

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Diplomová práce

Content management systémy

Autor DP: Bc. Petr Pechek
Vedoucí DP: Ing. Simona Pejsarová

2009 ČZU v Praze

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Content management systémy" jsem vypracoval samostatně pod vedením vedoucí diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 14.4.2009

Petr Pechek

Poděkování:

Děkuji Ing. Simoně Pejsarové za cenné rady a za pomoc při řešení problémů spojených s vypracováním diplomové práce.

Content management systémy

Content management systems

Souhrn

Tato práce pojednává o problematice content management systémů. Práce popisuje content management systémy a technologie i nástroje využívané jimi a pro jejich tvorbu. Z velké množiny content management systémů jsou vybrány dva zástupci a je proveden jejich popis a následně jejich porovnání.

Vybranými zástupci content management systémů jsou Drupal a Joomla. Pro porovnání je předem stanoveno několik vlastností nebo požadavků. Tyto vlastnosti jsou na základě stanovených kritérií bodově ohodnoceny.

Praktická část práce se zabývá tvorbou modulu pro CMS Drupal. Pomocí modulu lze zobrazit fotogalerii z webového alba Google picasa. Dále umožňuje přidávání a ubírání fotogalerií jako ostatních článků, také ukazuje výhody použití API Drupalu pro zjednodušení tvorby webových stránek. Zde jsou popsány základní principy tvorby modulů pro CMS Drupal. V závěru této části jsou zhodnoceny přínosy Drupal API pro budoucí využití a rozšířitelnost.

Klíčová slova

Content management systémy, PHP, MYSQL, HTML, Drupal, Joomla

Summary

This work deals with the issue of content management systems. The work first describes what is content management systems, what technology they use and what tools can be used for their creation. From a large set of content management system, selected two representatives and the descriptions and then comparing them.

Selected representatives of content management systems are Drupal and Joomla. For comparison it is selected some features or requirements. These properties or requirements are evaluated by points based on established criteria.

In practical part of work was created a module for Drupal CMS. The module allows you to view the gallery of Google Picasa Web Album. It makes possible to create or delete albums like an article, also shows advantages of using Drupal API to simplify creation of web pages. This describes the basic principles of creation modules for Drupal CMS. At the end of this part is to promote the benefits of Drupal API for future use and extensibility.

Key words

Content management systems, PHP, MYSQL, HTML, Drupal, Joomla

Obsah

1 Úvod.....	6
2 Cíl práce a metodika.....	6
3 Charakteristika CMS.....	7
3.1 Redakční systém.....	7
3.2 Základní funkce CMS.....	8
4 Nástroje pro tvorbu CMS.....	9
4.1 Klientská vrstva CMS.....	10
4.1.1 CSS.....	11
4.1.2 Javascript.....	12
4.1.2.1 Technologie AJAX.....	12
4.1.3 SEO.....	13
4.2 Aplikační vrstva CMS.....	14
4.2.1 Autentizace uživatele.....	14
4.2.2 Zabezpečení proti CSRF.....	15
4.2.2.1 Referer.....	15
4.2.2.2 Autorizační token.....	15
4.3 Datová vrstva CMS.....	16
4.3.1 Objektový datový model.....	16
4.3.2 Relační datový model.....	17
4.3.3 Relační algebra.....	17
4.3.4 Jazyk SQL.....	18
5 Přehled nejznámější zástupců a jejich popis.....	18
5.1 Výběr dvou zástupců.....	19
5.2 CMS drupal.....	19
5.2.1 Historie Drupalu.....	19
5.2.2 Bezpečnost systému Drupal.....	19
5.2.2.1 Chyba ve verzi 4.7.11 až 5.6.....	20
5.2.2.2 Bezpečnostní chyba v modulu Simple Access.....	20
5.2.2.3 Zneužitelnost systému menu Drupalu.....	21
5.2.2.4 Chyba v instalátoru.....	21
5.2.3 Způsob ověření identity uživatele.....	22
5.2.3.1 Bezpečnost hashovacího algoritmu md5.....	22
5.2.4 Zabezpečení přístupu do databáze.....	23
5.2.5 Identifikace již přihlášeného uživatele.....	23
5.2.6 Práva přístupu k systému.....	23
5.2.7 Překlady.....	24
5.2.8 WYSIWYG editor.....	25
5.2.9 Vzhled systému Drupal.....	25
5.2.9.1 PHPTemplate.....	27
5.2.9.2 Změna barev v motivu vzhledu.....	28
5.2.10 Úprava vzhledu v TinyMCE.....	28
5.3 CMS Joomla.....	28
5.3.1 Historie.....	29

5.3.2	Původ názvu.....	30
5.3.3	Bezpečnost CMS Joomla!.....	31
5.3.3.1	Bezpečnostní chyby v CMS Joomla!.....	31
5.3.4	Přístupová práva k systému.....	32
5.3.5	Překlady.....	32
5.3.6	WYSIWYG editor.....	34
5.3.7	Vzhled systému Joomla!.....	34
6	Porovnání vybraných CMS.....	37
6.1	Porovnávání vlastností obou systémů.....	37
6.2	Systém Drupal.....	38
6.2.1	Instalace.....	38
6.2.2	Nastavení čistých URL.....	39
6.2.3	Přidání modulu.....	39
6.2.4	Přidání článků.....	39
6.2.5	WYSIWYG editor.....	39
6.2.6	Instalace a provoz e-shopu.....	40
6.2.7	Tvorba anket.....	40
6.2.8	Instalace a provoz fotogalerie.....	41
6.2.9	Správa uživatelů.....	42
6.3	Systém Joomla!.....	42
6.3.1	Instalace.....	42
6.3.2	Nastavení čistých URL.....	43
6.3.3	Přidání modulu.....	43
6.3.4	Přidání článků.....	44
6.3.5	WYSIWYG editor.....	44
6.3.6	Instalace a provoz e-shopu.....	45
6.3.7	Tvorba anket.....	45
6.3.8	Instalace a provoz fotogalerie.....	45
6.3.9	Správa uživatelů.....	46
6.4	Shrnutí porovnání.....	47
7	Tvorba modulu pro vybraný systém.....	48
7.1	Funkce které bude modul realizovat.....	48
7.2	Implementace modulu.....	48
7.2.1	Obsah souboru gpicasa.info.....	49
7.2.2	Obsah souboru gpicasa.install.....	49
7.2.3	Obsah souboru gpicasa.module.....	52
7.2.3.1	Funkce popisující položku obsahu.....	53
7.2.3.2	Další důležité funkce.....	57
7.3	Zobecněte přínosy zvoleného nástroje pro další možná použití.....	60
8	Závěr.....	60
9	Literatura.....	61
10	Přílohy.....	63
10.1	Vysvětlení zkratk.....	63
10.2	Zdrojový kód souboru gpicasa.module.....	64

1 Úvod

Internet je v poslední době nejvíce dynamicky se rozvíjícím médiem. Je využíván k prezentaci většiny firem. Stránky na internetu již nejsou statické ale dynamické. Jejich obsah se mění daleko více a častěji než dřív. Internet se také hojně využívá k provozu elektronických obchodů. Je trend, aby správa webových stránek byla snazší a nevyžadovala příliš velké znalosti o technologiích.

V této době se hojně využívají také content management systémy (zkráceně CMS), které umožňují zpracovat a tvořit obsah webových stránek bez znalosti programování a syntaxe jazyků, které se k tvorbě webových stránek používají. CMS umožňují publikovat články, vytvářet fotogalerie, tvořit a spravovat elektronické obchody. CMS umožňují řídit a kontrolovat publikování těchto prvků obsahu.

2 Cíl práce a metodika

Cílem práce je:

- Popsat CMS a nástroje pro jejich tvorbu
- Popsat a porovnat dva vybrané zástupce CMS
- Vytvořit modul pro jeden z content management systémů

V úvodu práce bude proveden rešeršní popis problematiky CMS, kde se bude vycházet z analýzy dostupných zdrojů a literatury. Dále bude proveden popis nástrojů a technologií pro jejich tvorbu. Z dostupných CMS budou vybrány dva zástupci a u nich provedeno porovnání.

Před porovnáním budou vybrány určité vlastnosti a funkce sledované u zvolených systémů. Vlastnosti a funkce získají body, dle určité stupnice hodnocení. Po ohodnocení jednotlivých vlastností a požadovaných funkcí se provede součet a systém, který získá více bodů, bude hodnocen v porovnávání lépe.

Dále bude popsána a realizována tvorba modulu pro jeden z porovnávaných systémů. A provedeno zhodnocení tvorby webových stránek pomocí API vybraného systému s ohledem na budoucí rozšiřitelnost.

3 Charakteristika CMS

CMS neboli Content Management System je všeobecný software na správu obsahu. Ten se může skládat z textů, obrázků a jiných mediálních elektronických souborů. Účelem CMS systému je přehledně spravovat obsah různého druhu a umožnit většímu počtu osob přístup k jistým materiálům. Toto značně ulehčuje komunikaci, zvláště ve firmách. Jednou z výhod mnohých CMS systémů je automatizace firemních procesů anebo pracovního toku (workflow). Pomocí CMS systému se přes různé lidi ve firmě může pohybovat určitý materiál, který mohou připomínkovat, měnit anebo jinak společně modifikovat.

CMS systém je častokrát používán i jako nástroj na archivaci dokumentů. Tuto možnost běžně využívají mediální firmy (televize, vydavatelství atd.), jejichž předmět podnikání je přímo spojený s poskytováním obsahu. CMS systémy však používají i všechny velké mezinárodní společnosti na správu své firemní komunikace. Tato funkce je však čím dále důležitější i pro malé a střední firmy. CMS systémy umožňují firmám neztratit se v dokumentech a nalézt je jednoduše a rychle na serveru, namísto klasického archivu. [1]

Pohledem jiného autora zkratka CMS pochází z anglického termínu Content management systems a označují se tak různé systémy pro správu obsahu. CMS je složitou webovou aplikací, používanou pro vytváření a úpravy obsahu webu bez znalostí tvorby WWW stránek přímo pomocí jazyka HTML a dalších jazyků. CMS tak umožňuje ušetřit finance, které by musely být zaplacený kodérovi. Další výhodou CMS je možnost rychlé reakce, web si můžete aktualizovat sami kdykoliv, ať je třeba neděle nebo svátek. [2]

3.1 Redakční systém

Podle různých autorů je redakčním systémem pro správu článků na webových stránkách. Podle toho lze redakční systém považovat za podmnožinu CMS.

Jiní autoři naopak považují redakční systém a CMS za synonyma. V chápání, co je to CMS, není u všech autorů a zdrojů informací shoda.

Podle Jana Bednaříka CMS umožňuje pohodlně spravovat obsah vaší webové prezentace v rámci nějakého webového rozhraní. [3]

Dle wikipedie je systém pro správu obsahu (CMS) software zajišťující správu dokumentů, nejčastěji webového obsahu. V dnešní době se jako CMS zpravidla chápou webové aplikace, někdy s případným doplňkovým programovým vybavením u klienta. [4]

Dle jiného zdroje "Redakční systém" představuje aplikaci, která spravuje data a informace různého charakteru a obsahu a zároveň se stará i o jejich efektivní využití a zobrazení na některém z předem připravených výstupů. V případě "webového redakčního systému" je za hlavní výstup považována internetová služba world wide web (tedy zkráceně "web"), jenž je každému čtenáři (uživateli) přístupná skrze webový / internetový prohlížeč – např. Internet Explorer, Mozilla, Opera, Netscape Navigator, atd. [5]

Nelze tedy jednoznačně stanovit, jaký je rozdíl mezi CMS a redakčním systémem. Někteří autoři toto považují za synonyma a někteří považují redakční systém za podmnožinu CMS. Lze tedy tvrdit spíše, že redakční systém je podmnožinou CMS a od CMS se liší tím, že neobsahuje některé specifické funkce, jako například funkce pro redakční zprávu a korektoru článků.

3.2 Základní funkce CMS

Mezi základní funkce CMS (obvykle se člení na administrátorské a uživatelské) patří:

- Tvorba, modifikace a publikace dokumentů (článků) zpravidla prostřednictvím webového rozhraní, často s využitím jednoduchého online WYSIWYG editoru nebo jednoduchého systému formátování textu (není nutná znalost HTML), [4]
- řízení přístupu k dokumentům, zpravidla se správou uživatelů a přístupových práv, často s funkcemi workflow či groupware, lze přidělovat práva přímo uživatelům nebo skupinám uživatelů
- správa diskusí či komentářů, ať už k publikovaným dokumentům nebo obecných,

- správa souborů,
- správa obrázků či galerií,
- kalendářní funkce,
- statistika přístupů. [4]

Dle Jana Bednaříka zajišťuje CMS tyto funkce:

- Uživatelské rozhraní - každý CMS má své uživatelské rozhraní pro správu obsahu. Práce s ním je snadná a intuitivní. Naučit se s ním pracovat jde rychle a je to pohodlné. [3]
- Přehledná správa obsahu - jednotlivé WWW stránky vidíte v jejich přirozené hierarchii a nemusí vás zajímat, jak a kde jsou uloženy. [3]
- Podpora různých rozšíření - anketa, diskuze, novinky a další jsou rozšíření CMS systému, které dodají vašim stránkám dynamiku. Se statickou prezentací nemožné věci. Navíc pomocí CMS velmi lehce ovladatelné. [3]
- Uživatelské účty - můžete snadno navolit pro každého zaměstnance (uživatele) práva a omezení pro správu obsahu. [3]

Je tedy zřejmé, že základní funkčností, kterou by měl CMS zajišťovat, je zjednodušení kompletní správy webové aplikace. Správa webu by měla být v CMS jednoduchá a intuitivní. Měla by se podobat ostatní práci počítačovými programy z pohledu uživatele.

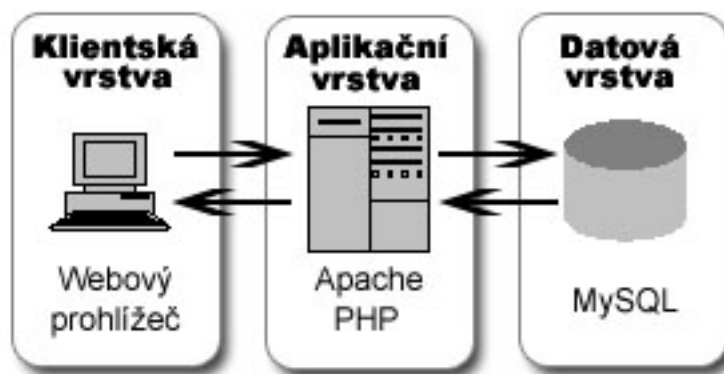
4 Nástroje pro tvorbu CMS

CMS je systém využívající výpočetní model client-server, konkrétně používá třívrstvou architekturu. Celá aplikace je rozdělena na tyto tři části:

- **Prezentační vrstva** – obsahuje funkce uživatelského rozhraní. Obvykle existuje několik prezentačních vrstev pro různé druhy zařízení, platformy a prostředí. Obvykle se stará o zobrazení grafického uživatelského rozhraní.
- **Aplikační vrstva** – tvoří prostředníka mezi vrstvou prezentační a vrstvou datovou. Obsahuje tzv. business logiku aplikace. V této vrstvě dochází k transformaci dat mezi vstupně-výstupními požadavky a datovou vrstvou.

- **Datová vrstva** – obsahuje funkce pro přístup k informacím v datovém úložišti [6]

Výhodou toho rozdělení je, že je umožněn do značné míry nezávislý vývoj jednotlivých částí aplikace. Dále je výhodou centrální uložení dat, kdy k datům může přistupovat více klientů najednou.



Obr. č. 1 - Třívrstvá architektura webové aplikace. Zobrazuje často používané technologie.

V systémech CMS se používají v jednotlivých vrstvách tyto technologie (v závorce jsou uvedeny konkrétní používané nástroje):

- Klientská vrstva: webový prohlížeč (Internet explorer, Mozilla, Opera a jiné). Nejrozšířenějším prohlížečem je dle více statistik Internet explorer od firmy Microsoft. [7]
- Aplikační vrstva: webový server (Apache, IIS) spolu se skriptovacím jazykem (PHP, ASP, Ruby a jiné) Podle společnosti Netcraft je nejrozšířenějším webovým serverem Apache nesledovaný IIS a GFE (od Google) [8]
- Datová vrstva: relační databáze (MYSQL, MSSQL, Oracle a jiné) nebo objektová databáze. Zřídka je využíváno ukládání dat přímo do souborů.

4.1 Klientská vrstva CMS

Jak bylo výše napsáno, aplikační vrstva je v CMS zastoupena webovým prohlížečem, který je v roli univerzálního klienta. Nejrozšířenějšími produkty jsou Internet explorer a Mozilla Firefox.

V klientské vrstvě se využívá hlavně těchto technologií: HTML, javascript, DOM a CSS.

HTML (HyperText Markup Language) je značkovací jazyk určený pro popis struktury dokumentu ve webovém. Lze ho použít i pro formátování vzhledu. Každý HTML element je určen párovými nebo nepárovými tagy. V současné době se většinou již html používá pouze pro zobrazení logické struktury webových stránek. Pro vytváření vzhledu se používá technologie CSS.

4.1.1 CSS

CSS (cascade style sheet) se používá k vytváření vzhledu webových stránek. Pomocí CSS lze určit vzhled jakéhokoliv elementu HTML. Lze určit vzhled konkrétních elementů i jejich tříd. Kaskádové styly slouží pro oddělení vzhledu dokumentu od jeho obsahu.

Kaskádové styly jsou vlastně sadou pravidel, které určují, jaké fonty a jiné formátovací charakteristiky se mají použít pro zobrazení určité části dokumentu. [9]

Praktický princip CSS, z kterého lze vyvodit smysl CSS, je popsán zde:

HTML je značkovací jazyk. Z toho vyplývá, že jednotlivé značky (tagy) mají vyznačovat význam jednotlivých částí textu. Přesto se v HTML v průběhu let objevilo několik atributů a elementů, které ovlivňují pouze grafický vzhled, a některé z původních tagů určených pro logické členění textu, jsou často účelově využívány k dosažení určitých grafických efektů. Aby se předešlo těmto nešvarům, a bylo možné jednoduše oddělit strukturální formátování textu od grafického formátování, definovalo v roce 1997 konsorcium W3C soubor metod pro grafickou úpravu webových stránek. Tento soubor metod se jmenuje Cascading Style Sheets (CSS), česky "kaskádové styly". [10]

Kaskádové styly umožňují definovat způsob zobrazení (velikost a druh písma, barvu, zarovnání, orámování, pozadí apod.) u každého html elementu. Tento předpis vzhledu není přímo součástí textu stránky, a díky tomu je zápis stránky přehlednější a dobře strukturovaný. Navíc styly umožňují definovat jednotný vzhled konkrétního elementu v celém dokumentu jedním zápisem, tzn. nemusí se opakovat u každého elementu.

Pokud styl uložíme do externího souboru, může ho využívat více stránek najednou. Definice vzhledu všech stránek je uložena na jednom místě. Při požadavku na změnu vzhledu stránek stačí upravit styl na jednom místě a změny se automaticky promítnou do všech stránek. [10]

4.1.2 Javascript

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, jehož autorem je Brendan Eich z tehdejší společnosti Netscape.

Jeho syntaxe patří do rodiny jazyků C/C++/Java. Slovo Java je však součástí jeho názvu pouze z marketingových důvodů a s programovacím jazykem Java jej vedle názvu spojuje jen podobná syntaxe. JavaScript byl v červenci 1997 standardizován asociací ECMA (European Computer Manufacturers Association) a v srpnu 1998 ISO (International Organization for Standardization). Standardizovaná verze JavaScriptu je pojmenována jako ECMAScript a z ní byly odvozeny i další implementace, jako je například ActionScript. [11]

Nyní se Javascript zpravidla používá jako interpretovaný programovací jazyk pro WWW stránky, často vkládaný přímo do HTML kódu stránky nebo načítaný ze samostatných souborů. Jsou jím obvykle ovládány různé interaktivní prvky GUI (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků nebo wysiwyg. Javascript nemůže z bezpečnostních důvodů pracovat s daty na straně klienta. Výjimkou v tomto směru jsou cookies, které pomáhají odstraňovat problém bezstavovosti protokolu HTTP. Práci s cookies pomocí javascriptu lze vytvoření cookie z bezpečnostních důvodů zakázat.

Javascript je jedním ze základů technologie AJAX.

4.1.2.1 Technologie AJAX

Technologie AJAX oproti dřívějšímu přístupu k tvorbě webových stránek způsobuje, že se přenáší mezi klientem a serverem pouze část stránky a ne celá stránka.

Oproti standardnímu modelu webové aplikace, kde je browser zodpovědný za iniciaci žádosti webovému serveru a její zpracování, Ajax model poskytuje zprostředkující

mezivrstvu – Ajax Engine, která zajišťuje tuto komunikaci. Ajax engine je obvykle jen JavaScriptový objekt nebo dokonce jenom funkce volaná, kdykoli mají být od serveru získány nějaké informace. To je zásadní rozdíl od klasického přístupu, kde dojde k poskytnutí odkazu na další (externí) zdroj. V pojetí Ajaxu každý odkaz na zdroj znamená volání Ajax engine.

Požadavek je zpracován asynchronně, což znamená, že požadavek bude vyslán, ale nebude se čekat na odpověď. Namísto toho se bude dále provádět kód klienta a odpověď bude zpracována později. Server, který obvykle dodává HTML dokumenty, CSS šablony vzhledu nebo JavaScriptový kód, bude nakonfigurován tak, že bude vracet data, která umí Ajax engine zpracovat. Tato data mohou být ve formě obvyčejného textu (plain text), XML dokumentů nebo cokoli, co bude potřeba. Jedinou podmínkou je, aby Ajax engine rozuměl a uměl interpretovat předávaná data. Nejčastěji se pro přenos dat při využití technologie AJAX používá formát dat JSON. JSON jsou data uložená ve formě javascriptového kódu, která AJAX engine zpracuje s využitím funkce eval().

V okamžiku, kdy obdrží Ajax engine odpověď ze serveru, může provést příslušnou akci – parsování dat, změna podoby uživatelského rozhraní, poskytnutí informací uživateli. Díky tomu, že tento proces zahrnuje přenos menšího množství informací než tradiční webový aplikační model, bude UI aktualizováno rychleji a bez nutnosti překreslování celého výsledného aplikačního okna. Uživatel může pracovat rychleji. [12]

4.1.3 SEO

Jednou z důležitých technik používaných na straně klienta je optimalizace stránek pro vyhledávače. Toto je obvykle označované anglickou zkratkou SEO (Search Engine Optimization).

Pro dobré SEO je ze všeho nejdůležitější zjistit, jestli je optimalizovaná stránka kompletně zaindexovaná. To je potřeba zjistit kvůli tomu, že lidé většinou nehledají informace, které jsou na úvodní stránce webu, ale většinou ty, ke kterým je potřeba se proklikat. Pokud webová stránka není dobře ve vyhledávačích zaindexovaná, je nutné prověřit, zda nepoužívá některé překážky, přes které nejsou vyhledávače schopny přejít. Obecně by se dalo shrnout, že pokud dokáže optimalizovaný web dobře přečíst textový

prohlížeč Lynx, dokáží ho přečíst a zaindexovat i vyhledávače. To znamená doporučení nepoužívat technologie typu Flash, Javascript, rámce atd. [13]

Jedním z důležitých předpokladů správné optimalizace pro vyhledávače je mít URL ve správném tvaru. Pro lepší zaindexování stránek vyhledávacími roboty je vhodné mít URL bez parametrů, tedy URL by neměla obsahovat například toto "?id=1". Dále je vhodné, aby HTML kód webu byl validní. Správná validita stránek je důležitá i z důvodů přístupnosti webu. Umožňuje číst webové stránky čtečkami pro zrakově postižené.

4.2 Aplikační vrstva CMS

Aplikační vrstva CMS je zastoupena webovým serverem a programem zpracovávajícím skriptovací jazyk na straně serveru. Nejrozšířenějšími zástupci jsou Apache, IIS a GFE. Skriptovací jazyk může být modulem webového serveru nebo se může jednat o samostatný program.

U web serveru Apache se využívá především jazyk PHP, dále pak lze využít Ruby nebo Perl. U IIS se nejčastěji využívá technologie ASP nebo ASP.NET, které jsou přímo vyvinuty pro spolupráci s IIS.

Prostřednictvím aplikační vrstvy se CMS připojuje k datové vrstvě reprezentované relační databází.

4.2.1 Autentizace uživatele

Pro ověřování uživatele je na úrovni aplikační vrstvy nejpoužívanější autentizační metodou, ověření pravosti uživatele pomocí session.

Session ukládá na webovém serveru informace o přihlášeném uživateli, o konkrétní aktivní relaci. Každé session je přiřazeno unikátní ID, pomocí kterého se určuje daná session. Na straně klienta je session ID ukládáno buď pomocí cookies nebo je předáváno jako proměnná pomocí metody GET nebo POST.

4.2.2 Zabezpečení proti CSRF

Zabezpečení proti CSRF (Cross-Site Request Forgery), tedy proti podvodnému vykonání nějakého útoku z jiné stránky je realizováno nejčastěji pomocí následujících metod:

4.2.2.1 Referer

Jedním z řešení jak zabránit CSRF je testovat, zda je v hlavičce Referer adresa mé stránky s administrací:

```
if ( !substr( $_SERVER[ "HTTP_REFERER" ], 0, 34 ) ==  
"http://www.domena.cz/administrace/" )  
{  
    echo "Mě neoblafneš!";  
    exit;  
}
```

Hlavičku Referer posílá prohlížeč (obsahuje adresu stránky, ze které jsem se dostal na aktuální) a není ji možné obelstít na útočnickově stránce. Bohužel posílání této hlavičky se dá vypnout v prohlížeči, tudíž se nelze na ni spolehnout. Další nevýhoda spočívá v nemožnosti vstoupit do administrace přímo přes externí odkaz – například jsem pořád přihlášen a v záložkách mám odkazy na pravidelné snížení prémie. Jsou to nevýhody relativně malé a toto řešení je i tak použitelné. [14]

4.2.2.2 Autorizační token

Lepší metodou k zabránění CSRF, ale náročnější na implementaci jsou tzv. autorizační tokeny. Před odkázáním na další stránku dáte do URL nebo formuláře autorizační proměnnou – např. md5 hash z cookie nebo náhodně vygenerovaný řetězec, který uložíte do databáze a v dalším skriptu porovnáte poslaný token s tokenem v databázi. [14]

Použití může vypadat nějak takto:

```
<?php  
session_start();  
?>  
<form action="delete.php" method="post">  
    <input type="hidden" name="id" value="2" />  
    <input type="hidden" name="token" value="<?php  
        echo sha1( session_id( ) + $_SESSION[ "password" ] ) ?>" />  
    <input type="submit" name="btn" value="Smazat" />  
</form>
```

Protože má session omezenou platnost (vyprší za několik minut), myslím, že toto opatření je dostačující. Proto je těžké a spíše nemožné, aby útočník zjistil dané session ID.

4.3 Datová vrstva CMS

Databáze je zjednodušeně řečeno přístup k datům, kdy data jsou oddělená od struktury programu a samotný program nemůže přistupovat k datům přímo. Pro přístup k datům je využíváno SŘBD, které poskytuje rozhraní pro komunikaci ostatním programům. Databáze umožňují víceuživatelský transakční přístup k datům. A v neposlední řadě také usnadňují programátorům aplikace práci s daty.

Zde se využívá relační databáze. Open source CMS využívají především MYSQL nebo postgresQL. Těchto systémů využívají především z důvodů, že jsou také distribuovány jako open source. Základem relační databáze je relační datový model.

Další datový model, který se používá na datové vrstvě CMS je objektový datový model.

4.3.1 Objektový datový model

Objektový datový model se od relačního datového modelu hodně liší. Všechna data v databázovém systému jsou uložena v objektech, které spolu komunikují prostřednictvím zasílání zpráv. V čistě objektovém jazyku jsou všechny datové prvky objekty.

Objektový model na rozdíl od relačního může využívat vlastností objektů, které nemá relační databázový systém. V relačním databázovém systému data tvoří pouze množinu a nelze je nějak jinak strukturovat. Jedinou možnou strukturou dat je relace. V objektovém datovém modelu mohou mít data složitější strukturu. Lze využívat dědičnosti, polymorfizmu a dalších vlastností objektového datového modelu.

Objektový datový model je vyspělejší a dokonalejší oproti relačnímu datovému modelu. Umožňuje snadněji tvořit dotazy využívající v relačním modelu operaci join. V některých případech je objektová databáze rychlejší než relační databáze. A to především v případě pokud relační databáze nevyužívá indexaci dat.

4.3.2 Relační datový model

Rozsáhleji bude popsán relační datový model než objektový datový model. Toto je především z důvodů, že v současných CMS se jako úložiště dat využívají především relační DB. Relační DB má také oproti objektové DB výhodu v dobré open source implementaci a také v rozšíření u hostingových společností. Společnosti poskytující webhosting pro provoz CMS používají především právě relační DB.

V roce 1969 přišel doktor E. F. Codd (*A relational data model for large shared data banks*) se svou představou o databázi založené na matematickém aparátu relačních množin a predikátové logiky. Databázová relace se od matematické poněkud liší. Má zavedený pomocný aparát nazvaný schéma relace. Schéma relace říká, jaký je název relace, kolik má sloupců a jaké jsou jejich názvy a domény (doména určuje přípustné hodnoty v daném sloupci). V databázích je schématem relace definice struktury tabulky. Ovšem relací není jenom tabulka, ale cokoliv strukturovaného do řádků a sloupců. Což znamená, že relací je i výsledek jakéhokoliv dotazu, a tak s ním můžeme pracovat. Velmi rozšířeným omylem je, že relační se model jmenuje podle vztahů mezi daty, což není pravda, protože název relační vychází z relací, na kterých je celý model založen. Relace je založena na relačním kalkulu, který vychází z relační algebry. [15]

V relačním modelu jsou data sdružená do relací, které mají n a m řádků. Za relaci označujeme souhrn řádků a sloupců. Data v relaci jsou identifikovaná pouze pomocí svých hodnot, tedy nelze na ně odkazovat jinak než prostřednictvím jejich hodnot. Aby byly řádky relace jednoznačně identifikovány, je potřeba, aby každý řádek relace obsahoval v rámci relace jednoznačný primární klíč, který ho určuje. [15]

4.3.3 Relační algebra

Relační algebra je popis operace s relační databází. Pro operace s relační DB se využívá tří základních operací relační algebry. Pomocí těchto operací a jejich kombinací lze realizovat jakoukoliv operaci s databází.

Operace projekce (project), slouží pro výběr, neboli označení některých sloupců relace. Neoznačené sloupce se ve výpisu, pokud k němu použijeme projekci, nebudou zobrazovat.

Operace selekce (select) slouží pro výběr některých řádků matice, které splňují zadanou podmínku.

Operace spojení (join) slouží ke spojení dvou různých relací. Pokud není zadaná podmínka a jedna relace má N řádků a druhá M, pak výsledná relace má N x M řádků. Tedy výsledná relace obsahuje všechny kombinace řádků spojovaných tabulek. Operace spojení je obvykle doplněna o podmínku. Proto rozlišujeme spojení s testováním na rovnost (equi-join) a testování s jinou podmínkou než rovností (theta-join). [16]

4.3.4 Jazyk SQL

Dotazovací jazyk SQL je jazyk založený na relačním kalkulu. Jeho hlavním rysem je, že se snaží napodobovat styl dotazování v anglickém jazyce. Původně vznikl v roce 1974, kdy se ještě nazýval SEQUEL. Mezi první implementace jazyka SQL patří systém R od firmy IBM. Původně nebyla syntaxe a sémantika jazyka SQL standardizována, jazyk SQL vznikl značně živelně. Jako nutnost se tedy jevila jeho standardizace, ke které následně také došlo v organizaci ANSI. Standardizace byla zejména nutná, protože postupně vznikalo dost databázových systémů, které využívaly jazyk SQL. Takovými systémy byl například DB/2 od IBM nebo databázový systém ORACLE od stejnojmenné firmy.

Pomocí jazyka SQL lze realizovat libovolné operace se systémem SŘBD. Je možné pomocí něho realizovat dotazy založené na relační algebře. [16]

5 Přehled nejznámější zástupců a jejich popis

Existuje okolo stovky open source CMS napsaných v jazyce PHP. Dále existuje přibližně 10 open source CMS naprogramovaných v jazyce ASP. K tomuto počtu existuje ještě velké množství komerčních CMS systémů. [17]

V rozsahu této práce není možné popsat všechny dostupné CMS. Z důvodů snadné instalace a dostupnosti pro více operačních systémů se zaměřím na kategorii open source systémů využívajících jazyk PHP. Ze všech zástupců budou pro podrobný popis vybrány dva systémy.

5.1 Výběr dvou zástupců

Z popsaných zástupců CMS byly vybrány systémy Joomla! a Drupal. Byly vybrány z důvodů univerzálnosti použití. Oba systémy jsou open source a jsou poskytovány zdarma. K oběma systémům je poskytováno velké množství doplňkových modulů rozšiřujících jejich funkcionalitu.

5.2 CMS drupal

Drupal je modulárním CMS pro zpravu webových stránek. Samotný Drupal se stará o zpravu spojení s databází, funkce pro práci s formuláři a další důležité funkce. Toto je nazýváno jádrem systému Drupal. Jádro systému zajišťuje API funkce pro práci s překladem, s databází a podobně. Moduly využívají těchto funkcí. Tyto funkce usnadňují vývojářům tvorbu modulů.

Vlastní stránky jsou realizovány pomocí modulů. Toto usnadňuje tvorbu modulů a pomáhá sjednotit vzhled systému Drupal. Programátor rozšíření již nemusí vše programovat od základu, může využívat API funkcí.

5.2.1 Historie Drupalu

Drupal je originálním dílem holandského studenta Driese Buytaerta. Když v roce 2000 Dries potřeboval sdílet informace s kolegy ze své koleje, vytvořil systém, který posléze (2001) pojmenoval Drop. Tento název vznikl z překlepu ze slova Dorp – holandsky vesnice. První veřejná verze se však již jmenovala Drupal a opět vychází z holandštiny, tentokrát z anglické výslovnosti slova drop – druppel. O vývoj Drupalu se stará několik hlavních vývojářů a více než 400 přispěvatelů, kteří poskytli své patche do jádra. Hlavní slovo má samozřejmě stále Dries Buytaert, ale o správu jednotlivých verzí se starají vybraní lidé. [18]

5.2.2 Bezpečnost systému Drupal

V této kapitole je popisován systém Drupal z hlediska bezpečnosti. Dále budou zdůrazněny a popsány jeho nejdůležitější bezpečnostní chyby. Většinu zde popisovaných chyb lze opravit pomocí upgrade CMS Drupal na jeho novější verzi.

5.2.2.1 Chyba ve verzi 4.7.11 až 5.6

Některé chyby zabezpečení byly hlášeny v systému Drupal, které mohou být využity škodlivým chování lidí, ke cross-site skriptingu, vložení nějakého skriptu nebo cross-site požadavku při útoku pomocí zneužití identity uživatele.

1) Vstup prošel přes nespecifikované parametry v tématu .tpl.php souborů není řádně ošetřen předtím než byl vrácen uživateli. To může být zneužito k provedení libovolného skriptu a HTML kódu do uživatelovi session v prohlížeči v postižené stránce.

Úspěšné využití této chyby vyžaduje, aby "register_globals" bylo povoleno a .htaccess soubor nedokáže zabránit přístupu do .tpl.php souborů a vypnutí "register_globals" také nezabrání této chybě. [19]

2) Chyba ve zdrojovém kódu filtrační funkce může být využívána k přemostění přes filtr neplatné UTF-8 sekvence. To může být zneužito k vložení libovolného HTML kódu a skriptu, který bude proveden v prohlížeči uživatele v souvislosti dotčených webových stránek, kdy jsou špatná data zobrazena pomocí prohlížeče Internet Explorer 6. [19]

3) Modul aggregator umožňuje uživatelům provádět určité akce prostřednictvím HTTP GET žádosti bez vykonávání jakékoli kontroly a ověření platnosti žádosti. To může být zneužito např. k odebrání položky z určité části, pokud uživatel navštíví speciálně vytvořenou stránku.

Modul aggregator se využívá k vytváření souhrnů zpráv z jiných webových stránek. Při vytváření těchto souhrnů využívá technologii RSS, která je postavená na jazyce XML. [19]

Slabiny jsou vykazovány v předcházejících verzích než 4.7.11 a 5.6. [19]

5.2.2.2 Bezpečnostní chyba v modulu Simple Access

Modul Drupalu Simple Access je bezpečnostně zneužitelný pomocí přemostění.

Simple Access pro Drupal v této verzi je náchylný k bezpečnostnímu přemostění, protože aplikace nedokáže správně zachovat soukromé informace o uzlu. [20]

Útočníci mohou využít tohoto problému k obejití některých bezpečnostní omezení a získat potenciálně citlivé informace, které mohou vést k jiným útokům. [20]

Verze modulu Simple Access před 5.x-1.3, jsou zranitelné.

Řešením tohoto problému je aktualizace modulu na novější verzi. V nové verzi je problém odstraněn.[20]

5.2.2.3 Zneužitelnost systému menu Drupalu

Některé bezpečnostní chyby, které byly odhaleny v Drupalu mohou být zneužity zlomyslnými uživateli k překonání bezpečnostních omezení.

Tato zneužitelnost je způsobena systémem menu, který nekontroluje důkladně přístupová práva k určitým stránkám. To může být využito například k editaci profilu ostatních uživatelů, editaci určitých proměnných pokud útočník vidí administrátorské stránky. To může odhalit potenciálně citlivé informace z blogu nebo jiných stránek navzdory chybějícím přístupovým právům.

Tato bezpečnostní chyba existuje ve verzích 6.x do verze 6.2.

Řešení aktualizace na verzi 6.2 nebo použití bezpečnostního patche.

5.2.2.4 Chyba v instalátoru

Instalátor systému Drupal umožňuje jakémukoliv návštěvníkovi poskytnout přístupové údaje pro databázi, když vlastník stránky nemá přístup k databázi. Toto povoluje útočníkovi spustit jakýkoliv kód na straně serveru. [21]

Okamžitým řešením problému je odstranění souboru install.php z kořenového adresáře CMS Drupal.

Tento problém se vyskytuje ve verzi 5.x, konkrétně ve verzích předcházejících verzi 5.3.

Řešením problému je aktualizace na verzi 5.3. Pokud uživatel nemůže systém aktualizovat okamžitě, je možné tento problém vyřešit pomocí aplikování bezpečnostního patche, který odstraní tuto chybu do té doby než bude moci aktualizovat celý systém Drupal. [21]

5.2.3 Způsob ověření identity uživatele

Každý uživatel je identifikován pomocí jedinečného přihlašovacího jména a pomocí jedinečného hesla.

Heslo je zakódováno pomocí hashovací funkce md5. Viz funkce z modulu jádra Users.

```
function user_pass_rehash($password, $timestamp, $login)
{
    return md5($timestamp . $password . $login);
}
```

Pro přihlášení do systému není omezen počet přihlášení, ale je možné nainstalovat do systému modul, díky kterému bude omezen počet přihlašovacích pokusů. Při jejich dosažení bude uživatelský účet zablokován. Také lze díky tomuto modulu nastavit zpoždění při přihlašování. Tento modul zabraňuje zneužití přihlašovacího formuláře k útokům robotů hrubou silou.

Bez jeho instalace lze zneužít přihlašovací formulář a roboti mohou získat přístup do systému.

5.2.3.1 Bezpečnost hashovacího algoritmu md5

Algoritmus MD5 vznikl v roce 1991 vytvořil jej Ronald Rivestem, aby nahradil, dosud používanou funkci MD4. V roce 1996 však byla objevena vada v návrhu MD5, to však neodradilo mnoho lidí, a tak je stále používaná až do roku 2004, kdy byly nalezeny daleko větší chyby, které zapříčinily upouštění od tohoto algoritmu v bezpečnostních aplikacích. Kryptologové z Japonska a posléze i z České republiky jej označili za nebezpečný, neboť objevili, zpětný algoritmus, kterým je možné vysledovat přibližnou množinu zpětných znaků – to pak má za následek menší časovou náročnost pro výpočet zpětného hashe i klasickým PC – pořád se sice jedná o velké množství kombinací – závislé na délce použitého hesla, ale i tak je to zjednodušení pro prolomení takto uloženého hesla.

Největší bezpečnostní riziko však spočívá ve stále rostoucím výkonu výpočetní techniky, a webech, zabývajících se prolamováním MD5. [22]

Z toho vyplývá, že jednoduché heslo není problém odhalit, pokud má člověk přístup do databáze. Proto je nutné zabezpečit databázi CMS Drupal před neoprávněným

přístupem. Také je dobré, aby si uživatelé systému vybírali hesla, která nejdou takto jednoduše prolomit. Aby si nezadávali jednoduchá hesla.

5.2.4 Zabezpečení přístupu do databáze

Systém si sám vytvoří soubor, ve kterém jsou uložena hesla k databázi a systém kontroluje, zda jsou dobře nastavena práva k tomuto souboru a případně tyto práva sám opraví. Administrátorovi systému je také k dispozici na stránce hlášení stavu informace zda je soubor chráněný.

5.2.5 Identifikace již přihlášeného uživatele

Identifikace již přihlášeného uživatele probíhá v CMS Drupal pomocí session. Identifikátor session je generován pomocí funkce jazyka php `session_id()`, které využívá k vytvoření jedinečného řetězce identifikujícího session hashovací algoritmus md5. Je velmi nepravděpodobné, aby více uživatelů bylo stejně identifikováno a stejně je obtížné odhalit cizí řetězec. Tento způsob lze z hlediska bezpečnosti označit jako dobrý.

5.2.6 Práva přístupu k systému

Každý uživatel má jasně stanovená práva, co smí v systému dělat. Existuje 1 uživatel, který má absolutní práva, může měnit cokoli a může spravovat účty ostatních uživatelů. Uživatel s maximálními právy je vytvořen při instalaci systému. Tento uživatel se označuje jako administrátor systému.

Dále je možné vytvářet skupiny uživatelů a jim přidělovat práva k určitým částem systému. Pro každou skupinu lze nastavit, zda může vykonávat určitá práva. Uživatelé mohou k systému přistupovat anonymně nebo se mohou registrovat. Registrace není omezená, ale je možné ji nastavit tak, aby bylo nutné jí schválit administrátorem systému. Práva lze přidělovat k jednotlivým modulům nebo k jejich částem. To závisí na složitosti modulu.

Lze editovat přímo tento soubor nebo lze využít webového rozhraní, kdy na překladu může spolupracovat více lidí.

Soubor s překladem je nutné naimportovat do Drupalu, pomocí modulu Locale v administrační sekci. Tento modul data ze souboru uloží do databáze, z které pak bude využívat data systém Drupal. V systému ve verzi 5.x se zdrojové texty ukládají do tabulky locales_source a překlady se ukládají do tabulky locales_target. Soubor .po slouží pro načtení dat modulem.

5.2.8 WYSIWYG editor

Pro systém Drupal lze použít WYSIWYG editor TinyMCE. Editor umožňuje nastavit každé skupině uživatelů určitý profil. Toto umožňuje určitým uživatelům omezit funkčnost editoru. Pro každý profil lze nastavit, která tlačítka v editoru se mají zobrazovat. Pro systém Drupal existuje modul, který umožňuje integraci TinyMCE do systému. Tento editor nahrazuje formulářové pole <textarea> a může fungovat v kterémkoliv tomto poli.

Editor TinyMCE je stejně jako celý Drupal Opensource. Editor je napsaný v JavaScriptu. Editor lze využívat i mimo systém Drupal. Může fungovat i v jiných CMS. [23]

5.2.9 Vzhled systému Drupal

Pro úpravu vzhledu využívá systém Drupal Motivů vzhledu (anglicky Templates). Instalace již existujících motivů vzhledu je jednoduchá. V administračním rozhraní stačí vybraný motiv aktivovat a určit ho jako výchozí. Jednotlivé motivy vzhledu je nutné před jejich instalací nahrát do složky themes/, kde jsou všechny dostupné motivy vzhledu.

Složka s motivem vzhledu obsahuje soubory s příponou .tpl.php, které definují vzhled jednotlivých sekcí webu.

Téma vzhledu (skin) je zjednodušeně řečeno sada souborů, které předefinovávají funkce theme_něco, ať už přímo nebo pomocí šablonovacích systémů, a dodávají další elementy vzhledu, jako například obrázky, CSS soubory a podobně. Skin může být

definován buď čistě v PHP nebo pomocí jednoho z několika šablonovacích systémů, jak bude uvedeno níže.

Když Drupal v průběhu zpracovávání stránky narazí na příkaz generovat zdrojový kód stránky, tedy na příkaz `theme(něco, ...)`, postupuje následovně:

- Pokud není používán šablonovací systém, podívá se do složky `skinu` a pokusí se najít předdefinování funkce `theme_něco()`.
- Pokud je používán šablonovací systém, Drupal ho požádá, aby dodal výsledný kód stránky. Šablonovací systém může projít uživatelsky definované soubory nebo se v případě neúspěchu podívat do vlastních výchozích implementací, má-li nějaké - to vše záleží čistě na něm.
- Pokud šablonovací systém žádný výstup nevrátí, Drupal použije výchozí implementaci funkce `theme_něco()`, která je obvykle uložena v souboru `includes/theme.inc` nebo ji dodává příslušný modul (všimněte si, že pomocí funkcí `theme_něco()` lze snadno skinovat také výstup generovaný moduly třetích stran, nikoli jen jádra Drupalu).

Vraťme se nyní k příkladu drobečkové navigace. Pokud chci změnit výchozí oddělovač `"»"` například na `">"`, základní postup je následující:

1. Vytvořím složku nového skinu v adresáři `themes/`, například `themes/mujskin`.
2. Do této složky přidám soubor `mujskin.theme`, který bude obsahovat běžný PHP skript (musí tedy obsahovat vsuvky `<?php` a `?>`).
3. Definuji novou funkci `mujskin_breadcrumb()`, která až na záměnu `theme ==>` `mujskin` bude mít stejnou signaturu jako funkce původní. Dobrým přístupem je okopírovat také původní kód, který u některých funkcí může být dost komplikovaný, a provést pouze požadované změny. Výsledný kód bude následující:

```
function theme_breadcrumb($breadcrumb)
{
    if (!empty($breadcrumb))
    {
        return '<div class="breadcrumb">'. implode(' > ',
        $breadcrumb) .'
```

```
}  
}
```

V dokumentaci k Drupal API najdete seznam skinovacích funkcí. Další podrobnosti popisuje stránka Plain PHP themes. Obvykle je jednodušší využít služeb nějakého šablonovacího systému.

5.2.9.1 PHPTemplate

Drupal podporuje řadu šablonovacích systémů, například XTemplate nebo Smarty, od verze 4.7 je však jako výchozí používán PHPTemplate. Základní ideou PHPTemplate je podporovat vkládání dynamického výstupu přímo přes PHP vsuvky bez nutnosti učit se nějakou speciální syntaxi, což je přístup pro řadu lidí jistě sympatický.

PHPTemplate je šablonovací systém vytvořený přímo pro CMS Drupal.

PHPTemplate pro předefinování skinovacích funkcí používá soubory pojmenované podle schématu `název_funkce.tpl.php`, přičemž je ve výchozím stavu možno použít pět následujících souborů. Názvy souborů jsou schodné se skinovací funkcí `themes_název_funkce()`.

- *page.tpl.php* - skinuje stránku jako celek, používá se pro úpravu celé stránky
- *block.tpl.php* - skinuje blok v postranním sloupci
- *box.tpl.php* - skinuje box používaný v hlavní oblasti s obsahem
- *comment.tpl.php* - skinuje oblast komentářů
- *node.tpl.php* - skinuje obsah nódu

V každém skinovacím souboru je možné využít předdefinovaných proměnných, které Drupal nahradí dynamickými částmi výstupu. Proměnné jsou zobrazovány pomocí php scriptu `<?php print $název_promenne; ?>`. Části kódu, které nejsou mezi značkami `<?php ?>` se odešlou na výstup vždy. Seznam dostupných proměnných lze nalézt v dokumentaci systému Drupal. [24]

V každé složce motivu vzhledu je soubor `style.css`, který určuje styly jednotlivých elementů webu. Lze pomocí něho do značné míry ovlivnit vzhled webu. Dále se ve složce mohou nacházet podsložky s obrázky a dalšími prvky webu. Obrázky se mohou nacházet i v kořenové složce motivu vzhledu. Také se v jeho kořenové složce nachází

soubor screenshot.png, který obsahuje náhled motivu vzhledu a zobrazuje se v administrační části Drupalu. [24]

5.2.9.2 Změna barev v motivu vzhledu

Některé motivy vzhledu umožňují měnit v administračním rozhraní svoje barevné schéma. Při změně barev se vytvoří v adresáři files/color/ (umístěn v kořenovém adresáři Drupalu) podadresář *jmeno_motivu_vzhledu-řetězec*. *Řetězec* je doplňkem názvu adresáře, který zajišťuje jeho jednoznačnou identifikaci.

Tyto motivy mají ve svém kořenovém adresáři adresář color/ a v něm soubor color.inc, ve kterém se definují jednotlivá barevná schémata pro určitý motiv vzhledu. Také se v něm definují obrázky, které se nemají přepsat při změně barevného schématu. Dále tento soubor obsahuje definici obrázků v jednotlivých vrstvách a jejich polohu. Dále obsahuje definici souborů, které budou na pozadí a nebo budou přeneseny beze změny.

V souboru color.inc je nastavený alfa kanál obrázku base.png (base.png je v adresáři color/). Pomocí alfa kanálu se určuje průhlednost v png obrázku.

Soubor base.png slouží ke generování motivu vzhledu. Pro generování náhledu motivu v administraci. Náhled se zobrazuje pod správou barev motivu vzhledu.

5.2.10 Úprava vzhledu v TinyMCE

TinyMCE umožňuje v administračním rozhraní nastavení možnosti využívat vlastní CSS soubor. Ve výchozím nastavení používá soubor editor_content.css umístěný v adresáři modules/tinymce/tinymce/jscripts/tiny_mce/themes/simple/css.

V iframe který generuje editor se k html kódu připojí soubor se styly z motivu vzhledu (soubor style.css). Lze tedy i pomocí tohoto souboru ovlivnit vzhled TinyMCE.

5.3 CMS Joomla

CMS Joomla je svobodný open source CMS pro účely publikování informací na internetu a intranetu. Je napsána v jazyce PHP a využívá databázi MySQL. Joomla! podporuje caching, indexaci stránek, RSS, tisknutelné verze stránek, zobrazování

novinek, blogy, diskusní fóra, hlasování, kalendář, vyhledávání v rámci webserveru, lokalizace a vícejazyčné verze. [25]

5.3.1 Historie

Joomla vycházela z CMS Mambo, které bylo původně vytvořeno v Melbourne (Austrálie) sídlící společnosti pojmenované Miro, v roce 2000. Zrodilo se v době, kdy na komunitních webech neohroženě vládl phpNuke. [26]

Miro zveřejnilo svůj kód nejdříve pod jménem Mambo Site Server a později jako Mambo Open Source ve verzi 3 v dubnu 2001 pod licencí GNU GPL. V této době až do poloviny roku 2002 bylo Miro jediným vývojářem Mamba, přidávalo však spíše jen opravy a bezpečnostní záplaty než nové funkce. Ve stejné době, v průběhu roku 2001, Miro také vyvinulo komerční odnož Mamba a později vydalo “Mambo 2002” v lednu 2002. V této době je Mambo 2002 a Mambo Open Source velmi podobné. [26]

Později v roce 2002 bylo Miro kontaktováno Robertem Castleyem, který viděl obrovský potenciál v aplikaci a byl ustanoven šéfem projektu. Castley převzal vývoj Mambo Open Source a dovedl jej k verzi 4 beta, ustanovením malého týmu nadšených a stejně smýšlejících vývojářů podporujících projekt.

Na začátku roku 2003 Miro dalo ruce pryč a zodpovědnost za kód přešla plně na Open Source project Development Team. Je to ve chvíli, kdy byly dokončovány práce na sérii 4.0 a bylo přepracováváno schéma a architektura kódu. Mezitím Miro pokračuje v práci na svém komerčním souputníkovi nezávisle na Open Source projektu a oba základy kódu se začínají výrazněji lišit. Je velmi důležité zmínit, že ani jedna verze spolu není kompatibilní.

Verze 4.5 byla vytvořena v prosinci 2003 s mnoha změnami, které můžeme považovat za samostatnou odvozeninu a originální Miro kód byl při úpravách v podstatě nahrazen. A je to právě tato doba, kdy se Mambo objevilo v hledáčku a bylo stále více spojováno s kvalitou a popularitou.

V roce 2004 Linux Format udělil Mambu ocenění “Best Free Software Project” (Nejlepší projekt svobodného software) roku a Linux User and Developer jej jmenoval “Best Linux or Open Source Software” (Nejlepší linuxový nebo open source software).

MamboForge.net je představeno v březnu 2004 díky frustraci z funkcemi nabízenými na SourceForge.net a také z důvodu stále většího zájmu trhu vývojářů třetích stran. Kromě toho nabídka Mambo šablon je nejbohatší a vzbuzuje závist různých dalších redakčních systémů v té době nabízených.

Září roku 2004 přineslo vydání verze 4.5.1, které zásadně rozšířilo funkční vylepšení a posílilo správce.

Pak, po následné rezignaci Robert Castleye na konci roku 2004 a jmenování jeho nástupce Andrew Eddieho, se Mambo Development Team a Miro rozhodli opět spojit a spolupracovat výkonněji a více formálně. Je ustaven The Mambo Steering Committee (Výbor pro řízení Mamba) ze zástupců Mira a Mambo Development Teamu.

V březnu 2005 byla vydána verze 4.5.2 s dobrými výsledky při vylepšování funkčnosti a stability bez radikálních změn ve vzhledu a ovládání rozhraní.

Rok 2005 přinesl více významnějších ocenění jako je “Best Open Source Solution” (Nejlepší open source řešení) a “Best of Show - Total Industry Solution” (Nejlepší na výstavě – Úplné průmyslové řešení) na LinuxWorld Boston a “Best Open Source Solution” (Nejlepší open source řešení) na LinuxWorld San Francisco.

Započaly práce na verzi 4.5.3, které urychlily mezinárodní podporu pro nyní již známého Mambo správce, vylepšily podporu přístupových úrovní a mnoho dalšího.

Bohužel, v srpnu 2005 se objevily spory v rozhodnutích uvnitř Mambo Steering Committee. Celý Core Team (základ vývojářského týmu) se rozhodl ukončit spolupráci s Miro a opustil projekt. Seskupili se pod hlavičkou Open Source Matters.

Po vypulírování a přípravách, přivedl Open Source Matters v září 2005 na svět Joomla! 1.0. V následujících měsících také Mambo zreformovalo svůj tým a podpůrné struktury, přerušilo práce na originální verzi 4.5.3 a soustředilo se na aktuální stabilní verzi základního kódu. [26]

5.3.2 Původ názvu

Název joomla je anglický fonetický přepis svahilského slova „jumla“, které znamená „všichni dohromady“ nebo „v celku“. Výslovnost slova Joomla je „džumla“. Tento

název byl vybrán jako závazek vývojářského týmu a komunity k tomuto projektu. První verze projektu Joomla! (Joomla! 1.0.0) byla vydána 16. září, 2005. Tato verze byla identická s produktem Mambo 4.5.2.3, byly pouze opraveny některé bezpečnostní chyby. Ve vývojovém plánu bude podle vývojářů Joomla! 2.0 kompletně přepsána do jazyka PHP 5. [25]

5.3.3 Bezpečnost CMS Joomla!

Tato kapitola se zabývá nejdůležitějšími bezpečnostními problémy v CMS Joomla!. Kromě toho zde budou řešeny další bezpečnostní hlediska jako například ukládání hesla uživatele a ověřování jeho identity. Také možnosti nastavení práv v systému.

5.3.3.1 Bezpečnostní chyby v CMS Joomla!

V této kapitole je uveden výpis některých bezpečnostních chyb, které se vyskytují v různých verzích CMS Joomla!. V kapitole budou uvedeny chyby objevené především v nedávné době. A také zde bude uveden způsob řešení daných bezpečnostních chyb.

Chyba v nastavení cookie

Když budete provozovat systém Joomla! zabezpečený pomocí SSL (celá webová stránka bude využívat SSL), Joomla! nenastaví SSL flag cookies. Toto umožní komukoliv monitorovat síť a najít cookie patřící k určité session. Všechna data pak nebudou přenášena bezpečně.

Postižené jsou verze 1.5.8 a nižší, u kterých běží celé stránky v režimu zabezpečení pomocí SSL a není k nim možný žádný nezabezpečený přístup.

Řešením je aktualizovat na poslední verzi systému Joomla! (alespoň 1.5.9) a nastavit volbu `force_ssl` v globálním konfiguračním souboru. Nebo je také možné aktivovat nastavení PHP `session.secure_cookie` v souboru `.htaccess` nebo v souboru `php.ini` a všechny verze systému Joomla! se pak budou řídit tímto nastavením. [27]

Chyba v generátoru náhodných čísel

Existuje mezera v generování náhodných čísel, která velmi snižuje entropii (neurčitost) v systému použití náhodných funkcí. To má účinek na generování tokenů a hesel. Oprava zvětšující entropii velmi snižuje šanci na uhádnutí tokenu. [28]

Pro generování náhodných hesel je dobré nahradit stávající metodu `JUserHelper::genRandomPassword()` metodou `Jutility::getHash(JuserHelper::genRandomPassword())`, která zajišťuje vyšší náhodnost hesel.

Tuto chybu mají verze řady 1.5.x ve verzi 1.5.6 a nižší.

Řešením je aktualizovat systém Joomla! na novější verzi (1.5.7 nebo novější). [28]

5.3.4 Přístupová práva k systému

Po instalaci systému existuje jeden uživatel, který má veškerá práva k systému. Tento uživatel nemá nijak svá práva omezena. Jednotliví registrovaní uživatelé se člení do skupin, jimž lze přidělovat určitá práva. V systému existují dvě základní a nezbytné skupiny uživatelů, veřejný a registrovaní. Ve skupině registrovaní jsou všichni uživatelé, kteří se zaregistrovali. Přístupová práva (přístupové úrovně) jsou v CMS Joomla! hierarchická. To znamená, že pokud je k modulu přiděleno nejnižší přístupové právo (veřejný), pak to znamená, že k modulu mají přístup i registrovaní a všechny další skupiny, které jsou výše v hierarchii.

Přístupová práva jednotlivým skupinám se nastavují v seznamu modulů nebo v seznamu zásuvných modulů. Každá komponenta má vlastní nastavení přístupových práv. Každá se může rozdělovat na různé moduly, jimž lze nastavit přístupová práva.

5.3.5 Překlady

Překlady systému Joomla! se nachází v adresáři `language/` nacházející se v kořenovém adresáři webu. Obdobně se nachází překlad administrátorské sekce v adresáři `administrator/language/`. V obou těchto adresářích se nachází pro každý jazyk jedna složka, která se jmenuje podle kódu jazyka. V této složce se nacházejí 2 XML soubory: soubor `instal.xml` a `kod_jazyka.xml`. Kód jazyka je specifický kód pro daný jazyk,

například pro češtinu se tento soubor jmenuje cs-CZ.xml. Struktura souborů v administrátorské sekci a na hlavních stránkách je shodná.

V souboru instal.xml se nachází informace o autorovi překladu, o názvu jazyka, o jeho kódu, kontakt na autora a další informace. Důležitou součástí tohoto souboru je seznam názvů souborů s překladem.

```
<?xml version="1.0" encoding="UTF-8"?>
<install type="language" version="1.5" client="site">
  <name>Czech (Cestina)</name>
  <tag>cs-CZ</tag>
  <version>1.5</version>
  <creationDate>2008-01-22</creationDate>
  <author>Svatopluk Vít</author>
  <authorEmail>svata@joomlaportal.cz</authorEmail>
  <authorUrl>www.joomla.cz</authorUrl>
  <copyright>>(C) 2005 - 2008 Open Source Matters. All rights
reserved.</copyright>
  <license>http://www.gnu.org/licenses/gpl-2.0.html
GNU/GPL</license>
  <description>Czech language files for Joomla! 1.5
Site</description>
  <files>
    <filename file="meta">cs-CZ.xml</filename>
    <filename>cs-CZ.ini</filename>
    <filename>cs-CZ.ignore.php</filename>
    <filename>cs-CZ.com_banners.ini</filename>
    <filename>cs-CZ.com_contact.ini</filename>
    <filename>cs-CZ.com_content.ini</filename>
    <filename>cs-CZ.com_frontpage.ini</filename>
    <filename>index.html</filename>
  </files>
  <params />
</install>
```

V ukázce je část souboru instal.xml. Pro přehlednost jsou vynechány některé názvy souborů s jazykovým překladem.

Soubor cs-CZ.xml obsahuje informace o kódování znaků a například o písmu pro export do formátu pdf.

V souborech s příponou ini se nachází překlad textů zobrazovaných uživateli. Soubory obsahují na každém řádku překlad jednoho výstupního textu ve formátu identifikátor_řetězce=překlad řetězce. Identifikátor jednoznačně určuje výstupní řetězec. Identifikátor je psán velkými písmeny.

```
# $Id: cs-CZ.plg_content_vote.ini 8234 2007-07-28 23:16:01Z svatas $
# Joomla! Project
# Copyright (C) 2005 - 2008 Open Source Matters. All rights reserved.
```

```
# License http://www.gnu.org/licenses/gpl-2.0.html GNU/GPL, see  
LICENSE.php  
# Note : All ini files need to be saved as UTF-8 - No BOM
```

```
ADD THE VOTING FUNCTIONALITY TO ITEMS=Přidá možnost hlasování v  
anketách do článků  
BEST=Nejlepší  
POOR=Nejhorší  
RATE=Hodnotit  
USER RATING=Hodnocení uživatelů
```

Obsah souboru cs-CZ.plg_content_vote.ini. Řádky začínající znakem # jsou komentáře.

5.3.6 WYSIWYG editor

Pro systém Joomla! existuje zabudovaný WYSIWYG editor založený na TinyMCE. Lze využít i jiný editor například FCK editor nebo JCE editor. Pro instalaci jiného WYSIWYG editoru je nutné do systému Joomla nainstalovat doplňkový modul. Ve výchozím nastavení je zapnut editor TinyMCE. V administrátorské sekci webu sekci webu lze toto nastavení změnit.

5.3.7 Vzhled systému Joomla!

Vzhled systému Joomla! lze měnit pomocí šablon. Šablony jsou v CMS Joomla! označovány anglickým termínem templates.

Všechny šablony najdete ve složce templates v kořenovém adresáři CMS Joomla!. Každý template má v adresáři templates/ vlastní složku, která se jmenuje stejně jako jeho název. Všechny templaty pro administrátorské rozhraní jsou ve složce administrator/templates.

Nejběžnější je mít v templatě minimálně 4 soubory:

index.php - poskytuje zobrazení a pozici modulů a komponentů.

template.css - ovládá prezentační aspekty templaty včetně specifikací mezer, fontů, okrajů obrázků, formátování seznamů atd.

templateDetails.xml - zde jsou zapsané meta-informace přidružené k templatě a používá ho installer a Template Manager.

template_thumbnail.ext - nahraďte .ext s formátem obrázku (.jpg, .png, .gif)

To je obecně 200x150 px obrázek, který se ukazuje, když kurzorem najedete na jméno šablony v Template Manageru.

V souboru templateDetails.xml jsou různá meta-data použitá v Template Manageru, při instalaci a pro údržbu. Je dobré rozeznat sekce tohoto souboru.

Počáteční zobrazení v Template Manageru ukazuje seznam dostupných šablon a základní informace o nich. Všechny tyto informace jsou umístěny v souboru templateDetails.xml. [29]

```
<install version="1.5" type="template">
  <name>rhuk_milkyway</name>
  <creationDate>11/20/06</creationDate>
  <author>Andy Miller</author>
  <authorEmail>rhuk@rockettheme.com</authorEmail>
  <authorUrl>http://www.rockettheme.com</authorUrl>
  <copyright></copyright>
  <license>GNU/GPL</license>
  <version>1.0.2</version>
  <description>TPL_RHUK_MILKYWAY</description>
```

Všechny soubory přidružené k šabloně zde musí být zapsány. A to takovým způsobem, že informace o souboru obsahuje plnou cestu začínající v rootu šablony. Installer používá tyto informace, když ukládá soubory během instalace. [29]

Malá ukázka z šablony rhuk_milkyway:

```
<files>
  <filename>index.php</filename>
  <filename>templateDetails.xml</filename>
  <filename>template_thumbnail.png</filename>
  <filename>params.ini</filename>
  <filename>images/arrow.png</filename>
  <filename>images/indent1.png</filename>
</files>
```

Některé šablony mohou zahrnovat podporu různých jazyků (překlady) statického textu v šabloně. Dva soubory jsou ukázány pod tímto odstavcem. V prvním jsou data, které uvidí uživatel, ve druhém jsou texty, které uvidí administrátor.

```
<languages>
  <language tag="en-GB">en-GB.tpl_beez.ini</language>
</languages>
<administration>
  <languages folder="admin">
    <language tag="en-GB">en-GB.tpl_beez.ini</language>
  </languages>
</administration>
```

Zde jsou definované dostupné pozice modulů, které jsou v šabloně použity.

```
<positions>
  <position>breadcrumb</position>
  <position>left</position>
  <position>right</position>
  <position>top</position>
  <position>user1</position>
  <position>user2</position>
  <position>user3</position>
  <position>user4</position>
  <position>footer</position>
  <position>debug</position>
  <position>syndicate</position>
</positions>
```

Parametry šablony nabízí možnost zobrazit volby, které mohou být zvoleny z administrátorského rozhraní v Template Manageru. Na příklad šablony `rhuk_milkyway` dovoluje administrátorovi změnit barvy rámečků, pozadí a definovat šířku stránky. Pro úpravu vzhledu v administraci je nutné mít práva k zápisu do souboru `params.ini` v adresáři daného šablony. [29]

```
<params>
  <param name="colorVariation" type="list" default="white"
label="Color Variation" description="Color variation to use">
    <option value="blue">Blue</option>
    <option value="red">Red</option>
    <option value="green">Green</option>
    <option value="orange">Orange</option>
    <option value="black">Black</option>
    <option value="white">White</option>
  </param>
</params>
```

Soubor `index.php` je kostra stránky. Každá stránka produkovaná Joomla! je zobrazena v `index.php`, ale s různým obsahem.

Soubor `index.php` zahrnuje směs kódu, který bude doručen prohlížeči tak, jak je a PHP kód, který bude modifikován před doručení. Kód obsahuje jednoduché HTML doplněné o příkazy PHP.

HTML kód přizpůsobený šablonovacímu systému CMS Joomla!

```
<head>
<jdoc:include type="head" />
<link rel="stylesheet" href="<?php echo $this->baseurl ?>
templates/mytemplate/css/css.css" type="text/css" />
</head>
```

Tedy místo toho, aby hlavička byla definovaná v souboru index.php, je generovaná z databáze pomocí kousku PHP kódu. Uživateli se bude zobrazovat pouze HTML a případně javascript. [29]

6 Porovnání vybraných CMS

Porovnání obou systémů bude provedeno tak, že bude porovnána složitost instalace obou systémů. Dále bude porovnána dostupnost potřebných modulů tak, aby systém splňoval danou funkcionalitu. Bude sledována dostupnost modulu pro tvorbu e-shopu, fotogalerie, diskuzního fóra a v neposlední řadě pro tvorbu novinek a článků.

Nejprve budou popsány porovnávané vlastnosti jednotlivých systémů. Popis porovnávaných vlastností se nachází v této kapitole z důvodů přehlednosti. Shrnutí bude provedeno ve formě tabulky, ze které bude vycházet výsledné hodnocení a porovnání obou systémů. Každá vlastnost bude ohodnocena na 5 bodové stupnici, kdy nejlepší hodnocení je 5 a nehorší jeden bod. Systém s více body bude vyhodnocen jako lepší.

U každé vlastnosti jsou stanovena kritéria, podle kterých se hodnotí, jestli je daná vlastnost lepší nebo horší. Kritéria jsou popsána v tabulce v následující kapitole.

6.1 Porovnávané vlastnosti obou systémů

Pro porovnání obou systémů bude porovnáována kvalita obou CMS v následujících vlastnostech:

Porovnávaná vlastnost	Hodnocená kritéria
Instalace systému	Počet kroků instalace, funkčnost systému po instalaci
Nastavení čistých URL	Složitost nastavení čistých URL, nutnost editace souboru .
Přidání modulu	Složitost přidávání modulu, počet kroků potřebných pro přidání modulu
Přidání článků	Složitost přidávání článků, jejich zobrazení na úvodní stránce
WYSIWYG editor	Počet úkonů nutných pro spuštění WYSIWYG editoru
Instalace a provoz e-shopu	Složitost instalace elektronického obchodu.

Porovnávaná vlastnost	Hodnocená kritéria
Instalace systému	Počet kroků instalace, funkčnost systému po instalaci
Tvorba anket	Integrace anket s ostatními články na webu.
Instalace a provoz fotogalerie	Instalace modulu fotogalerie. Počet dostupných galerií.
Správa uživatelů	Jednotnost správy uživatelů (společná správa uživatelů)

6.2 Systém Drupal

V této kapitole jsou popsány porovnávané vlastnosti systému Drupal.

6.2.1 Instalace

Instalace systému Drupal probíhá v následujících krocích. Nejprve je nutné nahrát soubory na server, kde se bude systém provozovat.

Kroky instalace:

- Výběr jazyka, pokud je k dispozici více jazykových verzí Drupalu.
- Ověření správného nastavení systému. Ověření nastavení práv k souborům a nastavení webového serveru a php.
- Nastavení přístupových údajů k databázovému serveru a nastavení jména používané databáze.
- Instalace webu a nastavení překladu. Tyto kroky proběhnou automaticky. V tomto kroku proběhne vytvoření tabulek v databázi a případně jejich naplnění daty (ne u všech tabulek).
- Nastavení webu, nastavení názvu webu. Jména a emailové adresy správce webu a hesla správce webu. A také emailu webu, na který budou chodit upozornění například o registraci nového uživatele. Případně lze v tomto kroku aktivovat využití čistých URL.
- Dokončení instalace a spuštění nového webu.

6.2.2 Nastavení čistých URL

Před nastavením čistých URL je nutné upravit soubor `.htaccess` (vyskytuje se přímo takto pojmenovaný). Je potřeba odkomentovat a upravit volbu `RewriteBase` tak, aby obsahovala adresář s umístěním webu.

Po tomto nastavení je pro využití čistých URL potřeba změnit jednu volbu v administraci. Volba půjde aktivovat pouze, pokud je správně nastavený `.htaccess` a server ho podporuje.

6.2.3 Přidání modulu

Pokud chceme přidat do CMS Drupal nějaký modul, je potřeba soubor s kódem modulu rozbalit do adresáře `modules/` nebo lépe do adresáře `sites/all/modules/` (kvůli oddělení přidávaných modulů od výchozích). Poté je potřeba modul aktivovat v administračním rozhraní v sekci Prvky webu, Moduly.

6.2.4 Přidání článků

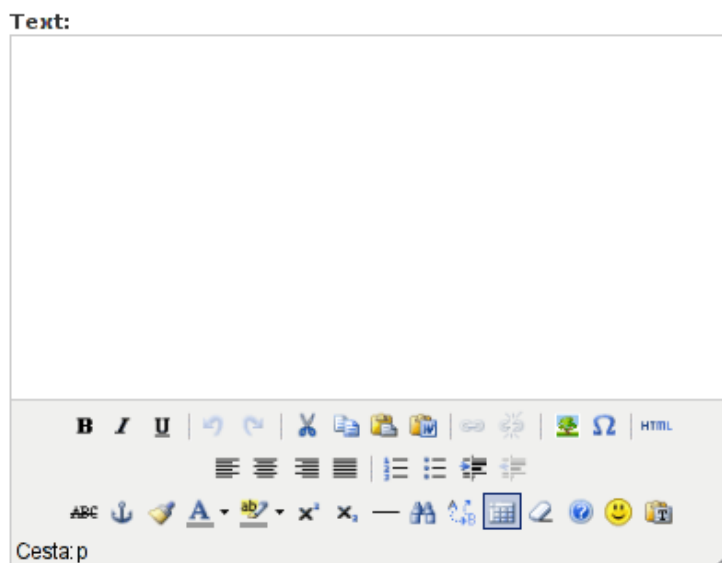
Pokud chceme zařadit článek do určité kategorie, je potřeba nejprve vytvořit slovník a spojit ho s články a pak přidat nějaké kategorie do daného slovníku.

Článek lze vytvořit po přihlášení v sekci vytvořit obsah, článek. V tomto formuláři lze upravit veškeré volby související s článkem. Článek se přidává v jediném kroku.

6.2.5 WYSIWYG editor

Pro využití WYSIWYG editoru je potřeba do systému Drupal nainstalovat nový modul. Lze například využít editor TinyMCE. Pro jeho využití je potřeba kromě modulu do systému Drupal stáhnout a nahrát do modulu vlastní editor. Modul jenom umožňuje využití editoru v CMS Drupal. A umožňuje jeho nastavení pro různé skupiny uživatelů (například aktivaci a deaktivaci tlačítek).

Samotný systém Drupal neobsahuje ve výchozí instalaci WYSIWYG editor.



Obr. č. 2 - WYSIWYG editor TinyMCE ve formuláři pro vkládání článků.

6.2.6 Instalace a provoz e-shopu

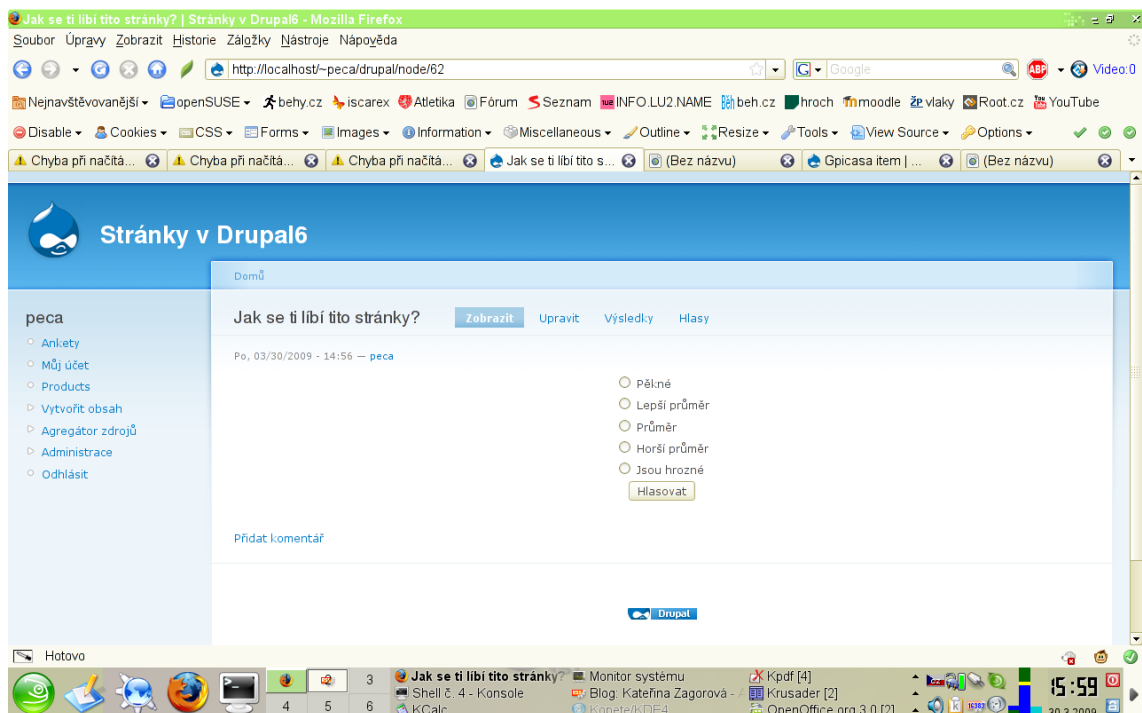
Pro využití e-shopu v CMS Drupal je nejprve nutné instalovat nový modul nebo více modulů. Pro provoz e-shopu v CMS existuje více modulů. Bylo rozhodnuto, že využijí modul E-commerce. E-commerce se skládá z 12 oddělených modulů. Moduly jsou rozdělené na jádro a další doplňkové moduly. Pokud se Drupal nachází ve výchozím stavu po instalaci, je potřeba doinstalovat ještě další moduly. Jejich seznam je u instalace modulů e-commerce.

Poté již je možné přidávat produkty a spravovat e-shop. E-shop se spravuje v administraci a produkty se přidávají podobně jako články.

6.2.7 Tvorba anket

Před vytvořením ankety v CMS Drupal je potřeba nejprve aktivovat modul Poll. A poté se anketa přidává jako kterýkoliv jiný článek. U ankety lze zadat libovolný počet odpovědí na vybranou otázku a odpovědí, lze i stanovit jejich výchozí hodnotu.

Anketa se v CMS zobrazuje na stejné úrovni jako ostatní články. Zobrazuje stejně jako jakýkoliv jiný článek.



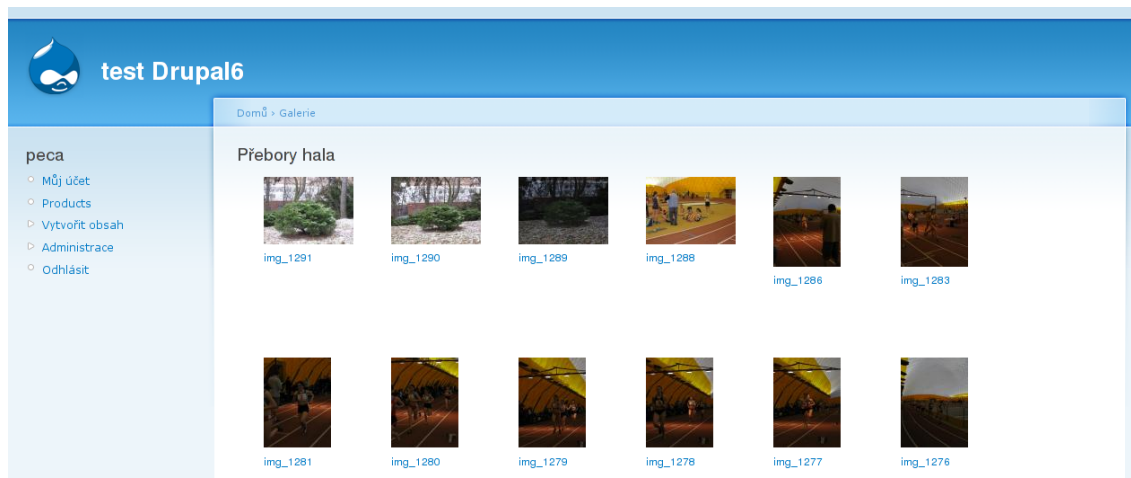
Obr. č. 3 - Příklad anketu vytvořené pomocí CMS Drupal

6.2.8 Instalace a provoz fotogalerie

Pro provoz fotogalerie existuje v CMS Drupal více modulů. Pro bližší popis byl vybrán modul image. Byl vybrán z důvodů snadné instalace a dostupnosti pro verzi 6.x CMS Drupal.

Před instalací modulu image je dobré vytvořit adresář image v adresáři sites/default/files a pak vytvořit adresář pro hromadný import obrázků. Modul image se skládá z 5 modulů, každý z nich se stará o nějakou část práce s obrázky. Po jejich instalaci běžným způsobem je potřeba vytvořit nějakou galerii. Galerie je z hlediska CMS Drupal podobná kategoriím článku a tedy je někdy nutné přiřadit galerie k obrázkům. Obrázky jsou na podobné úrovni jako články, přidávají se obdobným způsobem.

Modul image umožňuje hromadný import obrázků do galerie. Před jejich importem se obrázky nahrají nejčastěji pomocí FTP serveru do určeného adresáře. Poté se v administraci provede vlastní import obrázků.



Obr. č. 4 - Fotogalerie v CMS Drupal vytvořená pomocí modulu image

6.2.9 Správa uživatelů

CMS Drupal umožňuje přiřazovat uživatelům libovolné role a to i více rolí jednomu uživateli. V systému existují speciální role - anonymní a přihlášený uživatel. Role přihlášený uživatel je automaticky přiřazena uživateli, který se v Drupalu zaregistruje. Jednotlivým rolím lze pak přiřazovat práva v rámci jednotlivých modulů nebo jejich částí.

V CMS Drupal ještě existuje speciální uživatel, který veškerá práva k systému. Tento uživatel se vytváří po instalaci.

6.3 Systém Joomla!

V této kapitole jsou popsány porovnávané vlastnosti systému Joomla!. Jedná se o stejné vlastnosti jako v předcházející kapitole.

6.3.1 Instalace

Instalace CMS Joomla! probíhá v následujících krocích.

Nahrání stránek na server. Pro lokalizaci je nutné vytvořit adresář v adresáři languages pod názvem "cs-CZ" a do něho zkopírovat lokalizační soubory.

Vlastní instalace, po nahrání souborů, probíhá v 7 krocích:

- Výběr jazyka

- Kontrola správného nastavení webového serveru
- Odsouhlasení licence. Joomla! je šířena pod licencí GNU/GPL verze 2
- Nastavení připojení k databázi. Určení jména databázového serveru, jména uživatele, hesla k databázi a určení názvu databáze pro systém Joomla!
- Nastavení FTP, pro lepší přístup pro práci se soubory.
- Nastavení názvu stránky a emailu a hesla účtu správce.
- Ukončení instalace, je nutné odstranit instalační složku "instalation/"

Po instalaci je nutné nastavit práva zápisu do adresářů, například pro uploadované soubory nebo pro instalaci nových modulů. Seznam adresářů, u kterých je potřeba nastavit práva pro zápis, lze nalézt v administraci v sekci Systémové informace, Práva složek. Adresářů je oproti systému Drupal více.

Pro využití českého překladu v CMS Joomla! je potřeba nahrát překlad administrátorského rozhraní a vlastních veřejně přístupných stránek do adresáře language v adresáři administration/ nebo v kořenovém adresáři webu.

6.3.2 Nastavení čistých URL

Pro nastavení čistých URL pro lepší SEO je v systému Joomla! nejprve nutné upravit soubor htaccess.txt tak, aby se změnil jeho název na .htaccess. Je potřeba v něm upravit volbu RewriteBase tak, aby obsahovala cestu adresáři, kde je umístěn na webu systém Joomla!.

V dalším kroku je potřeba aktivovat volbu v konfiguraci v administraci volbu "Search Engine Friendly URLs" a "Use Apache mod_rewrite".

6.3.3 Přidání modulu

Po stažení souboru s modulem se modul instaluje přes administrátorské rozhraní, ve kterém se do systému nahraje soubor s vybraným modulem. V případě, že tato instalace selže, lze nahrát soubory s kódem modulů přímo do určitých adresářů na webu přes FTP. Po nahrání je potřeba spustit instalační script zadáním speciální URL, které by měla být určena v nápovědě k modulu.

V CMS Joomla! lze přidávat komponenty, moduly a zásuvné moduly. Tyto mají podobný význam jako moduly v CMS Drupal, ale v určitých věcech se liší. Instalace všech probíhá stejným způsobem.

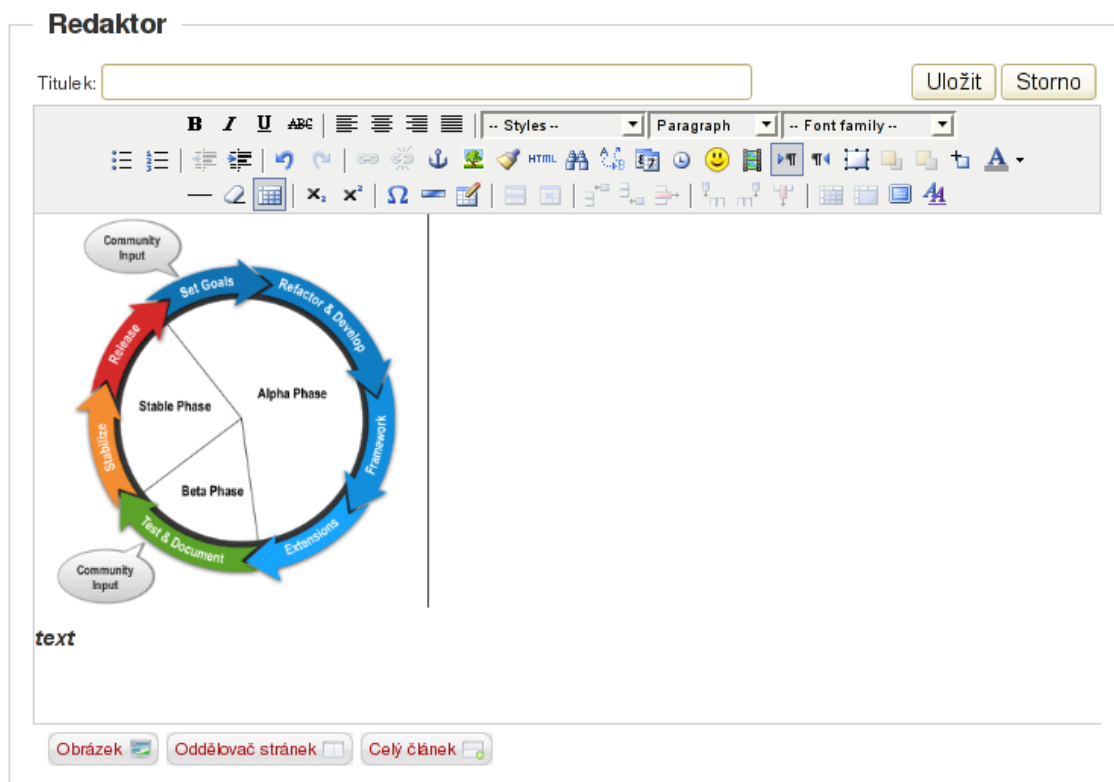
6.3.4 Přidání článků

Přidání článků do systému probíhá v následujících krocích:

- Vytvoření potřebné kategorie (pokud článek bude chceme mít v nějaké kategorii)
- Vytvoření článku a jeho zařazení do kategorie.
- Případné vytvoření odkazu na kategorii článků v menu.

6.3.5 WYSIWYG editor

Systém Joomla! již po korektně provedené instalaci nabízí pro editaci článků WYSIWYG editor TinyMCE. Editor je plně funkční. WYSIWYG editor je spuštěn ve výchozím nastavení.



Obr. č. 5 - Výchozí WYSIWYG editor v CMS Joomla!

6.3.6 Instalace a provoz e-shopu

Pro vytvoření elektronického obchodu lze v CMS Joomla! využít modulu VirtueMart. Tento modul lze instalovat přímo přes administrátorské rozhraní systému nebo v případě selhání lze modul nainstalovat manuálně přímým nahráním souborů modulu do příslušných adresářů Joomla. Po instalaci modulu lze do elektronického obchodu vložit ukázková data, která pomohou se správou obchodu.

6.3.7 Tvorba anket

Komponenta pro tvorbu ankety je v CMS Joomla! po instalaci již připravena k použití. U každé ankety je možné zvolit až 12 otázek. Je možné i zvolit interval jednotlivých odpovědí, kdy bude možné vícekrát odpovědět z jedné IP adresy. Vícenásobná možnost se využije pro uživatele, kteří jsou za NAT a používají jednu veřejnou IP adresu.

Pro zveřejnění ankety je potřeba přidat anketu v modulu Polls, který se ve výchozím nastavení zobrazuje na titulní stránce a zároveň je nutné aby anketa byla veřejná.

6.3.8 Instalace a provoz fotogalerie

Pro využití v CMS Joomla! bylo naprogramováno hodně modulů, které lze využít pro tvorbu fotogalerie. Dle www.joomla.org existuje 42 modulů pro tvorbu fotogalerie. Pro další popis byla vybrána komponenta Joom gallery. Komponenta byla vybrána především z důvodů dobrého hodnocení uživateli na webových stránkách www.joomla.org.

Komponenta umožňuje různé způsoby dávkového nahrávání fotografií. Lze využít Java upload, dávkový upload (fotografie se nahrají v jednom komprimovaném .zip souboru) nebo FTP upload. FTP upload je podobný fotogalerii v CMS Drupal. Aby se fotografie mohly zařadit do nějaké galerie, je potřeba tuto galerii před jejich uploadem vytvořit a při nahrávání fotografií zvolit danou galerii.

Pro uveřejnění fotogalerie na titulní stránce CMS Joomla! je potřeba přidat odkaz na fotogalerii do menu.

6.3.9 Správa uživatelů

Uživatelé se v CMS Joomla! člení do rolí. Uživatelské role nelze v Joomla přidávat. Lze k jednotlivým uživatelům role přiřazovat, ale nelze přímo v administrátorském rozhraní vytvořit nové uživatelské role. Práva k jednotlivým modulům a případně jejich částem se přiřazují uživatelským rolím přímo u nastavení modulu. V CMS Joomla není centrální správa přístupových práv.

6.4 Shrnutí porovnání

Porovnávaná vlastnost	Lepší v systému(CMS):		Zdůvodnění volby
	Drupal	Joomla!	
Instalace systému	4	3	Instalace CMS Drupal probíhá v méně krocích než instalace CMS Joomla!.
Nastavení čistých URL	4	2	Nastavení čistých URL je jednodušší v systému Drupal, stačí na to méně kroků a méně změn v administraci.
Přidání modulu	3	5	V CMS Joomla! lze přidat modul přímo v administraci bez nutnosti ručně nahrávat soubory.
Přidání článků	4	3	V systému Drupal je možné přidávat články ihned po instalaci systému. V CMS Joomla! je nejprve nutné vytvořit kategorie.
WYSIWYG editor	2	5	V CMS Joomla! je WYSIWYG editor připravený k použití hned po instalaci.
Instalace a provoz e-shopu	3	4	V CMS Joomla! je jednodušší instalace elektronického obchodu. Na rozdíl od Drupalu k tomu dostačuje jeden modul.
Tvorba anket	5	3	V CMS Drupal je jednodušší publikace ankety na hlavní stránce webového sídla, publikuje se podobně jako článek.
Instalace a provoz fotogalerie	3	4	V CMS Joomla! je větší výběr modulů zajišťujících správu fotografií.
Správa uživatelů	5	1	V Drupalu je jednotná správa nastavení uživatelských práv a lze vytvářet uživatelské role.
Souhrn	33	30	

Tabulka hodnocení porovnání jednotlivých vlastností obou systémů.

Z tabulky vyplývá, že lze lépe hodnotit systém Drupal, ale rozdíl mezi oběma systémy není příliš výrazný. Oba systémy se liší v určitých vlastnostech. V některých je lepší CMS Joomla! a v jiných je lepší CMS Drupal.

7 Tvorba modulu pro vybraný systém

Pro tvorbu modulu byl vybrán systém Drupal z důvodu, že systém Drupal byl vyhodnocen jako lepší než systém Joomla.

Systém Drupal neobsahuje modul pro správu fotogalerií, který by dosahoval podobných kvalit a možností jako má správce fotogalerií plicasa od firmy Google. Proto bylo rozhodnuto, že modul bude spojovat webové stránky vytvořené systémem Drupal a správce fotogalerií Google plicasa. Modul bude zobrazovat náhledy fotografií z dané fotogalerie. Ke každé fotogalerii bude připojen její nadpis a popis.

7.1 Funkce které bude modul realizovat

Modul bude zajišťovat následující funkce:

- Přidávání, úpravy a mazání galerií.
- Zobrazení 10 nejnovější galerií.
- Správa galerií podobná jako správa článků.
- Automatické nebo ruční obnovení fotogalerie.
- Nastavení počtu náhledů zobrazovaných na jednom řádku tabulky.
- Správa galerií bude podobná jako správa článků, galerie se budou zobrazovat spolu s články na úvodní stránce.

7.2 Implementace modulu

Zdrojový kód modulu se skládá ze třech základních souborů. Soubory se jmenují stejně jako název modulu, ale liší se svojí příponou. Soubory mají příponu .info, .install a .module. Vytvářený modul se bude jmenovat gplicasa.

Při tvorbě modulu v CMS Drupal se využívají hook funkce. Hook funkce jsou funkce jejichž název obsahuje název modulu a název konkrétní hook funkce oddělené podtržítkem. Hook funkce se může například jmenovat gplicasa_install(). Hook funkce zavolá Drupal vždy při určité činnosti, když vyžaduje od modulu nějaké informace nebo je předpokládána nějaká funkčnost modulu. Funkce gplicasa_install() je zavolána při instalaci modulu.

Kromě zmíněných tří souborů obsahuje ještě modul gpicasa další soubory v adresáři, které zajišťují zpracování RSS kanálu z webu Google do správné podoby. Také zajišťují získání potřebných informací, jako je datum a čas poslední změny fotogalerie. Pro vlastní parsování RSS kanálu je využito již hotové třídy, které šířena pod licencí GPL a lze jí využít pro modul.

7.2.1 Obsah souboru gpicasa.info

V souboru gpicasa.info jsou informace o názvu a o verzi modulu. V tomto souboru se nachází i popis modulu. Mohou se zde nacházet i informace o modulech, které jsou nutné pro instalaci modulu. Informace z tohoto souboru se zobrazují v administraci CMS Drupal v sekci, kde je možné aktivovat i deaktivovat rozšiřující moduly.

Zdrojový kód souboru gpicasa.info:

```
; $Id: gpicasa.info,v 0.1 2009/03/08 06:10:18 Peca Exp $
name = "Gpicasa"
description = "Aggragate photos form google picasa web album"
core = "6.x"
version = "6.x-0.1-beta"
```

Druhý řádek označuje název modulu, další popis modulu. Kód core="6.x" Určuje pro kterou verzi CMS Drupal je daný modul vytvořen. V tomto případě je modul vytvořen pro Drupal 6.

7.2.2 Obsah souboru gpicasa.install

Soubor gpicasa.install obsahuje informace nutné pro instalaci a smazání modulu.

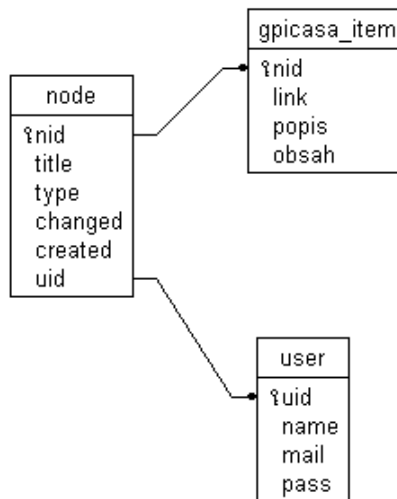
Soubor obsahuje hook funkci gpicasa_install(), tato funkce se spouští při instalaci modulu.

```
/**
 * Implementation of hook_install().
 */
function gpicasa_install()
{
  //vytvoreni tabulek v db a pripadne dalsich dat a konstant
  drupal_install_schema("gpicasa");
  variable_set('row_num',3); //vychozi nastaveni promene
}
```

Funkce `drupal_install_schema("gpicasa");` vytvoří tabulku v DB dle funkce `gpicasa_schema()`;

```
function gpicasa_schema()
{
  //vytvoreni tabulek v DB
  $schema=array();
  $schema["gpicasa_item"]=array(
    "description" => t("Table of google picasa feeds"),
    "fields" => array(
      'nid' => array(
        'description' => t('The primary identifier for a node.'),
        'type' => 'serial',
        'unsigned' => TRUE,
        'not null' => TRUE),
      'link' => array(
        'description' => t('Link to photos RSS'),
        'type' => 'varchar',
        'length' => '500',
        'not null' => TRUE,
        'default' => ''),
      'popis' => array(
        'type' => 'text',
        'length' => '2000'),
      'obsah' => array(
        'type' => 'text',
        'not null' => TRUE),
    ),
    "unique keys" =>array(
      'nid' => array('nid'),
    ),
    'primary key' => array('nid')
  );
  return $schema;
}
```

Funkce vrací asociativní pole složené s asociativních polí. Každá položka v poli znamená jednu tabulku v databázi. Pole "fields" v položce pole obsahuje jednotlivé atributy dané relace (tabulky). U každého pole je specifikovaný jeho datový typ a další potřebné vlastnosti.



Obr. č. 6 - ERD modulu s vazbami na CMS Drupal. Modul přidává tabulku *gpicasa_item*. Tabulky *user* a *node* nemají z důvodů přehlednosti zobrazeny všechny atributy.

Funkce `gpicasa_uninstall()` se vykoná při odinstalování modulu.

```

function gpicasa_uninstall()
{
    //mazani vseh zaznamu node
    $result=db_query("SELECT nid FROM {gpicasa_item}");
    while ($pole_vys=db_fetch_array($result))
    {
        //mazani nodu
        node_delete($pole_vys['nid']);
    }
    //zruseni tabulky v DB
    drupal_uninstall_schema('gpicasa');
}
  
```

Funkce nejprve vymaže všechny přidané fotogalerie. Odstranění databáze tabulky v databázi by nebylo dostatečné, protože se při data o fotogaleriích ukládají i v jiných tabulkách, které jsou využívány pro články. Po vymazání fotogalerií je provedeno odstranění tabulky v databázi.

Funkce `db_query()`; zajišťuje provedení dotazu nad databází. Dotaz může být proveden nad různými databázemi. Funkce i zajišťuje ochranu proti SQL injection.

7.2.3 Obsah souboru gpicasa.module

Soubor gpicasa.module obsahuje vlastní kód modulu, který je potřebný pro jeho správnou funkci. V této kapitole budou vysvětleny důležité funkce obsažené v souboru. Celý zdrojový kód lze nalézt v příloze práce.

V úvodní části souboru jsou definovány důležité konstanty a je k souboru připojen soubor s knihovnou realizující parsování RSS nebo ATOM kanálu. Kód tohoto souboru není přímo závislý na CMS Drupal a lze ho využít i jinde.

```
define('ADMIN_ONE_ITEM','administer one gpicasa item'); //name of one
gpicasa admin item
define('DEF_ROW_NUM',3); //výchozí počet obrázků na řádku
require_once "inc/magpie_simple.php"; //php knihovna pro nacteni
ATOM(RSS) feed
```

Důležitou hook funkcí v souboru gpicasa.module je funkce gpicasa_menu(). Funkce zajišťuje přiřazení konkrétních zobrazovaných stránek k určitým relativním odkazům CMS Drupal. Struktura menu v CMS Drupal je hierarchická.

```
function gpicasa_menu()
{
  $items = array();
  $items['admin/content/gpicasa'] = array(
    'title' => t('Gpicasa'),
    'description' => t('List of all of the gpicasa feed'),
    'access callback' => '_gpicasa_test_access',
    'access arguments' => array(ADMIN_ONE_ITEM,'administer all
gpicasa'),
    'page callback' => 'gpicasa_list_item'
  );
  $items['admin/content/gpicasa/%'] = array(
    'title' => t('Gpicasa admin one feed'),
    'description' => t('Admin one Google picasa feed'),
    'page callback' => 'gpicasa_list_one_item',
    'page arguments' => array(3),
    'access callback' => '_gpicasa_test_access',
    'access arguments' => array(ADMIN_ONE_ITEM,'administer all
gpicasa'),
    'type' => MENU_CALLBACK
  );
  $items['admin/gpicasa/ref/%'] = array(
    'title' => t('Gpicasa refresh item'),
    'page callback' => 'gpicasa_refresh_feed',
    'page arguments' => array(3),
    'access arguments' => array('administer all gpicasa'),
    'type' => MENU_CALLBACK
  );
  $items['admin/settings/gpicasa'] = array(
    'title' => t('Gpicasa settings'),
    'description' => t('Set basic variables'),
```

```

    'page callback' => 'drupal_get_form',
    'page arguments' => array('gpicasa_settings'),
    'access callback' => '_gpicasa_test_access',
    'access arguments' => array(ADMIN_ONE_ITEM, 'administer all
gpicasa')
  );
  return $items;
}

```

Funkce vrací asociativní pole složené z asociativních polí, kde každá položka je jednou z položek, které je přiřazena nějaká funkce. Může a nemusí být zobrazována v menu. V řetězcovém indexu je uvedena relativní drupal cesta k položce menu. Důležitým prvkem položky menu je 'page callback', který značí funkci, která je zavolána při navštívení položky a stará se o zobrazení obsahu. Pomocí položky 'page arguments' jsou předávány argumenty funkce určené v položce 'page callback'.

Důležitým prvkem je i 'access callback', který určuje jaká funkce bude zavolána pro kontrolu přístupových práv. Výchozí funkcí je funkce Drupalu `user_access()`. Obsahem prvku 'access arguments' je pole argumentů předávané funkci kontrolující práva.

Položky se rozdělují na několik typů. Typ položky je určen pomocí prvku pole 'type'. Výchozím typem je `MENU_NORMAL_ITEM`, který označuje, že se položka zobrazí v navigaci nebo v menu v administrátorské sekci. Typ `MENU_CALLBACK` značí, že se položka nezobrazuje v navigaci. Tento typ slouží pro určení funkce, která se vykoná na daném relativním odkazu.

7.2.3.1 Funkce popisující položku obsahu

Další důležitou součástí souboru `gpicasa.module` je soubor hook funkcí starající se přidávání obsahu a další práci s obsahem.

```

//implementation of hook_node_info
function gpicasa_node_info()
{
  return array(
    'gpicasa' => array(
      'name' => t('Gpicasa item'),
      'module' => 'gpicasa',
      'description' => t('Gpicasa item is ATOM(RSS) feed from google
picasa web album'),
      'title_label' => t('title label'),
      'has_body' => FALSE,
    )
  );
}

```


Hook funkce `gpicasa_node_info()`; zajišťuje vytvoření nového typu obsahu. Určuje například jméno typu obsahu, jeho popis a kterého modulu se daný typ obsahu týká. Tyto informace se zobrazují na stejném místě, na kterém se vytváří články.

```
//implementation of hook_access, access to node
function gpicasa_access($op, $node, $account)
{
    global $user;
    if (($op=='create') || ($op=='delete') || ($op=='update'))
    { //create node
        if ((user_access(ADMIN_ONE_ITEM) && ($user->uid == $node->uid))
|| user_access('admin all gpicasa'))
        {
            return TRUE;
        }
        else return FALSE;
    }
    else
    {
        return TRUE;
    }
}
}
```

Funkce `gpicasa_access($op, $node, $account)`; určuje přístupová práva k danému typu obsahu. Parametr `$op` určuje typ operace s obsahem. Pokud funkce vrátí TRUE, tak lze s obsahem provádět danou operaci. Tato funkce vyžaduje mít administrátorská práva k operacím delete, update a create, ale pro zobrazení obsahu žádná práva nejsou vyžadována.

```
/**
 * Implementation of hook_form().
 */
function gpicasa_form(&$node, $form_state) {
    global $user;
    // name field
    $form['title'] = array(
        '#type' => 'textfield',
        '#title' => t('Feed name'),
        '#size' => 60,
        '#maxlength' => 100,
        '#default_value' => $node->title,
        '#weight' => -10,
    );
    //normal text field
    $form['link']=array(
        '#type' => 'textfield',
        '#title' => t('Feed'),
        '#size' => 60,
        '#maxlength' => 500,
        '#description' => t('Link to ATOM(RSS) feed from gpicasa
album'),
        '#required' => TRUE,
```

```

        '#default_value' => $node->link,
        '#weight' => -9
    );
    //popis fotogalerie
    $form['popis'] = array(
        '#type' => 'textarea',
        '#title' => t('Feed description'),
        '#cols' => 60,
        '#rows' => 10,
        '#default_value' => $node->popis,
        '#weight' => -7
    );
    return $form;
}

```

Funkce `gpicasa_form()`; je funkcí, která rozšiřuje nebo upravuje formulář pro vkládání typu obsahu. Funkce vrací asociativní pole, kde každá položka specifikuje jedno pole formuláře. V tomto případě je formulář rozšiřován o tři položky title, link a popis. Title a link jsou jednořádkové textové položky a popis je víceřádkový text. Typ položky formuláře `textarea`.

Obr. č. 7 - Formulář na vkládání nových fotogalerií. Druhá položka formuláře slouží pro vložení odkazu na RSS kanál fotogalerie.

```

//implementation of hook_load
function gpicasa_load($node)
{ //nacteni obsahu feedu
  $dotaz="SELECT obsah,popis,link FROM {gpicasa_item} WHERE nid=%d";
  $object=db_fetch_object(db_query($dotaz,$node->nid));
  return $object;
}

```

Hook funkce `gpicasa_load($node)`; určuje, jaké další položky mají být načteny při načtení daného obsahu. Funkce vrací objekt, jehož atributy jsou položky rozšiřující typ obsahu. Zde funkce vrací objekt rozšířený o `link`, `obsah` a `popis`.

```

//implementation of hook_validate

```

```

function gpicasa_validate($node,&$form)
{
  if (!ereg('http://picasaweb\.google\.com/data/feed/base/user/.*',
$node->link))
  {
    form_set_error('link',t('You have to well-fill feed element'));
  }
}

```

Hook funkce `gpicasa_validate($node,&$form)` ověřuje správnost dat vyplněných ve formuláři. Pro vložení fotogalerie Google pिकासa je nutné vložit odkaz ve správném formátu.

```

//implemantation of hook_node
function gpicasa_insert($node)
{
  //global $user;
  $hodnoty=gpicasa_fetch_feed($node->link);
  $dotaz="INSERT INTO {gpicasa_item} (nid,link,obsah,popis) VALUES
(%d,'%s','%s','%s)";
  db_query($dotaz,$node->nid,$node->link,$hodnoty['obsah'],$node->
popis);
}

```

Hook funkce `gpicasa_insert($node)` rozšiřuje proces vkládání typu obsahu. Zde je navíc uládán obsah, odkaz(link) a popis do tabulky modulu `gpicasa_item`.

```

// Implementation of hook_update()
function gpicasa_update($node)
{
  //aktualazice feedu
  $hodnoty=gpicasa_fetch_feed($node->link);
  $dotaz="UPDATE {gpicasa_item} SET link='%s', obsah='%s', popis='%s'
WHERE nid=%d";
  db_query($dotaz,$node->link,$hodnoty['obsah'],$node->popis,$node->
nid);
}

```

Hook funkce `gpicasa_update($node)`; rozšiřuje proces aktualizace obsahu. Funkce přidává aktualizaci položek v tabulce `gpicasa_item`.

```

//implementation of hook_view
function gpicasa_view($node, $teaser = FALSE, $page = FALSE,
$block=FALSE)
{
  if ($teaser)
  { //strucny vypis bez obrazku, pouze popis souboru
    $telo['#value']=$node->popis;
    $node->content['teaser']=$telo;
  }
  else
  {
    $telo['#value']=$node->popis."<br />".$node->obsah;
    $node->content['body']=$telo; //prevede obsah do tela dokumentu
  }
}

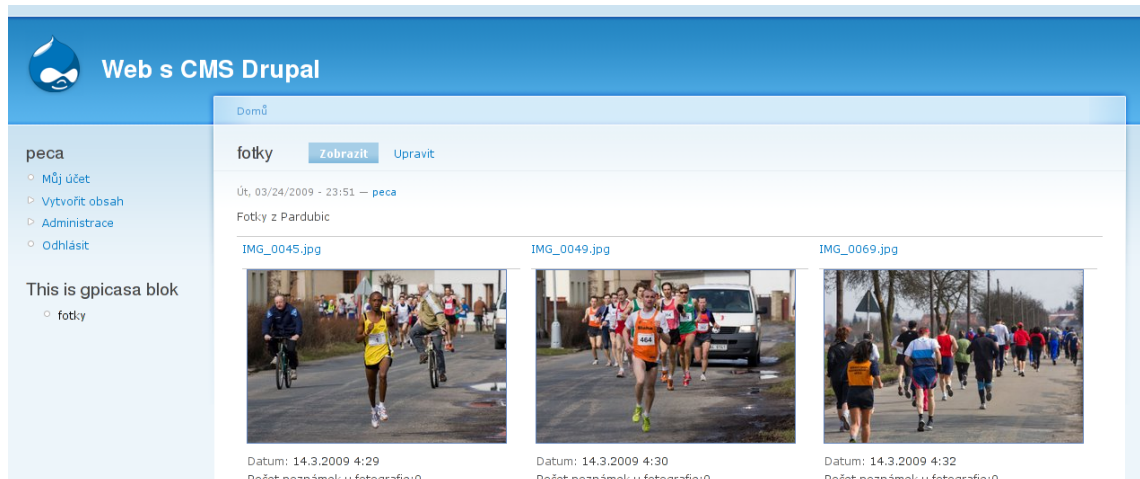
```

```

return $node;
}/**/

```

Funkce `gpicasa_view()`; určuje, jak se má typ článku zobrazit. Funkce vrací objekt, kde prvkem určujícím obsah je asociativní pole, kde klíčová hodnota '#value' určuje obsah, který se má zobrazit. Nejprve je nadefinováno stručné zobrazení na úvodní stránce a potom obdobný výpis.



Obr. č. 8 - Vzhled vložené fotogalerie. Počet fotogalerií na řádku lze nastavit v administraci.

```

//implementation of hook_delete
function gpicasa_delete
{ //smazání obsahu
  db_query("DELETE FROM {gpicasa_item} WHERE nid=%d", $node->nid);
}

```

Funkce `gpicasa_delete($node)`; určuje jaké operace mají být provedeny při smazání fotogalerie. Při jejím smazání je navíc provedeno smazání řádku v tabulce `gpicasa_item`.

7.2.3.2 Další důležité funkce

V souboru `gpicasa.module` se nachází ještě další důležité funkce nezbytné pro správnou funkci modulu.

```

//implementation of hook_cron
//this function run always WHEN cron running...
function gpicasa_cron()
{
  $result=db_query("SELECT changed,link,{gpicasa_item}.nid FROM
{gpicasa_item} INNER JOIN {node} ON {node}.nid={gpicasa_item}.nid");
  while ($pole_vys=db_fetch_array($result))
  { //obnova vseh feedu
    //načtení hlavičky ze servru googlu
    $hodnoty=gpicasa_fetch_header($pole_vys['link']);
    $zmeneno=strtotime($hodnoty['change']);
  }
}

```

```

if (($pole_vys['changed'])<$zmeneno)
{ //je potreba update feedu, zatim jenom obsah
  $hodnoty=gpicasa_fetch_feed($pole_vys['link']);
  $dotaz="UPDATE {gpicasa_item} SET obsah='%s' WHERE nid=%d";
  db_query($dotaz,$hodnoty['obsah'],$pole_vys['nid']);
  //aktualizace data zmeny
  $dotaz="UPDATE {node} SET changed=%d WHERE nid=%d";
  db_query($dotaz,date(),$pole_vys['nid']);
}
}
}

```

Funkce gpicasa_cron(); se vykoná vždy, když je spuštěn skript cron.php nacházející se v kořenovém adresáři CMS Drupal. Pokud na serveru, kde je web umístěn, běží cron démon lze zajistit automatické periodické spouštění tohoto skriptu. Cron lze spustit i ručně přímo z administračního rozhraní CMS.

Zde funkce aktualizuje fotogalerie, pokud byly změněny. Změnu si skript zjistí načtením hlavičky RSS kanálu. Pokud je potřeba obnovit fotogalerii, je provedeno načtení dat ze serveru googlu a aktualizace potřebných položek v tabulkách node a gpicasa_item.

```

//implementation of hook_block
function gpicasa_block($op = 'list', $delta = 0, $edit = array())
{ //blok generuje 10 nejnovějších RSS feedu
  $blok_title=t('This is gpicasa blok');
  switch ($op)
  {
  case 'list':
    $block[0]["info"] = t($blok_title);
    return $block;
    break;
  case 'view':
    //zde bude generovaniobsahu modulu
    $dotaz="SELECT title,{node}.nid,popis FROM {gpicasa_item} INNER
JOIN {node} ON {node}.nid={gpicasa_item}.nid ORDER BY {node}.created
DESC LIMIT 10";
    $result=db_query($dotaz);
    $blok['subject']=$blok_title;
    $pom=0;
    $blok['content']='<ul>';
    while ($pole_vys=db_fetch_array($result))
    {
      $blok['content'].='<li>'.l($pole_vys["title"],"node/"
$pole_vys["nid"],array('attributes' => array('title' =>
drupal_substr($pole_vys['popis'],0,50)))."</li>"; //link na konkretni
feed
      $pom=1;
    }
    if (!$pom)
    {
      $blok['content'].='<li>'.t('Nothing feed add')."</li>\n";
    }
  }
}

```

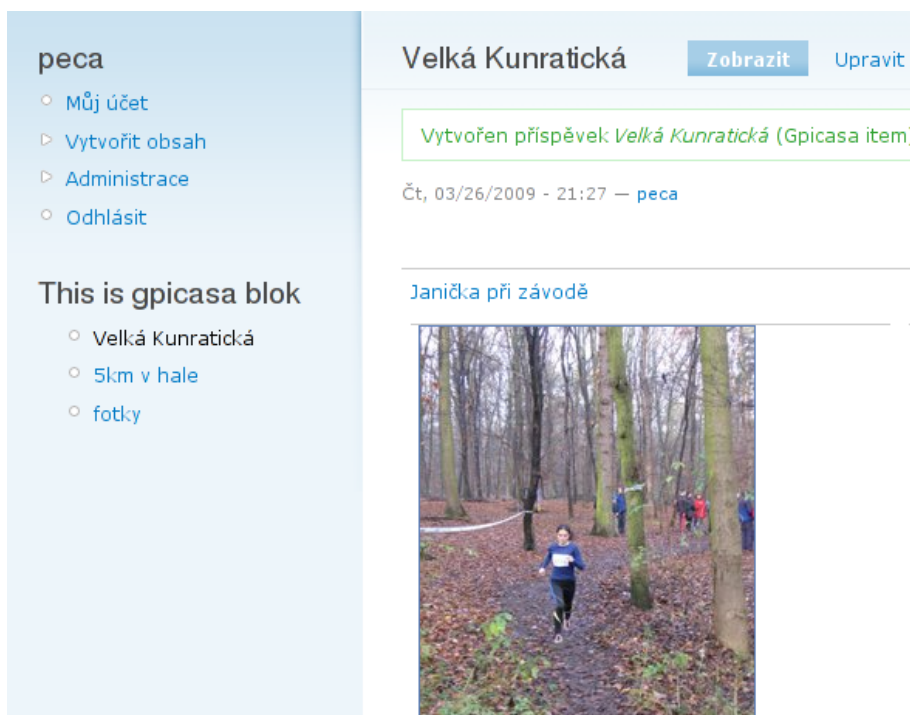
```

    }
    $blok['content'] .= '</ul>';
    return $blok;
    break;
}
}

```

Hook funkce `gpicasa_block($op = 'list', $delta = 0, $edit = array());` zajišťuje potřebné informace pro vytvoření bloku. Funkce vrací asociativní pole, které může mít dva rozměry, pokud má modul více bloků. Index pole 'content' určuje obsah bloku. Index 'subject' určuje předmět bloku a index 'info' určuje informace o bloku, které budou zobrazovány v seznamu bloků. Parametr `$op` určuje typ operace, která je s blokem prováděna. Když se parametr `$op` rovná 'list', tak toto určuje, jaké informace budou o bloku zobrazeny v seznamu bloků. Pokud se rovná 'view', tak určuje, co je zobrazováno, pokud je blok povolen.

Blok modulu `gpicasa` zobrazuje 10 nejnovější přidaných galerií. Po kliknutí na název galerie je zobrazena celá konkrétní fotogalerie.



Obr. č. 9 - V levé části výřezu stránky lze vidět blok modulu `gpicasa`. Blok zobrazuje 10 nejnověji přidaných galerií. Galerie jsou řazeny od nejnověji přidané.

7.3 Zobecněte přínosy zvoleného nástroje pro další možná použití

Modul pro Drupal umožňuje zjednodušit tvorbu webových stránek. Systém CMS Drupal a jeho API lze využít pro tvorbu téměř jakékoliv web side. Výhodou je, že umožňuje využít API funkcí a není nutné vše programovat od základů. Moduly lze do systému přidávat za jeho běhu. Webové stránky vytvořené v CMS Drupal jsou tedy snadno rozšiřitelné.

8 Závěr

CMS jsou v poslední době velmi využívané systémy. CMS jsou nástrojem usnadňujícím správu webů. Umožňují spravovat bez znalosti technologií. Využívají se pro provoz jak malých webových stránek, tak rozsáhlých portálů nebo elektronických obchodů. Důležitým technologickým prvkem pro fungování CMS je využití třívrstvé architektury.

Z mnoha zástupců Content management systémů byly vybrány systémy Drupal a Joomla. Byly vybrány na základě využití stejných technologií a na základě podobného zaměření obou systémů.

Dle stanovených kritérií je lépe bodově hodnocený CMS Drupal. Rozdíl mezi systémy však není výrazný a pro konkrétní nasazení může být lepší i systém Joomla!. Systém Drupal má především lepší správu uživatelů a způsob vkládání článků a dalšího obsahu. V CMS Drupal jsou lépe integrovány různé typy článků. Lepší je také instalace systému. V CMS Joomla! je lepší WYSIWYG editor a správa fotogalerií.

Byl vytvořen modul pro CMS Drupal, který umožňuje zobrazit fotogalerie z webového alba Google picasa. Umožňuje spravovat galerie podobně jako články. Modul dobře demonstruje výhody využití API Drupalu pro zjednodušení a usnadnění tvorby webových stránek.

9 Literatura

1. Co je to CMS? [online], [cit. 2.10.2008], dostupné z <<http://www.cms-systemy.cz/o--portali/co-je-cms-redakcni-system--/>>
2. CMS [online], [cit. 2.10.2008], dostupné z <<http://www.adaptic.cz/znalosti/slovnicek/cms.htm>>
3. Jan Bednařík Co je to CMS [online], [cit. 2.10.2008], dostupné z <<http://www.bednarik.org/typo3-uvod/co-je-cms/>>
4. Systém pro správu obsahu [online], [cit. 10.2.2008], dostupné z <http://cs.wikipedia.org/wiki/Redakční_systém>
5. Jiří Lukáš Co je to redakční systém? [online], [cit. 10.2.2008], dostupné z <<http://www.supersvet.cz/view.php?navezclanku=co-je-to-redakcni-system&cislocclanku=2005050501>>
6. Třívrstvá architektura [online], [cit. 10.10.2008], dostupné z <<http://www.itexpert.cz/trivrstva-architektura>>
7. Browser statistics [online], [cit. 26.3.2008], dostupné z <<http://marketshare.hitslink.com/browser-market-share.aspx?qprid=0>>
8. Netcraft Web Server Survey [online], [cit. 10.10.2008], dostupné z <<http://survey.netcraft.com/Reports/200809/>>
9. Vilém Sklenák XML a knihovny [online], [cit. 10.11.2008], dostupné z <knihovny.cvut.cz/akp/clanky/05.pdf>
10. Linuxsoft [online], [cit. 10.10.2008], dostupné z <http://www.linuxsoft.cz/article.php?id_article=267>
11. Javascript [online], [cit. 12.10.2008], dostupné z <<http://cs.wikipedia.org/wiki/Javascript>>
12. David Toth [online], [cit. 12.10.2008], dostupné z <http://64.233.183.104/search?q=cache:yw0pA_67VMUJ:webing.felk.cvut.cz/hs/download/DT-ajax-CZ-art.pdf+Princip+AJAXu+filetype:pdf&hl=cs&ct=clnk&cd=11&gl=cz&client=firefox-a>
13. Ing. Radim Smička Optimalizace pro vyhledávače - SEO [online], [cit. 10.2.2009], dostupné z <<http://knihy.root.cz/kniha/optimalizace-pro-vyhledavace-seo/stahnout/>>
14. Martin Grames Hackerem sám sobě (3) - CSRF [online], [cit. 6.2.2009], dostupné z <<http://www.chapadlo.cz/weblog/clanek/hackerem-sam-sobe-3-csrf>>
15. Relační model [online], [cit. 10.3.2009], dostupné z <http://cs.wikipedia.org/wiki/Rela%C4%8Dn%C3%AD_model>
16. **Merunka Vojtěch**, *Objektový přístup v databázových systémech*, ČZU PEF, 2002 ISBN 80-213-0882-6

17. Opensource CMS [online], [cit. 9.3.2009], dostupné z <<http://php.opensourcecms.com/>>
18. O systému Drupal [online], [cit. 20.1.2009], dostupné z <<http://www.drupal.cz/o-systemu-drupal>>
19. Drupal Multiple Vulnerabilities [online], [cit. 25.5.2008], dostupné z <<http://secunia.com/advisories/28422/>>
20. Drupal Simple Access Module Security Bypass Vulnerability [online], [cit. 25.5.2008], dostupné z <<http://www.securityfocus.com/bid/28720/discuss>>
21. Drupal core - Arbitrary code execution via installer [online], [cit. 25.5.2008], dostupné z <<http://drupal.org/node/184316>>
22. Nebezpečnost MD5 - chraňte svá hesla! [online], [cit. 25.5.2008], dostupné z <<http://www.owebu.cz/bezpecnost/vypis.php?clanek=1156>>
23. TinyMCE [online], [cit. 9.3.2009], dostupné z <<http://tinymce.moxiecode.com/index.php>>
24. Drupal - tvorba vlastního vzhledu [online], [cit. 10.1.2009], dostupné z <<http://interval.cz/clanky/drupal-tvorba-vlastniho-vzhledu/>>
25. Jiří Vaněk Co je to Joomla [online], [cit. 2.2.2009], dostupné z <<http://jirivanek.eu/co-je-to-joomla>>
26. Svatopluk Vít Jak se liší mambo a Joomla? [online], [cit. 2.2.2009], dostupné z <<http://www.joomlportal.cz/content/view/180/2/>>
27. Hanno Boeck JSession SSL Session Disclosure [online], [cit. 6.2.2009], dostupné z <<http://developer.joomla.org/security/news/287-20090101-core-jsession-ssl-session-disclosure.html>>
28. Stefan Esser Random Number Generation Flaw [online], [cit. 6.2.2009], dostupné z <<http://developer.joomla.org/security/news/272-20080902-core-random-number-generation-flaw.html>>
29. Pochopení Joomla! 1.5 templatů [online], [cit. 28.3.2009], dostupné z <<http://www.escope.cz/index.php/Template/pochopeni-joomla-templat.html>>

10 Přílohy

10.1 Vysvětlení zkratk

API (*Application Programming Interface*)

Programové rozhraní aplikace. Rozhraní pro programování nějaké aplikace. Poskytuje programátorovi funkce nebo třídy, které lze využít při programování.

HTML (*Hypertext markup language*)

Hypertextový značkovací jazyk. Vychází z jazyka SGML. Hlavní jazyk používaný pro tvorbu webových stránek.

URL (*Uniform resource location*)

Je řetězec znaků s definovanou strukturou, který slouží k přesné specifikaci umístění zdrojů informací (ve smyslu dokument nebo služba) na Internetu.

FTP (*File transfer protokol*)

Je protokol aplikační vrstvy z rodiny TCP/IP. Je určen pro přenos souborů mezi počítači, na kterých mohou běžet rozdílné operační systémy (je platformně nezávislý). Definován je v RFC 959 a rozšířen byl v RFC 2228.

WYSIWYG (*What you see is what you get*)

Typ editoru, který umožňuje vizuální editaci souboru. Při editaci je přímo vidět výsledný vzhled souboru.

JSON (*JavaScript Object Notation*)

Odlehčený formát pro výměnu dat, která mohou být organizována v polích nebo agregována objektech. Vychází ze syntaxe jazyka Javascript.

NAT (*Network address translation*)

Je zařízení pro překlad privátních IP adres. Umožňuje rozšířit počet adresovaných počítačů v protokolu IPv4.

GPL (*Genral public licence*)

Je licence pro svobodný software, původně napsaná Richardem Stallmanem pro projekt GNU. GPL je nejpopulárnějším a dobře známým příkladem silně copyleftové licence, která vyžaduje, aby byla odvozená díla dostupná pod toutéž licencí.

SŘBD (*Systém řízení báze dat*)

Je softwarové vybavení, které zajišťuje práci s databází, tzn. tvoří rozhraní mezi aplikačními programy a uloženými daty.

MD5 (*Message-Digest algorithm 5*)

Je hashovací algoritmus, který ze vstupního řetězce vytvoří unikátní řetězec délky 32 znaků. Výstupní řetězec bude mít vždy délku 32 znaků.

MD4 (*Message-Digest algorithm 4*)

Starší verze algoritmu MD5.

RSS (*Really Simple Syndication*)

RSS je kanál, který je podmnožinou XML, je určen pro poskytování obsahu. RSS umožňuje uživatelům z jediného místa sledovat informace zveřejňované na různých webech. Nástrojem pro sledování obsahu RSS jsou desktopové RSS čtečky nebo některé z mnoha webových agregátorů.

SEQUEL (*Structured English Query Language*)

Původní zkratka pro jazyk SQL. V průběhu vývoje se jazyk SEQUEL přejmenoval na SQL.

10.2 Zdrojový kód souboru gpicasa.module

```
<?php
define('ADMIN_ONE_ITEM','administer one gpicasa item'); //name of one
gpicasa admin item
define('DEF_ROW_NUM',3);
require_once "inc/magpie_simple.php"; //php knihovan pro nacteni
ATOM(RSS) feed
//implemantation of hook_help
function gpicasa_help($path,$arg)
{
    switch ($path)
    {
        case "gpicasa":
            $vystup=t("List of all google picasa feeds");
```

```

break;
case "admin/gpicasa/list":
    $vystup=t("You can see here all gpicasa rss feeds");
break;
case "node/add/gpicasa";
    $vystup=t("Gpicasa item is ATOM(RSS) feed from google
picasa<br />
    This display photo lika an artikle");
break;
case "admin/settings/gpicasa";
    $vystup=t("Setting some Gpicasa variables");
break;
}
return $vystup;
}
//implementation of hook_menu
function gpicasa_menu()
{
    $items = array();
    $items['gpicasa'] = array(
        'title' => t('All gpicasa item'),
        'page callback' => 'gpicasa_show_all_items',
        'access callback' => '_gpicasa_test_access',
        'access argument' => array('view gpicasa','access content'),
        'type' => MENU_SUGGESTED_ITEM
    );
    $items['admin/content/gpicasa'] = array(
        'title' => t('Gpicasa'),
        'description' => t('List of all of the gpicasa feed'),
        'access callback' => '_gpicasa_test_access',
        'access arguments' => array(ADMIN_ONE_ITEM,'administer all
gpicasa'),
        'page callback' => 'gpicasa_list_item'
    );
    $items['admin/content/gpicasa/%'] = array(
        'title' => t('Gpicasa admin one feed'),
        'description' => t('Admin one Google picasa feed'),
        'page callback' => 'gpicasa_list_one_item',
        'page arguments' => array(3),
        'access callback' => '_gpicasa_test_access',
        'access arguments' => array(ADMIN_ONE_ITEM,'administer all
gpicasa'),
        'type' => MENU_CALLBACK
    );
    $items['admin/gpicasa/ref/%'] = array(
        'title' => t('Gpicasa refresh item'),
        'page callback' => 'gpicasa_refresh_feed',
        'page arguments' => array(3),
        'access callback' => array('administer all gpicasa'),
        'type' => MENU_CALLBACK
    );
    $items['admin/settings/gpicasa'] = array(
        'title' => t('Gpicasa settings'),
        'description' => t('Set basic variables'),
        'page callback' => 'drupal_get_form',
        'page arguments' => array('gpicasa_settings'),

```

```

        'access callback' => '_gpicasa_test_access',
        'access arguments' => array(ADMIN_ONE_ITEM, 'administer all
gpicasa')
    );
    return $items;
}
//test prav
function _gpicasa_test_access()
{ //vraci true pokud mam alespon 1 z prav
  //global $user; //info o uzivateli
  $pole_prav=func_get_args();
  for ($i=0;$i<count($pole_prav);$i++)
  {
      if (user_access($pole_prav[$i]))
      {
          return true;
      }
  }
  return false;
}
//implementation of hook_perm
function gpicasa_perm()
{ //module has two perms
  return array(ADMIN_ONE_ITEM, 'administer all gpicasa', 'view gpicasa');
}
//implementation of hook_block
function gpicasa_block($op = 'list', $delta = 0, $edit = array())
{ //blok generuje 10 nejnovjsich RSS feedu
  $blok_title=t('This is gpicasa blok');
  switch ($op)
  {
  case 'list':
    $block[0]["info"] = t($blok_title);
    return $block;
    break;
  case 'view':
    //zde bude generovaniobsahu modulu
    $dotaz="SELECT title,{node}.nid,popis FROM {gpicasa_item} INNER
JOIN {node} ON {node}.nid={gpicasa_item}.nid ORDER BY {node}.created
DESC LIMIT 10";
    $result=db_query($dotaz);
    $blok['subject']=$blok_title;
    $pom=0;
    $blok['content']='<ul>';
    while ($pole_vys=db_fetch_array($result))
    {
        $blok['content'].='<li>'.l($pole_vys["title"],"node/".
$pole_vys["nid"],array('attributes' => array('title' =>
drupal_substr($pole_vys['popis'],0,50)))."</li>"; //link na konkretni
feed
        $pom=1;
    }
    if (!$pom)
    {
        $blok['content'].='<li>'.t('Nothing feed add')."</li>\n";
    }
  }
}

```

```

        $blok['content']..='</ul>';
        return $blok;
        break;
    }
}
//implementation of hook_cron
//this function run always WHEN cron running...
function gpicasa_cron()
{
    $result=db_query("SELECT changed,link,{gpicasa_item}.nid FROM
{gpicasa_item} INNER JOIN {node} ON {node}.nid={gpicasa_item}.nid");
    while ($pole_vys=db_fetch_array($result))
    { //zde bude refresh vsech feedu
        $hodnoty=gpicasa_fetch_header($pole_vys['link']);
        $zmeneno=strtotime($hodnoty['change']);
        if (($pole_vys['changed'])<$zmeneno)
        { //je potreba update feedu, zatim jenom obsah
            $hodnoty=gpicasa_fetch_feed($pole_vys['link']);
            $dotaz="UPDATE {gpicasa_item} SET obsah='%s' WHERE nid=%d";
            db_query($dotaz,$hodnoty['obsah'],$pole_vys['nid']);
            //aktualizace data zmeny
            $dotaz="UPDATE {node} SET changed=%d WHERE nid=%d";
            db_query($dotaz,date(),$pole_vys['nid']);
        }
    }
}
function gpicasa_list_item()
{ //list all gpicasa item
    $dotaz="SELECT title,{node}.nid FROM {gpicasa_item} INNER JOIN
{node} ON {node}.nid={gpicasa_item}.nid ORDER BY {node}.created";
    $result=db_query($dotaz);
    $text="<ul>";
    $pom=0;
    while ($pole_vys=db_fetch_array($result))
    {
        $text.="<li>".l($pole_vys["title"],"admin/content/gpicasa/"
$pole_vys['nid'])." | ".l(t("Refresh"),"admin/gpicasa/ref/"
$pole_vys['nid'])."</li>";
        $pom=1;
    }
    if (!$pom)
    { //zadny prizpevek
        $text.="<li>".t('Nothing feed add')."</li>\n";
    }
    $text.="</ul>";
    return $text; //vystup formatovany podle sablony
}
function gpicasa_list_one_item($id)
{
    $result=db_query("SELECT title,popis,obsah FROM {gpicasa_item} INNER
JOIN {node} ON {node}.nid={gpicasa_item}.nid WHERE {node}.nid=%d",
$id);
    $pole_vys=db_fetch_array($result);
    $text="<h3>".$pole_vys["title"]."</h3>";
    $text.=$pole_vys["popis"]."<br />";
    $text.=$pole_vys["obsah"];
}

```

```

    return $text;
}
//implemantation node system (add new node)
//implementation of hook_node_info
function gpicasa_node_info()
{
    return array(
        'gpicasa' => array(
            'name' => t('Gpicasa item'),
            'module' => 'gpicasa',
            'description' => t('Gpicasa item is ATOM(RSS) feed from google
picasa web album'),
            'title_label' => t('title label'),
            'has_body' => FALSE,
        )
    );
}
//implementation of hook_access, access to node
function gpicasa_access($op, $node, $account)
{
    global $user;
    if (($op=='create') || ($op=='delete') || ($op=='update'))
    { //create node
        if ((user_access(ADMIN_ONE_ITEM) && ($user->uid == $node->uid))
|| user_access('admin all gpicasa'))
        {
            return TRUE;
        }
        else return FALSE;
    }
    else
    {
        return TRUE;
    }
}

/**
 * Implementation of hook_form().
 */
function gpicasa_form(&$node, $form_state) {
    global $user;

    // $admin = user_access('administer nodes') || user_access('edit any
poll content') || (user_access('edit own poll content') && $user->uid
== $node->uid);

    // $type = node_get_types('type', $node);

    // name field
    $form['title'] = array(
        '#type' => 'textfield',
        '#title' => t('Feed name'),
        '#size' => 60,
        '#maxlength' => 100,
        '#default_value' => $node->title,
        '#weight' => -10,
    );
}

```

```

);
//normal text field
$form['link']=array(
  '#type' => 'textfield',
  '#title' => t('Feed'),
  '#size' => 60,
  '#maxlength' => 500,
  '#description' => t('Link to ATOM(RSS) feed from gpicasa
album'),
  '#required' => TRUE,
  '#default_value' => $node->link,
  '#weight' => -9
);
//popis fotogalerie
$form['popis'] = array(
  '#type' => 'textarea',
  '#title' => t('Feed description'),
  '#cols' => 60,
  '#rows' => 10,
  '#default_value' => $node->popis,
  '#weight' => -7
);
return $form;
}

//implementation of hook_load
function gpicasa_load($node)
{ //nacteni obsahu feedu
  $dotaz="SELECT obsah,popis,link FROM {gpicasa_item} WHERE nid=%d";
  $object=db_fetch_object(db_query($dotaz,$node->nid));
  return $object;
}
//implementation of hook_validate
function gpicasa_validate($node,&$form)
{
  if (!ereg('http://picasaweb\.google\.com/data/feed/base/user/.*',
$node->link))
  {
    form_set_error('link',t('You have to well-fill feed element'));
  }
}
//implemantation of hook_node
function gpicasa_insert($node)
{
  //global $user;
  $hodnoty=gpicasa_fetch_feed($node->link);
  $dotaz="INSERT INTO {gpicasa_item} (nid,link,obsah,popis) VALUES
(%d,'%s','%s','%s)";
  db_query($dotaz,$node->nid,$node->link,$hodnoty['obsah'],$node-
>popis);
}
// Implementation of hook_update()
function gpicasa_update($node)
{
  //aktualazice feedu
  $hodnoty=gpicasa_fetch_feed($node->link);

```



```

    $dotaz="UPDATE {gpicasa_item} SET link='%s', obsah='%s', popis='%s'
WHERE nid=%d";
    db_query($dotaz,$node->link,$hodnoty['obsah'],$node->popis,$node-
>nid);
}

//implementation of hook_view
function gpicasa_view($node, $teaser = FALSE, $page = FALSE,
$block=FALSE)
{
    if ($teaser)
    { //strucny vypis bez obrazku, pouze popis souboru
        $telo['#value']=$node->popis;
        $node->content['teaser']=$telo;
    }
    else
    {
        $telo['#value']=$node->popis."<br />".$node->obsah;
        $node->content['body']=$telo; //prevede obsah do tela dokumentu
    }
    return $node;
}/**/
//implementation of hook_delete
function gpicasa_delete($node)
{ //smazani obsahu
    db_query("DELETE FROM {gpicasa_item} WHERE nid=%d",$node->nid);
}
//formular pro nastaveni promene
function gpicasa_settings($form_state)
{
    $form['row_num']=array(
        '#type' => 'textfield',
        '#title' => t('Pictures in row'),
        '#description' => t('Number of pictures in one row'),
        '#size' => 60,
        '#maxlength' => 2,
        '#required' => TRUE,
        '#default_value' => variable_get('row_num',DEF_ROW_NUM).""
    );
    //prevod na retezec
    );
    $form['save']=array(
        '#type' => 'submit',
        '#value' => t('Save')
    );
    return $form;
}
//overeni formulare
function gpicasa_settings_validate($form,&$form_state)
{
    //$form_state['values']['row_num']+=0; //prevod na cislo
    if (!is_numeric($form_state['values']['row_num']))
    { //neni vlozeno cislo
        form_set_error('row_num',t('Pictures have to be number'));
    }
}
function gpicasa_settings_submit($form,&$form_state)

```

```

{
  variable_set('row_num', $form_state['values']['row_num']);
  drupal_set_message(t('Variable was save'));
}
//refresh one gpicasa item
function gpicasa_refresh_feed($nid)
{
  $pole_vys=db_fetch_array(db_query("SELECT link FROM {gpicasa_item}
WHERE nid=%d", $nid));
  $pole=gpicasa_fetch_feed($pole_vys['link']);
  $dotaz="UPDATE {gpicasa_item} SET obsah='%s' WHERE nid=%d";
  $result=db_query($dotaz, $pole['obsah'], $nid);
  //refresh na admin/content/gpicasa
  header("Location: admin/content/gpicasa");
}

```