat operační systém, na kterém budeme provozovat službu webového serveru. Podle operačního systému a serveru, případně podle oblíbenosti programátora, vybereme skriptovací jazyk a databázový server, ve kterém budou uložena data.

Pokud ovšem hledáme co nejlevnější řešení, tak v nabídce produktů můžeme vybírat mezi OpenSource nebo Free softwarem. U tohoto typu licencí je pořizovací cena velice nízká nebo úplně nulová.

3.1.1 Operační systém

V současné době je na trhu velké množství operačních systémů, na kterých je možné provozovat webový server. U firem, které poskytují hosting webových stránek, se nejčastěji setkáme s konfigurací Linux, Apache, PHP a MySQL. Některé hostings poskytují prostor pro stránky i na platformě Windows, kde se jako skriptovací jazyk na straně serveru používá ASP, případně ASP.NET a pro uložení dat databázový server Microsoft SQL Server.

Názory na to, který z obou operačních systémů je bezpečnější, nebo které řešení nabízí větší výkon, je nad rámec této práce. Podle mého názoru záleží v první řadě na penězích, protože řešení čtveřice Linux, Apache, PHP, MySQL vychází podstatně levněji oproti cenám licencí serverového operačního systému Windows a databázového serveru SQL. V druhé řadě záleží na lidech, kteří budou systém vyvíjet, a na jejich znalostech, jestli jsou zvyklí pracovat spíše v prostředí Windows nebo Linux.

Já osobně jsem si pro vývoj systému vybral operační systém Windows, protože nemám zkušenosti s konfigurací a správou Linuxu.

3.1.2 Webový server

Abychom mohli vytvořit informační systém, potřebujeme k tomu také nějaký webový server, který bude poskytovat naše stránky uživatelům v síti. V současné době se mezi nejpoužívanější servery řadí produkt firmy *Apache Software Foundation* s názvem Apache HTTP Server Version 2.0 a produkt od firmy Microsoft pod zkratkou IIS (*Internet Information Services*). Výběr webového serveru je úzce svázan s výběrem operačního systému. Pokud budeme umisťovat náš informační systém na server, který máme v naší firmě, vybereme přednostně ten server, který bude kompatibilní s operačním systémem používaným na firemním serveru. Serve Apache je možné nainstalovat jak na operační

systemu typu Unix, tak i na operační systém MS Windows. Další, kritériem je skriptovací jazyk, ve kterém budeme systém vytvářet. Oba servery je možné nastavit tak, aby interpretovaly jak PHP, tak i ASP skripty. Nicméně každý z jazyků obsahuje sadu funkcí, které jsou spíše orientované na operační systém, pro který byly původně určeny. Například jazyk PHP obsahuje funkce pro zjištění vlastníka souboru, čísla skupiny, nebo zaslání emailů přes Postfix. Tyto funkce ale na operačním systému Windows nemají žádný význam.

Pro řešení této práce jsem zvolil webový server Apache. Neodpovídá to sice modelovému řešení Linux, PHP a MySQL, ale s provozem serveru Apache na systému Windows mám dobré zkušenosti.

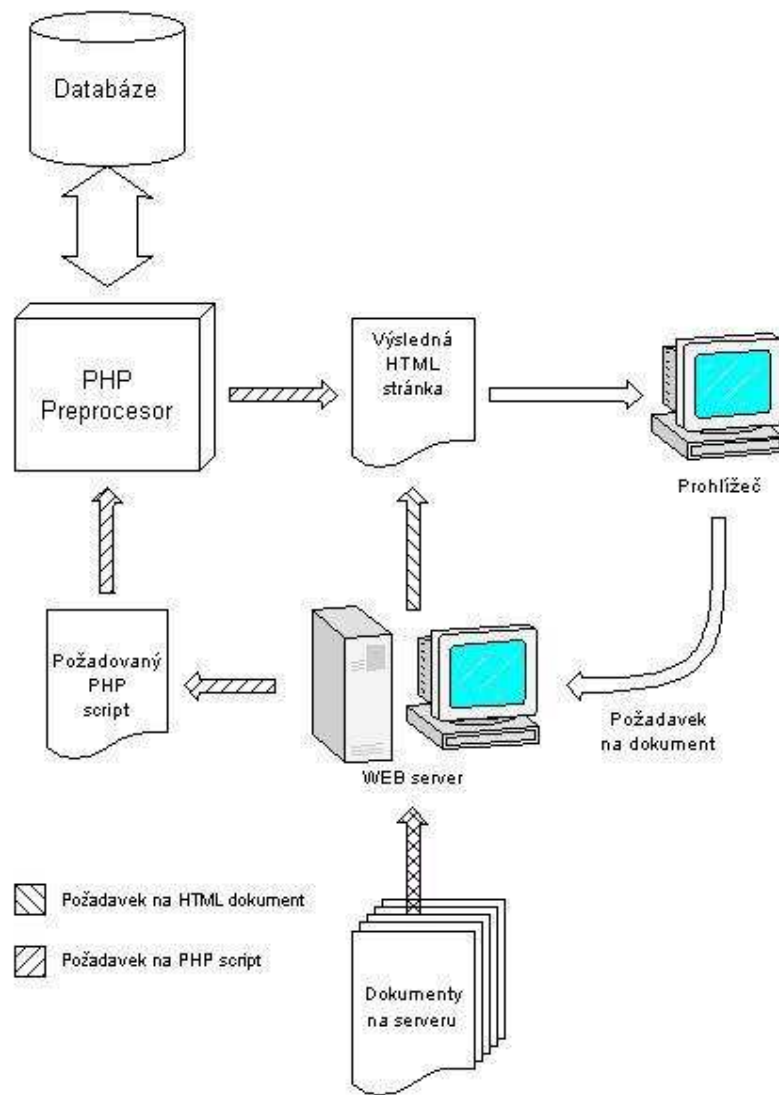
3.1.3 Skriptovací jazyk

Mezi nejpoužívanější serverové skriptovací jazyky patří, jak již bylo zmíněno dříve, jazyk PHP a jazyk ASP. Každý z těchto dvou jazyků má své výhody i nevýhody, ale v podstatě splní všechny požadavky programátorů. Hlavním rozdílem je nedostatečná podpora objektově orientovaného programování v jazyku PHP verze 4.0. Další nevýhodou jazyka PHP je absence přívětivého vývojového prostředí. Ovšem i přes všechny uvedené nedostatky je jazyk PHP stále nejpoužívanější, protože většina hostingů poskytuje právě podporu jazyka PHP.

Jazyk PHP je také oblíbený pro svou jednoduchou syntaxi, která vychází z jazyků Perl a C. Středně zkušený programátor si na syntaxi jazyka PHP zvykne rychleji, než u jazyka ASP. Hlavní výhodou jazyka PHP je dlouhá historie jeho používání, takže existuje spousta knihoven, které lze zdarma použít.

Na obrázku 3.1 je znázorněna činnost serveru při příchodu požadavku na určitý dokument. Pokud se jedná o skript PHP, předá se dokument ke zpracování preprocesoru PHP a ten potom vrací výsledný HTML dokument. V případě, že se o PHP skript nejedná, odesílá se dokument přímo klientovi.

Pro řešení této práce jsem zvolil skriptovací jazyk PHP, protože s ním mám několikaleté zkušenosti.



Obrázek 3.1 – zpracování požadavku klienta na dokument

3.1.4 Databázový systém

Jak již bylo řečeno, pro chod systému je velmi podstatný databázový systém, který bude uchovávat data zobrazované v prezentaci. Různé systémy nabízejí různé funkce, za různou cenu. Na trhu je k dostání spousta kvalitních databázových systémů, které jsou určeny pro velké podnikové systémy, jako např. Oracle. Pro použití ve webových prezentacích se nejčastěji používají databázové systémy MySQL, MS SQL, nebo Postgre.

Rozdíl mezi jmenovanými systémy je hlavně v ceně licence. Protože server MySQL je pro nekomerční použití poskytován zdarma, zvolil jsem si jej pro řešení své bakalářské práce.

3.1.5 Hosting a doména

Pokud nevlastníme vlastní server, můžeme využít nabídky firem, které poskytují hosting webových prezentací na svých serverech. Mezi poskytovateli hostingů je možné hodně vybírat, protože je jich velké množství. Každý poskytovatel nabízí různé služby, různě veliký prostor pro uložení dat prezentace a hlavně různý software. Pokud budeme vybírat pouze s hostingů nabízející podporu jazyka PHP, první, co by nás mělo zajímat, je verze PHP nainstalovaného na jejich serveru a moduly, které jsou do něj nainstalované.

Spolu s výběrem hostingů často souvisí i registrace domény. Výběru doménového jména je také důležité věnovat určitý čas, protože zvolením vhodné kombinace slov v doméně se může naše prezentace umístit na lepší pozici při vyhledávání, viz. kapitola 1.1.5. Při registraci je možné vybírat z několika domén prvního řádu. V české republice se nejvíce používá doména .CZ, která je však oproti ostatním doménám jako např. *.INFO*, *.NET*, *.COM*, podstatně dražší.

3.2 Model systému

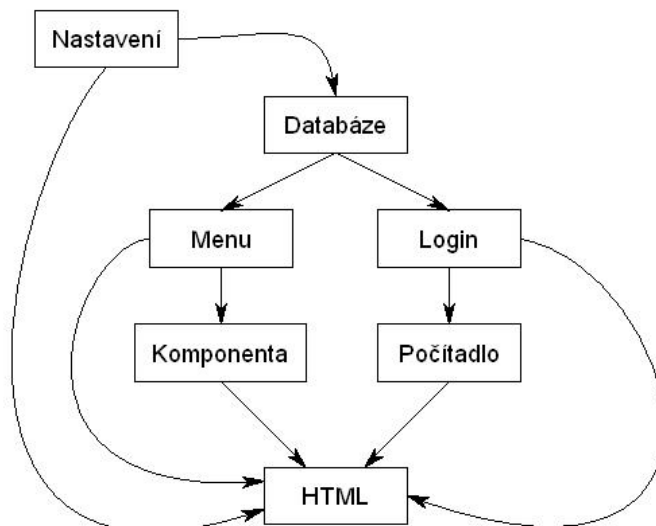
Systém se skládá z několika základních částí. Funkcionalita celého systému je rozložena mezi jádro systému, viz. obrázek 3.2, moduly, komponenty a knihovny obecných funkcí. Jádro systému se stará o načítání a zavádění souborů, které obsahují obecné funkce, moduly nebo komponenty.

Knihovny obecných funkcí jsou funkce, které se vyskytují v různých částech systému a kdyby byly součástí komponenty nebo modulu, tak by se zbytečně opakovaly. Jedná se například o funkci pro zaslání e-mailů, formátování datumu, apod.

Moduly jsou části systému, které je možné přidávat do jádra systému podle požadavku zákazníka na funkcionalitu systému. Moduly pro základní funkčnost systému jsou moduly nastavení, menu, login, html a počítadlo.

Komponenty jsou větší funkční bloky, které je možné přidávat do systému stejně jako moduly, ale jedná se o funkce, které nejsou systémového charakteru, jako je např. přístup k databázi, ale o funkce, se kterými pracuje návštěvník prezentace přímo.

Mezi hlavní součásti jádra systému patří objekt udržující informace o konfiguraci systému, objekt, který se stará o spojení s databází, dále objekt obsahující výsledný HTML dokument, objekt generující menu, objekt umožňující přihlášení uživatele, pak objekt, který uchovává informace o přístupu jednotlivých návštěvníků, a nakonec komponenta, která zobrazí to, co návštěvník vybere pomocí menu.



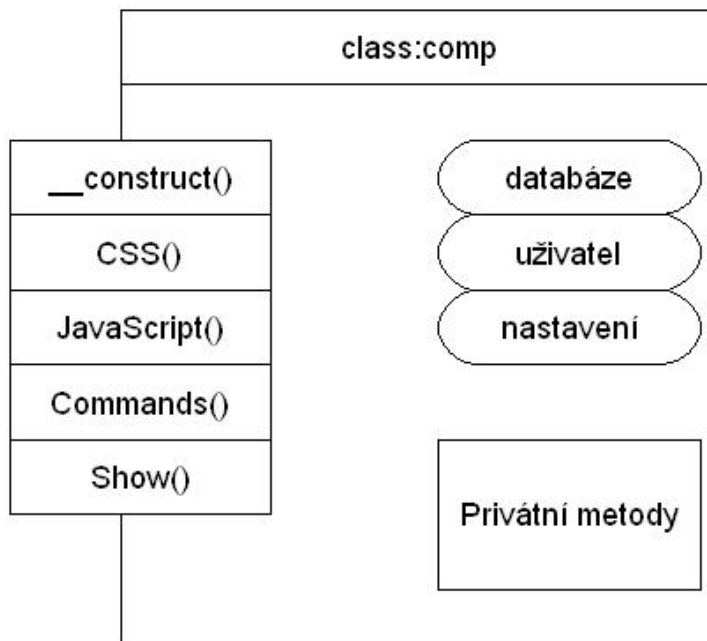
Obrázek 3.2 – diagram jádra systému

3.2.1 Komponenta

Hlavní výhodou celého systému je to, že se skládá z množství komponent, které je možné do systému přidávat a odebírat podle požadavků zákazníka. Komponenty umožňují jednoduše přidávat do systému kompletní funkční bloky, jako například katalog výrobků, elektronický obchod nebo diskusní fórum. Jednotlivé komponenty jsou realizovány pomocí třídy *comp*, která má jednotné rozhraní.

Rozhraní komponent viz. Obrázek 3.3 je definováno pomocí několika metod a konstruktoru. Konstruktor zajišťuje přístup komponenty k databázi, konfiguraci a informacím o přihlášeném návštěvníkovi. *css()* je metoda, která vrací cestu k souboru s lokálními definicemi kaskádových stylů pro vzhled uživatelského rozhraní komponenty. Metoda *javascript()* vrací JavaScriptový kód, který je potřebný pro vylepšení funkčnosti uživatelského rozhraní komponenty, na straně klienta. Metoda *commands()* se stará o vykonání požadované funkce, např. uložení dat do databáze. A poslední z metod rozhraní je metoda *show()*, která se volá na místě, kde se má zobrazit uživatelské rozhraní komponenty.

Metoda *show()* může být rozdělena do dvou sekcí podle toho, jestli jde o zobrazování informací, které komponenta spravuje, nebo o jejich administraci.



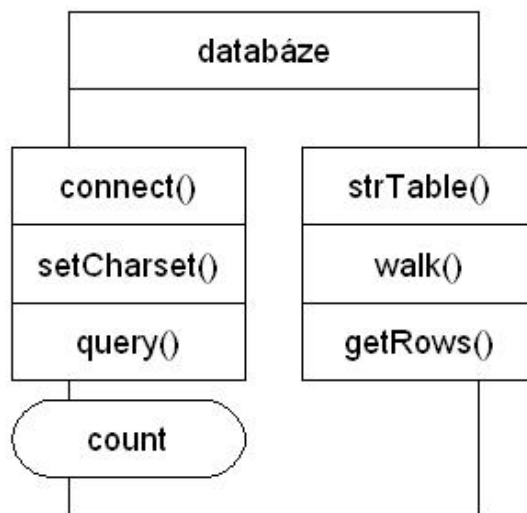
Obrázek 3.3 - komponenta

Na obrázku 3.3 je znázorněna abstraktní komponenta, která definuje rozhraní a privátní metody. Privátní metody jsou odlišné pro každou komponentu podle jejího návrhu. Metody mohou přistupovat k databázi, k informacím o právě přihlášeném uživateli nebo objektu, který udržuje konfiguraci systému..

3.2.2 Třída databáze

Aby bylo možné komunikovat s databází, je do systému zahrnut i objekt, který provádí dotazy nad databází. Jeho rozhraní je zobrazeno na obrázku 3.6. Po vytvoření objektu databáze se objekt pokusí spojit s databázovým serverem. Pokud se mu to nepodaří, vypíše chybovou hlášku a zastaví další provádění skriptu. Po připojení k serveru metodou *connect()* třída vybere databázi, se kterou bude systém pracovat, a nakonec nastaví pomocí metody *setCharset()* znakovou sadu, ve které se budou ukládat zpracovávaná data.

Mezi základní metody objektu databáze patří metoda *query()*. Tato metoda provede dotaz nad databází. Dotazy se metodě předávají jako parametr typu string. Další ze základních metod je metoda *getRows()*, která při každém zavolání vrátí jeden řádek výsledné tabulky. Počet záznamů ve výsledné tabulce je možné zjistit přečtením atributu *count*.



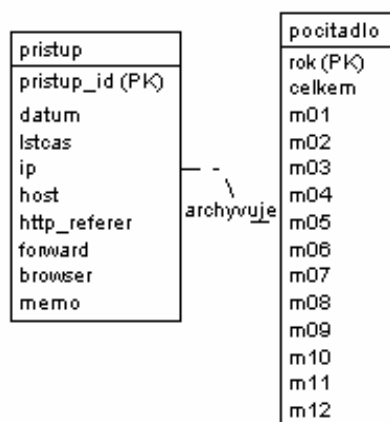
Obrázek 3.4 – třída databáze

Jednou z rozšiřujících metod je metoda *strTable()*. Tato metoda vrátí html kód výsledné tabulky a jako hlavičku použije jména sloupců tabulky v databázi. Další rozšiřující metoda *walk()* umožňuje definovat funkci, která se použije při zpracování jednotlivých řádků tabulky.

3.2.3 Počítadlo přístupů

Modul, který se stará o evidenci přístupu návštěvníků v systému, je umístěn v souboru *mod_pocitadlo.php*. Informace o jednotlivých přístupech se uchovávají v datové struktuře, která je znázorněna na obrázku. Tabulka *pristup* obsahuje informace o jednotlivých přístupech návštěvníků v aktuálním roce. Mezi evidované údaje patří datum a čas prvního a posledního požadavku návštěvníka. IP adresa počítače, který zaslal požadavek. Informace o prohlížeči, adresa, ze které klient přichází, a pole pro poznámku.

Na obrázku 3.5 je znázorněna datová struktura, která v databázi udržuje uvedené informace.



Obrázek 3.5 – datová struktura počítadla

Informace o přístupech se využívají i v ostatních komponentách, kde je potřeba mít možnost zjistit nějaké informace o návštěvníku. Například ve spolupráci s modulem pro vkládání ankety zajišťuje, aby jeden návštěvník nehlasoval více než jednou. Tato funkce není zrovna jednoduchá, protože prozatím neexistuje spolehlivá metoda, mimo autentizace, jak přesně identifikovat unikátnost jednotlivých návštěvníků. Při použití cookies na straně klienta nemůžeme zaručit, že se informace o hlasování doopravdy zaznamená. Naopak při vazbě na IP adresu znemožníme hlasovat uživatelům, kteří jsou připojeni pomocí routeru a mají stejnou veřejnou IP adresu.

V tabulce *pocitadlo* se pak uchovává statistika přístupů pro celý rok a pro každý měsíc zvlášť. Tyto informace pak umožňují sledovat popularitu stránek v průběhu několika let.

3.2.4 Uživatelé a uživatelské skupiny

Jelikož tento systém má umožňovat některým návštěvníkům větší práva, než ostatním, musí umožňovat odlišit jednotlivé uživatele. Systém umožňuje definovat neomezený počet uživatelských skupin a jednotlivé uživatele do nich přiřazovat. Tento systém je pouze jednoúrovňový, takže neumožňuje zanořování skupin do sebe. Datová struktura uživatelů a skupin je vidět na obrázku 3.6.

V základní konfiguraci systém vyžaduje přítomnost čtyř základních skupin.

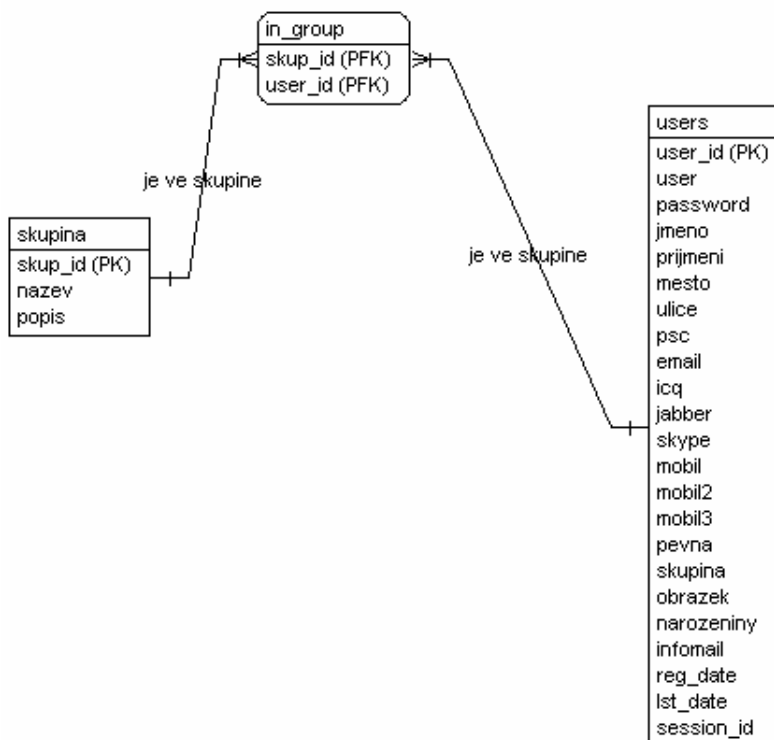
- Skupina *guest* umožňuje definovat práva přístupu neregistrovaným anonymním návštěvníkům.
- Skupina *administrator* seskupuje uživatele v nejsilnějším právu přístupu právem vše.
- Skupina *superadmin* je vyhrazena pro uživatele s většími právy než obyčejný registrovaný uživatel, ale ne s absolutním přístupem jako administrátor. Uživatel ve skupině *superadmin* bývá většinou zákazník.
- Skupina *reguser* pak obsahuje registrované uživatele. Většinou jsou to uživatelé, kteří se musí zaregistrovat, aby mohly využívat hlavní funkci prezentace. Např. u elektronického obchodu je to odeslání objednávky.

Každý uživatel, kterému chceme přidělit nějaké právo přístupu, musí být členem některé skupiny, protože práva je možné přidělovat pouze na úrovni skupin. Na obrázku je znázorněna struktura tabulek v databázi, které se starají o uchovávání informací o uživatelských účtech.

Tabulka *user* obsahuje informace o jednotlivých uživateli. Uchovávané informace jsou hlavně uživatelské jméno, otisk hesla, jméno, příjmení, datum a čas registrace uživatele, datum a čas poslední návštěvy uživatele a kontaktní informace jako je telefon a adresa.

Tabulka *skupina* obsahuje informace o uživatelských skupinách v systému. Tato tabulka vždy obsahuje základní čtyři skupiny, jak bylo uvedeno výše. Pomocí komponenty *com_user.php* je možné v administrační části prezentace vytvořit a spravovat uživatelské skupiny a přiřazovat do nich jednotlivé uživatele.

Tabulka *in_group* pak slouží jako vazební tabulka mezi tabulkou *user* a tabulkou *skupina*. Umožňuje přidávat uživatele do skupin.



Obrázek 3.6 – datová struktura uživatelů

3.2.5 Login

Pokud má uživatel v systému vytvořený účet, má možnost se přihlásit do systému pod svým uživatelským jménem a heslem. Přihlašování uživatelů do systému zajišťuje modul uložený v souboru *mod_login.php*. Tento modul zároveň vytvoří objekt, který nese informace o přihlášeném uživateli a poskytuje je ostatním částem systému.

Objekt *login* má několik metod, pomocí nichž se dá zjistit uživatelské jméno, id uživatele, nebo jestli je členem některé skupiny. Metoda pro zjišťování, zda je uživatel členem nějaké skupiny, se volá ve všech komponentách, které se starají o nějakou funkci, ke které mají přístup pouze registrovaní uživatelé. Příkladem takové komponenty může být komponenta pro správu uživatelských účtů, ke které mají přístup pouze členové skupiny *administrator*.

Metoda pro kontrolu přihlašovacích údajů po přijetí požadavku na přihlášení postupuje v několika následujících krocích.

1. převede uživatelské jméno na malá písmena a vyhledá ho v tabulce uživatelů
2. převede zadané heslo na otisk algoritmem MD5
3. porovná otisk hesla získaný z databáze a otisk vygenerovaný v těle skriptu
4. pokud se otisky shodují provede přihlášení uživatele a zjistí do kterých skupin patří

5. pokud zadané údaje nesouhlasí, informuje o tom uživatele

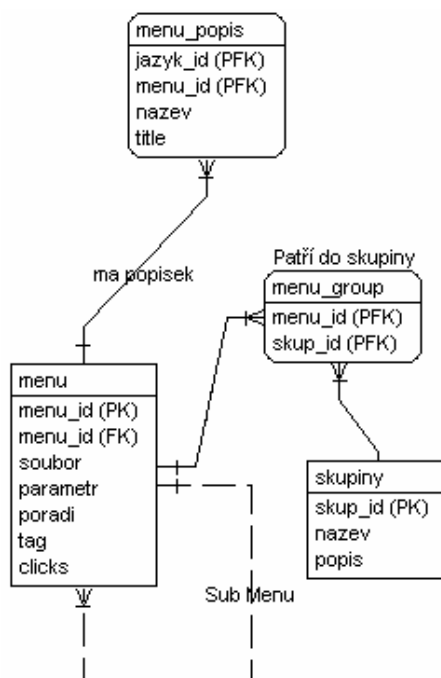
3.2.6 Menu

Menu je realizováno modulem *mod_menu.php*. Modul si při vytvoření instance jeho objektu načte za tabulky menu informace o položkách, ke kterým má uživatel přístup. Modul *menu* vytváří navigační menu, které umožňuje návštěvníkovi přecházet mezi jednotlivými částmi prezentace.

V současné verzi systému není obsažena komponenta pro správu menu, protože ke změně obsahu položek menu nedochází tak často a ve většině případů zůstanou ve stejném stavu, v jakém byly do systému vloženy.

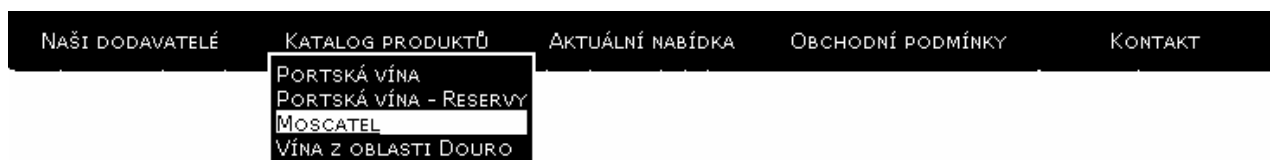
Modul se skládá ze tří hlavních tabulek, viz. obrázek 3.7, tabulky *menu*, tabulky *menu_group* a tabulky *menu_popis*. Zatímco tabulka *menu* se stará o udržování informací jednotlivých položek menu, tabulka *menu_group* přiřazuje jednotlivé položky do uživatelských skupin. Poslední tabulka připravuje systém na možnost rozšíření, aby každá položka menu mohla mít popisek v několika jazycích.

Nejdůležitější tabulkou je však tabulka *menu*, která udává, co se vlastně vloží po kliknutí na některou z položek menu. Modul umožňuje vložit přímo nějaký soubor (většinou stránku html, nebo php skript), komponentu, nebo aby se po kliknutí vykonal nějaký příkaz systému, např. odhlášení uživatele. Aby bylo možné vytvořit několikaúrovňové menu, je v tabulce zaveden atribut *parent_id*, který udává *id* nadřazené položky.



Obrázek 3.7 – datová struktura menu

Menu zobrazené v prezentaci pak může vypadat podobně jak na obrázku 3.8.



Obrázek 3.8 – ukázka menu

3.2.7 Šablony

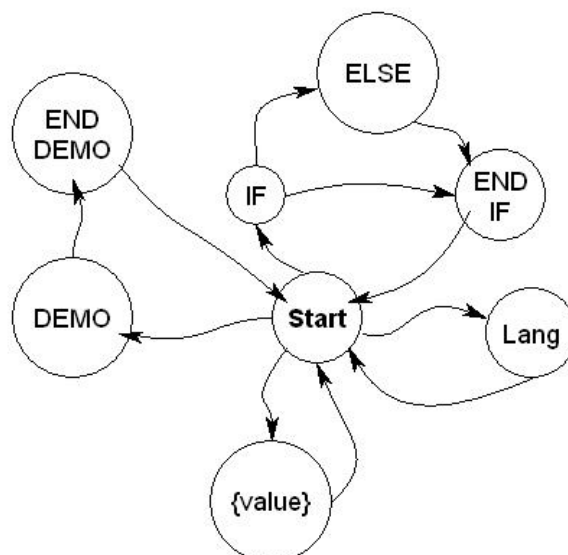
Protože jeden z požadavků na vlastnosti systému bylo jednoduché vytváření nových modulů, obsahuje systém modul pro použití šablon. Šablony umožňují oddělit vzhled prezentace od obsahu. Modul využívá pro popis šablony značku komentáře v jazyce HTML a tagy HTML. Značka pro komentář se využívá proto, že umožňuje do jejího těla zapsat jakýkoliv text, aniž by se změnil zobrazený obsah HTML stránky. Aby se usnadnilo grafikovi navrhování šablony v běžném prohlížeči, používají se pro vyznačení prvků šablony klasické HTML tagy. Pokud by byla šablona realizována v XML, tak by její prohlížení v běžném prohlížeči nebylo možné. V následující tabulce je uveden seznam značek, které je modul schopen zpracovat.

značka	význam
<!--{navez}-->	Značka obsahu. V šabloně se potom nahradí hodnotou přiřazenou klíči název
<!--[prikaz parametr]-->	Značka příkazu

název příkau	význam
<!--[IF navez]-->	Pokud je nastaven klíč název vypíše se obsah následující tuto značku
<!--[ELSE]-->	Tato značka uvádí alternativní obsah pokud klíč název není nastaven
<!--[END IF]-->	Tato značka ukončuje podmínku
<!--[DEMO]-->	Demo hodnoty pro usnadnění tvorby šablony
<!--[END DEMO]-->	Konec demo hodnoty

Tabulka 3.1 – značky a příkazy šablony

Na obrázku 3.9 je diagram konečného automatu zpracovávajícího šablony.



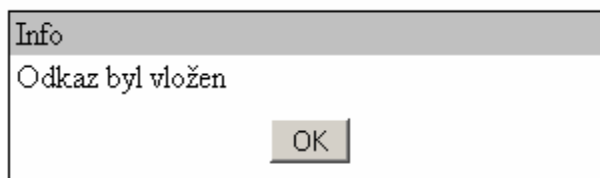
Obrázek 3.9 – automat pro zpracování šablony

3.2.8 Zobrazení zpráv

Nedílnou součástí systému je také modul, který se stará o zobrazování informačních zpráv. Modul se jmenuje *alert* a je uložen v souboru *mod_alert.php*. Zajišťuje centrální správu zpráv, pomocí níž bude uživatel informován o chybách nebo stavech provedení některé z funkcí.

Objekt *alert* má metodu pro vložení zprávy a metodu pro vypsání všech zpráv. Do objektu je možné vložit několik typů zpráv, které se později vloží na určené místo ve vygenerované stránce. Objekt umožňuje vkládání zpráv typu *message*, což je informační zpráva o provedení nějaké operace, a zprávy typu *error*, což jsou zprávy o chybách.

Metoda vypisující zprávy do těla dokumentu pak může podle typu zprávy rozlišit, jak se má zpráva zobrazit. Možností, jak zobrazit zprávu, je mnoho. V tomto systému jsem použil způsob zobrazení jako popup okno. Přibližně uprostřed obrazovky se zobrazí malé okénko s textem zprávy a tlačítkem OK pro skrytí zprávy, viz. obrázek 3.10. Další způsob, jak zobrazit zprávu, je vypsání ji někde do těla dokumentu.



Obrázek 3.10 – popup zpráva

Text zpráv se ukládá do relace (session), tento způsob uložení dat zajistí, že data zůstanou v systému do té doby, než je objekt zpracuje a z relace odstraní. Toto řešení umožní použití funkce *redirect*.

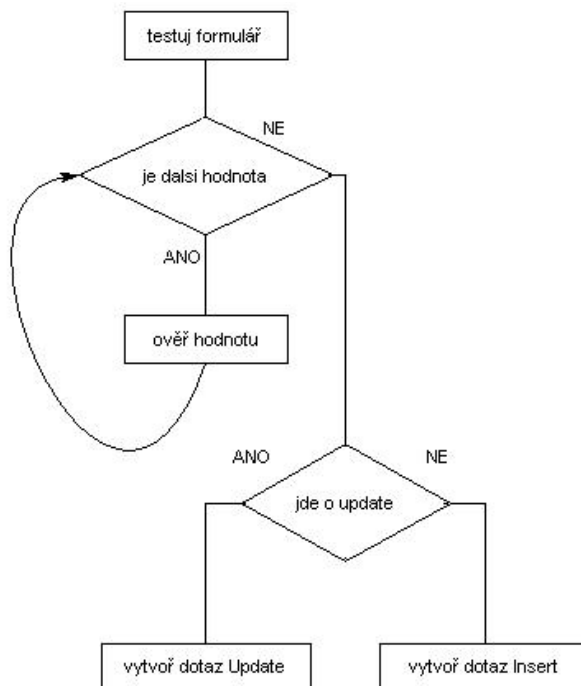
Funkci *redirect* je vhodné použít v kódu na místech, kde dojde k vložení zpracovávaných dat do databáze. Při volání funkce *redirect* server odešle klientovy hlavičku o přesměrování, a ten uvolní z lokální cache data, které právě odeslal prostřednictvím formuláře. Tímto způsobem se zamezí opakovanému odeslání dat z formuláře a jednoduchým způsobem se zamezí vzniku duplicitních záznamů v databázi.

3.2.9 Zpracování formulářů

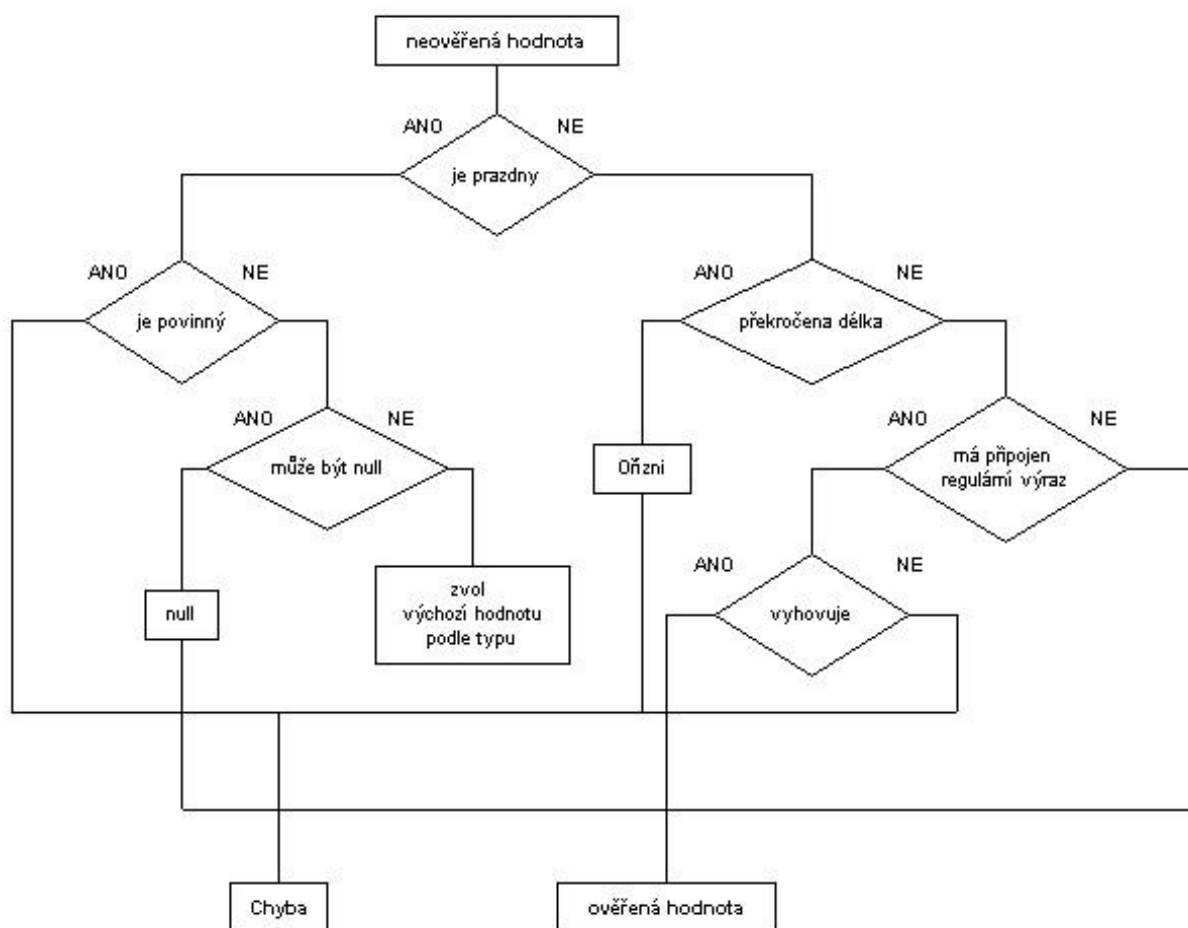
Protože každý informační systém obsahuje velké množství formulářů, které většinou dělají stejnou nebo podobnou činnost, je v systému implementován mechanismus pro automatizaci kontroly zadaných údajů. Třída *form*, uložená v souboru *form.php*, zahrnuje funkce pro vytvoření HTML kódu formuláře a metodu, která zkontroluje a vytvoří dotaz pro zápis dat do databáze. Třída *form* dědí metody od třídy *html* a rozšiřuje jejich vlastnosti. Třída *html* umožňuje pouze vkládat do formuláře ovládací prvky jako např. textové políčko, tlačítka, apod.

Třída *form* umožňuje vytvořit pojmenovaný formulář, do kterého uživatel vyplní požadované informace. Pro kontrolu dat odeslaných formulářem musíme předat objektu *form* informace o jednotlivých polích formuláře. Každému poli je možno přidělit jméno, obecnou chybovou zprávu, regulární výraz, kterým se pole ověří a parametr, zda je pole nutno vyplnit. Pokud dojde k odeslání formuláře pomocí tlačítka *submit*, musí být někde v `commands()` metodě komponenty podmínka, která rozpozná odeslaný formulář a obslouží jeho zpracování.

Na obrázku 3.11 je znázorněn postup při kontrole formuláře. Prochází se seznam zadaných polí a podle parametrů k nim připojených se vyhodnotí. Pokud se jedná o pole, které je nutné vyplnit, aby mohl být formulář zpracován, a jeho hodnota je prázdná, dojde k chybě. Stejně tak dojde k chybě, pokud je k poli připojen regulární výraz a hodnota v něm obsažená mu nevyhoví. Postup při kontrole jednoho pole formuláře je znázorněn na obrázku 3.12.



Obrázek 3.11 – diagram kontroly formuláře



Obrázek 3.12 – diagram kontroly jednoho pole formuláře

3.3 Implementované funkce

Prezentace je rozdělena do dvou bloků. První blok uložený v souboru *index.php* slouží k zobrazení prezentace návštěvníkům, kde mohou prohlížet její obsah. Druhý blok je uložený v souboru *admin.php*, a ten umožňuje přihlášení uživatelům, kteří mají práva k editaci obsahu nebo ostatním administračním úkonům.

Oba dva bloky jsou v podstatě totožné, s tím rozdílem, že se do každého načítá různý hlavní kaskádový styl a každý pracuje se svým vlastním menu. V bloku *anmin.php* je dále zavedena speciální konstanta.

Komponenty mohou být specializované, což znamená, že obstarávají pouze zobrazení spravovaných informací, nebo kompletní, které obstarávají jak zobrazení spravovaných, tak i rozhraní pro jejich administraci. Aby bylo možné odlišit, zda jde o mód administrace, nebo prohlížení, je v bloku *admin.php* zavedena konstanta, pomocí níž se testuje, o jaký mód se jedná.

3.3.1 Správa uživatelů

Komponenta *uživatelé* se stará o správu uživatelů a uživatelských skupin. Umožňuje přidávat uživatele do jednotlivých skupin, vytváření a mazání nových skupin. Editační rozhraní je zobrazeno na obrázku 3.13. Vytvoření skupiny a přiřazení uživatelů však nemá žádný efekt, pokud danou skupinu nedefinuje některá komponenta, nebo modul. Uživatelé by byli členy skupiny, kterou ale nikdo nevyžaduje.

V další verzi systému by bylo dobré implementovat mechanismus, který by umožňoval přidělovat modulům a komponentám skupiny. Podle přiřazených skupin by se pak určilo, kteří uživatelé mají právo s touto komponentou pracovat. Současně by měl zavést i seznam operací, které by člen skupiny mohl vykonávat.

3.3.2 Nahrávání souborů

Protože většina uživatelů nebude mít možnost přistupovat k souborům prezentace pomocí FTP protokolu, musejí mít možnost nějakým způsobem nahrávat soubory na webový server. Pokud je uživateli umožněn přístup ke komponentě souborového manažeru, může nahrávat soubory do

Skupiny

- administrator
- guest
- ps1234b
- superuser

vložit

[Zpět](#)

administrator

Uživatelé ve skupině		dostupní uživatelé
KoNoPaS Skudce	<< >>	Bišof Brož Buday Bužga dablikcz Dejwyd Duba

Název

[Zpět](#)

Obrázek 3.13 – správa uživatelských skupin

určitých složek na webu. Ke kterým složkám bude mít uživatel přístup, je určeno také na základě toho, do které skupiny patří. Na obrázku 3.14 je vidět uživatelské rozhraní komponenty.

Souborový manager



Obrázek 3.14 – souborový manager

Například uživatel ve skupině fotograf bude mít možnost zapisovat soubory do složky galerie, kde bude mít právo vytvořit i novou složku. Program automaticky rozpozná, že se jedná o ukládání fotografie do složky galerií fotek, a postará se o vytvoření dvojice fotografií ve správném rozlišení.

Funkce *image_resize*, která se stará o změnu velikosti nahrávané fotky, je uložena v souboru *gd2.php*. Tato skupina funkcí vyžaduje, aby bylo PHP na serveru nainstalováno spolu s modulem GD2. Modul GD2 obsahuje funkce pro práci s obrázky.

3.3.3 Správa galerie

Modul správy galerie umožňuje hromadně vytvořit z velkého množství fotek, uložených v nějakém adresáři na serveru, galerii fotek. V modulu se jednoduše vybere adresář, ze kterého se má galerie vytvořit, vyplní se pole s informacemi, kdy a kým byly fotografie vytvořeny, a po odeslání formuláře se galerie vytvoří.

3.3.4 Správa obsahu

Jelikož systém má umožňovat jednoduché vytváření textového obsahu i uživatelům, kteří nemají zkušenosti se značkovacími jazyky, je do systému začleněn i jednoduchý wysiwyg editor. Zkratka

wysiwyg (*What You See Is What You Get*) v překladu znamená „co vidíš, to dostaneš“. Na stejném principu funguje i známý textový editor Word od firmy Microsoft.

Formátování textu se neprovádí zapisováním textových značek mezi užitečný text, ale klikáním na ikony na nástrojové liště.

Komponenta starající se o obsah slouží jak k jeho zobrazení, tak i k jeho editaci. Pokud se k editaci dostaneme z administrační části, komponenta nám nabídne veškeré položky obsahu uložené v databázi. Výběrem jedné z položek se dostaneme k wysiwyg editoru, ve kterém se nám zobrazí text obsahu, který hodláme editovat.

3.3.5 Ankety a jejich správa

Anonymní anketa může být dobrým zdrojem informací od návštěvníků prezentace. Umožňuje totiž jednoduchým způsobem zjistit odpovědi na často kladené otázky, např. jestli návštěvníkům vyhovuje grafické zpracování prezentace, jaký obsah by chtěli v prezentaci nalézt, nebo co jim v prezentaci vadí. Na obrázku 3.15 je znázorněn formulář pro zadání ankety.

Realizace anket se skládá ze tří tabulek. Tabulka *ank_otz* uchovává jednotlivé otázky a datumy, kdy má skončit její dotazování. Tabulka *ank_odp* uvádí seznam možností pro odpověď na jednu z otázek. A poslední tabulka *ank_stat* obsahuje statistiku hlasování pro jednotlivé odpovědi.

Aby se zamezilo vícenásobnému hlasování jednoho uživatele, ukládá se v tabulce *ank_stat* i id přístupu z modulu počítačového přístupu na stránky.

Editace ankety - Pouzivate RSS?

Odpověď	Počet hlasů
<input type="checkbox"/> ANO	2
<input type="checkbox"/> NE	4
<input type="checkbox"/> Nevím co to je	2

Obrázek 3.15 – správa anket

3.3.6 Správa odkazů

Základní myšlenkou webu je propojovat dokumenty umístěné na internetu. Součástí většiny prezentací často bývá seznam odkazů na weby s podobnou tematikou. O údržbu tohoto seznamu se stará komponenta *odkazy*. Komponenta umožňuje organizaci odkazů uložených v tabulce *odk_odkazy* do skupin. Seznam skupiny je pak uložen v tabulce *odk_skupiny*. Na obrázku je zobrazeno uživatelské rozhraní, pomocí něhož se odkazy do systému vkládají. Každému odkazu je možno přiřadit název, adresu, popis a obrázek. Na obrázku 3.16 je vidět jeho uživatelské rozhraní.

Kategorie 'Skupina odkazů'

Název kategorie:

Název	
<input type="checkbox"/>	odkaz
<input type="checkbox"/>	odkaz 2



Vložit odkaz

Název:	<input type="text"/>
Link:	<input type="text"/>
Popis:	<input type="text"/>
Obrázek:	<input type="text" value="-- I k o n a --"/>



Obrázek 3.16 – správa odkazů

3.3.7 Vzkazy

Někdy je vhodné návštěvníkům umožnit vyjádřit svůj názor. Modul *vzkazy* umožňuje návštěvníkům zanechávat na webu krátké vzkazy a reagovat na vzkazy ostatních návštěvníků. Aby se zabránilo vkládání zpráv spamovacími roboty, je při vkládání vzkazu nutné zadat náhodně vygenerovaný kód.

Správa vzkazů se neprovádí v administrační části, ale přímo při jejich prohlížení. Pokud má přihlášený uživatel dostatečná práva, systém mu umožní vzkazy editovat nebo mazat.

Vzkazy se řadí pod sebe podle času, kdy byly do systému vloženy viz. obrázek 3.17.

Přidat vzkaz

Strana 1 »

Skudce	Odpověď
Zrovna pracuju na seznamu 'členů' třídy a jeima kontaktama, zároveň formulář pro registraci ...	
Satek	
:) jeйда :)	
	22.Dubna 200711:56
13.Dubna 200716:55	

Skudce	Odpověď
Začínám pracovat na nové verzi WEBu ... tak snad už se konečně podaří a budete moct být na mě hrdí ;)	
Misha	
jsme na tebe hrdí, jsi šikula :)	
	10.Dubna 200717:26
9.Dubna 200700:00	

Satek	Odpověď
Rokac byl good, mmch fotak tam nebyl?	
Skudce	
jo ... ale né náš :) bohužel	
	8.Dubna 200700:00
31.Března 200700:00	

Obrázek 3.17 - vzkazy

3.3.8 Zaslání zprávy

Aby mohly návštěvníci komunikovat s vlastníky prezentace, implementoval jsem formulář pro odeslání e-mailu. Pokud zveřejníte svou e-mailovou adresu na stránce, která je přístupná komukoli, můžete počítat s tím, že se brzy ve vaší schránce objeví nemalé množství nevyžádané pošty. Pokud má internet nějaké špatné vlastnosti, tak mezi první z nich patří SPAM - nevyžádaná pošta. Aby mohl návštěvník kontaktovat některou z osob, které se starají o chod prezentace, umožňuje formulář vybrat osobu, na jejíž e-mail se zpráva doručí.

Napište nám

komu

správci webu ▾

jméno, příjmení

e-mail

telefon

zpráva

Odeslat

Obrázek 3.18 – formulář pro odeslání zprávy

Závěr

Tento systém pro správu obsahu webové prezentace umožňuje jednoduše vytvořit webovou prezentaci, ke které mohou přistupovat návštěvníci pomocí standardního internetového prohlížeče. V administračním rozhraní prezentace systém umožňuje vytvářet uživatele a uživatelské skupiny, které mohou mít různé možnosti práce se systémem.

Systém má být jednoduše modifikovatelný pro webové prezentace menších firem podle požadavků zákazníka. Jednoduché úpravy systému umožňuje způsob jeho navržení, které rozprostírá specializované části do funkčních bloků – komponent.

Pro ověření použitelnosti systému v reálném provozu byly pomocí tohoto systému vytvořeny dvě internetové aplikace. Testovací provoz prokázal, že systém je díky komponentám velice rychle modifikovatelný a variabilní. Jednotlivé komponenty se díky šablonám dají velice rychle upravovat podle návrhů designéra.

V dalších verzích systému by mělo dojít k rozšíření systému o možnost vkládání obsahu v několika jazykových verzích. Další vylepšení by si zasloužil jednoúrovňový model skupina. Uspořádání skupin do stromové struktury by usnadnilo organizaci uživatelů v systému a distribuování práv přístupu k jednotlivým částem systému.

Literatura

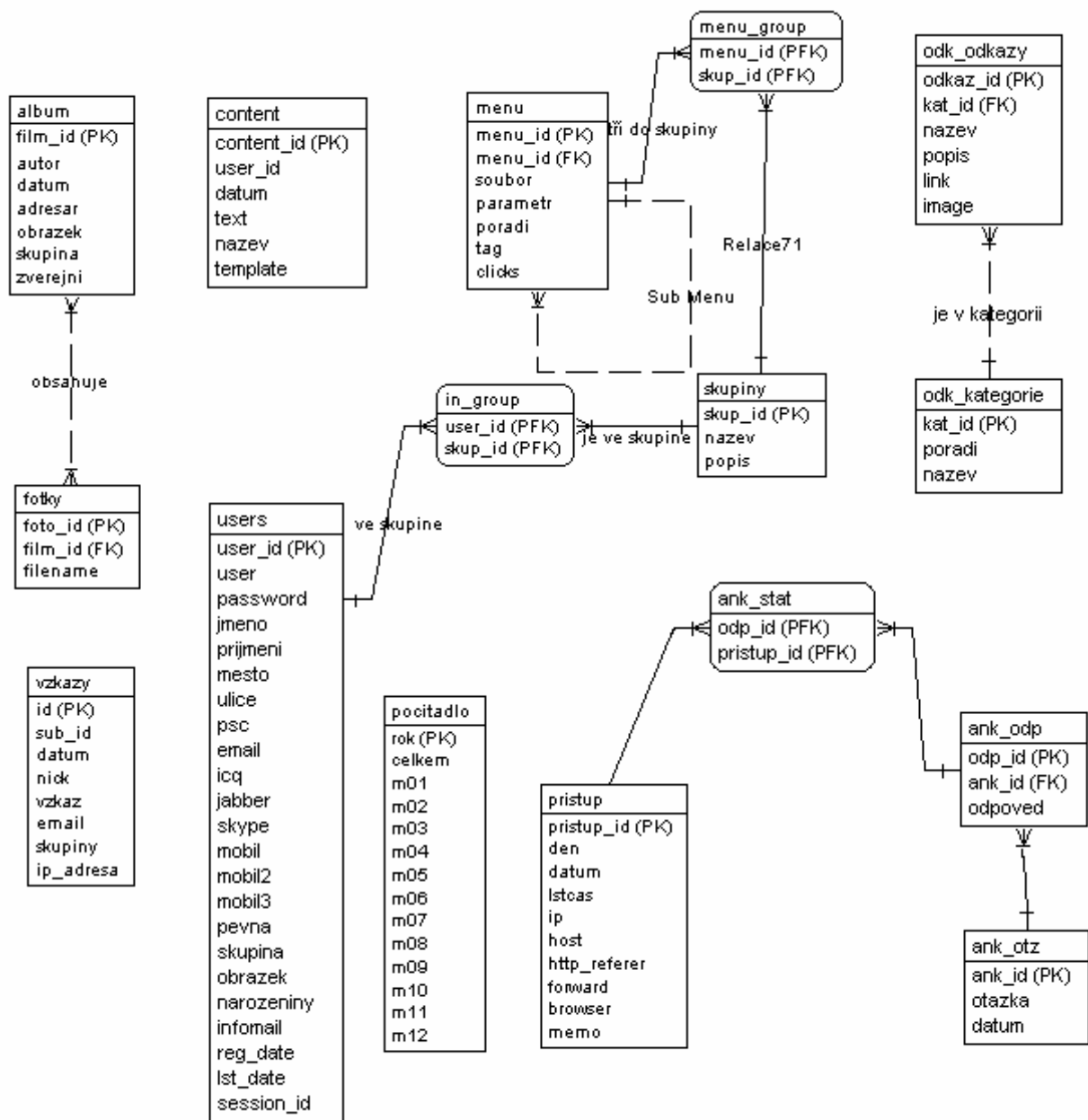
- [1] JANOVSKEÝ D. *Jak psát web* [online]. 2007 [cit. 2007-04-11]. Dostupný z WWW: <<http://www.jakpsatweb.cz>>.
- [2] *Wikipedie* [online]. 2007 [cit. 2007-05-11]. Dostupný z WWW: <<http://cs.wikipedia.org>>.
- [3] ULLMAN, L., *PHP a MySQL*. Brno : Computer Press, 2004. 534 s. ISBN 80-251-0063-4.
- [4] PROKOP, M., *CSS - kaskádové styly pro webdesignéry*. Brno: CP Books, 2005. 288 s. ISBN 80-251-0487-7
- [5] DARIE, Ch., *Ajax a php*. Zonner Press, 2006. 320 s. ISBN: 80-86815-47-1.

Seznam příloh

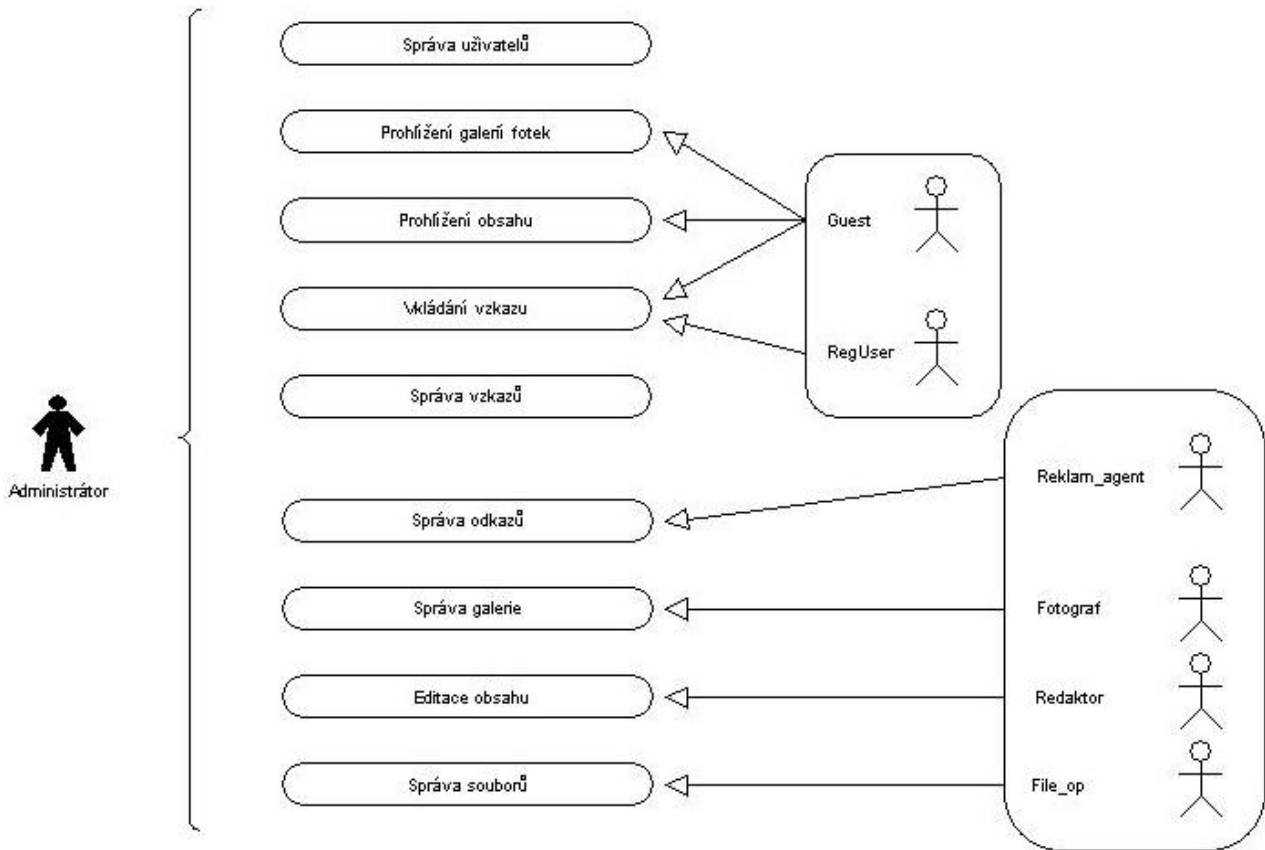
Příloha 1. CD

Příloha 2. Schéma databáze

Příloha 3. UseCase Diagram



Příloha 2 – schéma databáze



Příloha 3 – UseCase diagram