

**Jihočeská univerzita v Českých Budějovicích**

Fakulta pedagogická – katedra fyziky

**Redakční systém realizovaný pomocí PHP a MySQL**

Bakalářská práce

Vedoucí práce: Ing. Michal Šerý

Autor: Filip Trnka

## **Anotace:**

Práce se zabývá problematikou a prostředky tvorby webových stránek a to jak historií, tak současnými trendy a požadavky. Velká část práce je věnována návrhu a realizaci redakčního systému, popisu jeho struktury a prostředků použitých k realizaci. Velká pozornost je udělena administraci obsahu redakčního systému se zaměřením na jednotlivé uživatelské role. Závěrem práce je popis několika konkrétních administračních úkonů se kterými se za běžného provozu redakčního systému setkáme nejčastěji.

## **Abstract:**

The work contains problems and sources making web pages. Deals with their history and recent trends and demands. Increased attention is granted to structure and realization Content Management System (CMS). Resources for realization CMS are detailed. Hight attention was given to content administration with adjustment from view of individual CMS users demands. In the end the work describes several common administrator stasks.

## **Prohlášení**

Prohlašuji že předloženou práci jsem vypracoval samostatně, pouze s použitím uvedené (citované) literatury.

Datum, podpis

.....

## **Poděkování**

Ing. Michalovi Šerýmu, vedoucímu bakalářské práce, za jeho čas a dobré věcné připomínky.

Děkuji Ing. Vladimíru Kobzevovi za poskytnutí užitečných materiálů a konzultace ohledně realizace redakčního systému.

# Obsah

<b>1 Úvod</b> .....	<b>9</b>
1.1 Cíl práce .....	9
1.2 Vymezení pojmů .....	9
1.2.1 Redakční systém .....	10
1.2.2 Databázový systém .....	10
1.2.3 Relační databáze .....	10
1.2.4 Internet .....	10
1.2.5 WWW, HTTP .....	10
1.3 Redakční systém .....	11
<b>2 Trendy vývoje webových stránek</b> .....	<b>13</b>
2.1 Statické www stránky .....	13
2.2 Dynamické webové aplikace .....	13
<b>3 Prostředky tvorby redakčního systému.</b> .....	<b>15</b>
3.1 Porovnání programovacích a skriptovacích jazyků .....	15
3.1.1 PHP .....	15
3.1.2 ASP.NET .....	16
3.1.3 JavaScript .....	17
3.1.4 Zhodnocení .....	17
3.2 Spojení s databázovým systémem .....	18
3.2.1 Možnosti přístupu k databázovým zdrojům .....	18
3.2.2 Zhodnocení .....	19
3.3 Možnosti autentizace .....	19
3.3.1 Základní HTTP .....	19
3.3.2 Pomocí formuláře a session .....	20
3.3.3 Zhodnocení .....	20
3.4 Validace vstupů .....	20
3.5 Generování výstupu .....	21
3.5.1 Vložené PHP .....	21
3.5.2 Oddělení společných částí .....	22
3.5.3 Šablonovací systémy .....	22
<b>4 Návrh a příprava realizace RS.</b> .....	<b>24</b>
4.1 Specifikace požadavků .....	24
4.1.1 Vývojář .....	24
4.1.2 Provozovatel / administrátor .....	25
4.1.3 Uživatel, návštěvník .....	25
4.1.4 Z pohledu bezpečnosti .....	25
4.2 Adresářová struktura .....	26
4.3 Konfigurační soubory .....	27
4.4 Vícejazyčné verze .....	27
4.5 Jádro a moduly .....	29
4.5.1 Sdílené prostředky jádra .....	29
4.5.2 Moduly .....	30

<b>5 Realizace ukázkového RS.....</b>	<b>32</b>
5.1 Validátor .....	32
5.2 Generování výstupního (X)HTML kódu .....	33
5.3 Jednotlivé obsahové bloky .....	34
5.4 Databázová abstrakční vrstva.....	35
5.5 Moduly .....	36
5.5.1 Stránka .....	36
5.5.2 Články .....	36
5.5.3 Novinky .....	37
5.5.4 Diskusní fórum.....	38
5.6 Administrace .....	39
5.6.1 Oprávnění a práva administrátorů .....	40
5.6.2 Správa článků .....	41
5.6.3 Reklama .....	42
5.7 Zhodnocení .....	43
<b>6 Závěr.....</b>	<b>45</b>
<b>7 Seznam literatury.....</b>	<b>46</b>
<b>A Seznam použitých zkratk.....</b>	<b>47</b>
<b>B Obsah příloženého CD.....</b>	<b>48</b>

## Seznam obrázků

3.1: Ukázka HTTP autentizačního dialogového okna . . . . .	19
3.2: Ukázka přihlašovacího HTML formuláře . . . . .	20
4.1: Ukázka části webového administračního rozhraní „Nastavení systému“ . . .	27
4.2: Ukázka výběru jazykové verze umístěného v hlavičce stránky . . . . .	28
4.3: Ukázka možností administrace jednotlivých modulů . . . . .	31
5.1: Příklad hlášení vygenerované při chybně vyplněném formuláři . . . . .	33
5.2: Schéma rozdělení stránky do jednotlivých obsahových bloků . . . . .	35
5.3: Ukázka grafického uspořádání novinek na úvodní stránce . . . . .	37
5.4: Příklad kontrolní otázky sloužící jako ochrana proti spamu . . . . .	38
5.5: Celkový pohled na administrační rozhraní redakčního systému . . . . .	39
5.6: Formulářové prvky použité při editaci článku . . . . .	41





# 1 Úvod

## 1.1 Cíl práce

Cílem práce je seznámit čtenáře s problematikou vývoje redakčních systémů v prostředí webu. Práce by měla shrnout možné přístupy k tvorbě www stránek a popsat a shrnout dostupné programové prostředky pro jejich realizaci. Taktéž by měla čtenáře seznámit se strukturou redakčních webových systémů (RS), poukázat na možná bezpečnostní rizika, vyhodnotit požadavky různých uživatelů systému (vývojářů, administrátorů a běžných uživatelů), navrhnout a implementovat ukázkový redakční systém.

## 1.2 Vymezení pojmů

### 1.2.1 Redakční systém

Úkolem redakčního systému je poskytnout jak provozovateli, tak uživateli komfortní a přehledný přístup k informacím v prostředí internetu. Provozovateli poskytuje pohodlnou a především rychlou tvorbu nového obsahu, případně změnu obsahu stávajícího. Uživateli potom umožňuje přehledné procházení tímto obsahem a současně mu nabízí ucelený souhrn funkcí, jako je vyhledávání obsahu, formátování pro tisk a další rozmanité služby.

### 1.2.2 Databázový systém

Databáze (DB) je kolekce trvalých informací (dat) uložených na paměťovém médiu. Softwarové prostředky pro správu a přístup k uloženým datům jsou označovány jako Systém řízení báze dat (DBMS) a slouží pro řízení přístupu k datům a správu reprezentace dat na úrovni paměťového média. DB a DBMS tvoří databázový systém.

Databázový systém zahrnuje:

- datové prvky – elementární hodnoty
- vztahy mezi prvky dat – složitější datové struktury, které tvoří jednotlivé záznamy
- integritní omezení – podmínky, které musí data uložená v databázi splňovat

- logické schéma – popis záznamů a jejich vazeb z uživatelského hlediska
- fyzické schéma – řeší uložení dat na paměťovém médiu

V současné době jsou nejvíce rozšířené relační a objektově-relační databázové systémy.

### 1.2.3 Relační databáze

Počátkem 70 let vznikají první databáze, které pohlížejí na data jako na tabulky. Kolem roku 1974 vzniká první verze dotazovacího jazyka SQL (Structured Query Language), který byl vyvinut právě pro komunikaci s relačními databázemi. Pomocí SQL popisujeme, jaká data chceme získat a nikoliv jakým způsobem. Výsledky dotazů provedených nad relační databází se vrací v podobě tabulek.

### 1.2.4 Internet

Internet je celosvětová „supersíť“, která spojuje menší lokální sítě pomocí protokolů TCP/IP. Název vznikl složením dvou slov *inter*, které znamená mezi, a *net* (od anglického network), které znamená síť. Internet vznikl z původně armádního projektu ARPANET. Počátkem 60. let americká armáda hledala způsob, jakým zajistit komunikaci armádních počítačů, které se nacházejí v různých městech nebo státech. V roce 1964 vzniká koncepce switching networks, která nemá žádné centralizované řízení. V roce 1966 vzniká první návrh sítě ARPANET a v roce 1969 je tato síť realizována čtyřmi uzly, které se nacházejí v různých městech. Pojem internet vzniká v roce 1987, kdy jím bylo propojeno 27 000 počítačů.

### 1.2.5 WWW, HTTP

WWW je zkratkou pro Word Wide Web, což v překladu znamená „Celosvětová pavučina“. Tento název je spojován se jménem Tim Barners-Lee, který v dubnu roku 1989 publikuje návrh vývoje WWW a v listopadu roku 1990 spouští na svém stolním počítači první webový server.

WWW je kombinací čtyř důležitých součástí:

- Hypertext – formát informací, který dovoluje „přechod jedním kliknutím“ na jinou část dokumentu nebo zcela jiný dokument
- Identifikátor zdroje (resource identifier) – unikátní identifikátor určující umístění serveru a dokumentu na něm uloženém (známé jako URL, dnes URI)

- Architektura Klient/Server – klient se pomocí URL dotazuje na dokumenty uložené na serveru, server je poskytne klientovy
- Značkovací jazyk (markup language) – jazyk, jehož zdrojový text obsahuje současně vlastní text i instrukce pro jeho zpracování

HTTP (Hyper Text Transfer Protocol) je internetový protokol původně určený pro přenos hypertextových dokumentů ve formátu HTML. Jeho počátku jsou někde kolem roku 1990. První rozšířenou verzí protokolu byla verze 0.9, která byla velmi jednoduchá a sloužila pouze k přenosu textu. To se brzy ukázalo jako velice limitující a vznikla verze 1.0, která umožňovala přenos nejen textu, ale i souborů. Současná verze protokolu je 1.1. Základní vlastnosti:

- neřeší transport dat mezi partnery
- nestará se o detekci přenosových chyb
- je bezstavový (model požadavek - odpověď)
- mezi jednotlivými spojeními není uchována žádná informace
- požadavek i odpověď mají shodnou strukturu
- délka zprávy nemá předepsány žádné meze
- délka řádku není omezena
- není omezen na znaky standardního ASCII kódu

### 1.3 Redakční systém

Redakční systém (RS) je informační systém typu klient/server využívající služeb tenkého klienta. Jako klient slouží webový prohlížeč a komunikačním protokolem jsou protokoly rodiny HTTP. Webové servery předávají klientům informace v podobě (X)HTML. Jak již bylo řečeno, protokol HTTP je bezstavový, tzn. mezi jednotlivými spojeními není uchována žádná informace. Tato skutečnost vyžaduje odlišný přístup při tvorbě takového systému. Je jasné, že pro ověření uživatele budeme muset provést autentizaci při každém požadavku uživatele a z tohoto požadavku vyplývá celá řada funkcionalit RS. Jejich podpora přímo v používaných programovacích jazycích často vůbec neexistuje, proto si ji programátor musí doprogramovat sám. Naštěstí pro řadu podobných úloh existují již hotové knihovny, které jsou volně k dispozici. U řady redakčních systémů bylo vyvinuto jednotné objektové jádro, které tyto knihovny zapouzdřuje a odstraňuje nevýhody prostředí webu.

Každá pokročilá webová aplikace by měla mít tyto vlastnosti:

- modularita
- zabezpečení aplikace
- validace vstupních dat
- uživatelské autentizace a správu uživatelů
- autorizace a přístupová práva
- jednotná databázová vrstva
- logování událostí a akcí
- validita výstupního kódu
- maskování URL

## 2 Trendy vývoje webových stránek

### 2.1 Statické www stránky

Základní tvorba www stránek vychází ze struktury HTML – jedná se o vlastní text doplněný značkovacími strukturami, které slouží v klientském prohlížeči k formátování výstupu. Dnes se s takovými stránkami samozřejmě také setkáváme, ale rozhodně se nehodí pro rychlou správu a údržbu obsahu. Také neumožňují uživateli vkládání informací (diskusní fóra, komentáře, ankety) s výjimkou uploadu souborů.

Stále však mohou velice dobře posloužit, jedná-li se o obyčejné informační stránky a díky jednoduchosti jejich vytvoření díky tzv. WYSIWYG (What you see is what you get, česky „co vidíš to dostaneš“) editorům. Z nejnámějších uveďme například MS Word nebo MS Front Page, kde jednoduše napíšete libovolný dokument a uložíte jako webovou stránku.

### 2.2. Dynamické webové aplikace

Díky nutnosti generovat výslednou www stránku v závislosti na aktuálně dostupných informacích / datech začaly vznikat skriptovací a programovací jazyky, vygenerují obsah stránek až v okamžiku žádosti uživatele.

Ve spojení s databázovým systémem tak vzniká velice mocný nástroj pro vytvoření libovolného informačního systému.

Obecně lze prostředky pro tvorbu dynamických stránek rozdělit na dvě skupiny

- serverové
- klientské

Serverové prostředky, nejčastěji skriptovací jazyky, umožňují vygenerovat unikátní obsah stránky odpovídající přesně dotazu odeslaného uživatelem. Tento vygenerovaný obsah je odeslán na klientskou stanici, kde se již pouze zobrazí. Na hardware a software klientské stanice tedy nejsou kladeny žádné zvýšené nároky. Do této skupiny patří například technologie PHP, ASP.NET, Python a další – budeme se jim věnovat v následující kapitole.

Mezi klientské prostředky můžeme zařadit například JavaScript. aby tato technologie mohla správně pracovat, musí ji klientská stanice podporovat. Kód, který je součástí webových stránek se tedy vykonává až na straně uživatele, což zvyšuje nároky na

schopnosti prohlížeče a samozřejmě tím také mohou nastat zvýšené nároky na výkon hardwaru.

## 3 Prostředky tvorby redakčního systému

Hlavní část této práce se věnuje realizaci redakčního systému za použití skriptovacího jazyka PHP ve spojení s databází MySQL. Tato kapitola by měla přiblížit další možné jazyky, které se dnes používají k tvorbě webových aplikací a na závěr porovnat jejich přednosti a nevýhody.

### 3.1 Porovnání programovacích jazyků

#### 3.1.1 PHP

PHP je skriptovací programovací jazyk, určený především k tvorbě dynamických webových stránek. Popularitu získal díky své jednoduchosti, rychlosti (ve srovnání s jinými interpretovacími jazyky), možnosti snadného rozšíření a v neposlední řadě díky komunitě, která jej stále vyvíjí, rozšiřuje a zdokonaluje.

Počátky PHP sahají do roku 1994, kdy Rasmus Lerdorf vytvořil sadu skriptů v Perlu pro počítání přístupů na jeho webových stránkách. O tyto skripty začal být zájem, proto je vydal přepsané do C jako balíček „Personal Home Page“ tools, zkráceně PHP. Vzhledem k zájmu napsal v roce 1995 skriptovací jádro spojené s jiným jeho nástrojem pro analýzu vstupu z HTML formuláře „Form Interpreter“ (zkráceně FI), které bylo nazváno PHP/FI nebo také PHP2. PHP3 vyšlo v roce 1998 a již bylo dílem celé skupiny vývojářů pod vedením Rasmuse Lerdorfa. V této verzi bylo přepracováno a rozšířeno jádro a přidáno jednoduché API, které umožňuje jiným programátorům přidávat do PHP vlastní funkce vytvářením modulů. V tuto chvíli kolem PHP vzniká velká komunita a PHP se stává nejpopulárnějším skriptovacím jazykem v prostředí webových aplikací. V roce 2000 vychází PHP4 a o čtyři roky později PHP5 s vylepšeným objektovým přístupem, podobným jazyku Java.

#### Výhody použití PHP:

- relativně jednoduché na pochopení
- má syntaxi velice podobnou jazyku C a je tedy většině programátorů dost blízký
- podporuje širokou škálu souvisejících technologií, formátů a standardů
- podporuje celou řadu internetových protokolů (HTTP, SMTP, FTP, IMAP, POP3, LDAP atd.)
- podporuje velké množství databází (MySQL, PostgreSQL, Oracle, MSSQL atd.)

- jedná se o otevřený projekt s rozsáhlou komunitou
- velké množství hotových knihoven a skriptů obvykle šířených pod některou z volných licencí
- dobré spojení s webovým serverem Apache (od verze 4 i jako modul Apache)
- je multiplatformní
- podpora téměř na všech webhostinzích

#### Nevýhody:

- PHP je interpretovaný, nikoliv kompilovaný (snížení rychlosti)
- kdokoliv s přístupem k serveru může nahlédnout do našich PHP skriptů
- vývojáři PHP ne zcela dodržují zpětnou kompatibilitu mezi verzemi (především mezi verzí 4 a 5)

### **3.1.2 ASP.NET**

ASP.NET je nadstavba .NET Frameworku firmy Microsoft pro tvorbu webových aplikací a služeb. Je nástupcem technologie ASP. ASP.NET je založen na CLR, který je sdílen všemi aplikacemi postavenými na .NET Frameworku. Programátoři tak mohou realizovat všechny své projekty v jakémkoliv jazyce podporujícím CLR, např. Visual Basic.NET, JScript.NET, C#, Managed C++, ale i mutace Perlu, Pythonu a další. Aplikace založené na ASP.NET jsou také rychlejší, neboť jsou předkompilovány do jednoho či několika DLL souborů.

ASP je skriptovací nástroj vyvinutý firmou Microsoft v roce 1996 pro tvorbu dynamických webových stránek. První verze ASP.NET byla vydána v lednu 2002 a byla součástí .NET Framework. Verze 1.1 byla uvolněna v dubnu 2003 a verze 2.0 v listopadu 2005.

#### Výhody použití ASP.NET:

- kompiluje zdrojové kódy a díky tomu jsou aplikace rychlejší
- díky kompilaci je více chyb zachyceno během vývoje
- rozděluje kód aplikace a HTML pomocí šablon
- na výběr z velkého množství programovacích jazyků
- automaticky řeší bezstavovost pomocí session

#### Nevýhody:

- uzavřený kód – nutnost spolehnout se na bezpečnost implementace



- nedokonalá nebo žádná podpora na jiných operačních systémech než MS Windows
- nekompatibilita session mezi jednotlivými verzemi
- závislost mnoha funkcionalit na JavaScriptu (z toho plyne horší dostupnost přes vyhledávače)

### 3.1.3 JavaScript

Ačkoliv to není přímo programovací jazyk pro hlavní tvorbu webových stránek, považují ho natolik za důležitý, že ho zde zmíním.

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, jehož autorem je Brendan Eich z tehdejší společnosti Netscape. V současné době je nejvíce používán jako interpretovaný programovací jazyk pro WWW vkládaný přímo do HTML kódu. Obvykle s ním jsou ovládány některé interaktivní prvky nebo tvořeny animace a efekty. Jako další velice často používaná funkce JavaScriptu na WWW stránce je kontrola hodnot zadaných do formulářů.

Program v JavaScriptu se obvykle spouští až po stažení WWW stránky z Internetu na klientský počítač, narozdíl od ostatních interpretovaných jazyků jako je např. PHP či ASP.NET. Z toho plynou některá bezpečnostní omezení jazyka, jako je například nemožnost práce se soubory, aby nedošlo k porušení soukromí uživatele.

### 3.1.4 Zhodnocení

**ASP.NET** Jedná se o jediný kompilovaný jazyk, z toho plyne jeho rychlost. Nevýhodou je uzavřený kód, který je vlastnictvím firmy Microsoft a z toho plyne špatná podpora na jiných OS než MS Windows. Hostingsy pro tento jazyk jsou ve srovnání s PHP málo rozšířené a drahé. Mnoho funkcí je závislých na JavaScriptu, tedy špatně se indexují vyhledávači, což může být v případě internetového obchodu zásadní překážka. Proto bych ho nedoporučil pro tvorbu projektů, které jsou na vyhledávacích závislé, naopak velice se hodí pro různé interní, veřejnosti uzavřené, systémy, kde lze možnosti JavaScriptu plně využít.

**PHP** je ideálním řešením pro malé a středně velké webové aplikace (u velkých je limitujícím faktorem rychlost). Je podporován téměř všemi hostingsy, cena takového

hostingu nebývá velická. Při použití správných knihoven usnadňujícími např. práci s databází, generování výstupu apod. s ním lze rychle a efektivně pokrýt většinu běžných požadavků na dnešní webové systémy.

Samozřejmě existuje mnoho dalších používaných technologií, jako jsou například interpretovací jazyky Perl či Python. Jejich použití ale není příliš časté a obvykle se používají na specializované projekty, kde se využívají některé jejich výhody jako může být například stabilita či rychlost.

## 3.2 Spojení s databázovým systémem

### 3.2.1 Možnosti přístupu k datovým zdrojům

**Nativní funkce.** Všechny DBMS jsou vybaveny nativními funkcemi, které zajišťují základní komunikační procesy – jedná se o komunikaci na nejnižší úrovni. Tyto funkce jsou programátory koncových aplikací málo využívány, není-li kladen na aplikaci požadavek extrémní rychlosti. Tyto funkce jsou tedy jakousi mezivrstvou, kterou používají zapouzdřené funkce.

**Zapouzdřené funkce** zachovávají strukturu daného programovacího jazyka a na pozadí volají funkce nativní. Programátor potom pracuje s funkcí či komponentou podle svých zvyklostí a již se nezajímá o komunikaci s databází. PHP podporuje velké množství databází. Problém je, že funkce pro práci s jednotlivými databázemi jsou pojmenovány různě, takže ztrácíme obecnost. Řešením je použití některého obecného propojovacího protokolu nebo abstrakční vrstvy.

**Obecný propojovací protokol ODBC.** Tento protokol vznikl kvůli potřebě jednoduchého přechodu z jedné databáze na jinou. ODBC obsahuje standardizované API, pomocí kterého je možné přistupovat k databázím shodným způsobem. Velkou nevýhodou je nízká výkonnost oproti standardním zapouzdřeným funkcím.

**Abstrakční vrstva.** Jde o sadu funkcí pro přístup k databázím, které podle typu používají příslušné zapouzdřené funkce. Jedná se o zobecnění zapouzdřených funkcí. Nevýhodou je možnost volání pouze těch zobecněných funkcí, které podporují všechny databáze a omezení společného SQL příkazu na společný standard. Pro příklad uveďme ADODB a Pear::DB.

### 3.2.2 Zhodnocení

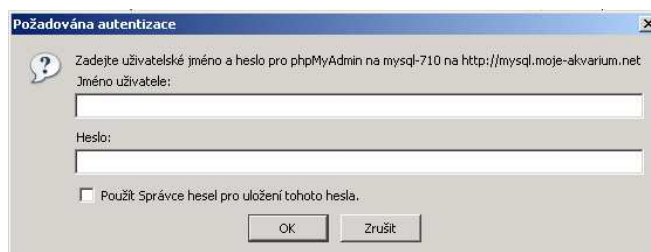
Pro potřeby redakčního systému se jeví jako nejlepší použít abstrakční vrstvy. Na jedné straně používá zapouzdřené funkce PHP pro konkrétní databázi (tzn. je mnohem rychlejší než ODBC), na straně druhé poskytuje obecný přístup k databázi. Některé abstrakční vrstvy dokonce řeší nesrovnalosti ve funkcionalitách jednotlivých databází, jako je například klíčové slovo LIMIT v databázi MySQL. Databáze MS SQL toto klíčové slovo nezná a používá místo něj klíčové slovo TOP, které má ve srovnání s LIMIT pouze omezené možnosti.

## 3.3 Možnosti autentizace

Autentizace je ověření identity uživatele, což je v prostředí víceuživatelských webových aplikací klíčové. Díky ní potom můžeme rozlišit jednotlivé uživatele, případně uživatelské role.

### 3.3.1 Základní autentizace HTTP

Jedná se o autentizační schéma, které je integrováno přímo v protokolu HTTP.



Obrázek 3.1: Ukázka HTTP autentizačního dialogového okna

Server si hlavičkou

```
WWW-Authenticate: Basic realm="RealmFoo"
```

vyžádá autentizaci uživatele. Prohlížeč zobrazí přihlašovací formulář a po zadání údajů je uloží v prohlížeči. Tyto údaje jsou poté při každém načítání dat odesílány zpět serveru, kde jsou znovu a znovu ověřovány.

Hlavní výhodou je, že lze tuto autentizaci požit k ochraně všech souborů na serveru, nejen pro skripty, protože jde o mechanismus na úrovni protokolu HTTP. Hlavní nevýhodou je možnost odchycení uživatelského jména a hesla při jejich opakovaném odesílání přes internet.

### 3.3.2 Autentizace pomocí formuláře a session

Pokud je vyžadována autentizace, webová aplikace vygeneruje HTML formulář s poli potřebnými pro úspěšné ověření uživatele. Formulář se odesílá výhradně HTTP metodou POST (metoda GET předává odeslané parametry v adresním řádku, takže se ukládají do historie prohlížeče a případných logů). Po úspěšné autentizaci vygeneruje server session token, který je odeslán uživateli a uložen v prohlížeči (nejčastěji ve formě cookies).



The image shows a simple login form with a yellow border. It contains two text input fields: the first is labeled 'Uživatelské jméno :' and contains the text 'Filip'; the second is labeled 'Heslo :'. Below the password field is a button labeled 'Přihlásit'.

Obrázek 3.2: Ukázka přihlašovacího HTML formuláře

Fakt, že jméno a heslo putuje internetem jen jednou (na začátku relace) je velkou výhodou. PHP od verze 4 přímo podporuje mechanismus session. Ty jsou ukládány v superglobálním poli `$_SESSION`, se kterým lze pracovat jako s jakýmkoliv jiným polem. Změny v tomto poli jsou zaznamenávány po skončení běhu skriptu.

### 3.3.3 Zhodnocení

Autentizace pomocí HTML formuláře a session se jeví jako lepší možnost pro webové aplikace a také je toho v praxi hojně využíváno. Zmenšuje možnost odposlechu citlivých údajů, programátorovi umožňuje větší variabilitu. Pro úplné zabezpečení údajů je možné použít šifrovaný protokol HTTPS.

## 3.4 Validace vstupů

Z klientského počítače se mohou serveru poslat data dvěma HTTP metodami (lze je i kombinovat). Metoda GET posílá data jako součást adresy (jako parametry oddělené „&“, respektive “?”). To se hodí např. pro posílání dat, která jsou součástí URI a říkají, jaký dokument / stránka se má načíst. Celý adresní řádek včetně parametrů se potom ukládá do historie prohlížeče.

Data, která nesouvisí s aktuálním stavem načtených stránek (nevyjadřují přesné umístění na serveru) je lepší posílat metodou POST. Pro delší texty, soukromé údaje a jiné citlivé informace je posílání dat metodou GET z hlediska zabezpečení rizikové.

#### Hlavní vlastnosti validátoru:

**Povolit pouze známá, validní data.** U vstupních dat můžeme znát jejich typ, rozsah nebo možnou podobu. Kontrola typu je velice důležitá obzvláště v případě PHP, které nemá povinnou deklaraci typu u proměnné a automaticky je přetypovává.

Pokud známe rozsah, měli bychom ověřit, že obdržená hodnota tomuto rozsahu odpovídá.

Nakonec známe-li podobu dat, provedeme kontrolu pomocí regulárních výrazů, případně porovnáme s výčtovou množinou.

**Nevalidní známá data.** Jedná se o data, o kterých víme, že validace skončila neúspěchem. V takovém případě můžeme data zrušit nebo místo nich použít bezpečnou (defaultní) hodnotu.

**Ostatní data zlikvidovat.** Pod ostatními daty rozumíme data v proměnných, které neočekáváme.

## 3.5 Generování výstupu

Výstupem webové aplikace jsou webové stránky ve formátu (X)HTML. Proberme techniky, kterými můžeme takového výstupu dosáhnout.

Ačkoliv další informace lze považovat za obecné, věnoval bych se v tomto okamžiku již konkrétním způsobům v PHP, se kterým mám největší zkušenosti.

### 3.5.1 Vložené PHP

Jedná se o techniku vkládání krátkých PHP skriptů přímo do (X)HTML stránky.

```
<html>
<body>
<?php
echo "Tento text vypsalo PHP.";
?>
</body>
</html>
```

To umožňuje rychlé vytváření prototypů a není nutné dodržovat strukturu. Výsledkem je smíšený kód HTML/PHP. Tento postup vede k nečitelnosti a velmi pracné udržitelnosti kódu. Je vhodný pro vytváření malých, rychlých skriptů, nikoliv pro vytváření ucelených webových systémů.

### 3.5.2 Oddělení společných částí

Tato technika vylepšuje předchozí způsob tím, že oddělí společné části kódu do samotných skriptů. V praxi to může být například záhlaví a zápatí stránek. Tento postup neodstraňuje smíšený kód, ale jistým způsobem ho strukturalizuje a umožňuje použití společných částí. Dříve byla tato technika často používána, dnes se hodí spíše jen na malé projekty a i tak bych se jí raději vyhnul.

### 3.5.3 Šablonovací systémy

Šablonovací systém zcela odděluje výstupní (X)HTML kód, který je reprezentován šablonami, a PHP kód, který je obsažen ve skriptech. Šablony i funkcionalitu webového systému je tak možno vyvíjet samostatně. Šablonovací systém umožňuje jednoduchou změnu vzhledu pouhou změnou šablony.

Nevýhodou šablonovacích systémů je nutnost naučit se pseudojazyk vkládaný do (X)HTML šablon (který bývá obvykle velmi jednoduchý) a pomalejší generování výstupu (šablony se musí nejprve zkompileovat a propojit s výstupem vygenerovaným z PHP skriptů). Pomalejší generování výstupu lze často urychlit cachováním předkompilovaných šablon. Toto řešení je ideální pro středně velké i velké projekty a umožňuje rozdělení práce v týmu.

K dispozici je celá řada šablonovacích systémů například Template Power, či Smarty. Asi nejznámějším a nejrozšířenějším šablonovacím systémem je právě Smarty, jehož vývoj je podporován samotným PHP týmem. Jeho hlavní výhody oproti jiným šablonovacím systémům jsou:

- jedná se o živý projekt podporovaný PHP týmem
- velké množství funkcionalit
- dobře zpracovaný debug výstup
- možnost rozšíření vlastními pluginy
- práce s cache na serveru

Hlavní nevýhodou je rychlost, kterou lze částečně zvýšit cachováním výstupu.

## 4 Návrh a příprava realizace RS

V této kapitole již přejdeme ke konkrétnímu popisu ukázkového redakčního systému. Nejprve shrneme požadavky jednotlivých skupin uživatelů (administrátoři, redaktoři, uživatelé), podíváme se na adresářovou strukturu RS, konfiguraci a přiblížíme si, jak funguje jádro celého systému.

Na závěr shrneme funkce některých základních modulů a podíváme se na administrační rozhraní systému.

### 4.1 Specifikace požadavků

Nejprve připomeňme základní požadavky (viz. kapitola 1.3), které by měla obecně webová aplikace splňovat:

- modularita
- zabezpečení aplikace
- validace vstupních dat
- uživatelské autentizace a správu uživatelů
- autorizace a přístupová práva
- jednotná databázová vrstva
- logování událostí a akcí
- validita výstupního kódu

#### 4.1.1 Vývojář

Z pohledu vývoje celého systému je velice důležité oddělit vlastní PHP skripty od výsledného (X)HTML kódu. Usnadní to případné ladění systému a odhalování chyb, zpřehlední PHP skripty, zjednoduší rozšiřování systému a změny výstupního (X)HTML kódu. Dále musí být systém do budoucna jednoduše rozšiřitelný, proto jsem se rozhodl ho vytvářet modulárně.



### **4.1.2 Provozovatel / administrátor**

Rozhodne-li se někdo zřídit redakční systém, zajímají ho jednotlivé moduly a jejich možnosti, případně jejich konfigurovatelnost a (jednoduchá) možnost přizpůsobení systému vlastním požadavkům.

Systém proto disponuje vlastním administračním rozhraním, kde je možné vypínat či zapínat a konfigurovat celé funkční bloky, spravovat obsah webu a uživatele.

### **4.1.3 Uživatel, návštěvník**

Koncový uživatel, návštěvník neví nic o vnitřní struktuře systému a pohlíží na něj jako celek. Používá jej k vyhledávání a sdílení informací s jinými uživateli. Pro odlišení jednotlivých uživatelů systém podporuje autentizaci a registraci uživatelů. Také uživatel očekává možnost jisté personalizace vzhledu a chování RS. Shrňme požadavky:

- RS poskytuje uživatelům služby
- autentizace pro rozlišení uživatelů
- registrace
- autorizace v případě poskytnutí některých služeb pouze autorizovaným uživatelům
- personalizace (nastavení vlastního vzhledu, případně chování RS)
- pohodlný přístup k dokumentům, službám, přehlednost

### **4.1.4 Z pohledu bezpečnosti**

Na jednotlivé bezpečnostní problémy jsme upozorňovali v průběhu jednotlivých kapitol a ještě se jí budeme věnovat v příštích kapitolách. Shrňme tedy jen základní požadavky na bezpečnost webové aplikace:

- validovat vstupní data od uživatele
- udělovat pouze nejnutnější přístupová práva
- používat jednoduché a přehledné řešení
- chránit pomocné skripty před samostatným spuštěním (použitím přesné URI na ně ukazující)

## 4.2 Adresářová struktura

Z hlediska přehlednosti a bezpečnosti je nutné navrhnout správnou adresářovou strukturu redakčního systému a předem si ujasnit, kde budou která data a skripty uloženy. Výpis adresářové struktury vypadá následovně:

```
/
/data/
/data/articles/
/data/articles/img/
/documentation/
/images/
/includes/
/install/
/languages/
/scripts/
/theme/
/theme/default/
/theme/default/templates/
```

Pojďme si nyní podrobně rozepsat, k čemu který adresář slouží.

`/` je kořenový adresář RS. Obsahuje pouze dva základní soubory, které se volají z prohlížeče: `index.php` a `admin.php`. Princip jejich činnosti si podrobněji popíšeme později.

**`/data/`** slouží jako úložiště uživatelských dat a podpůrnému obsahu, který není možné uložit do databáze. Konkrétními podsložkami jsou například *articles* a *articles/img*, které slouží k ukládání obsahu souvisejícího s články. Jednotlivé moduly mají pokud je to potřeba vlastní podsložky.

**`/documentation/`** zde asi není potřeba něco rozepisovat. Adresář obsahuje jednoduchou dokumentaci k redakčnímu systému, případně nápovědu.

**`/includes/`** je klíčovým adresářem jádra a obsahuje všechny sdílené skripty, třídy a funkce. Veškerý obsah je chráněn proti samostatnému spuštění.

**`/install/`** Jsou v něm uloženy všechny soubory a procedury potřebné k nainstalování systému – tedy vytvoření databáze, vložení inicializačních dat, základní nastavení, případně jiné procedury.

**`/languages/`** obsahuje potřebné jazykové překlady jednotlivých předem definovaných textových řetězců

**`/theme/`** obsahuje šablony XHTML výstupu, potřebné grafické prvky dané šablony a vše co pevně souvisí s konkrétní šablonou. Jednotlivé šablony vzhledu

redakčního systému mají vlastní podadresáře (standardní je šablona v podadresáři *default*).

### 4.3 Konfigurační soubory

Konfigurační soubory lze v tomto případě rozdělit na základní, který obsahuje minimum informací o nastavení systému, ale obsahují vše potřebné k jeho správné činnosti. Jedná se především o údaje nutné k připojení k databázi, prefix v názvech tabulek v databázi a šifrovací klíč, který se přidává na konec všech zadaných hesel, ještě před jejich zašifrováním, pro zvýšení bezpečnosti.

**Nastavení systému (globálních proměnných)**

Index sekce	Obnovit soubor s nastavením	Přidat položku			
<a href="#">Přepnout</a> <a href="#">Editovat</a> <a href="#">Odstranit</a>			<b>Jméno proměnné</b>	<b>Hodnota</b>	<b>Popis</b>
			advertisement	TRUE	Zobrazovat reklamu ano x ne
<a href="#">Přepnout</a> <a href="#">Editovat</a> <a href="#">Odstranit</a>			advertisement_bottom	TRUE	Zobrazovat reklamu dole (pokud je reklama povolena)
<a href="#">Přepnout</a> <a href="#">Editovat</a> <a href="#">Odstranit</a>			advertisement_left	TRUE	Zobrazovat reklamu vlevo (pokud je reklama povolena)
<a href="#">Přepnout</a> <a href="#">Editovat</a> <a href="#">Odstranit</a>			advertisement_right	TRUE	Zobrazovat reklamu vpravo (pokud je reklama povolena)

Obrázek 4.1: Ukázka části webového administračního rozhraní „Nastavení systému“

Druhým konfiguračním souborem je soubor, vlastně cachovaný soubor, s uživatelským nastavením, konfigurací vlastního redakčního systému, nastavením pravidel a omezeních, konfigurace jednotlivých modulů apod. Toto nastavení je uloženo v databázi, ale pro zvýšení rychlosti systému se při jakékoliv změně nastavení vytvoří soubor na disku obsahující toto nastavení. Narozdíl od prvního konfiguračního souboru probíhá nastavení proměnných obsažených zde přes webovou administraci redakčního systému (webové administraci bude věnována samostatná kapitola)

Oba konfigurační soubory jsou umístěny v adresáři */includes/*.

### 4.4 Vícejazyčné verze

Protože v dnešní době o návštěvnosti Vašich stránek rozhodují především internetové vyhledávače, je velice pravděpodobné, že na stránky zavítají i návštěvníci z cizích zemí. Redakční systém proto umožňuje přepínání předem neomezeného počtu jazyků (záleží pouze, zda je nalezen potřebný jazykový „slovník“).

Redakční systém umožňuje toto přepnutí jazyka nejen v uživatelském rozhraní pro návštěvníky, ale také v administračním rozhraní pro správce.



Obrázek 4.2: Ukázka výběru jazykové verze umístěného v hlavičce stránky

RS ve svém kódu používá místo textových řetězců v konkrétním jazyce pouze zástupné proměnné, které jsou v závislosti na aktuálním jazyce předdefinovány konkrétními textovými hodnotami.

Soubory s jednotlivými jazykovými překlady jsou uloženy v adresáři */languages/*, jejich název je dán ve formátu *lang\_xx.php*, kde *xx* představuje zkratku jazyka (např. *cz*, *en*, *de*). Jednotlivé textové řetězce jsou definovány jako konstanty následovně:

```
define ('zastupny_retezec', 'text_retezec');
```

Redakční systém neumožňuje ukládat vícejazyčné verze jednotlivého obsahu (článků apod.). Pokud by takové řešení bylo vyžadováno, znamenalo by to zcela odlišnou strukturu databáze.

Realizovaný redakční systém jde ale v jednotlivých jazykových verzím ještě dále a umožňuje dokonce změnit některá klíčová slova používaná v adresovém řádku prohlížeče. Jako příklad uveďme typicky požadavek na zobrazení statické stránky, které se provede jako *index.php?stranka=alias\_stranky*. Protože ne každému bude vyhovovat pojmenování proměnné *stranka*, může toto klíčové slovo v administraci zaměnit třeba za anglické *page*.

Nutno upozornit, že podobná nastavení se provádějí pouze před ostrým spuštěním RS, protože při provedení této změny za provozu RS by způsobilo neplatnost odkazů zaindexovaných ve vyhledávačích.

## 4.5 Jádru a moduly

V této kapitole se pokusím přiblížit strukturu jádra systému, obsah sdíleného kódu a posloupnosti jeho vykonávání. Podíváme se také na přídavné moduly systému, jejich instalaci a inicializaci.

### 4.5.1 Sdílené prostředky jádra

Jádru redakčního systému je tvořeno souborem *index.php* a obsahem adresáře */includes/*, kde jsou uloženy veškeré stěžejní skripty a třídy pro běh aplikace. Zmíním pouze ty nejdůležitější:

Třída pro generování výstupu podle šablony, třída pro generování a validaci formulářů, třída pro upload souborů, třída a podpůrné funkce pro práci s uživateli, třída pro generování drobečkové navigace, třída počítadla návštěvnosti, funkce pro inicializaci připojení databáze, inicializaci šablony, generování menu, funkce pro generování kódu jednotlivých bloků (hlavička, patička, informační sloupec, informační řádek, generování reklamy), inicializace modulů a další.

Popis obsahu jednotlivých částí kódu je jistě nad rozsah této práce, proto popíši pouze postup vykonávání jednotlivých procesů při vykonávání skriptu.

- v prvním kroku dojde k vložení potřebného kódu do jednoho skriptu (funkce *include*), čímž dojde k načtení obou konfiguračních souborů
- připojení k databázi a nastavení znakové sady ve které bude probíhat komunikace
- inicializace tříd pro práci s uživateli, drobečkovou navigací a počítadla návštěvnosti
- zapnutí cacheování výstupu a inicializace výstupní XHTML šablony – načte se šablona dle konfigurace nebo podle nastavení aktuálního uživatele, pokud není nalezena, načte se defaultní šablona, která musí být vždy přítomna
- dojde k inicializaci jazyka – opět podle nastavení aktuálně přihlášeného uživatele nebo defaultně nastaveného jazyka v konfiguraci
- ověření uživatele – zda se jedná o přihlášeného uživatele nebo pouze anonymního návštěvníka

- v tomto okamžiku začíná generování obsahu a jeho vkládání do XHTML šablony (HTML hlavička)
- zjištění požadavku na případné zobrazení obyčejné „statické“ stránky a vygenerování obsahu
- v případě, že se nejedná o „statickou“ stránku, dojde k inicializaci požadovaného modulu a pokud je nalezen a aktivní, k vygenerování obsahu modulem
- generování podpůrného obsahu jednotlivých bloků, pokud jsou aktivní. Mezi základní bloky patří především přepínání jazyka, zobrazení menu, zobrazení informací o uživateli a umožnění přepnutí výstupní šablony, vyhledávání apod. V administraci je možné vytvářet nové bloky, které mohou mít obsah obyčejného textu, HTML kódu nebo vloženého PHP kódu (což umožňuje velkou variabilitu)
- generování obsahu informačního řádku a sloupce – zde se mohou zobrazovat kontextové tipy, tipy s odkazy na jednotlivé články, některá zajímavá statistická vyhodnocení aj.
- generování několika bloků reklamy
- na závěr dojde ke vygenerování výsledného XHTML kódu k zobrazení, zrušení potřebných tříd a ke spuštění některých servisních funkcí

## 4.5.2 Moduly

Aby byla zajištěna co největší variabilita systému, lze jeho funkce rozšiřovat o nové moduly. Aby byl systém konzistentní, musí být každý modul dodržovat určitá pravidla jako například:

- vrací vygenerovaný XHTML kód do několika proměnných – hlavní obsah, hlavní nadpis stránky, ze kterého se generuje také titulek prohlížeče
- plní drobečkovou navigaci
- pokud je s ním počítáno ve výstupní šabloně, může plnit i další bloky s obsahem

### Instalace modulu

Instalace nového modulu probíhá v několika krocích.

- 1) nakopírování adresáře modulu do složky */modules/* a případné vytvoření dalších potřebných adresářů pro ukládání dat
- 2) vytvoření nových tabulek v databázi a přednapištění daty

3) vytvoření nového řádku v tabulce *modules*, kde je uvedena cesta k modulu, popis aj.

4) na závěr je nutné přihlášení do administrace a aktivace daného modulu

7	Status uživatele	Vložený soubor : ./modules/uzivatele/block_user_status.php	ALL	70	1	<a href="#">Zakázat</a> <a href="#">Editovat</a> <a href="#">Odstranit</a>
1	Vyhledávání	Vložený soubor : ./includes/blocks/search.php	ANONYMOUS REG_USER	60	1	<a href="#">Zakázat</a> <a href="#">Editovat</a> <a href="#">Odstranit</a>
8	Online users	Vložený soubor : ./modules/uzivatele/block_online_users.php	ALL	60	1	<a href="#">Zakázat</a> <a href="#">Editovat</a> <a href="#">Odstranit</a>

Obrázek 4.3: Ukázka možností administrace jednotlivých modulů

### **Inicializace modulu před jeho spuštěním**

- 1) zjištění statusu k aktivaci požadovaného modulu, ověření jeho existence na disku
- 2) načtení jeho jazykových souborů
- 3) vložení kódu modulu a jeho vykonání

## 5 Realizace ukázkového RS

Po zdoluhavém teoretickém rozboru redakčního systému, přistoupíme k jeho realizaci. Protože již víme, co vše náš redakční systém umožňuje a známe i jeho základní strukturu a postup jednotlivých procedur systému při spuštění, podíváme se v následující kapitole na několik konkrétních programových řešení.

Bude se jednat z hlediska bezpečnosti aplikace o velice důležitý validátor HTML formulářů, třídu pro generování výstupního XHTML kódu, použitou databázovou abstrakční vrstvu a vybrané moduly.

Redakční systém má zjednodušit především vkládání a práci s obsahem, proto se podíváme na možnosti administrace systému, přiblížíme si jednotlivé uživatelské role administrátorů a shrneme možnosti práce s obsahem.

### 5.1 Validátor formulářů

V kapitole 3.4 jsme si již řekli, jaké jsou na validátor kladeny požadavky. Nyní se podíváme blíže jak tato třída pracuje.

Pro validátor je použita mnou modifikovaná třída EasyFormProc, která je za určitých podmínek volně k dispozici na <http://www.nazly.net>. Nebudu zde ukazovat jednotlivé PHP příkazy a funkce vlastní třídy, protože celá třída je poměrně obsáhlá, ale naznačím jednotlivé kroky.

- inicializace třídy a vlastního formuláře
- vytvoření jednotlivých polí (v tomto kroku se nastavuje typ editovaného pole, název, CSS, pravidla pro kontrolu vložených hodnot, případně další parametry které má každý typ pole jiné)
- vygeneruje se XHTML kód formuláře
- v případě odeslání formuláře se provede kontrola zadaných hodnot – pokud je tato kontrola v pořádku pokračuje skript se zpracováním informací, v opačném případě opět vypíše formulář s již zadanými hodnotami

Použití validátoru s těmito funkcemi značně zjednoduší práci s HTML formuláři, zrychlí kontrolu vložených dat do formuláře a sníží velikost výsledného kódu aplikace.



Formulář nevyhovuje v následujících bodech :

- Nevyhovuje uživatelské jméno (délka 5 až 15, povolené znaky a-z, A-Z, 0-9, podtržítka)
- Heslo nesplňuje povinné parametry nebo se liší
- Nevyhovuje délka celého jména
- Nekorektní emailová adresa
- Vyberte typ administrátorského účtu!

Obrázek 5.1: Příklad hlášení vygenerované při chybně vyplněném formuláři

## 5.2 Generování výstupního (X)HTML kódu

V kapitole 3.5 jsme si popsali možné způsoby, jak lze v PHP generovat HTML kód a vysvětlili jsme si funkci šablonovacích systémů. Nyní si popíšeme konkrétní šablonovací systém použitý v ukázkovém redakčním systému : Template Power

Prvním krokem při použití tohoto šablonovacího systému je vytvoření samotné HTML nebo chcete-li XHTML šablony, která může vypadat například takto:

```
<!-- START BLOCK : bl_head -->
<div id="head">
  <!-- START BLOCK : bl_search -->
  <div id="search">
    {SEARCH}
  </div>
  <!-- END BLOCK : bl_search -->
  {HEAD}
</div>
<!-- END BLOCK : bl_head -->
```

Klíčovými slovy jsou tedy příkazy START BLOCK a END BLOCK, které vyznačují konkrétně v ukázkovém kódu dvě různé (vnořené) části HTML kódu označené jako *bl\_head* a *bl\_search*. Tyto části potom obsahují proměnné *{HEAD}* a *{SEARCH}*.

A nyní jak se s takto připravenou šablonou pracuje v PHP:

V prvním kroku inicializujeme k použití a zobrazení hlavní blok *bl\_head*, pokud chceme (nemusíme), inicializujeme také blok pro vyhledávání *bl\_search* a jednotlivé proměnné označené složenými závorkami naplníme potřebnými textovými řetězci, které mohou obsahovat pouhé popisky či složitější HTML struktury.

Podobným způsobem můžeme strukturovat výstupní HTML kód do téměř neomezené hloubky, což nám později umožní pouhým přepsáním jediné šablony zcela změnit výsledný HTML kód, tedy výsledný vzhled webu. Jako příklad je vhodné

například menu, které v jedné šabloně může být na levé straně stránek realizované seznamem, zatímco v jiné šabloně se může jednat o jednořádkovou tabulku umístěnou v horní části stránek.

### 5.3 Jednotlivé obsahové bloky

Popišme si nyní blokovou strukturu obsahu generovaného redakčním systémem. Systém je velice variabilní a mnoho bloků s obsahem lze zcela zakázat, či libovolně modifikovat. U jednotlivých bloků tedy také zmíním jeho subbloky s obsahem, které se zde ve standardní šabloně vyskytují. Tato struktura jak ji budu popisovat je obsažena v šabloně, proto ji lze v jiných šablonách libovolně modifikovat.

Jednotlivé bloky s obsahem tedy jsou:

- Hlavička spolupracujeme - jedná se o informační řádek který se nachází ještě nad hlavičkou stránky (jeho použití je volitelné, primárním účelem je zobrazení reklamy)
- Hlavička - obsahuje logo stránek, případně slogan stránek. Volitelnými bloky jsou vyhledávání a výběr jazyka
- Hlavní informační panel - zde je vypsáno volitelně hlavní menu stránek, status uživatele, online uživatelé, libovolný blok vytvořený v administraci, reklama
- Drobečková navigace - aby byl uživatel stále v obraze a věděl, kde se ve struktuře stránek nachází, nad hlavním obsahem je zobrazen navigační řádek
- Hlavní obsah - v tomto základním bloku je uveden nadpis stránky, popis stránky, hlavní text stránky, volitelně potom 2 bloky reklamy a veškeré další bloky, které na stránku vypíše aktuální modul
- Informační řádek - volitelně zapínatelný panel pod hlavním textem stránky, kde se automaticky generují tipy na články, poslední komentáře, příspěvky z diskusního fóra a další (celkem asi 15 různých možností)
- Informační sloupec - je opět možné zcela vypnout. Může volitelně obsahovat kontextové tipy, reklamu, ankety
- Patička - slouží především ke grafickému ukončení stránky, vypsání autora, kontaktu, případně práv, volitelně může zobrazovat čas generování stránky a celkový počet přístupů.



Obrázek 5.2: Schéma rozdělení stránky do jednotlivých obsahových bloků

## 5.4 Databázová abstrakční vrstva

Pro vykonávání SQL příkazů jsou v tomto případě použity funkce, které nám nabízí PHP. Jedná se zejména o:

- `mysql_connect()` - inicializace připojení k databázi
- `mysql_select_db()` - výběr konkrétní databáze
- `mysql_query()` - vykonání SQL příkazu / dotazu
- `mysql_fetch_array()` - získání jednoho řádku z vrácené tabulky dat
- `mysql_num_row()` - počet vrácených řádků tabulky

Z pohledu PHP se jedná o základní funkce pro práci s databází MySQL, takže se jedná o velice jednoduchou abstrakční vrstvu, která neumí pracovat s jinými databázemi ani neumí sama provádět složitější operace s výsledky dotazů. S postupem času jsem zjistil, že by se dalo velké množství práce ušetřit použitím sofistikovanější DB abstrakční vrstvy jako je například ADO DB.

## 5.5 Moduly

Samotné jádro redakčního systému umí zobrazovat velké množství podpůrných informací, ale hlavní obsahová část stránek je přenechána modulům. V této kapitole se seznámíme s několika důležitými moduly redakčního systému a jejich funkcemi. Zdaleka se nebudeme věnovat všem modulům ani jejich možnostem, protože to je nad rozsah této práce.

### 5.5.1 Stránka

Modul stránek má na starosti jednu ze základních funkcí jakéhokoliv webu – prezentovat statické HTML stránky. Protože se jedná o sěžejní funkci redakčního systému, je i tento modul implementován jinak než ostatní. Zatímco jiné moduly je třeba inicializovat v adresovém řádku typicky *mod=nazev\_modulu*, v tomto případě stačí zadat požadovaný alias stránky: *stranka=alias\_stranky*.

Stránka uložená v databázi obsahuje několik klíčových polí:

- titulek, název stránky
- URL alias
- klíčová slova
- informační text (zde je shrnut obsah stránky)
- vlastní text

Vlastní text stránky je již v databázi uložen jako XHTML kód.

### 5.5.2 Články

Články se v mnohém podobají statickým stránkám popsaným v předešlé kapitole. Obsahují ale některá užitečná rozšíření, která usnadňují začlenění článků do kategorií, případně seriálů. Shodné prvky jsou tedy *název článku*, *URL alias*, *klíčová slova*, *informační text* a *text*. Kromě těchto položek obsahuje článek následující rozšíření:

- *kategorie* slouží k začlenění článku do konkrétní kategorie
- *série* v případě, že je článek přiřazen do série více článků, automaticky dochází ke generování odkazů na ostatní články ze stejné série
- *styl* zde je možné nadefinovat vlastní kaskádové styly pro grafické formátování článku

- anketa definice ID ankety, která se bude zobrazovat u článku
- autor rozšířené informace o autorovi článku
- status rozšířené možnosti stavu článku (publikovaný, čeká na publikování, koncept, publikace zastavena, smazaný)
- galerie každý článek může obsahovat galerii fotek, která se zobrazuje pod článkem
- komentáře prostor pro vyjádření názorů návštěvníků k danému článku

Pro názornost rozdílů mezi sérií a kategorií uvedu jednoduchý příklad:

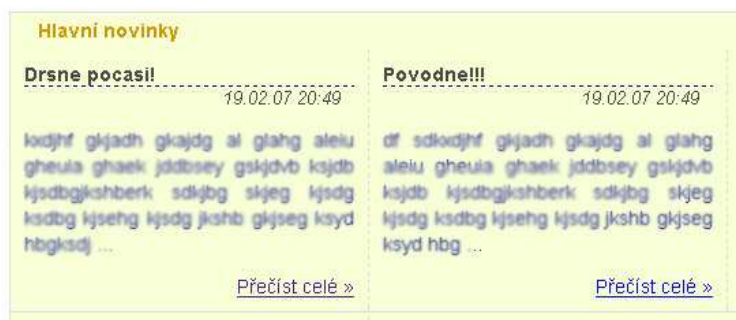
Můžeme mít kategorii pojmenovanou „Tvorba webu“. V této kategorii nyní můžeme vytvořit několik sérií článků, které se budou zabývat tvorbou webových stránek, které budou nazvané například „Základy XHTML“, „CSS snadno a rychle“, „JavaScript pro každého“.

Aby to nebylo příliš jednoduché, platí ještě některá zvláštní pravidla mezi uživateli (podrobně si to probereme v dalších kapitolách). Jen stručně již zde zmíním, že článek může editovat superadministrátor, administrátor s oprávněním pro úpravy článků nebo redaktor.

Modul článků považuji za zcela stěžejní část redakčního systému a velké množství funkcionalit je navázáno na jednotlivé kategorie, článků či články samotné.

### 5.5.3 Novinky

Modul novinek umožňuje vytvářet krátké textové informace zobrazované na úvodní stránce redakčního systému. Jeden z příkladů využití novinek je například informování návštěvníků o změnách webu, připravovaných službách, ale klidně také mohou být použity k pravidelnému zobrazování nejnovějších událostí.



Obrázek 5.3: Ukázka grafického uspořádání novinek na úvodní stránce

Každá novinka obsahuje následující informace:

- název
- textový obsah
- nepovinně fotografii
- odkaz na článek, stránku nebo zcela libovolný odkaz
- prioritu

Podle priority jsou novinky na úvodní stránce zobrazeny buď s delším textem nebo jsou uvedeny pouze jako odkaz k jejich úplnému zobrazení. Každá novinka lze samozřejmě rozkliknout na samostatnou stránku.

### 5.5.4 Diskusní fórum

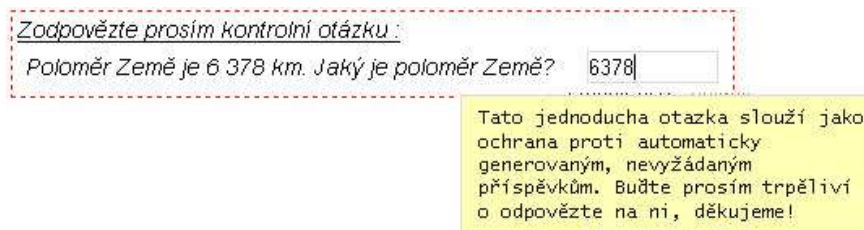
Diskusní fórum slouží k volnému debatování na určité téma. Obecně se dělí do několika kategorií, přičemž každá kategorie má vlastní fotografii, popisný text a status, který říká, kdo může či nemůže do dané kategorie vkládat příspěvky.

Status může být:

- zakázaná
- pouze pro čtení
- čtení všichni, zápis je pro registrované
- zápis i čtení pro všechny

Při vkládání příspěvku musí uživatel zadat následující údaje: email, text příspěvku, volitelně také jméno a předmět. V případě, že příspěvek vkládá přihlášený uživatel se mu automaticky jméno a email předvyplní.

Pokud příspěvek do diskusního fóra vkládá nepřihlášený uživatel, musí odpovědět na jednoduchou kontrolní otázku, která slouží jako ochrana před komentářovým spammem automaticky vkládaným na stránky tzv. roboty. Tato otázka se náhodně mění a zajistí, že příspěvek vkládá osoba, nikoliv automatický skript.



Obrázek 5.4: Příklad kontrolní otázky sloužící jako ochrana proti spamu

## 5.6 Administrační rozhraní RS

Doposud jsme se věnovali pouze generování obsahu, který se zobrazuje obyčejnému návštěvníkovi či uživateli stránek. Jednou z nejdůležitějších funkcí redakčního systému je ale umožnit jednoduchou správu obsahu webu jeho správci případně celé skupině administrátorů.

Proto se v této kapitole podíváme, jak se vytváří a spravuje obsah jednotlivých sekcí, vysvětlíme si také základní uživatelské role správců v administračním rozhraní a nakonec si ukážeme jako názorný příklad vytvoření nového článku.

The screenshot displays the 'Správa hlavního menu' (Main Menu Management) interface. At the top, there is a navigation menu with links like 'Uživatelé', 'Administrátoři', 'Práva administrátorů', 'Bloky', 'Kategorie (Menu)', and 'Nastavení systému'. Below this is a breadcrumb trail: 'Navigace : Domů > Administrace > Správa hlavního menu'. The main content area features a table of menu items:

ID	Název	Web adresa	Volání	Popis	Priorita	Status			
1	Úvodní stránka	index.php		Odkaz na úvodní stránku webu	100	1	<a href="#">Editovat</a>	<a href="#">Odstranit</a>	<a href="#">Zakázat</a>
2	Novinky	index.php?mod=news		Nejnovější události	90	1	<a href="#">Editovat</a>	<a href="#">Odstranit</a>	<a href="#">Zakázat</a>
3	Články	index.php?mod=clanky		Publikované články	80	1	<a href="#">Editovat</a>	<a href="#">Odstranit</a>	<a href="#">Zakázat</a>
4	Diskuzní fórum	index.php?mod=diskuze		Diskuzní fórum	50	1	<a href="#">Editovat</a>	<a href="#">Odstranit</a>	<a href="#">Zakázat</a>
6	Inzerce	index.php?mod=inzerce		Inzerce	45	1	<a href="#">Editovat</a>	<a href="#">Odstranit</a>	<a href="#">Zakázat</a>
5	Ankety	index.php?mod=ankety		Ankety	40	1	<a href="#">Editovat</a>	<a href="#">Odstranit</a>	<a href="#">Zakázat</a>
7	Uživatelé	index.php?mod=uzivatele		Seznam uživatelů a jiné služby	20	1	<a href="#">Editovat</a>	<a href="#">Odstranit</a>	<a href="#">Zakázat</a>
8	Uživatelská podpora	index.php?mod=support		Uživatelská podpora	10	1	<a href="#">Editovat</a>	<a href="#">Odstranit</a>	<a href="#">Zakázat</a>

On the right side, there is a sidebar with the following information:

- Přihlášen : root
- Čas přihlášení : 13:39:40 (24.04.2007)
- Počet online uživatelů serveru : 1
- Kritické logy :**
  - 2007-04-12 14:04:03 Admin - neplatný pokud o přihlášení (root : paSSword457)
  - 2007-02-27 20:48:05 Admin - nedostatečné oprávnění pro vstup do sekce (section=,logs,)
  - 2007-02-21 19:33:18 Admin - neplatný pokud o přihlášení (root : paSSword)
- [Více >>](#)

At the bottom, there is a footer with the text: 'Články a fotografie nelze bez souhlasu autora publikovat nebo zveřejňovat mimo tento server. Copyright © Filip Trnka, 2005-2006. Realizace : Filip Trnka'. It also shows 'Počet přístupů : 399' and 'Stránka vygenerována za : 2.959 s'.

Obrázek 5.5: Celkový pohled na administrační rozhraní redakčního systému

Výpis nejdůležitějších sekcí a funkcí administrace:

- správa uživatelů
- správa administrátorů a redaktorů
- práce s jednotlivými obsahovými bloky
- editace hlavního menu
- globální nastavení systému

- správa statických stránek
- správa kategorií článků, článků, galerií k článkům, sériím článků
- redaktorské rozhraní
- sledování logů redakčního systému
- správa novinek
- správa reklamy
- správa diskusního fóra a kategorií fóra
- správa antispamových kontrolních otázek

### 5.6.1 Oprávnění a práva administrátorů

Administrační rozhraní je navrženo pro práci jednoho i většího počtu správců najednou, přičemž každý správce může mít nastavena určitá omezení.

Základní rozdělení správců je

- administrátoři
- redaktoři

Začneme od toho jednoduššího, což je v našem případě redaktor.

**Redaktor** má v administraci umožněnu jednu jedinou a přesto zásadní činnost: vytváření nových článků. Jeho vstup je striktně omezen pouze do redaktorského rozhraní, což je modifikované administrační rozhraní článků, kde lze vytvářet články nové a editovat články vlastní. Samozřejmě je také umožněno redaktorovi nahrání potřebných grafických podkladů pro vytvoření článku.

Co je důležité, redaktor nemůže článek bez povolení administrátora publikovat ani odstranit.

**Administrátor** již může mít na rozdíl od redaktora přístup neomezený nebo limitovaný jen na určité sekce administrace. Každý administrátor má tedy nastavená práva, která mu garantují vstup do jednotlivých sekcí, speciálním případem oprávnění je potom *superadmin*, který garantuje administrátorovi přístup do všech sekcí bez ohledu na další nastavená oprávnění.

Tato struktura práv administrátorů dovoluje například následující omezení funkcí správců:



- 1 superadministrátor – má kontrolu nad celým systémem bez omezení
- několik administrátorů – podle oprávnění mohou hlídat podřízené redaktory a kontrolovat publikování článků, spravovat uživatele či administrátory, vytvářet stránky, spravovat reklamu a mnoho dalšího.
- větší množství redaktorů – může se jednat i o externí spolupracovníky, kteří pouze připravují obsah článků

## 5.6.2 Správa článků

Již několikrát jsem zde zmínil specifický postup při vytváření nového článku a rozdílná oprávnění redaktora. Nyní si předvedeme jako názorný příklad vytvoření nového článku redaktorem, jeho schválení administrátorem k publikaci a případně další možnosti související se správou článků. Tuto sekci administrace jsem vybral záměrně neboť ji považuji za jednu z nejdůležitějších v celém redakčním systému a zároveň za trochu specifickou.

The image shows a web form for editing an article. On the left, there are several input fields: 'Název', 'URL Alias', 'Klíčová slova', 'Kategorie', 'Série', 'Informační text', and 'Styl'. The 'Kategorie' field is a dropdown menu with the selected option 'Akvaristika obecně Pěstování akvarijních rostlin'. The 'Série' field is a dropdown menu with the selected option 'none'. Below these fields is a rich text editor with a toolbar containing icons for bold, italic, underline, text color, background color, bulleted list, numbered list, link, unlink, and a 'Format' dropdown. The main text area of the editor is empty. At the bottom right of the editor, there is a 'Path:' label and an 'Odeslat' button.

Obrázek 5.6: Formulářové prvky použité při editaci článku

Obrázek, který vidíte neodpovídá skutečnosti – hlavní část s textem je umístěna pod textovým polem styl.

Nový článek může vytvořit jak administrátor, tak redaktor. My si popíšeme tuto činnost z pohledu redaktora, což je zajímavější.

V prvním kroku redaktor vstoupí do rozhraní pro vytváření článků zobrazeného na obrázku. Zde vyplní všechny povinné položky, a v jednoduchém editoru podobného podobném rozhraní MS Word vytvoří vlastní obsah článku. Pokud chce zároveň vložit nějaké obrázky, uloží nejprve článek bez obrázků, poté přes rozhraní uploadu obrázků

nahráje obrázky na server do datového úložiště konkrétního článku. V tomto kroku dochází automaticky ke generování náhledů velkých obrázků.

Nyní administrátor poupraví článek v tom smyslu, že vloží požadované obrázky, provede konečné formátování obsahu a uloží.

Zde možnosti redaktora končí a přichází na řadu administrátor, který má povolen vstup do sekce článků, kde může provést potřebnou kontrolu článku a pokud je spokojen s výsledkem, změní status článku na publikovaný. Od tohoto okamžiku je článek zobrazován návštěvníkům stránek.

V dalším kroku je možné nahrát do galerie článku několik fotek, které se budou zobrazovat pod samotným obsahem článku.

#### Možnosti redaktora

- vytvořit a v budoucnu upravovat vlastní článek

#### Možnosti administrátora

- vytvořit a upravovat jakýkoliv článek
- měnit jeho status (především publikovat článek)
- smazat článek
- pracovat s galerií článku

### **5.6.3 Reklama**

Nyní se budeme ještě jednu kapitolu věnovat popisu konkrétního modulu, respektive funkci RS. Protože je nemožné věnovat se všem funkcím a modulům, vybral jsem jeden, který považuji za velice zajímavý svými možnostmi administrace a nastavení a tím je funkce zobrazování reklamy.

Úmyslně jsem použil slovo funkce, protože se nejedná o samostatný modul – zobrazování reklamy v redakčním systému je natolik provázané s jádrem a že jsem ho nakonec vytvořil jako jeho součást.

Pro zobrazení reklamy jsou v RS určeny 4 pozice definované ve výstupní šabloně. Standardně je to pod hlavním titulkem stránky, v hlavním informačním bloku, v pravém informačním sloupci a pod koncem hlavního textového obsahu stránky. Každé z těchto umístění lze volitelně vypnout či zapnout v administraci RS, stejně jako lze zcela zakázat zobrazování reklamy.

Formát zobrazované reklamy může být zadán dvěma způsoby

- přímo zadaný text nebo HTML kód uložený v databázi
- adresář, který obsahuje soubor index.xhtml

První možnost zadání kódu reklamy umožňuje rychle vytvářet jednoduché reklamní slogany a popisky, zatímco druhá možnost umožňuje zobrazovat libovolnou grafiku, JavaScript, Flash animace a obecně jakýkoliv obsah použitelný na webu.

Zobrazování reklamy se řídí několika pravidly

- blok ve kterém se reklama zobrazuje
- datum od kterého se má reklama zobrazovat
- limit jak dlouho se může reklama zobrazovat – ten je určen buď datem ukončení zobrazování nebo počtem zobrazení reklamy
- nastavenou prioritou reklamy – v případě, že je jednomu bloku reklamy přiřazeno více reklam na zobrazení, dochází k jejich střídání určeného právě zadanou prioritou. Například pokud budou v jednom bloku dvě zobrazované reklamy s prioritami 20 a 80, odpovídá tato priorita jejich počtu zobrazení ze 100 načtení stránek.
- čas zobrazení reklamy – každá reklama může mít určený seznam hodin, v které se bude v průběhu dne zobrazovat

## **5.7 Zhodnocení realizovaného RS**

Při návrhu redakčního systému jsem postupoval následujícím způsobem:

- souhrn funkcionalit a vlastností navrhovaného systému
- návrh databáze, propojení jednotlivých tabulek, integritní omezení některých sloupců
- návrh základní struktury skriptů a adresářové struktury
- vytvoření základních funkcí a tříd včetně zprovoznění cizích tříd
- postupný vývoj a rozšiřování funkcí jádra, současně práce na jednotlivých modulech a administraci

Vývoj celého redakčního systému trval více než 2 roky a odhadem jsem na něm mohl strávit 200 až 300 hodin. Přesto je na něm stále mnoho věcí k rozšiřování a vylepšování. Během té doby jsem se velice zlepšil v mých programátorských technikách

a naučil se velice dobře vyvíjet aplikace pro prostředí webu, což jsem samozřejmě  
zhodnotil na mnoha jiných projektech.

Rozsah aplikace je docela dobře zřejmý z následujících údajů:

- mnou napsané PHP skripty : 450 kB
- počet tabulek v databázi : cca 40

## 6 Závěr

Tato práce byla zaměřena na přiblížení trendů a možností vývoje internetových stránek se zaměřením na dostupné prostředky pro tvorbu redakčního systému. Nejprve jsem se věnoval definici technologií, nad kterými jsou redakční systémy postaveny a také jsem se zabýval jejich historickým vývojem. Dále jsem popsal důležité funkční celky redakčního systému a rozebral možné způsoby jejich realizace.

Praktickou částí realizace redakčního systému, poskytujícího všechny potřebné funkce, které by měl ucelený webový systém nabízet. Nejprve jsem rozebral požadavky několika skupin uživatelů ze kterých jsem dále vycházel. Dále jsem navrhl adresářovou strukturu pro uložení samotných skriptů potřebných k běhu redakčního systému a také úložišť různých typů dat. V dalších krocích jsou popsány důležité funkční celky a funkce, které systém nabízí a je navržen způsob rozšiřování systému pomocí jednotlivých modulů.

Celá kapitola 5 se již věnuje funkcím a strukturám hotového redakčního systému a popisuje způsob vytváření výsledného kódu, strukturu obsahových bloků a popisuje jednotlivé moduly. Zaměřil jsem se pouze na ty nejdůležitější moduly, za které považuji zobrazování stránek, články, diskusní fórum a novinky. Ačkoliv systém nabízí spoustu dalších zajímavých funkcí, nebylo by možné veškerou funkcionalitu popsat v této práci. Zvláštní část je ještě věnována administračnímu rozhraní redakčního systému s několika názornými ukázkami postupu správy obsahu.

## 7 Seznam literatury

- [1] Wikipedia: CMS. <http://cs.wikipedia.org/wiki/CMS>
- [2] Wikipedia: CLR.  
[http://cs.wikipedia.org/wiki/Common\\_Language\\_Infrastructure](http://cs.wikipedia.org/wiki/Common_Language_Infrastructure)
- [3] Wikipedia: JavaScript <http://cs.wikipedia.org/wiki/Javascript>
- [4] Wikipedia: PHP <http://cs.wikipedia.org/wiki/Php>
- [5] Wikipedia: ASP.NET <http://cs.wikipedia.org/wiki/ASP.NET>
- [6] Wikipedia: Internet <http://cs.wikipedia.org/wiki/Internet>
- [7] Wikipedia: WWW <http://cs.wikipedia.org/wiki/Www>
- [8] V. Kobzev. Diplomová práce: Vývoj výkonného jádra informačního systému
- [9] J. Castagnetto a kol.: Programujeme PHP profesionálně
- [10] B. Bernard. Bakalářská práce: Využití objektových technologií při vývoji webových aplikací: Srovnání ASP.NET a PHP
- [11] M. Macík. Bakalářská práce: Obecný internetový obchod
- [12] P. Zajíc. Linuxsoft.cz – seriál o PHP <http://www.linuxsoft.cz/php/>
- [13] M. Olšavský. Linuxsoft.cz – php knihovny: Smarty  
<http://www.linuxsoft.cz>

## **A Seznam použitých zkratek**

API - Application Programming Interface  
ASCII - American Standard Code for Information Interchange  
ASP - Active Server Pages  
CGI - Common Gateway Interface  
CLR - Common Language Runtime  
CMS - Content Management System  
CSS - Cascading Style Sheets  
DB - Database  
DBMS - Database Management System  
DLL - Dynamically Linked Library  
FTP - File Transfer Protocol  
HTML - Hypertext Markup Language  
HTTP - Hypertext Transfer Protocol  
HTTPS - Hypertext Transfer Protocol Secure  
IMAP - Internet Message Access Protocol  
LDAP - Lightweight Directory Access Protocol  
MS - Microsoft  
ODBC - Open Database Connectivity  
POP3 - Post Office Protocol  
RS - redakční systém  
SMTP - Simple Mail Transfer Protocol  
SQL - Structured Query Language  
TCP/IP - Transmission Control Protocol and Internet Protocol  
URL - Uniform Resource Locator  
WWW - World Wide Web  
WYSIWYG - What you see is what you get  
XHTML - Extensible Hypertext Markup Language

## **B Obsah přiloženého CD**

Na přiloženém CD se nachází zdrojové kódy realizovaného redakčního systému zkomprimované do jednoho ZIP archivu a text této práce ve formátu PDF:

*Bakalářská práce - redakční systém PHP,MySQL.pdf*

*Redakční systém.zip*

Struktura ZIP archivu s redakčním systémem obsahuje adresářovou strukturu popsanou v kapitole 4.2.