

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

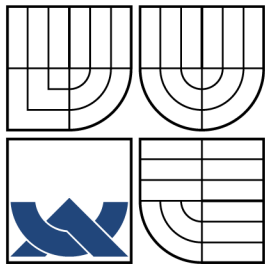
PUBLIKAČNÍ A REDAKČNÍ SYSTÉM V JAZYCE PHP

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

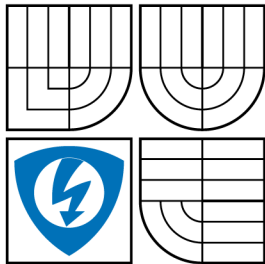
AUTOR PRÁCE
AUTHOR

PETR VELKÝ

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

PUBLIKAČNÍ A REDAKČNÍ SYSTÉM V JAZYCE PHP

PUBLISHING AND EDITORIAL SYSTEM DESIGNED WITH PHP LANGUAGE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

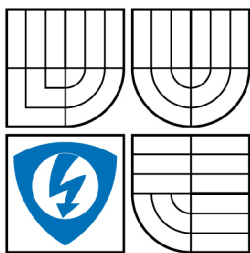
AUTOR PRÁCE
AUTHOR

PETR VELKÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAN KACÁLEK

BRNO 2009



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Petr Velký

ID: 102410

Ročník: 3

Akademický rok: 2008/2009

NÁZEV TÉMATU:

Publikační a redakční systém v jazyce PHP

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se programovacím jazykem PHP a s principy CMS systémů. Navrhněte datový model modulárního CMS systému. Při návrhu CMS systému se řiďte principy objektově orientovaného programování. Navržený systém realizujte.

DOPORUČENÁ LITERATURA:

[1] Gilmore J. W., Velká kniha PHP a MySQL 5, ZONER Press 2008, ISBN 80-86815-53-6

[2] Castagnetto J. et al., Programujeme PHP, Computer Press 2004, ISBN 80-7226-310-2

Termín zadání: 9.2.2009

Termín odevzdání: 2.6.2009

Vedoucí práce: Ing. Jan Kacálek

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

ZDE VLOŽIT PRVNÍ LIST LICENČNÍ
SMOUVY

Z důvodu správného číslování stránek

ZDE VLOŽIT DRUHÝ LIST LICENČNÍ
SMOUVY

Z důvodu správného číslování stránek

ABSTRAKT

Tato práce se zabývá vývojem redakčního systému. V úvodní části dokumentu se zabývá návrhem vlastního systému, následuje výčet použitých technologií a implementace vlastního systému.

KLÍČOVÁ SLOVA

Redakční systém, php, mysql, ajax, TinyMCE

ABSTRACT

This work deals with development of content management system. It consists of design of own system, list of used technologies and implementation.

KEYWORDS

Content management system, php, mysql, ajax, TinyMCE

VELKÝ P. *Publikační a redakční systém v jazyce PHP*. Místo: Vysoké učení technické. Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací, 2009. Počet stran 42. Počet stran příloh 3. Bakalářská práce. Vedoucí práce byl Ing. Jan Kacálek.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Publikační a redakční systém v jazyce PHP“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

(podpis autora)

Děkuji všem, kteří mi jakkoli pomáhali k úspěšnému dokončení této práce. Zvláště bych chtěl poděkovat Ing. Janu Kacálkovi a serveru grafika.cz za poskytnuté články.

OBSAH

Úvod	12
1 Redakční systém	13
1.1 Webový redakční systém	13
2 Použité technologie	14
2.1 Jazyk PHP	14
2.1.1 Historie	14
2.1.2 Seznámení s jazykem PHP	14
2.1.3 Možnosti PHP	15
2.1.4 Objektově orientované programování v PHP	15
2.1.5 Výhody a nevýhody PHP	17
2.1.6 Požadavky na systém	17
2.2 Databáze MySQL	17
2.2.1 Co je to databáze	18
2.2.2 Databáze MySQL	18
2.3 Wysiwig editor TinyMCE	19
2.3.1 Co je to wysiwyg editor	19
2.3.2 TinyMCE	19
2.4 AJAX framework jQuery UI	20
2.4.1 Co je to AJAX	20
2.4.2 Výhody a nevýhody	20
2.4.3 jQuery UI	21
3 Návrh vlastního CMS	22
3.1 Uživatelské rozhraní	22
3.2 Rozdělení rolí	23
3.3 Stručný popis jednotlivých modulů	23
3.3.1 Uživatelé	24
3.3.2 Moduly	24
3.3.3 Články	24
3.3.4 Kategorie	24
3.3.5 Komentáře	24
3.4 Životní cykly článku a komentáře	24
3.5 Datový model	26

4	Redakční systém iCMS	28
4.1	Struktura systému	28
4.1.1	Index	29
4.1.2	Jádro	29
4.1.3	Menu	30
4.1.4	Moduly	30
4.1.5	Rozšíření	31
4.1.6	Kaskádové styly	32
4.2	Moduly	32
4.2.1	Systémové moduly	33
4.2.2	Nadstavbové moduly	34
4.3	Frontend	35
5	Závěr	36
	Literatura	37
	Seznam symbolů, veličin a zkratk	38
	Seznam příloh	39
A	První příloha	40
A.1	Třída MySql	40
A.2	Třída Cms	41
B	CD příloha	42

SEZNAM OBRÁZKŮ

2.1	Wysiwyg editor TinyMCE se všemi jeho rozšířeními	19
3.1	Návrh uživatelského rozhraní redakčního systému	22
3.2	Rozdělení rolí pro jednotlivé moduly	23
3.3	Životní cyklus entity článek	25
3.4	Životní cyklus entity komentáře	25
3.5	Datový model jádra CMS	26
3.6	Datový model publikačního systému	27
4.1	Prostředí redakčního systému iCMS	28
4.2	Diagram struktury systému	29
4.3	Formulář pro přidání článku vygenerovaný pomocí šablony	31
4.4	Modální dialog pro potvrzení smazání volaný pomocí jQuery UI	32
4.5	Prohlížení článků s rolí šéfeditor	34
4.6	Frontend publikačního systému	35
A.1	Proměnné a metody třídy Mysql	40
A.2	Proměnné a metody třídy Cms	41

ÚVOD

V této práci se zabývám redakčním systémem. Snažím se popsat jak by takový redakční systém měl vypadat a co by měl umět. Postupně procházím jednotlivé technologie, které jsem použil při vývoji vlastního systému, rozebírám jejich výhody a nevýhody. Návrh vlastního systému se skládá z návrhu vzhledu uživatelského rozhraní, rozdělení rolí jednotlivým operacím, stručného popisu modulů, diagramů životních cyklů hlavních entit a z datového modelu celého systému. Poslední část práce se věnuje vlastní implementaci systému - popisuje strukturu aplikace a způsob jakým byl vytvořen modulární systém pro správu publikačního systému. Kromě redakčního systému je součástí aplikace také frontend publikačního systému.

1 REDAKČNÍ SYSTÉM

Pro CMS „Content Management System“ se často používají termíny redakční či publikační systém. Jedná se o software zajišťující správu dokumentů, uživatelů, diskuzí, souborů, obrázkových galerií a mnoho dalšího. Uživatelům daného CMS jsou pak přiřazována práva, podle kterých má možnost měnit obsah jednotlivých kategorií.

1.1 Webový redakční systém

Web CMS někdy taky WCMS je software, obvykle implementován jako webová aplikace, pro tvorbu a správu HTML obsahu. Měl by být navržen tak, aby uživatelům s žádnou nebo minimální znalostí programovacích či značkovacích jazyků, umožnil relativně jednoduše vytvářet a spravovat obsah. Data tohoto systému pak ve většině případů bývají uchovávána v databázi.

Důležité vlastnosti dobrého WCMS:

- lehce aplikovatelná forma (obvykle HTML)
- jednoduše spravovatelný obsah
- modularita (možnost přidání doplňků, pluginů)
- udržování systému v souladu se současnými webovými trendy
- workflow management (rozdělení pravomocí uživatelům)
- virtualizace obsahu (možnost každého uživatele pracovat s virtuální kopií celého webu)

WCMS se dá ještě dále rozdělit do kategorií podle toho, kde systém zpracovává obsah. Existují tři možnosti:

- online
- offline
- kombinace obojí

Mezi nejznámější CMS patří např. Mambo, Joomla!, Drupal, Vignette CMS a Bricolage.

2 POUŽITÉ TECHNOLOGIE

Aby mnou navrhovaný CMS co nejvíce splňoval základní požadavky dobrého systému, které jsem zmiňoval výše, rozhodl jsem se do systému implementovat několik volně dostupných technologií. Informace k této kapitole jsem čerpal z internetových dokumentací jednotlivých projektů viz lit. [2], [4], [5] a [6].

2.1 Jazyk PHP

Redakční systém, který navrhuji bude implementován pomocí objektově orientovaného programování v PHP. V této kapitole popíšu vlastnosti tohoto jazyka a pokusím se objasnit jeho výhody a nevýhody.

2.1.1 Historie

PHP „Hypertext Preprocesor“, původně „Personal Home Page“ vytvořil Rasmus Lerdorf v roce 1994. V tomtéž roce autor přidal program Form Interpreter a tak vznikla verze 2. Zeev Suraski a Andi Gutmans v roce 1997 dosavadní PHP přepsali a dali tak základ PHP 3, které vyšlo v roce 1998. Podporovalo již systém Windows, obsahovalo více funkcí a bylo rychlejší. V roce 2000 bylo PHP opět rozšířeno ve verzi 4, kde byli uvedeny prvotní náznaky objektově orientovaného programování (dále jen OOP). Důležitý zlom nastal v roce 2004, kdy byla vydána verze 5 s plnou podporou OOP. V současné době se spíše opravují chyby a zvyšuje bezpečnost. Aktuální verze je PHP 5.2.7.

2.1.2 Seznámení s jazykem PHP

PHP je velmi rozšířený open source skriptovací programovací jazyk využívaný hlavně při tvorbě dynamických HTML stránek. Jeho syntaxe je velmi jednoduchá, vychází z jazyků C, Java a Perl.

Ukázka kódu:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <?php echo "<p>Hello world!</p>"; ?>
```

```
</body>
</html>
```

Jak je vidět na příkladu, PHP skript se vkládá do klasického HTML kódu a začíná instrukcí `<?php` a končí `?>`, přičemž nepsaným pravidlem je, že hlavní část skriptu se nachází na začátku a v HTML kódu se objevují už jen výstupy skriptu. Tento jednoduchý skript vypíše do HTML dokumentu text „Hello world!“. PHP skript na rozdíl od JavaScriptu, který je zpracováván na straně klienta, je většinou zpracováván na straně serveru. Server vygeneruje HTML kód, jehož součástí je také PHP skript a pošle ho klientovi. Klient dostane výsledek PHP skriptu ale neuvidí jeho vlastní kód. Programování v PHP je zároveň velmi snadné pro nově příchozí začátečníky a současně nabízí pokročilé možnosti a techniky pro profesionální programátory. PHP je používáno díky podpoře velkého množství operačních systémů, internetových protokolů, a přístupu k většině databázových systémů a také díky své „přizpůsobivé“ syntaxi.

2.1.3 Možnosti PHP

Nejvíce možností PHP nabízí ve zpracování skriptů na straně serveru – server-side scripting. Dá se v něm dělat vše co v jakémkoli jiném CGI „Common Gateway Interface“ programu. V PHP se tedy dá:

- zpracovávat data z formuláře
- vytvářet dynamický obsah stránky
- posílat a přijímat cookies
- a mnoho dalšího ...

PHP je nezávislé jak na operačním systému tak webovém serveru. Nabízí možnost jak procedurálního tak OOP nebo jejich kombinace. Mnohem propracovanější podpora OOP byla implementována od verze 5. Velká výhoda PHP je také v podpoře většiny současných databázových systémů jako jsou MySQL, MS-SQL, Oracle, ODBC a další.

2.1.4 Objektově orientované programování v PHP

Při studování principů OOP v PHP jsem čerpal z knihy Velká kniha PHP a MySQL (viz lit. [1]).

OOP mění pohled na programování aplikace. Ze psaní logických procedur přechází k entitám skutečného života, které v závěrečné fázi modeluje. Výsledná aplikace pak více připomíná svět okolo nás. Mezi klíčové pojmy OOP patří třídy, objekty,

metody a členské proměnné. Třídy mají za úkol definovat entitu sadou chování a charakteristik. Třída je zároveň něco jako šablona, ze které můžeme vytvářet specifické instance entit – objekt. Metoda je vlastně funkce, jejímž účelem je ale definovat chování v konkrétní třídě. Poslední klíčový parametr členská proměnná je vlastně běžná proměnná v PHP, která ovšem popisuje atributy konkrétní třídy. K OOP patří tři základní pojmy – zapouzdření, dědění a polymorfismus. Zapouzdření je metoda, při které se vnitřní fungování aplikace schová za dobře známá rozhraní, z nichž se dá přistoupit ke každé komponentě aplikace. Každá komponenta by měla být co nejvíce nezávislá na ostatních, díky čemuž se dá nejen opětovně použít ale umožňuje také vývojáři sestavovat komponenty dohromady jako skládačku. Těmto dobře definovaným rozhraním se říká objekty. Objekty se vytvářejí ze šablon, kterým se říká třídy, do kterých se implementují data i chování, které očekáváme u konkrétní entity. Třídy vykazují jistá chování prostřednictvím funkcí, kterým se říká metody. Metody zase naopak manipulují s charakteristikami tříd pomocí atributů, kterým se říká členské proměnné. Tato strategie přináší dvě velké výhody:

- vývojář může změnit implementaci aplikace, aniž by to mělo vliv na uživatele objektu
- redukuje se počet potenciálních uživatelských chyb, protože uživatel komunikuje s aplikací prostřednictvím rozhraní

Dědění je strategie, kterou se prosazuje opětovná využitelnost kódu. Dobře navržené třídy bude totiž možné využít v dalších aplikacích, protože jejich kód bude natolik abstraktní, že opětovné použití takových tříd nebude problém. Polymorfismus definuje schopnost OOP předefinovat charakteristiku či chování nějaké třídy v závislosti kontextu, ve kterém se používá.

Při vytváření nebo likvidaci objektů, vývojář často potřebuje provést řadu úkonů. Když se vytváří nový objekt, často chceme inicializovat jisté členské proměnné nebo dokonce spustit nějaké metody. K tomu slouží mechanismus s názvem konstruktor. Je to definovaný blok kódu, který se vykoná automaticky, při vytvoření nové instance objektu. Naopak destruktory nabízejí stejné možnosti při skončení skriptu.

Další funkce, kterou nabízí OOP je klonování objektů. Neklonovaný objekt dědí hodnoty členů z klonovaného objektu.

Při navrhování modelu, ve kterém jsou objekty na sobě různě závislé se využívají dva postupy. Pokud budeme tvořit model, v němž předpokládáme mnoho úzce provázaných objektů, použijeme abstraktní třídu. Je to taková třída, u které se nepředpokládá, že se z ní nebudou vytvářet instance. Slouží jen jako základní třída, z níž budou dědit jiné třídy. Pokud se budeme tvořit funkcionalitu, kterou budou

postupně vybavovány nezávislé objekty, použijeme rozhraní. Rozhraní definuje všeobecnou specifikaci pro implementaci nějaké konkrétní služby, deklaruje požadované funkce a konstanty, aniž by se zabývalo tím, jak se přesně mají implementovat.

Poslední funkce, kterou se budeme zabývat je reflexe. Reflexe je třída, která obsahuje celou řadu metod, prostřednictvím kterých můžeme prozkoumávat objekt, abychom se o něm dozvěděli více informací. Pomocí reflexe se můžeme dotazovat nejen na třídy a metody, ale také na funkce, rozhraní a rozšíření.

2.1.5 Výhody a nevýhody PHP

Některé hlavní výhody a nevýhody.

Výhody:

- základní instalace obsahuje široký seznam funkcí
- podpora mnoha databázových serverů
- široká podpora hostingu

Nevýhody:

- nativní funkce nepoužívají objektový přístup a většina z nich nepodporuje výjimky
- neexistuje přímá podpora pro vzdálené volání procedur
- ve standardní distribuci chybí debugovací nástroj

2.1.6 Požadavky na systém

Pro plnohodnotné využití všech funkcí PHP je nutné mít nainstalovaný http server a databázový server. Pro práci na projektu jsem si zvolil volně dostupný http server Apache a databázi MySQL. Nastavení PHP se pak provádí v konfiguračním souboru PHP.INI. Postup instalace a konfigurace lze najít na domovských stránkách jednotlivých projektů (viz lit. [2], [3] a [4]).

2.2 Databáze MySQL

Veškeré data, ať už o užívatelích, modulech nebo člancích budou ukládána prostřednictvím PHP do databáze. Databáze je jeden ze stěžejních prvků celého systému.

2.2.1 Co je to databáze

Databáze je určitá uspořádaná množina informací (dat) uložená na paměťovém médiu. Někdy se myslí pod pojmem databáze nejen data ale také software umožňující práci s daty. Tento software se v české odborné literatuře nazývá systém řízené báze dat (SŘBD).

Databáze se dají rozdělit podle způsobu ukládání dat a podle vazeb mezi jednotlivými daty na:

- Hierarchická databáze
- Síťová databáze
- Relační databáze
- Objektová databáze
- Objektově relační databáze

Mezi nejznámější databázové systémy patří – MySQL, PostgreSQL, Oracle a Microsoft SQL.

2.2.2 Databáze MySQL

Autoři této databáze jsou David Axmark, Allan Larsson a Michael Widenius. V současné době stojí za vývojem Sun Microsystems. Je šířena pod bezplatnou i pod komerční licenci a aktuální verze má číslo 5.1. MySQL je součástí tzv. „svaté trojice“ – MySQL, PHP, Apache, využívané pro tvorbu webových aplikací.

Databáze MySQL se z rozdělení v předchozí kapitole řadí mezi databáze relační. Relační databáze uchovávají data (záznamy) v tabulkách, které se skládají z řádků (záznamů) a sloupců, přičemž mezi jednotlivými záznamy jsou definovány vztahy (relace). Je to multiplatformní databáze. Jak název napovídá vychází z databáze SQL „Structured Query Language“ a obsahuje pár rozšíření. SQL znamená strukturovaný dotazovací jazyk. Práce s databázemi a tabulkami probíhá prostřednictvím SQL dotazů. Mezi ty nejzákladnější dotazy patří např. SELECT, INSERT, UPDATE a DELETE. Od začátku vývoje byla tato databáze optimalizována hlavně pro rychlost. Příklad SQL dotazu:

```
SELECT 'name', 'login', 'password' FROM 'Users'
```

Tento dotaz vybere jméno, login a heslo z tabulky, která má název „Users“. Dotaz se dá rozdělit do 4 částí. V první části (SELECT) se zadá operace, kterou chceme provést. Následuje (*), která zastupuje názvy sloupců, které chceme vybrat.

V druhé části (FROM) vybereme tabulku ('table'). Dále je pak třetí nepovinná část (WHERE), kde udáváme podmínky pro záznamy, které chceme vybrat. Poslední čtvrtá část (ORDER BY) také nepovinná, slouží k určení jak a podle kterého sloupce budeme záznamy řadit.

2.3 Wysiwig editor TinyMCE

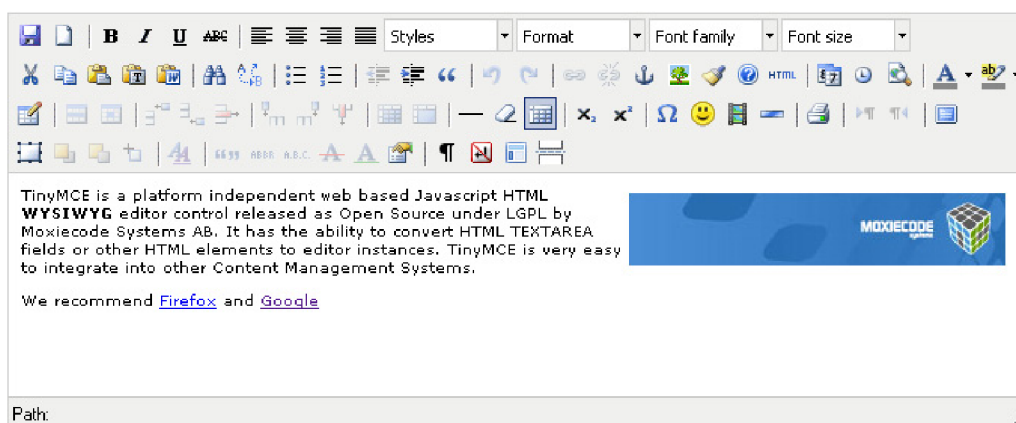
Wysiwig editor využijí při vkládání formátovaného textu např. text článku. Díky němu bude správa obsahu jednodušší a hlavně tím umožním spravovat a stylovat obsah uživatelům, kteří nemají žádné znalosti HTML jazyka.

2.3.1 Co je to wysiwyg editor

Wysiwyg je zkratka, která vznikla s anglické věty „What you see is what you get“. Tyto editory umožňují rychlejší tvorbu HTML dokumentů, aniž bychom museli znát principy HTML kódu. V praxi to znamená, že uživatel píše text a upravuje jej pomocí funkcí, které jsou známe např. z aplikace Word. Editor pak tento text zpracuje do HTML kódu a CSS stylů. Nevýhoda wysiwyg editorů je, že takto vygenerované dokumenty často obsahují některé zbytečné HTML tagy nebo jsou nevhodně umístěny, tento kód také bývá objemnější.

Mezi současně nejlépe vyvinuté wysiwyg editory patří FCKEditor a TinyMCE, oba tyto editory jsou na srovnatelné úrovni.

2.3.2 TinyMCE



Obr. 2.1: Wysiwyg editor TinyMCE se všemi jeho rozšířeními

Jedná se open source JavaScript/HTML wysiwyg editor. Pomocí něj se jakýkoliv textfield na HTML stránce dá proměnit v HTML wysiwyg editor. Podporuje většinu

webových prohlížečů a OS. Je snadno implementovatelný do redakčních systémů. Podporuje spousty jazyků, včetně češtiny. Jeho výstup je plně validní XHTML, CSS.

Návod instalace se dá nalézt na jeho domovských stránkách (viz lit. [7]).

2.4 AJAX framework jQuery UI

Implementací tohoto frameworku bude systém více interaktivní a vzhledově přívětivější.

2.4.1 Co je to AJAX

AJAX je technologie, která se využívá při tvorbě interaktivních internetových stránek. Tato technologie umožňuje měnit obsah stránky bez nutnosti jejího opětovného načtení, protože přenáší pouze změněná data a tudíž nedochází ke generování celého HTML dokumentu. Poskytuje možnost pro vývojáře vytvořit příjemnější prostředí. K tomu však potřebují novější verze prohlížečů, což v současné době není nepřekonatelný problém. AJAX aplikace jsou vyvíjeny pomocí těchto technologií:

- HTML nebo XHTML a CSS (zobrazení informací)
- DoM a JavaScript (vlastní implementace operací)
- XMLHttpRequest (asynchronní výměna dat)

Mezi současné internetové aplikace využívající technologii AJAX patří například GoogleMail nebo FaceBook.

2.4.2 Výhody a nevýhody

Výhody:

- odstranění potřeby znovunačtení stránky po provedení operace
- při vhodné implementaci menší zátěž serveru i sítě

Nevýhody:

- webové stránky se chovají jako aplikace a ne jako posloupnost stránek (z čehož plyne např. znemožnění používání tlačítek Zpět a Vpřed, ačkoliv moderní AJAX aplikace již tento problém částečně vyřešili)
- nevhodnost protokolu http pro aplikace komunikující intenzivně se serverem (při nevhodné implementaci může dojít k přetížení serveru)
- nemožnost samovolného kontaktování uživatele serverem

2.4.3 jQuery UI

jQuery UI je interaktivní knihovna postavena na JavaScript knihovně jQuery. Pomocí této knihovny se dají snadno tvořit interaktivní aplikace. Přímo na stránkách tohoto projektu [6] si lze nadefinovat vlastní vzhled ještě před jeho stáhnutím. Obsahuje několik komponentů, které jsou rozdělené do tří skupin:

- Interactions
- Widgets
- Effects

V mém systému využiji pouze skupinu widgets, která obsahuje několik komponent:

- Accordion
- Datepicker
- Dialog
- a další

Nejzajímavější jsou komponenty Datepicker a Dialog. Datepicker umožňuje vložit jakékoli datum do textového vstupu přes kalendář. Dialog se dá využít pro zobrazování informačních zpráv nebo pro získání potvrzení pro nějakou událost.

3 NÁVRH VLASTNÍHO CMS

Redakční systém, který navrhují, bude řešit publikační server, na kterém budou moct uživatelé číst články, které budou tříděné v kategoriích. Jednotlivé články budou moct uživatelé diskutovat v komentářích. Systém se bude skládat z nadstavbových modulů a jádra, ve kterém se budou spravovat uživatelé a nadstavbové moduly. Nadstavbové moduly budou – články, kategorie a komentáře.

3.1 Uživatelské rozhraní

Na hlavní obrazovce redakčního systému, kterou můžete vidět na obr. 3.1, bude *uživatelské menu*, *hlavička*, *hlavní menu* a *editační obrazovka*.

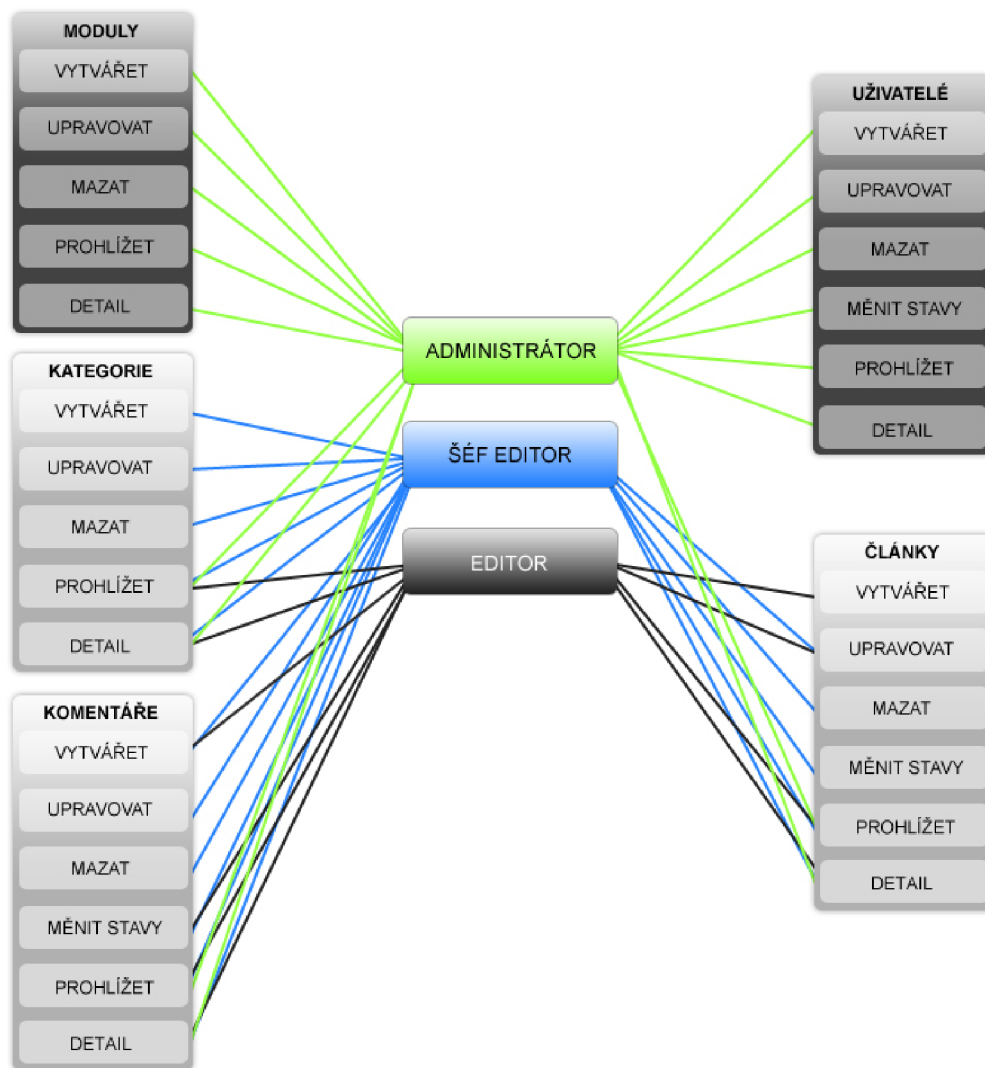


Obr. 3.1: Návrh uživatelského rozhraní redakčního systému

V *uživatelském menu* bude nabídka pro konkrétního přihlášeného uživatele. V *hlavičce* bude název webu, který redakční systém spravuje. *Hlavní menu* bude nabízet správu všech modulů (jak těch systémových tak těch nadstavbových) a v *editační obrazovce* se budou provádět veškeré operace s daty.

3.2 Rozdělení rolí

System práv bude definovatelný dynamicky. Pro publikační systém v něm budou nastaveny role *Editor*, *Šéf editor* a *Administrátor*. Podrobný přehled toho co daná role bude moct vykonávat je vyobrazen na obr. 3.2.



Obr. 3.2: Rozdělení rolí pro jednotlivé moduly

3.3 Stručný popis jednotlivých modulů

Jak již bylo zmíněno výše systém bude spravovat 3 nadstavbové moduly. Jejich popis včetně těch 2 systémových je uveden níže.

3.3.1 Uživatelé

Jedná se o modul, který bude patřit do jádra systému a budou se v něm spravovat uživatelé a jejich role.

3.3.2 Moduly

V tomto modulu, který bude rovněž implementován do jádra, se budou spravovat veškeré moduly uvedené níže.

3.3.3 Články

Tento modul bude obsahovat základní údaje o jednotlivých článcích. Kromě nadpisu a textu článku, bude navíc obsahovat informace o klíčových slovech, datu přidání článku, kdo byl jejím autorem a do jaké kategorie daný článek spadá.

3.3.4 Kategorie

Kategorie poměrně jednoduchý modul, budou obsahovat pouze název a jeho stručný popis.

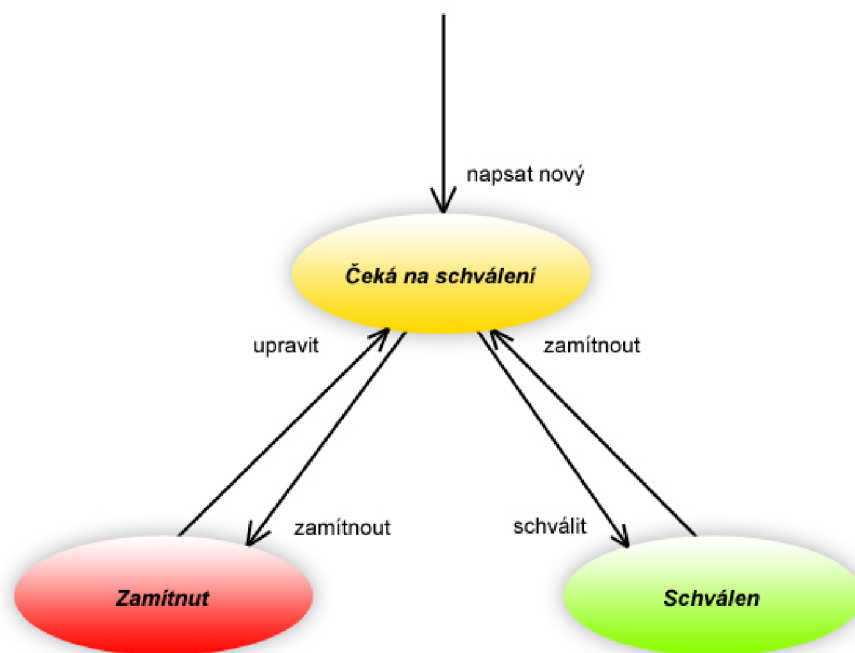
3.3.5 Komentáře

Kromě textu a údaje o autorovi komentáře, bude ještě uchovávat data o tom, ke kterému článku byl komentář přidán a kdy.

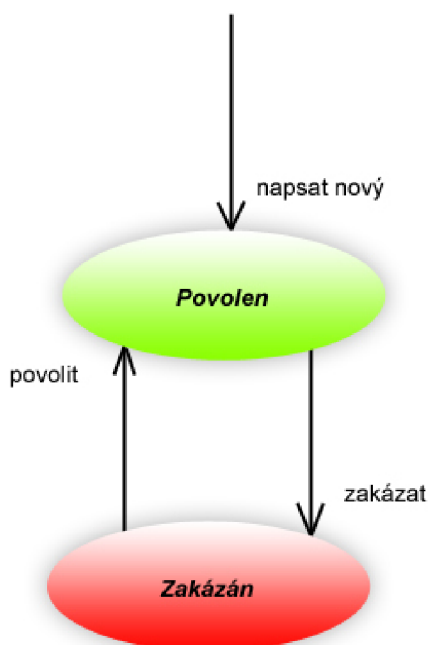
3.4 Životní cykly článku a komentáře

Hlavní entita celého systému je článek. Bude mít 3 stavy. Po napsání článku a odeslání do databáze přichází do stavu 1 – *Čeká na schválení*. V tomto stavu může být článek smazán ale nedá se upravit. Čeká na schválení nebo zamítnutí. Při schválení přechází do stavu 2 – *Schválen*, nedá se vymazat ani upravovat ale může být znovu zamítnut a tím přejde do stavu 3 – *Zamítnut*. V tomto stavu může být článek upravován i mazán. Pokud dojde k jeho úpravě dostane se znovu do stavu 1 tedy *Čeká na schválení*. Celý tento cyklus je na obr. 3.3.

Životní cyklus komentáře na obr. 3.4 je jednodušší. Bude mít pouze dva stavy a to *Povolen* a *Zamítnut*. Po odeslání komentáře do databáze přijde do stavu 1 – *Povolen*. Bude moct být zakázán tím přejde do stavu 2 *Zakázán*. A z tohoto stavu bude moct být opět povolen. Upravovat a mazat komentář bude možné v obou stavech.



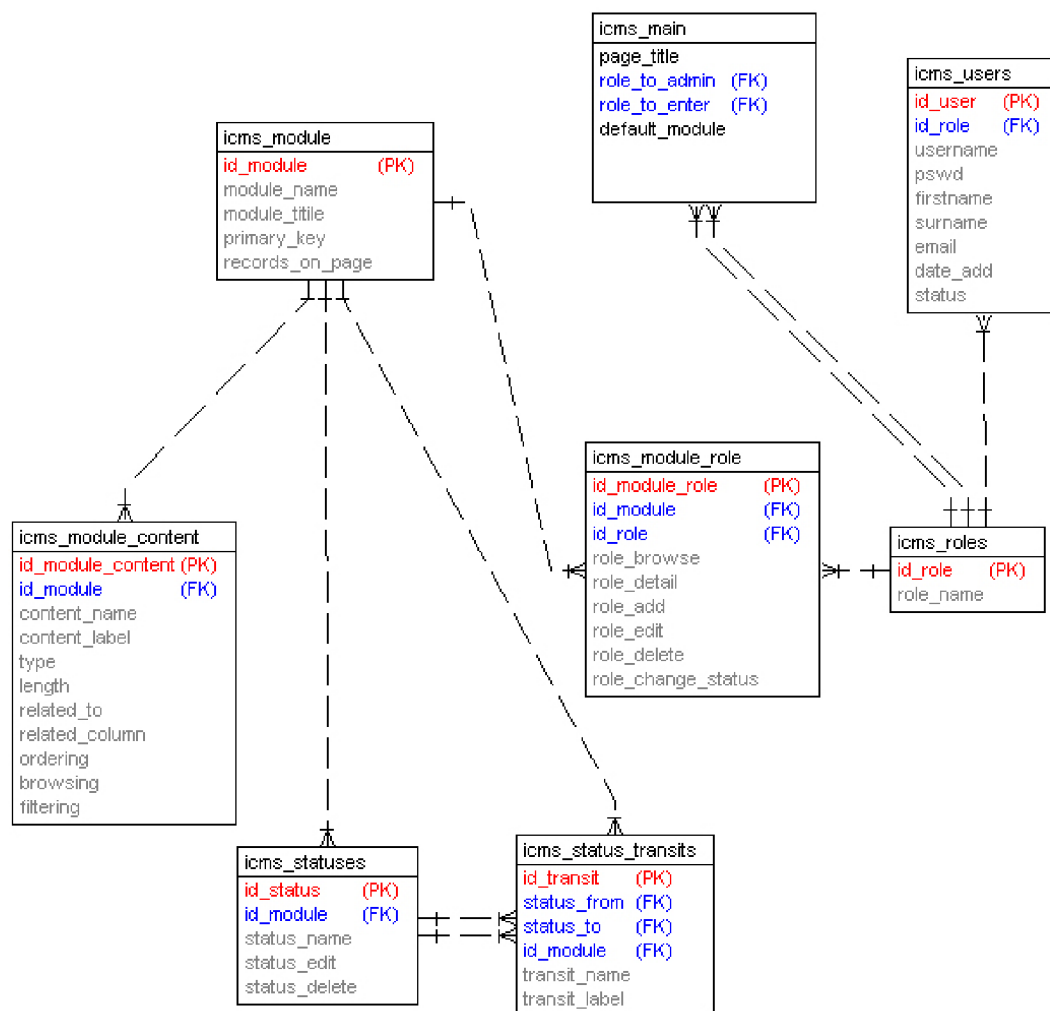
Obr. 3.3: Životní cyklus entity článek



Obr. 3.4: Životní cyklus entity komentáře

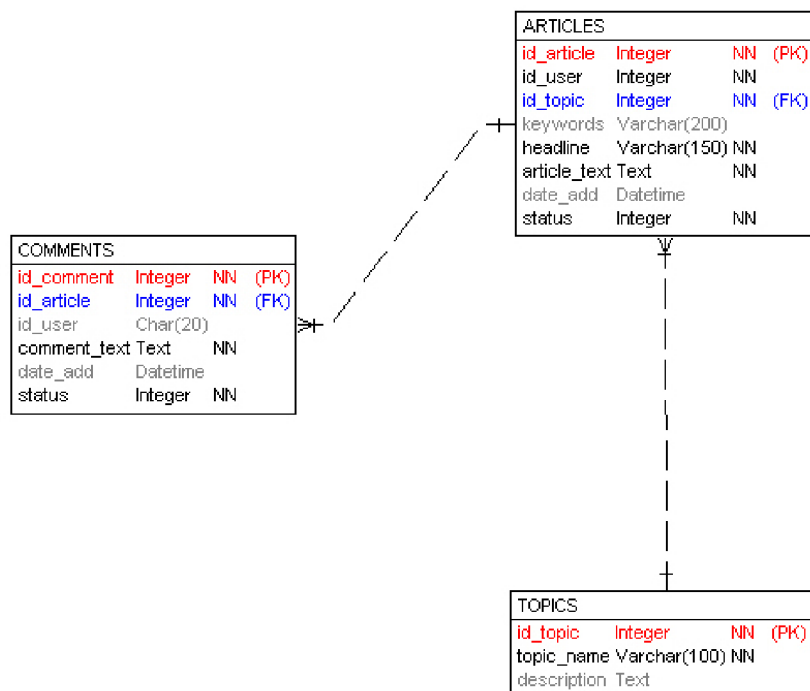
3.5 Datový model

Datový model je pro lepší orientaci rozdělen na dva obrázky – model jádra 3.5 a model publikačního systému 3.6.



Obr. 3.5: Datový model jádra CMS

Hlavní entity jádra jsou moduly (`icms_module`) a uživatelé (`icms_users`). Na moduly mají relaci entity obsah modulu (`icms_module_content`) – obsahuje veškerá data o modulu, stavy (`icms_statuses`) a přechody stavů (`icms_status_transits`) – řeší stavy modulu a způsob přechodů mezi nimi, a entita `icms_module_role` – popisuje jaké operace může s modulem dělat daná role. Na entitu uživatelé má relaci pouze role a ty se opět vážou na entitu `icms_module_role`. Na konec je třeba ještě zmínit entitu `icms_main`, ve které jsou atributy titulek stránky, jaká role potřebná pro vstup do systému, jaká pro administraci a také nastavení defaultního modulu, tedy který modul se zobrazí po přihlášení do systému.

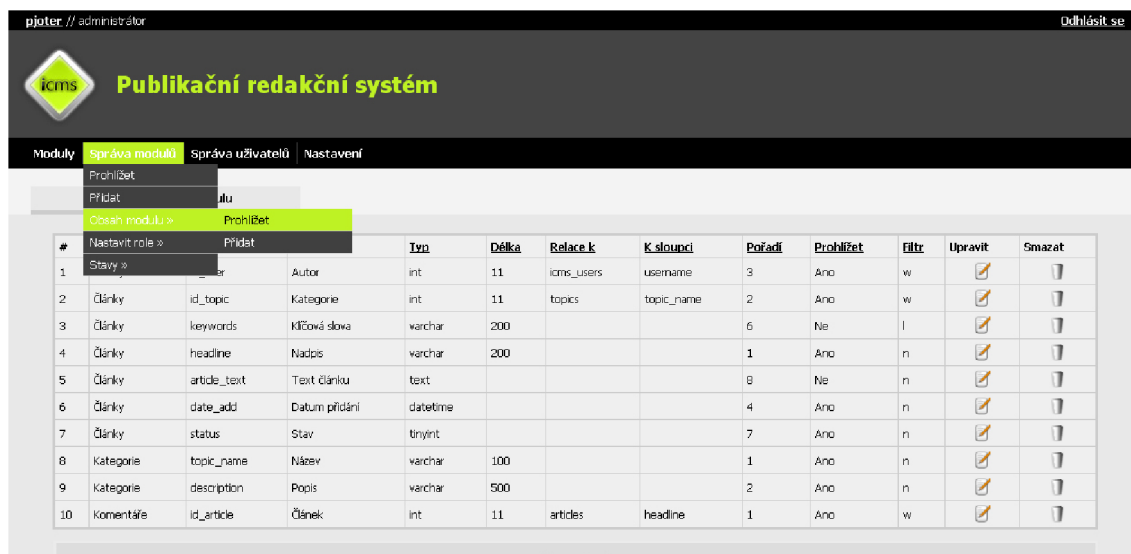


Obr. 3.6: Datový model publikačního systému

Model publikačního systému popisuje 3 entity – články (ARTICLES), kategorie (TOPICS) a komentáře (COMMENTS). Na článek má relaci, kromě kategorie do které spadá, taky entita z modelu jádra uživatelé jako autor článku. Komentáře mají relaci k článku a také k autorovi stejně jako článek.

4 REDAKČNÍ SYSTÉM ICMS

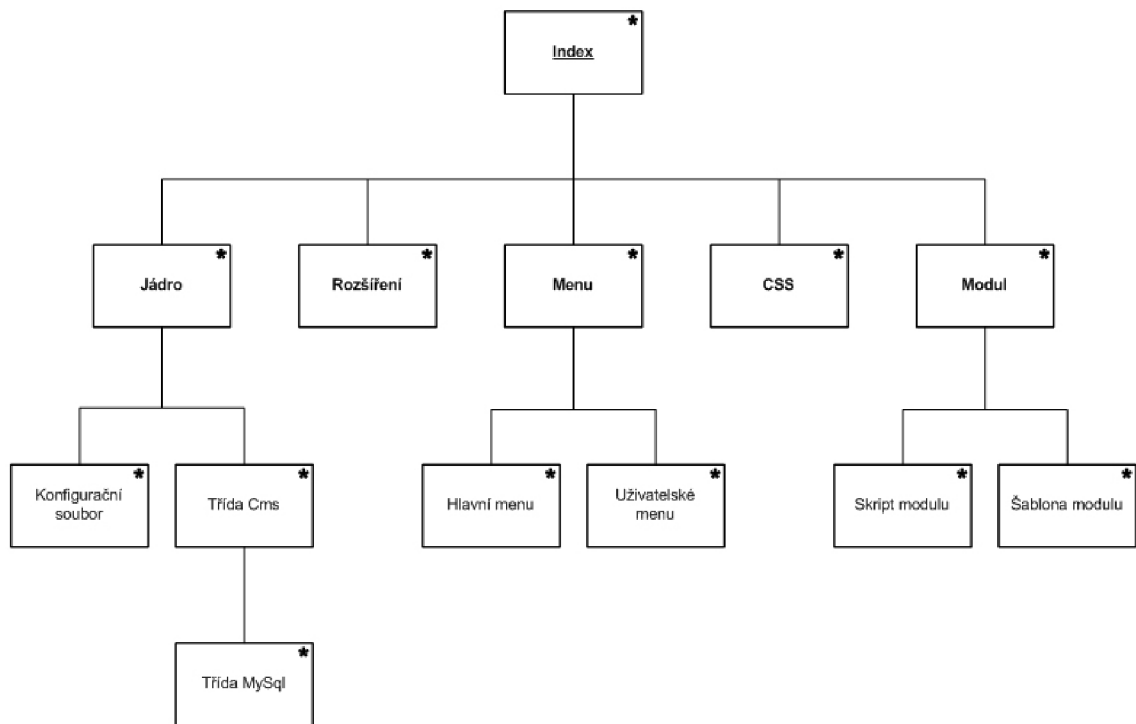
Realizovaný CMS pomocí meta dat umožňuje spravovat jednotlivé nadstavbové moduly. Systém rolí pak umožňuje vymežit operace, které daný uživatel s rolí bude moci vykonávat s daným modulem. CMS se skládá z jádra, které tvoří 7 systémových modulů a 3 nadstavbové moduly. Součástí aplikace je také frontend, který zobrazuje tyto nadstavbové moduly a umožňuje běžnému uživateli prohlížet články a komentovat je.



Obr. 4.1: Prostředí redakčního systému iCMS

4.1 Struktura systému

Celá aplikace se vkládá do hlavního souboru `index.php`. Jádro inicializuje skript `init.php` – načte údaje z konfiguračního souboru `config.php` a nahraje a inicializuje třídy `MySQL` (`mysql.class.php`) a `Cms` (`cms.class.php`). Dále se v indexu načítají a konfigurují javascripty pro rozšíření jQuery UI (`jquery-ui-1.7.1.custom.min.js`) a TinyMCE (`tiny_mce.js`). Pro jQuery UI se navíc načítají šablony pro všechny dialogy ze souboru `dialogs.php`. Další části systému jsou kaskádové styly (`main.css` a `menu.css`), které definují vzhled hlavního menu a celého systému. Jednotlivé moduly načítá skript `main.php`, např. skript pro obsluhu modulu je zpracovávány ve skriptu (`module.php`), který generuje pomocí tříd a databáze data pro šablonu, která pomocí CSS tyto data naformátuje a zobrazuje. Celá struktura systému je přehledně zobrazena v diagramu na obrázku 4.2. Jednotlivé části systému jsou podrobněji popsány níže.



Obr. 4.2: Diagram struktury systému

4.1.1 Index

Hlavní stránka, do které se vkládají všechny níže uvedené části systému. Před jeho html částí se vkládají php skripty. V html části se pak načítají rozšíření, kaskádové styly a šablona s jejíž pomocí se zobrazují vygenerované data. Jako výchozí se načte modul pro přihlášení uživatele.

4.1.2 Jádro

Jádro systému je postaveno celé na php. Skládá se ze dvou tříd MySql a Cms. Výčet všech proměnných a metod je uveden v příloze.

Inicializační skript

Tento skript načte konfigurační soubor, ve kterém jsou nadefinované přihlašovací údaje do databáze. Také načte obě třídy, provede připojení do databáze a další nastavení (kódování dotazů a další).

Třída MySql

Skládá se z metod a atributů, pomocí kterých se dá jednoduše pracovat s databází. Obsahuje metody pro vkládání, čtení, úpravu a mazání dat z databáze MySQL. Čtení umožňuje v několika podobách např. jako asociativné pole pokud jde o více záznamů. Soupis atributů a metod je v příloze A.1.

Třída Cms

Tato třída dědí třídu MySql aby mohla jednoduše komunikovat s databází. Tvoří hlavní část celého systému. Pomocí metod a atributů této třídy je implementován celý redakční systém. Např. vytváří tabulku pro prohlížení dat, generuje řazení podle jednotlivých sloupců, zpracovává její filtr a obsluhuje stránkování. Soupis atributů a metod je v příloze A.2.

4.1.3 Menu

Hlavní menu

Generuje se automaticky, zobrazuje všechny moduly, které jsou nadefinovány v systému. Podle rolí povoluje jednotlivé akce s modulem. Generuje se ve dvou režimech – administrátorský a editační. Administrační navíc obsahuje navigaci pro správu všech systémových modulů. Jaká role je pro daný režim potřeba se dá nakonfigurovat v nastavení.

Uživatelské menu

Toto umožňuje uživateli kliknutím na svůj login přejít do editační obrazovky svého profilu. Obsahuje také informaci o roli jakou má přihlášený uživatel a tlačítko pro odhlášení se ze systému.

4.1.4 Moduly

Moduly jsou implementovány jako php skript a šablona. Pro redakční systém jsem naprogramoval skripty a šablony pro správu systémových a nadstavbových modulů které do indexu vkládá skript `main.php`. Přihlášení a odhlášení uživatele provádí skripty `login.php` a `logout.php`.

Skripty

Skript modulu provádí 6 typů operací – *prohlížet*, *detail*, *přidat*, *upravit*, *smazat* a *přechody stavů* – pomocí třídy Cms. Pro každou z uvedených operací vygeneruje

Obr. 4.3: Formulář pro přidání článku vygenerovaný pomocí šablony

data, které se pak zobrazují v šabloně. Skript také zpracovává data z šablon a ukládá je do databáze.

Šablony

Šablona zobrazuje 4 základní obazovky – *prohlížení*, *přidat*, *upravit a detail*, data pro ně připravuje skript modulu. Prohlížení zobrazí vygenerovanou tabulku, pro práci s položkami modulu, ze skriptu modulu, včetně stránkování a filtru. Přidat zobrazí vygenerovaný formulář. Upravit také zobrazí formulář vyplněný aktuálními daty. Detail zobrazí pouze aktuální data položky.

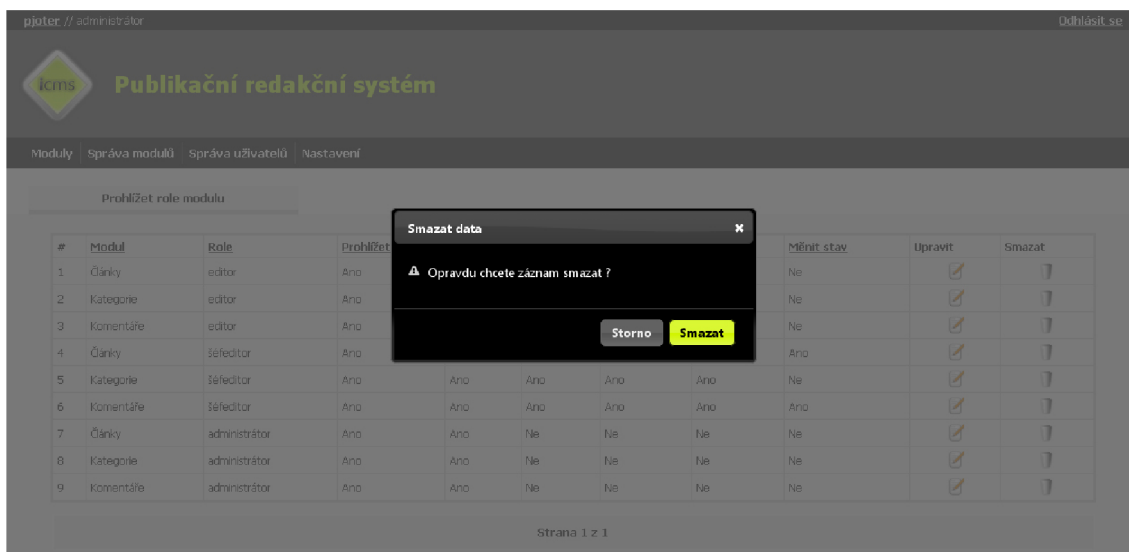
4.1.5 Rozšíření

Všechny rozšíření se vkládají jako javasripty do indexu. Kromě jQuery UI a TinyMCE systém také využívá framework jQuery mimo jiné pro zobrazení hlavního menu.

jQuery UI

```
<script type="text/javascript"
src="./extensions/jquery/js/jquery-ui-1.7.1.custom.min.js"></script>
```

Využívá se pro zobrazování potvrzovacích a informačních dialogů. Například potvrzovací dialog pro smazání záznamu (na obrázku 4.4) přes Ajax odebere řádek



Obr. 4.4: Modální dialog pro potvrzení smazání volaný pomocí jQuery UI

tabulky a spustí php skript pro vymazání záznamu z databáze. Dialogy se do indexu vkládají jako šablona dialogů. Tento framework používá vlastní CSS téma nadefinované skrze rozhraní na stránkách tohoto projektu.

TinyMCE

```
<script language="javascript" type="text/javascript"
  src="./extensions/tiny_mce/tiny_mce.js"></script>
```

Tento wysiwyg editor se používá v šabloně pro obrazovky upravit a přidat. V konfiguraci jsou nadefinované jaká tlačítka a pluginy se mají zobrazovat v tomto editoru.

4.1.6 Kaskádové styly

Hlavní soubor `main.css` se využívá k formátování celého systému. Definuje vzhled layoutu systému, prohlížeč tabulky a editačního formuláře. Dále se používá soubor `menu.css` pro definování vzhledu hlavního menu.

4.2 Moduly

Moduly obsluhují tabulky databáze prostřednictvím třídy `Cms`, kterou načte a inicializuje inicializační skript. Z datového modelu (obrázek 3.5 a 3.6) vyplývá rozdělení modulů na systémové a nadstavbové. Kromě všech modulů uvedených níže je v systému zvlášť modul nastavení systému `settings.php`.

4.2.1 Systémové moduly

Všechny tyto moduly umožňují akce přidat, upravit a mazat. Slouží pro správu 7 tabulek z datového modelu jádra CMS. Jejich skript spouští a data z něj zpracuje šablona `core.php`.

Uživatelé

Uživatelé kromě akcí přidat, upravit, mazat se dají také aktivovat a deaktivovat. Pomocí těchto akcí se dá zamezit nebo povolit uživateli přístup do systému. Umožňuje také editaci vlastního profilu uživateli s dostatečnou rolí pro vstup do systému.

Moduly

V modulech se uchovává informace o názvu tabulky z databáze, kterou spravuje a její primární klíč. Obsahuje také atribut, který definuje počet záznamů na stránku.

Obsah modulu

Chování nadstavbových modulů se konfiguruje v tomto modulu. Zde se nastavují jednotlivé atributy a jejich chování. Např. atribut s názvem `date_add` chápe systém jako datum přidání záznamu do tohoto modulu. Kromě `date_add` existuje ještě jeden zvlášť zpracovávaný atribut `status` kvůli definici životního cyklu modulu. Atributu se dá taky nastavit relace k jinému z dalšího modulu.

Stavy

Tento modul popisuje stavy modulu. Definuje jaká operace může být prováděna v jednotlivých stavech.

Přechody stavů

Životní cykly všech nadstavbových modulů se vytvářejí pomocí tohoto modulu. Systém navíc rozeznává dva speciální typy přechodů – `on_add` a `on_edit` – pomocí kterých jde nastavit chování modulu po vložení nového záznamu nebo jeho úpravě.

Role

Modul definuje názvy rolí. Je důležité aby id role rostlo přímo úměrně s pravomocemi role.

Role modulu

Pomocí tohoto modulu se dá nastavit, co může dělat uživatel s danou rolí v daném modulu.

4.2.2 Nadstavbové moduly

Publikační systém článků je realizován pomocí těchto 3 nadstavbových modulů. Skript a šablona pro tyto moduly jsou soubory `module.php`

Články

Tento modul spravuje základní údaje o jednotlivých článcích. Kromě nadpisu a textu článku, obsahuje informace o klíčových slovech, datu přidání článku, kdo byl jejím autorem a do jaké kategorie daný článek spadá. Pomocí systémových modulů stavy a přechody stavů je zaručen životní cyklus článku tak jak je uveden na obr. 3.3.

#	Nadpis	Kategorie	Autor	Datum přidání	Stav	Detail	Upravit	Smazat	Schwít	Zamítnout
1	Apple uvedl nové servery Xserve	Hardware	pioter	2009-04-08 19:05:33	●	🔍				✖
2	Microsoft Silverlight 3: ve znamení multimédií, HD videa a 3D grafiky	Webdesign	pioter	2009-04-08 19:16:02	●	🔍		🗑️	✔	✖
3	Google SketchUp: Rotační modely pomocí Follow Me	Digitální grafika	fenda	2009-04-08 19:25:22	●	🔍				✖
4	Ohlédnutí za historií aneb jak vypadal první „digitál“ na světě	Fotografie	meta	2009-04-08 20:04:27	●	🔍				✖
5	Společnost Google uvolnila bezplatnou lite verzi nástroje pro tvorbu scénérií GroundWizzarddd	Digitální grafika	pioter	2009-04-09 00:25:31	●	🔍		🗑️	✔	✖
6	Apple vydal Mac OS X 10.5.7	Software	pioter	2009-05-17 21:19:47	●	🔍				✖
7	Smashing Magazine, aneb centralace, která potěší	Software	meta	2009-05-24 23:28:47	●	🔍				✖
8	Červený trpaslík: Zpátky na zemi - a digitálně	Digitální grafika	meta	2009-05-24 23:37:17	●	🔍	✍️	🗑️		
9	Pulitzerovy ceny 2009: Jak vypadají vítězné fotografie	Fotografie	ave	2009-05-24 23:39:37	●	🔍				✖

Obr. 4.5: Prohlížení článků s rolí šéfeditor

Kategorie

V kategoriích je možné definovat pouze informace o názvu kategorie a její stručný popis.

Komentáře

Komentář obsahuje data o tom, ke kterému článku patří, kdo je jeho autorem a hlavně text komentáře. Také komentáře mají nadefinovaný svůj životní cyklus z obr. 3.4.

4.3 Frontend

Webdesign Digitální grafika Hardware Software Fotografie

uživatel: pjofer
odhlásit se

Rozhovor: Webdesigner a grafik Daniel "AGARWAEN" Střelec

Webdesign

Napsáno: 2009-04-08 19:23:56, autor: František Kubalík

World Wide Web se neustále rozrůstá o obrovský počet nových stránek, bannerů, obrázků a videí. Coby reklamní medium je Internet v dnešní době již na třetím místě za televizí a tiskem, za sebou nechal poprvé i rozhlasovou reklamu, z důvodu přilákání zákazníka se hlavně komerční stránky stávají stále dokonalejšími a to jak po stránce grafické, tak po té programátorské. Tento vývoj ovšem nevyhnutelně přináší neustále se zvyšující požadavky na jejich tvorbu a provoz, což může potvrdit i webdesigner a grafik **Daniel "AGARWAEN" Střelec**, o jehož zkušenostech se dočtete v tomto článku.



Obr. 4.6: Frontend publikačního systému

Frontend je webová aplikace, která zobrazuje články z databáze. Pro komunikaci s databází využívá třídu Mysql. Navigace se generuje jako odkazy na jednotlivé kategorie. Kliknutím na nadpis článku se uživatel dostane na jeho detail. Ve frontendu je také registrační a přihlašovací modul, díky kterému mohou přihlášení uživatelé komentovat články. Po přihlášení může uživatel také měnit svůj profil.

5 ZÁVĚR

Cílem této práce bylo vytvořit redakční systém, který bude modulární a bude spravovat publikační systém článků. CMS se skládá z backendu a frontendu. Backend je prostředí, ve kterém se konfiguruje celý systém. Také se v něm spravuje publikační systém článků. Frontend pak zobrazuje tento spravovaný publikační systém a umožňuje tak běžnému uživateli procházet články a komentovat je.

Datový model redakčního systému se skládá ze 7 tabulek, které společně tvoří jádro systému, a 3 tabulek nadstavbových. Tyto tři tabulky slouží pro ukládání dat publikačního systému článků. Redakční systém jsem implementoval pomocí jazyka PHP. Naprogramoval jsem dvě třídy – jednu pro obsluhu databáze a druhou pro obsluhu redakčního systému. Systém je modulární, tudíž umožňuje přidávání dalších nadstavbových modulů. Celý backend se skládá z modulů pro přihlášení, pro konfiguraci systému a pro správu nadstavbových modulů. CMS také umožňuje definovat životní cykly modulů pomocí stavů a přechodů mezi nimi. Další částí je systém rolí, pomocí kterého se rozdělí jednotlivé úkony s nadstavbovými moduly do několika skupin a členství v této skupině pak umožní uživateli provádět její úkony.

Frontend tohoto systému umožňuje běžnému uživateli registrovat se a po přihlášení může komentovat články. Menu frontendu jsou odkazy na jednotlivé kategorie.

Povedlo se mi tedy vytvořit modulární redakční systém, který spravuje publikační systém článků. Systém je plně funkční, takže si myslím, že se mi zadání podařilo úspěšně splnit.

LITERATURA

- [1] Gilmore J. W. *Velká kniha PHP a MySQL 5*. ZONER Press 2007, ISBN 80-86815-53-6.
- [2] *PHP Manual* [online].
2008, poslední aktualizace 8. 11. 2008 [cit. 2008-08-11].
Dostupné z URL: <<http://www.php.net/manual/en/>>.
- [3] *Apache HTTP Server Version 2.2 Documentation* [online].
2008, poslední aktualizace 8. 11. 2008 [cit. 2008-08-11].
Dostupné z URL: <<http://httpd.apache.org/docs/2.2/>>.
- [4] *MySQL 5.0 Reference Manual* [online].
2008, poslední aktualizace 20. 11. 2008 [cit. 2008-20-11].
Dostupné z URL: <<http://dev.mysql.com/doc/refman/5.0/en/>>.
- [5] *TinyMCE Wiki Documentation* [online].
2008, poslední aktualizace 30. 10. 2008 [cit. 2008-01-12].
Dostupné z URL: <<http://wiki.moxiecode.com/index.php/TinyMCE:Index>>.
- [6] *jQuery user interface Demos & Documentation* [online].
2009, poslední aktualizace 07. 05. 2009 [cit. 2009-14-05].
Dostupné z URL: <<http://jqueryui.com/demos/>>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

CMS redakční systém – Content Management System

CSS kaskádové styly – Cascading Style Sheets

HTML hypertextový značkovací jazyk – HyperText Markup Language

OOP Objektově orientované programování

SQL strukturovaný dotazovací jazyk – Structured Query Language

UI uživatelské rozhraní – User Interface

SEZNAM PŘÍLOH

A První příloha	40
A.1 Třída MySql	40
A.2 Třída Cms	41
B CD příloha	42

A PRVNÍ PŘÍLOHA

A.1 Třída MySql

MySql
+conn +dbName +dbPassword +dbServer +dbUser +err_handler +insert_addslashes +log_query +result
+close() +conf(in a) +connect() +delete(in tbl : char, in where : char) +err(in r : char, in q : char, in e : char) +fetch_assoc(in r : char) +free_result(in r : char) +insert(in tbl : char, in cols : char) +insert_id() +num_rows(in r : char) +query(in r : char, in q : char) : char +query_fetch_assoc(in q : char) : char +query_fetch_assoc_all(in q : char, in index : int) : char +query_fetch_row_all(in q : char) : char +query_result(in q : char, in line : int, in r : int) : char +query_result_all(in q : char, in r : int) : char +replace(in tbl : char, in cols : char) +result(in r : char, in f, in row : int) +update(in tbl : char, in cols : char, in where : char)

Obr. A.1: Proměnné a metody třídy MySql

A.2 Třída Cms

Cms
+browse +defaultRecordsOnPage : int +mode : char +moduleId : int +moduleName : char +moduleRecordsOnPage : int +moduleTitle : char +primaryKey : char +role : int +roleToAdmin : int +roleToEnter : int +tableData : char +tableHead : char +userId : int +userRoles
+addModule(in data : char) +browseCoreModule(in f : int) : char +browseModule(in f : int) : char +changeStatus(in id : int, in transitName : char) +checkFilter() +deleteModule(in id : int) +detailModule(in id : int) : char +editModule(in id : int, in data : char) +getCoreHead() : char +getFilter() : char +getMainMenu() : char +getModulePars() +getPagesButtons(in first : int) : char +getRecordIds() : char +getSelect(in tableHead : char, in for : char) : char +getStatusRoles(in status : int) : char +getStatusTransits(in status : int) : char +getTableData() +getTableHead(in for : char) +getUserRoles(in idModule : int) +setFilter(in data : char) +setOrder(in orderBy : char) +setPages(in first : int)

Obr. A.2: Proměnné a metody třídy Cms

B CD PŘÍLOHA

Na přiloženém CD se kromě elektronické verze tohoto dokumentu nachází zdrojové kódy redakčního systému s vyexportovanou databází. Systém byl testován na PHP verzi 5.2.9, Mysql 5.0.51a a Apache verze 2.11.