

# **Webový portál firmy pro sdílení dat**

Web based data sharing server

Bc. Svatopluk Hastík

---

Diplomová práce  
2009



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav aplikované informatiky

akademický rok: 2008/2009

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Svatopluk HASTÍK**

Studijní program: **N 3902 Inženýrská informatika**

Studijní obor: **Počítačové a komunikační systémy**

Téma práce: **Webový portál firmy pro sdílení dat**

Zásady pro vypracování:

1. Seznámení s technologiemi a jejich zhodnocení.
2. Význam šablonovacích systémů.
3. Analýza informačních systémů pro sdílení dat a redakčních systémů.
4. Návrh a analýza potřebné funkčnosti.
5. Proces vývoje a využívání informačního systému.
6. Zhodnocení přínosu práce a možný rozvoj do budoucna.

Rozsah práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. AJAX a PHP : tvoříme interaktivní webové aplikace profesionálně /. Brno : Zoner Press, 2006. 320 s.
2. Learning PHP and MySQL. Sebastopol, Calif. O'Reilly 2007. 411 s.
3. PHP and Smarty on large scale web development. Sebastopol, Calif. : O'Reilly, 2007
4. Programming PHP. Sebastopol, Calif. O'Reilly 2006. 521 s.
5. Databázové systémy MySQL + PHP /. Zlín : Univerzita Tomáše Bati ve Zlíně. Fakulta aplikované informatiky, 2006. 126 s.
6. Velká kniha PHP 5 & MySQL : kompendium znalostí pro začátečníky i profesionály /. Brno : Zoner Press, 2005. 711 s.
7. Web database applications with PHP and MySQL. Beijing. ; Farnham : O'Reilly, c2004. 796 s. 24 cm.
8. Podnikové informační systémy : podnik v informační společnosti /. Výrazně přeprac. a rozš. vyd.. Praha : Grada, 2008. 283 s.

Vedoucí diplomové práce:

**Ing. Petr Šilhavý**

Ústav aplikované informatiky

Datum zadání diplomové práce:

**20. února 2009**

Termín odevzdání diplomové práce:

**27. května 2009**

Ve Zlíně dne 13. února 2009



prof. Ing. Vladimír Vašek, CSc.  
*děkan*



doc. Ing. Ivan Zelinka, Ph.D.  
*ředitel ústavu*

## **ABSTRAKT**

Cílem této práce je vytvořit informační systém jehož úkolem bude efektivní možnost přístupu k FTP serveru, správa podnikových projektů, komunikace mezi zaměstnanci a obecná možnost sdílení všech dat a informací, které budou v systému zadány. Důležitá je rovněž bezpečnost při používání celého systému a především pak přístupu k webovému rozhraní FTP serveru.

Klíčová slova: informační systém, FTP server, projekty, sdílení, PHP

## **ABSTRACT**

The focus of this project is to create an information system, which will serve as an effective mean to access FTP server, as a business project management, an easy communication tool for employees and data and information sharing service throughout the whole system. Security in global, but mainly in the web based interface for accessing the FTP server, is also very important.

Keywords: information system, FTP server, projects, share, PHP

Na tomto místě bych chtěl poděkovat Ing. Petru Šilhavému za ochotu, připomínky, cenné rady a vedení při tvorbě práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval.  
V případě publikace výsledků budu uveden jako spoluautor.

Ve

.....

Zlíně 20.5.2009

Podpis diplomanta

**OBSAH**

<b>ÚVOD</b> .....	<b>10</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>11</b>
<b>1 AJAX</b> .....	<b>12</b>
1.1 HISTORIE .....	12
1.2 VLASTNOSTI.....	14
1.3 POUŽITÍ .....	14
1.4 VÝHODY A NEVÝHODY .....	15
1.5 SHRNUTÍ.....	15
<b>2 ŠABLONOVACÍ SYSTÉMY</b> .....	<b>17</b>
2.1 ŠABLONOVACÍ SYSTÉMY V PHP .....	17
2.1.1 Smarty .....	18
2.1.2 HTMLtmpl .....	20
<b>3 WEBOVÉ APLIKACE</b> .....	<b>22</b>
3.1 INFORMAČNÍ SYSTÉMY .....	22
3.1.1 Specifikace informačního systému.....	24
3.2 REDAKČNÍ SYSTÉMY.....	24
3.2.1 Funkce redakčního systému .....	25
3.2.2 Rozdělení redakčních systémů .....	25
3.2.3 Srovnání redakčních systémů .....	26
3.2.3.1 phpRS.....	26
3.2.3.2 Drupal .....	27
3.2.3.3 Joomla.....	27
3.2.4 Redakční systémy tabulkové srovnání .....	27
<b>4 PROGRAMOVACÍ PROSTŘEDKY</b> .....	<b>30</b>
4.1 MYSQL.....	30
4.1.1 Tabulky a vztahy mezi nimi .....	31
4.1.2 Normy při stavbě databáze .....	32
4.1.2.1 První normální forma (1.NF) .....	33
4.1.2.2 Druhá normální forma (2.NF).....	33
4.1.2.3 Třetí normální forma (3.NF).....	33
4.1.3 Datové typy .....	33
4.2 PHP.....	34
4.2.1 Charakteristika jazyka .....	35
4.3 SOUČASNÝ STAV .....	37
<b>II PRAKTICKÁ ČÁST</b> .....	<b>39</b>
<b>5 NÁVRH INFORMAČNÍHO SYSTÉMU</b> .....	<b>40</b>

5.1	SYSTÉM ROZDĚLENÍ UŽIVATELŮ .....	40
5.2	WEBOVÉ ROZHRANÍ FTP SERVERU.....	41
5.3	POZNÁMKY.....	41
5.4	JEDNODUCHÉ ÚKOLY V KALENDÁŘI.....	41
5.5	ZASÍLÁNÍ ZPRÁV MEZI UŽIVATELI .....	42
5.6	SPRÁVA ZPRACOVÁVANÝCH PROJEKTŮ .....	42
5.7	PRACOVNÍ PLOCHA.....	42
<b>6</b>	<b>NÁVRH DATABÁZE .....</b>	<b>44</b>
6.1	TABULKA DIRFILE_OPERACE .....	44
6.2	TABULKA HODNOSTI.....	44
6.3	TABULKA KALENDAR_UKOLY .....	45
6.4	TABULKA LOG .....	45
6.5	TABULKA POZNAMKY .....	46
6.6	TABULKA POZNAMKY_SDILENI.....	46
6.7	TABULKA PRISTUPY_ADR .....	46
6.8	TABULKA PROJEKTY .....	47
6.9	TABULKA PROJEKTY_HODINY_LIDE .....	47
6.10	TABULKA PROJEKTY_KATEGORIE.....	48
6.11	TABULKA PROJEKTY_POVOLENI .....	48
6.12	TABULKA PROJEKTY_SUBPRACE .....	48
6.13	TABULKA PROJEKTY_SUBPRACE_LIDE .....	49
6.14	TABULKA UZIVATELE .....	49
6.15	TABULKA UZIVATELE_PRAVA .....	50
6.16	TABULKA ZPRÁVY .....	50
<b>7</b>	<b>TVORBA A VLASTNOSTI INFORMAČNÍHO SYSTÉMU.....</b>	<b>52</b>
7.1	ROZHRANÍ FTP SERVERU .....	52
7.1.1	Zpracování funkcí webového rozhraní FTP serveru .....	53
7.2	ZASÍLÁNÍ ZPRÁV MEZI UŽIVATELI .....	54
7.3	KALENDÁŘ A ÚKOLY .....	55
7.4	SYSTÉM POZNÁMEK.....	56
7.5	REALIZACE SEKCE „PROJEKTY“ .....	56
7.6	REALIZACE PRACOVNÍ PLOCHY.....	59
7.7	PRÁCE V INFORMAČNÍM SYSTÉMU .....	60
7.8	POUŽÍVÁNÍ INFORMAČNÍHO SYSTÉMU.....	64
	<b>ZÁVĚR .....</b>	<b>67</b>



ZÁVĚR V ANGLIČTINĚ.....	68
SEZNAM POUŽITÉ LITERATURY.....	69
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	71
SEZNAM OBRÁZKŮ.....	73
SEZNAM TABULEK.....	74
SEZNAM PŘÍLOH.....	75

## ÚVOD

V době, kdy se žádná firma neobejde bez správy svých dat prostřednictvím počítačů a vlastních podnikových informačních systémů přichází možnost vyměňovat a sdílet důležité data prostřednictvím internetu. Správa těchto dat je zprostředkovávána formou interního softwaru, do kterého mají přístup jak vedoucí pracovníci firmy, tak i zaměstnanci uvnitř podnikové intranetové sítě.

V rámci diplomové práce „Webový portál firmy pro sdílení dat“ bude vyvíjen informační systém (IS), jehož úkolem je efektivní a jednoduchá správa projektů firmy za použití internetového prohlížeče.

Před samotnou praktickou tvorbou IS je potřeba zvolit výběr programovacích prostředků. V tomto případě byla zvolena kombinace PHP, MySQL vzhledem k možnostem hostingu na němž má IS běžet a licenčním podmínkám těchto technologií. Pro rychlejší práci je častým společníkem při procházení webu Ajax a jeho schopnost provádění některých skriptů na pozadí, aniž by bylo nutno provést znovunačtení celé stránky.

Provoz IS, který bude umístěn na internetu s sebou nese samozřejmě rizika v podobě bezpečnosti, na kterou musí být vždy kladen velký důraz. Zvláště v případě, pokud se jedná o systém jehož úkolem bude i správa adresářového prostoru FTP serveru. Uživatelé tedy budou rozděleni do několika skupin, které budou symbolizovat práva přístupu do jednotlivých sekcí IS. Dalším důležitým blokem bude správa podnikových projektů a prací všech uživatelů IS. Díky těmto položkám bude mít každý uživatel přehled nejen o všech pracích, na kterých se kdy podílel, ale i o počtu hodin, které odpracoval, popř. finančním vyrovnání.

V rámci rozmanitosti uživatelů je brána v potaz i multijazyčnost celého systému, která může být řešena formou speciálního souboru, v němž se budou nacházet všechny texty, které se v IS mají vyskytovat přímo v HTML kódu. Tento soubor lze pak ručně editovat, popř. vytvořit administrační rozhraní na jeho editaci.

Pro správnou funkčnost webového IS je třeba zajistit správný provoz v nejpoužívanějších internetových prohlížečích (Mozilla Firefox, Internet Explorer, Opera) a na případnou nekompatibilitu s IS upozornit uživatele předem.

## **I. TEORETICKÁ ČÁST**

# 1 AJAX

## 1.1 Historie

Termín Ajax se poprvé objevil v únoru 2005 v článku Jesse James Garretta (*Ajax: A New Approach to Web Applications*), avšak jeho počátky spadají do hlubší historie.

První webové prezentace byly výhradně statického charakteru. Každá HTML stránka se načítla celá a byla unikátní svou URL adresou, která ji identifikovala v rámci celého internetu. S dalším vývojem rostl i zájem o dynamické weby, tedy takové u nichž docházelo k průběžným změnám na stránce bez nutnosti obnovovat celou stránku. Tato technologie se nyní dostává stále více do popředí ať už díky velkému rozvoji těchto prezentací nebo díky usnadnění a urychlení prohlížení webových stránek.

Zásadním mezníkem byl vývoj technologie JavaScript (původní název LiveScript), který byl poprvé podporován prohlížečem Netscape v roce 1995. Tato technologie umožňuje zpracovávat klientské dotazy přímo v prohlížeči, kdy není nutno mít trvale vytvořené spojení mezi serverem a klientem. Podpora a funkčnost JavaScriptu v prohlížeči je zásadní pro správnou funkci dynamických aplikací, které jsou založeny na Ajaxu.

Dalším krokem přechodu k dynamického načítání části webové prezentace bylo zavedení rámců (neviditelných rámců). Díky tomu šlo dosáhnout načtení obsahu webové stránky do jednotlivých rámců zavoláním příslušné JavaScriptové funkce. Respektive neviditelné rámce umožňovaly nastavení minimálních rozměrů za účelem vytvoření spojení se serverem, díky kterému bylo možno odesílat a přijímat obsah rámce na server a zpět.

V roce 1996 zavedl Microsoft dynamické HTML (DHTML), jehož principem bylo za pomoci JavaScriptu měnit kteroukoliv část načtené stránky. Navzdory snaze Microsoftu se tato technologie nikdy nestala standardem, avšak její rozvoj byl dalším krokem směrem k Ajaxu.

Vyvinutím prvku `<iframe>` v roce 1997 se vývojářům dostal do ruky další nástroj jak na stránku umístit neviditelný plovoucí rámec, jenž umožňoval spojení mezi klientem a serverem.

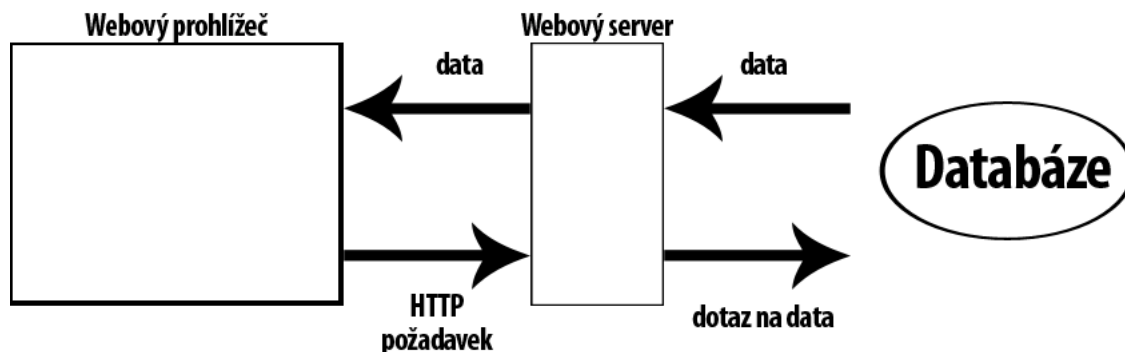
V roce 2001 vyvinul Microsoft ve formě objektu ActiveX (patentovací programovací prvek jako standard od Microsoftu) nástroj XMLHttpRequest, který dokázal vytvořený http požadavek

ovládat JavaScriptem. Data mohou být ve tvaru XML nebo HTML nebo v jiném formátu, který je požadován vývojářem.

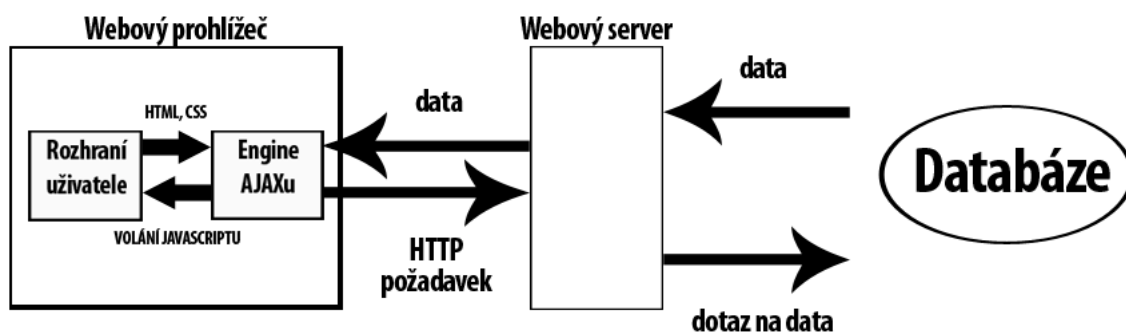
Díky velké popularitě XMLHttpRequest se rozhodla vývojářská sekce projektu Mozilla vytvořit vlastní podobný objekt, jehož nejdůležitější objektové metody a vlastnosti byly převzaty z XMLHttpRequest. Tento objekt se nazývá XMLHttpRequest. Stejně tak i další prohlížeče (Opera, Safari) byly nuceni zvolit si podporu objektu, který by dokázal zpracovat http požadavek na pozadí. Oba se rozhodly pro XMLHttpRequest od Mozilly. Paradoxem je, že tento objekt začal Microsoft používat rovněž od verze Internet Explorer 7.

Objekt XMLHttpRequest se stal srdcem celého moderního Ajaxu, který lze tedy definovat jako přístup k otázce interakce mezi klientem a serverem. Tento přístup je založen na přenosech malého množství informací k serveru a od něj, aby interakce s uživateli byla co nejrychlejší. [1]

Ajax tedy není nic jiného než nějaký přístup k otázce interakce mezi klientem a serverem. [1]



Obrázek 1: Schéma klasického dotazování prostřednictvím webového prohlížeče [1]



Obrázek 2: Schéma provedení dotazu za pomoci AJAXU [1]

## 1.2 Vlastnosti

Server je nastaven tak, aby vracel taková data, které dokáže Ajax využít. Tato data můžou být ve formátu čistého textu, HTML, XML. Ajaxové aplikace by měly mezi klientem a serverem posílat a přijímat co nejméně informací.

## 1.3 Použití

Drtivě nepoužívanější formou Ajaxu je pomocí objektu XMLHttpRequest za pomoci JavaScriptu a voláním do elementu HTML stránky. Podstatou je tedy zavolání JavaScriptové funkce, která na pozadí odešle žádost ke zpracování požadovaného PHP souboru, přičemž je možno využít vstupních parametrů pomocí `$_GET` nebo `$_POST` proměnných.

Pomocí XMLHttpRequest vytvoříme objekt Request, následně pomocí `request.open("GET", url, true)` načteme stránku, která je obsažena v proměnné url. Poté pomocí `request.readyState` zjistíme zda-li je akce hotova (roven hodnotě 4). Posléze je prováděna kontrola `request.Status`, který, pokud je roven 200 (což je hodnota, kterou vrací server), označuje správné provedení. Nakonec funkce `request.responseText` vypíše načtený obsah do požadovaného elementu.

Velmi efektivním a jednoduchým řešením je použití Ajaxu za pomoci vytvořených JavaScriptových knihoven. Mezi dvě nepoužívanější patří knihovna Prototype a knihovna jQuery. Jejich výhodou je, že nemusíme znát do detailu parametry objektu

XMLHttpRequest, ale postačí nám znát funkci příslušné knihovny, která Ajaxový skript vyvolává.

Nevýhodou tohoto způsobu řešení je nutnost načtení dalších zdrojových dat knihovny, která mohou zbytečně zatěžovat internetové připojení uživatele.

## 1.4 Výhody a nevýhody

Mezi hlavní výhody Ajaxu patří odstranění nutnosti znovunačtení celých stránek, což má příznivý vliv nejen na snížení zátěže na servery, ale i na uživatelskou pohodlnost. Typické použití naznačuje např. anketa u níž uživatel při kliknutí nemusí čekat na znovunačtení celé stránky, ale pouze části dokumentu, který obsahuje anketu.

Při nevhodném použití Ajaxových skriptů může dojít dokonce i ke zvýšení nároků na server díky velkému množství http požadavků, které se provádějí na pozadí. Implementací Ajaxu na celé webové stránky dochází také k nefunkčnosti klasických tlačítek v prohlížeči Zpět/Vpřed díky struktuře, která mění hlavní obsah stránky bez změny url adresy webu. Takovýmto použitím Ajaxu se také webová stránka stává ohrožena kvůli indexaci do vyhledávačů. Při zpracování takovýchto požadavků dochází většinou k mírnému zpoždění a při načítání velkého množství dat na pozadí stránky může vzniknout pro uživatele deziluze, která povede k odchodu z webové prezentace.

Z tohoto důvodu je doporučeno používat Ajax na části webových prezentací nikoliv na celé stránky webové aplikace.

## 1.5 Shrnutí

Mezi průkopníky celowebových ajaxových aplikací je společnost Google, jejíž e-mailová schránka Gmail je zdárným příkladem, že vše může fungovat velmi efektivně. Při prvním připojení do uživatelského rozhraní schránky je nahrán celý engine do jednoho z mála plovoucích rámců a další požadavky jsou již prováděny pomocí objektu XMLHttpRequest. Aplikace využívá několik klasických rámců (a také plovoucích rámců) pro správu a vyrovnávací paměť pro případ velkých změn v uživatelském rozhraní. Právě složitá struktura rámců umožňuje správnou funkčnost tlačítek Zpět a Vpřed, což je jedna z výhod použití rámců a plovoucích rámců ve spojení s objektem XMLHttpRequest. [1]

Ajax úzce souvisí i s nástupem tzv. Web 2.0 nebo-li termínu, který označuje novou generaci webových aplikací. Web je smýšlen jako služba a je rozvíjen za pomoci uživatelů, kteří se aktivně zapojí do vytváření obsahu (blogování). Data jsou oddělena od prezentace a ve finále tedy mohou být reprezentována různými způsoby (RSS).



## 2 ŠABLONOVACÍ SYSTÉMY

Na vývoji webové aplikace se podílejí čtyři typy pracovníků:

- Grafik
- Kodér (XHTML, HTML)
- Programátor (PHP, MySQL, ASP)
- Copywriter

Grafik se stará o kompletní návrh designu. Jeho úkolem je tedy zpracovat grafickou šablonu v grafickém programu, kde bude vzhled rozdělen do jednotlivých vrstev. Takovýto soubor je následně předáván kodérovi jehož úkolem je vytvořený design zpracovat a nakódovat tak, aby se primárně zobrazoval stejně ve všech prohlížečích a byl pokud možno validní podle vybrané normy.

Úkolem programátora je vytvořit funkční aplikaci, která bude na webové XHTML šabloně zobrazovat požadované informace (výpis textů, anketní systém, redakční systém). Nakonec přichází na řadu copywriter, který naplní, popřípadě zkontroluje web po textové stránce. Jeho úkolem je zajistit, aby byly texty stručné, čitelné, výstižné a aby dokázaly čtenáře na stránce připoutat.

Je velmi obtížné najít člověka, který by zvládal výborně dělat všechny tyto činnosti. Tedy byl jak výborný grafik, tak skvělý kodér a profesionální programátor a copywriter. Na každé rozsáhlé aplikaci proto pracuje tým lidí, který se neustále doplňuje a předává si vlastní výstupy mezi jednotlivými vrstvami spolupráce (grafik dává šablonu kodérovi).

Šablony v PHP mají za úkol rozlišit práci kodéra a programátora. Je tedy nezbytné, aby byl XHTML kód striktně oddělený od PHP kódu. Tohoto lze dosáhnout právě použitím šablon. Jednoduše řečeno kodér si do svého XHTML souboru přidá speciální značky pomocí níž pak PHP programátor nastaví šablonovací systém tak, aby danou značku vyhodnotil a vypsal na její pozici požadovanou informaci.

### 2.1 Šablonovací systémy v PHP

Existuje nespočet řešení, jakým způsobem oddělit XHTML kód od toho programovacího. Bylo vyvinuto velké množství šablonovacích systémů, ať už od těch megalomanských nebo po jednoduché projekty, které postačí pro malé webové aplikace.

Každý má také samozřejmě možnost si vytvořit vlastní systém šablon. Pro příklad uvedu nejjednodušší ukázkou použití vlastní šablony. Kdy budeme mít nejdříve XHTML soubor ve kterém bude v titulku stránky speciální značka, kterou PHP vyhodnotí a vypíše místo ní požadovanou informaci.

```
<html>
  <head>
    <title>{titulek}</title>
  </head>
  <body>
  </body>
</html>
<?php
  $file = file("sablon.html");
  str_replace("{titulek}", "Titulek stránky", $file);
?>
```

Obrázek 3: Jednoduchý příklad použití skriptu šablonovacího systému

Toto triviální řešení pouze naznačuje základní význam šablon a to oddělení PHP kódu od XHTML. Pokud potřebujeme výstupní textové informace načítat z databáze nemusíme ve výsledku vůbec sáhnout do XHTML šablony a kódování práce zůstane neporušena.

Velkou výhodou je tedy to, že nám tento efektivní způsob zpracování webové aplikace umožní v budoucnu jednoduše změnit vzhled webové stránky, aniž bychom jakkoliv museli konzultovat změny s PHP programátorem, který by byl bez šablonovacího systému nucen vložit XHTML kód do PHP skriptů.

### 2.1.1 Smarty

Každému, kdo se někdy o šablonovací systémy zajímal se jistě při prvním zmínění tohoto tématu vybaví slovo Smarty. Jde o největší šablonovací systém, který můžeme v PHP najít. Jeho silnou stránkou je velmi mnoho užitečných funkcí a efektivní systém cachování, kdy šablona není zpracována při každém obnovení stránky ale využívá již uložených souborů v cache paměti na FTP serveru, čímž výrazně urychluje načítání stránek.

Vše shrnuje následující příklad:

Grafik vytvoří HTML šablonu:

```
<html>
  <head>
    <title>{$titulek}</title>
  </head>
  <body>
  </body>
</html>
```

Obrázek 4: HTML šablona pro SMARTY

Všimněme si, že tentokrát je jako unikátní operátor v šabloně řešen podobně jako v PHP, tedy se značkou dolar na začátku, avšak uzavřen ve složených závorkách.

Tuto šablonu tedy uloží grafik do souboru body.tpl. Následuje práce PHP programátora, který vytvoří následující skript index.php.

```
<?php
require ("smarty.php");
//volani zdrojove smarty knihovny
$smarty = new Smarty(); //definice objektového prvku
//kontrola, zda existuje Cookie titulek
$titulek = array_key_exists('titulek', $_COOKIE)?
$_COOKIE['titulek']: 'Úvodní stránka';
$smarty->assign('titulek', $titulek);
$smarty->display('body.tpl');
?>
```

Obrázek 5: PHP kód volání šablony pomocí SMARTY

Spuštěním výše uvedeného skriptu dojde k načtení naší vytvořené šablony a nahrazení výrazu {\$titulek} za skutečnou hodnotu. Ve skutečnosti bude vygenerován soubor ve kterém už bude použit klasický php příkaz echo, který zajistí vypsání požadované hodnoty.

```
<html>
  <head>
    <title>
    <?php echo $this->_tpl_vars['titulek']; ?>
    </title>
  <body>
  </body>
</html>
```

Obrázek 6: Konečné zobrazení šablony po zpracování pomocí SMARTY

Ze všech příkladů je patrné, že základní princip šablonovacího systému je vždy stejný. Tudíž nahrazení unikátního výrazu reálnou hodnotou pomocí PHP skriptu. Hlavní výhodou rozsáhlejších šablonovacích systémů jsou velké možnosti při programování HTML, kdy je

kodérovi dán do ruky silný nástroj, kterým může ovlivnit výsledný vzhled stránky (použitím, funkcí, cyklů, podmínek). Právě použití cyklů je dáváno jako příklad při generování dynamického menu, kde se jinak při práci bez šablon nevyhneme začlenění HTML kódu do PHP skriptů.

U Smarty to ilustruje následující příklad:

```
<body>
  <ul>
    {section name=id loop=$odkazy}
      <li><a href="{ $url[id] }">{$odkazy[id]}</a></li>
    {/section}
  </ul>
</body>

<?php
$odkazy = array('Odkaz 1','Odkaz 2');
$url = array('http://www.seznam.cz/', 'http://www.centrum.cz/');
$smarty->assign('odkazy', $odkazy);
$smarty->assign('url', $url);
?>
```

Obrázek 7: Použití cyklů u SMARTY

V HTML šabloně nadefinujeme pomocí smarty požadovaný cyklus, který nám zajistí zpracování polí \$odkazy a \$url a vypíše `<ul><li></li></ul>` menu strukturu. Opět tedy vidíme zjevnou výhodu v žádném použití HTML značek v PHP.

### 2.1.2 HTMLtmpl

Použití Smarty se v mnoha projektech volí jako optimální, protože umožňuje prakticky všechno co si může člověk od šablonovacího systému představit. Avšak velké pravomoce pro HTML kodéra nemusí být žádoucí, proto je mnohdy pro menší projekty vhodnou volbou jednoduchý šablonovací systém.

HTMLtmpl je vyvíjen od roku 2001 a jeho možnosti plně postačují pro nenáročný projekt, přičemž jsou splněny všechny základní pravidla specifikující používání šablon.

Hlavní výhodou oproti SMARTY je 40x menší velikost požadované třídy, která zpracovává HTML šablonu, což může být pro datově objemné projekty výrazný faktor. Níže uvedené příklady znázorňují jednoduché použití HTMLtmpl při přepisu proměnné v šabloně, popř. výpisu menu ve smyčce.

```
<html>
  <head>
    <title><TMPL_VAR NAME=TITULEK></title>
  </head>
  <body>
  </body>
</html>

$template->param(
  HOME => 'Homepage'
);

print $template->output;
```

Obrázek 8: Příklad přepsání proměnné u HTMLtmpl

### Šablona

```
<ul>
  <TMPL_LOOP NAME=MENU>
  <li>
    <a href="<TMPL_VAR NAME=URL>"><TMPL_VAR NAME=ODKAZ></a>
  </li>
</TMPL_LOOP>
</ul>
```

### PHP skript

```
$template->param(MENU =>
[
  { url => 'http://www.seznam.cz', odkaz => 'Seznam.cz' },
  { url => 'http://www.centrum.cz', odkaz => 'Centrum.cz' },
]);

print $template->output();
```

### Výstup

```
<ul>
  <li><a href="http://www.seznam.cz">Seznam.cz</a></li>
  <li><a href="http://www.centrum.cz">Centrum.cz</a></li>
</ul>
```

Obrázek 9: Použití cyklu v šablonovacím systému HTMLtmpl

## 3 WEBOVÉ APLIKACE

### 3.1 Informační systémy

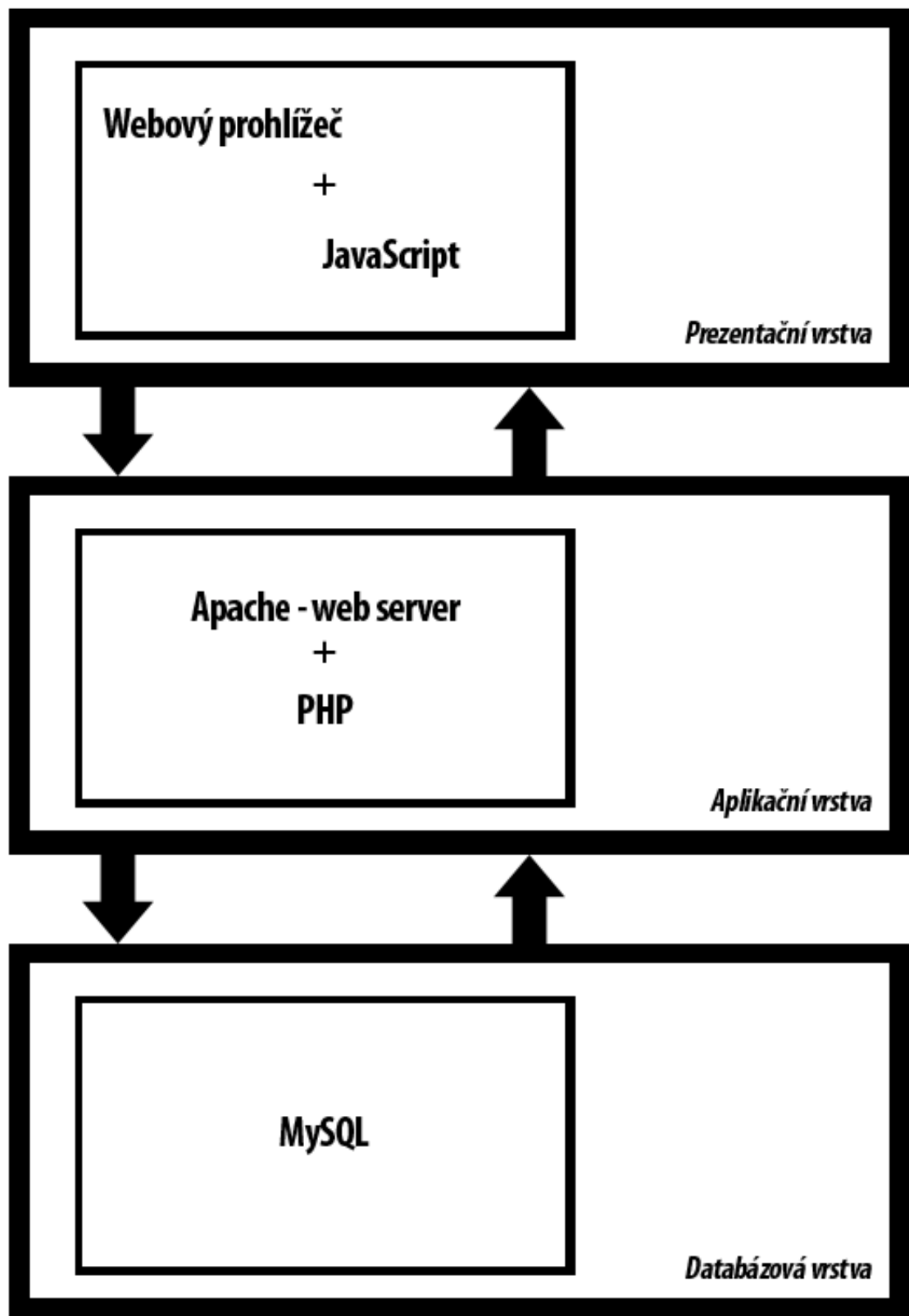
Informačním systémem (IS) máme na mysli systém, který umožňuje vnitřní propojení uživatelů do sítě, v níž mají všichni možnost sdílet data, zpracovávat a poskytovat informace nebo spolu vzájemně komunikovat. Architektura IS se obvykle skládá ze tří částí:

- Prezentační (zaručuje propojení s uživatelem – uživatelské prostředí)
- Funkční - aplikační (jednotlivé aplikace informačního systému, bezpečnost)
- Datová - databázová (konkrétní data se kterými se pracuje)

Volba vhodného IS je pro mnoho firem důležitým krokem ke splnění efektivních podmínek pro práci svých zaměstnanců. V minulosti bylo obvyklé, že se o vývoj takového vlastního systému starala každá firma jednotlivě a co podnik, to jiný systém. Dnes už se toto odvětví stalo natolik důležitým, že existuje mnoho specializovaných firem, které se dokážou plně postarat nejen o správnou instalaci a funkčnost, ale i o následnou správu a odladění chyb, které vznikly při používání. Nasnadě je také potřeba zmínit fakt, že takovéto řešení obnáší většinou velkou finanční náročnost na začátku, avšak do budoucna přináší mnoho výhod oproti řešení, kdy by se o vývoj informačního systému staral jednotlivec nebo skupina zaměstnanců firmy, jejichž případný odchod a následné nečekané problémy při používání systému by vedly ke komplikacím ve snaze je vyřešit.

Webový informační systém lze tedy pojmenovat jako informační systém, který se stará o správu informací a dat mezi uživateli prostřednictvím internetu. Jeho základní architektura je tedy podobná klasickému IS, avšak veškerá obsluha je prováděna skrz internetový prohlížeč odkudkoliv, kde má uživatel přístup na internet.

Jádrem systému je používaná technologie pro skriptování (PHP), technologie umožňující ukládání dat do strukturovaných tabulek (MySQL), technologie zobrazující data uživateli (HTML). IS na webu lze tedy považovat za webovou aplikaci, jež potřebuje ke své správné funkčnosti výše zmíněné technologie.



Obrázek 10: Schématické znázornění vrstev informačního systému založeném na klientské části formou webového prohlížeče

### 3.1.1 Specifikace informačního systému

Firemní informační systém (FIS) je interně obsluhován specializovaným softwarem, který je nainstalovaný na každém počítači v síti. Takovýto systém se tedy nejen stará o správu dat a informací, které uživatelé sdílí mezi sebou v dané síti intranetu, ale dokáže tyto data sdílet formou webového prohlížeče dál do internetu.

Proces návrhu FIS

- *První fáze* je identifikace obecných požadavků na informační systém. [5]
- *Druhá fáze* spočívá ve volbě vhodných technologií, návrhu jejich využití a implementace obecného prostorového i atributového rozhraní. Důraz je přitom kladen na robustnost, bezpečnost, nízkou nákladovost a v neposlední řadě na uživatelskou jednoduchost zvoleného řešení. [5]
- *Třetí fáze* je proces tvorby informačního systému ve vybraném programovacím prostředí, jeho testování a první nasazování do ostrého provozu
- *Čtvrtá fáze* je zaškolení vedoucích pracovníků pro práci a administraci informačního systému, prezentace funkčnosti FIS zaměstnancům

Způsoby připojení do FIS.

- **Silný klient** – připojení do systému skrz softwarové rozhraní. Klient se dokáže připojit přímo k databázové vrstvě a používá se protokol TCP/IP
- **Tenký klient** – připojení do systému přes rozhraní webového prohlížeče. Takovýto klient komunikuje s aplikační vrstvou a využívá protokolu http

Při tvorbě rozhraní pro webový prohlížeč je brán v potaz fakt, že na trhu je používáno několik různých verzí prohlížečů (Internet Explorer, Mozilla Firefox, Opera, Safari).

## 3.2 Redakční systémy

Doby, kdy byl internet složen výhradně ze statických webových stránek jsou už dávno pryč. Vše fungovalo na principu, kdy uživatel vytvořil HTML stránku, do ní přidal speciální značky tohoto jazyka, které vymezovaly její vzhled a samotný text, který reprezentoval obsah stránky. Takovýto soubor byl nahrán na FTP server a webová stránka byla na světě.



Potřeba pravidelně aktualizovat data na webových stránkách vedla k nutnosti vytvořit nástroj, který bude umožňovat měnit tyto informace přímo v rozhraní webových stránek. A tak začaly vznikat jednoduché redakční systémy jejichž prvním úkolem bylo prosté nahrazení textu na stránce, přičemž zdrojová data se ukládala do \*.txt souborů. Budoucnost pokročila a dnes vše funguje na principech ukládání informací do rozsáhlých databází (MySQL, PostgreSQL, Microsoft SQL server,..) při zpracování pomocí základních skriptovacích jazyků (PHP, ASP).

Charakteristika dnes hojně používaného termínu Web 2.0 sama říká, že obsah webu je tvořen dynamicky blogováním, psaním příspěvků a všechny tyto operace jsou uskutečňovány určitým redakčním systémem. V angličtině se používá termín CMS (Content management system – systém pro správu obsahu).

### **3.2.1 Funkce redakčního systému**

Základní operací, kterou každý redakční systém obsahuje je přidávání příspěvků. Ať už tedy píšete blog nebo publikujete nové články na web, vlastně tím přidáváte příspěvek. Variant, jak dané řešení probíhá, je nespočet. Obvyklým řešením je zadání nadpisu stránky, textu který se zobrazí ve zkrácené formě a textu, který se zobrazí po rozkliknutí článku. Užitečným řešením je použití speciálních JavaScriptových editorů (TinyMCE, FCK editor), které umožňují jednoduchou editaci vloženého textu a její úpravu do požadovaného formátu a v zásadě tak poskytují funkci textového editoru v prostředí internetových stránek.

Plnohodnotý redakční systém, ale umožňuje kompletní správu webu od vytváření anket, fotogalerií, e-shopu, novinek, správu fór po kompletní úpravu zdrojového HTML kódu přímo v rozhraní redakčního systému.

U menších firemních prezentací se stává velmi populární možnost použít redakční systém pouze na část webu, která bývá pravidelně aktualizována (novinky, články), zatímco na zbytek postačí statické stránky, u kterých se nepředpokládá, že by se měnily.

### **3.2.2 Rozdělení redakčních systémů**

Při výběru CMS pro svou webovou prezentaci je potřeba se rozhodnout pro jaký typ prezentace redakční systém potřebujeme. Nemá smysl používat rozsáhlá řešení na malé

podnikové stránky, naopak při tvorbě rozsáhlého portálu je vhodné použít také rozsáhlý redakční systém.

Na internetu existuje obrovské množství tzv. open source řešení redakčních systémů. Jejich přehled lze nalézt na stránkách [opensourcecms.com](http://opensourcecms.com). Výhodou použití takovýchto řešení je hlavně cena, jelikož jsou tyto projekty šířeny zdarma. K těm největším (Drupal, Joomla) lze nalézt i českou lokalizaci. Systémy jsou rozlišeny funkčností, možnostmi nastavení, způsobem instalace a požadovanými dostupnými prostředky, na kterých jsou schopny běžet. Ve většině případů stačí nahrát zdrojové soubory na FTP server, postupovat v instalaci podle pokynů, zadat správné údaje pro přihlášení do databáze a redakční systém je připraven k obsluze webu.

### 3.2.3 Srovnání redakčních systémů

V následujícím přehledu zmíním klíčové vlastnosti redakčních systémů, které obsahují českou lokalizaci a které jsou volně šířitelné pod licencí GNU. V tomto odvětví existuje velké množství produktů a jejich výběr by se měl volit podle velikosti předpokládané webové prezentace. V každém případě může u jakéhokoliv opensource řešení vzniknout otázka bezpečnosti vzhledem ke zveřejněným zdrojovým kódům. Zakoupením redakčního systému u specializované firmy však riziko prolomitelnosti plně nevyrušíme. Existují i firmy, které dokonce volně šířitelné systémy upravují pro své potřeby, načež je pak vydávají za své. Tím nejen že porušují původní licenční podmínky, ale nepotírají tím plně původní bezpečnostní riziko, které může vzniknout při použití opensource řešení.

#### 3.2.3.1 *phpRS*

PhpRS je prvotní český redakční systém vyvíjený od roku 2001 a postavený na jazyku PHP a databázi MySQL. Mezi jeho přednosti patří kompletní česká lokalizace a to nejen z hlediska uživatelského prostředí nebo dokumentace, ale i v komentářích zdrojových kódů. Administrační systém umožňuje snadnou správu uživatelů a jejich členění do příslušných sekcí (administrátor, publicista). V základní verzi lze vkládat články, vytvářet více úrovněvé menu rubrik. Systém podporuje instalaci modulů, ať už jde o správu anket, krátkých novinek, vytvoření galerie nebo diskusní fórum.

### 3.2.3.2 Drupal

První zástupcem rozsáhlého open source redakčního systému, vyvíjeného od roku 2001 je Drupal. Jeho hlavním rysem je velké množství modulů, které jej činí velmi mocným nástrojem při správě webového obsahu. Jeho jádro potřebuje pro svůj chod PHP a MySQL (PostgreSQL). Mezi další výhody patří úprava HTML šablon přímo v prostředí administrace. S pomocí Drupalu je tedy možné vytvořit plnohodnotný portál pro přidávání článků do vlastních kategorií, diskusní fórum, galerii nebo vlastní blog.

### 3.2.3.3 Joomla

Pokud bychom hledali komplexní a rozsáhlý redakční systém pro náročnou webovou prezentaci bude Joomla tím pravým řešením. Její vývoj započal v roce 2005 a od té doby se komunita jejích uživatelů početně rozrostla. V administraci lze upravit prakticky vše na co si vzpomenete a na internetu existují stovky dostupných modulů, které tento redakční systém dělají ještě víc použitelnější.

Do Joomla se kromě klasického diskusního fóra, blogu, galerie dá doinstalovat i velmi kvalitní modul pro e-shop (Virtuemart), který umožňuje velmi mnoho konfiguračního nastavení se kterými se se nesetkáme ani u samostatných opensource e-shopů.

Vzhledem k rozsáhlosti je menší nevýhodou Joomla její náročnost na použitý hosting a díky špatné konfiguraci webového serveru se nám ji nemusí podařit ani nainstalovat.

## 3.2.4 Redakční systémy tabulkové srovnání

V následující tabulce je zobrazeno tabulkové srovnání třech výše zmíněných redakčních systémů.

tabulka 1: Srovnání funkcí redakčních systémů Drupal, Joomla, Wordpress [6]

Redakční systém	Drupal 6.10	Joomla 1.5.10	Wordpress 2.7.1
Aplikační server	PHP 4.3.5 +	4.3.10 +	4.3 +
Databáze	MySQL 4.1 + PostgreSQL 7.4 +	MySQL 3.23 +	MySQL 4.0 +
Licence	GNU GPL	GNU/GPL v2	GNU GPL
Operační systém	Všechny	Všechny	Všechny
Webový server	Apache, IIS	Apache	Apache, IIS
Cena	zdarma	zdarma	zdarma

Redakční systém	Drupal 6.10	Joomla 1.5.10	Wordpress 2.7.1
<b>Bezpečnost</b>			
Prověřování záznamu	Ano	Ne	Omezený
Captcha	Modul	Modul	Modul
Schválení obsahu	Ano	Ano	Ano
Verifikace e-mailu	Ano	Ano	Modul
Historie přihlášení	Ano	Ano	Modul
SSL kompatibilní	Ano	Ano	Ano
SSL přihlášení	Ne	Ano	Modul
<b>Podpora</b>			
Komerční manuály	Ano	Ano	Ne
Komerční podpora	Ano	Ano	Ne
Komunita vývojářů	Ano	Ano	Ano
Online pomoc	Ano	Ano	Ano
Testovací rozhraní	Modul	Ano	Ano
<b>Jednoduchost použití</b>			
Drag n Drop obsah	Modul	Ne	Ano
Přátelské URL	Ano	Ano	Ano
Zmenšování obrázků	Modul	Ano	Limitované
Makro jazyk	Modul	Ano	Modul
Hromadný upload	Modul	Ano	Modul
WYSIWYG Editor	Modul	Ano	Ano
<b>Výkon</b>			
Cachování stránek	Ano	Ano	Modul
Export do HTML	Ne	Ne	Limitované
<b>Správa</b>			
Správa reklam	Modul	Ano	Ne
Administrace v prohlížeči	Ano	Ano	Ano
Substránky	Ano	Ano	Ne
Skiny, vzhledy	Ano	Ano	Ano
Vícejazyčnost	Ano	Modul	Modul
Obsah mezi servery	Modul	Ne	Ne
<b>Součinnost</b>			
RSS	Ano	Ano	Ano
FTP	Limitované	Ano	Limitované
UTF	Ano	Ano	Ano
XHTML	Ano	Ne	Ano
<b>Zabudované aplikace</b>			
Blog	Ano	Ano	Ano
Chat	Modul	Modul	Modul
Diskuze, fóra	Ano	Modul	Modul
Kalendář událostí	Modul	Modul	Modul
Správa souborů	Modul	Modul	Modul

<b>Redakční systém</b>	<b>Drupal 6.10</b>	<b>Joomla 1.5.10</b>	<b>Wordpress 2.7.1</b>
Grafy	Ne	Modul	Ne
Správa odkazů	Modul	Ano	Ano
Novinky e-mailem	Modul	Modul	Modul
Fotogalerie	Modul	Modul	Modul
Ankety	Ano	Ano	Modul
Vyhledávání	Ano	Ano	Ano
Mapa stránek	Modul	Modul	Modul
Čtečka RSS	Ano	Ano	Ano
Wiki	Modul	Modul	Modul

## 4 PROGRAMOVACÍ PROSTŘEDKY

### 4.1 MySQL

Projekt je postavený na základech databázového prostředí MySQL, které je právem označováno jako nejpoužívanější open source řešení při tvorbě webových aplikací.

První zmínka o MySQL spadá do jara roku 1995, kdy jej přivedla na svět švédská firma MySQL AB a od této doby bylo vydáno několik dalších verzí, které jsou aktualizovány dodnes. Poslední stabilní verze je 5.1.32. MySQL spadá do kategorie relačních databází. Tím je myšlen způsob zpracování dat, které jsou ukládány strukturovaně do tabulek do struktur řádků a sloupců, přičemž sloupce symbolizují jednotlivé parametry. Rozložení řádků není nijak určeno, nelze definovat, který řádek je první nebo poslední. Proto je vhodné přiřadit každé tabulce primární klíč, který jednoznačně identifikuje jednotlivé řádky. Díky relacím je tedy možné jednotlivé tabulky propojovat a ty pak mohou sdílet stejné data. Kromě primárního klíče existuje tudíž ještě cizí klíč, který zobrazuje informaci o sdílených datech z jiné tabulky.

Jak je již z názvu patrné, MySQL využívá pro komunikaci jazyk SQL, standardizovaný programovací jazyk, který vznikl v 70 letech 20. století. Jeho velkou předností je syntaxe podobná přirozenému jazyku (angličtině). Většina produktů založených na SQL dnes využívá standard SQL-99.

MySQL nabízí uživateli při vytváření tabulek několik jejich typů:

- MyISAM
- InnoDB
- MEMORY
- BLACKHOLE
- ARCHIVE
- FEDERATED
- MRG\_MYISAM

Nejpoužívanějším typem je InnoDB a MyISAM, které od sebe liší podporou, resp. nepodporou transakcí (skupina příkazů, které mají tu vlastnost, že se provedou v jeden okamžik všechny najednou).

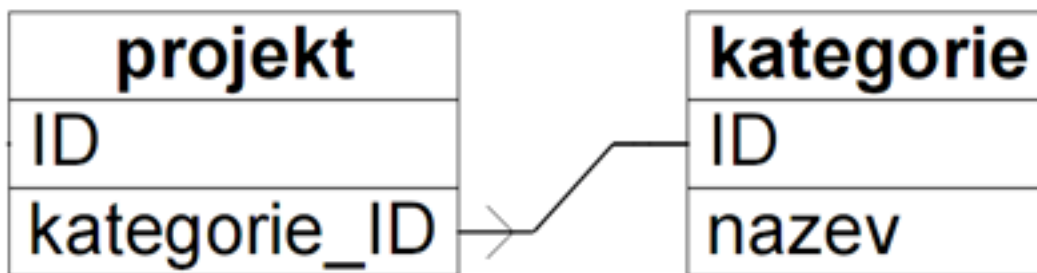
#### 4.1.1 Tabulky a vztahy mezi nimi

Při rozsáhlém projektu o databázi, která obsahuje velké množství tabulek, do nichž se určitá data ukládají opakovaně je už nutno dopředu uvažovat o co největší eliminaci redundance. Zbytečná velikost databáze pak zpomaluje vybírání cílových dat a při spojení s PHP nebo jiným programovacím webovým jazykem výrazně omezuje webové aplikace. Abychom tomuto artefaktu zabránili je potřeba data, která se opakují nechat sdílet prostřednictvím tabulkových vztahů.

Nejpoužívanější typy jsou 1:N a M:N. Spojení 1:N reprezentuje takové spojení, kdy jeden identifikátor v tabulce jedné se může vícekrát opakovat v tabulce druhé.

*Příklad:*

Máme tabulku *projekty*, do které se ukládají veškeré projekty, které vedoucí pracovník zadá svým zaměstnancům. Každý projekt může být zařazen do různých kategorií (Reklama, Dokument, Film) a hrozí tedy nebezpečí, že by se v tabulce projekty mohlo zobrazovat vícekrát jedno a to samé slovo symbolizující danou kategorii. Z tohoto důvodu tedy vytvoříme další tabulku, kterou pojmenujeme *kategorie*. Do této tabulky uložíme všechny známé kategorie, které budou zároveň reprezentovány vlastním primárním klíčem. Do tabulky *projekty* pak uložíme pouze tento klíč, který bude odkazovat na tabulku *kategorie*. V tomto případě se stane v tabulce *projekty* klíčem cizím. Tímto způsobem jsme si nejen že zabezpečili nižší redundanci při vkládání nových dat do tabulky *projekty*, ale také jednodušší aktualizaci v případě přejmenování jedné kategorie. Změna v tabulce *kategorie* totiž ve finále postačí pro změnu parametrů kategorií i v tabulce *projekty*. Kdybychom ukládali názvy kategorií přímo do tabulky *projekty*, nejen že by se nám zkomplikovala případná aktualizace kategorií, ale také by se tím znesnadnil výpis všech dostupných kategorií.

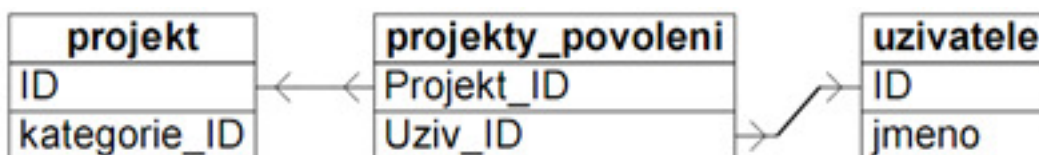


Obrázek 11: Ukázka vztahu tabulek 1:N

Dalším často používaný typem je spojení M:N. V tomto případě dochází k propojení dvou tabulek pomocí třetí pomocné tabulky, která však nemusí obsahovat žádný primární klíč a slouží pouze jako „sběr dat“.

*Příklad:*

Máme tabulku *projekty* a tabulku *uzivatele*. Obě tabulky mají svůj primární klíč, jimiž identifikují jednotlivé položky, které obsahují. Nyní budeme chtít definovat, do kterých projektů bude moci, který uživatel vkládat jednotlivé pracovní úkony. K tomuto účelu nám poslouží třetí tabulka *projekty\_povoleni* do níž si uložíme potřebné primární klíče. Tedy primární klíč projektu a primární klíč uživatele. Tato tabulka nemusí obsahovat vlastní primární klíč. Nyní při přístupu uživatele do příslušné sekce daného projektu dojde k porovnání jeho identifikátoru s identifikátorem projektu a nalezení dané shody v tabulce *projekty\_povoleni* povede k umožnění přístupu k vložení práce.



Obrázek 12: Ukázka vztahu M:N

#### 4.1.2 Normy při stavbě databáze

Každá databáze by měla mít takové vlastnosti, aby byla jednoduše aktualizovatelná, nebyla redundantní, byla bezpečná, přehledná a rychlá. Ne vždy je dosaženo všech těchto parametrů. Kompromisem je dodržování normálních forem, které definují pravidla, jakým se má vytvářet efektivní databáze.



#### 4.1.2.1 První normální forma (1.NF)

Tato norma zavádí pravidlo ve kterém každý sloupec (atribut) musí obsahovat pouze jednu hodnotu. Např. tedy v případě ukládání jména by mělo být jméno a příjmení uloženo ve zvlášť sloupcích příslušné tabulky.

#### 4.1.2.2 Druhá normální forma (2.NF)

Druhá normální forma jasně definuje používání vztahů 1:N a eliminuje redundanci databáze. Tedy v případě zakládání nového projektu se nebude do tabulky *projekty* ukládat jméno, příjmení a další údaje o uživateli, který projekt založil, ale uloží se do ní pouze identifikátor (v tomto případě cizí klíč), který bude odkazovat do tabulky *uzivatele* na konkrétního uživatele, který bude mít své unikátní parametry. Tímto předejdeme problémům při případné aktualizaci nebo mazání, kdy bychom po smazání projektu přišli i o informace o uživateli, který projekt založil.

#### 4.1.2.3 Třetí normální forma (3.NF)

Pokud se relace nachází v 1.NF a 2.NF může se přistoupit k třetí normální formě, která stanovuje takzvanou tranzitivní závislost.

Tím je myšlen vztah dvou atributů (sloupců) jedné položky, přičemž jeden je závislý na samotném klíči a druhý je závislý na prvním atributu. Pokud takovéto tranzitivní závislosti oddělíme do samostatných tabulek, dosáhneme splnění 3.NF.

### 4.1.3 Datové typy

Jednotlivé sloupce (atributy), kterými je každá položka v tabulce reprezentována mají svůj charakteristický datový typ podle toho, jaký druh dat se má do databáze uložit. Datové typy rozdělujeme do skupin na celočíselné, čísla s pohyblivou desetinnou čárkou, textové, zobrazující datum a čas a modifikátory reprezentující unikátní vlastnost položky v tabulce. Přehled těch nejpoužívanějších zobrazuje následující tabulka.

tabulka 2: Výpis vybraných datových typů databáze MySQL [7]

Datový typ	Popis
<i>Celočíselné</i>	
TINYINT	Rozsah -128 až 127
INT	Rozsah hodnot od -2147483648 do +2147483647
<i>S pohyblivou desetinnou čárkou</i>	
FLOAT	Rozsah hodnot od -3.402823466E+38 do 3.402823466E+38
DOUBLE	Rozsah hodnot od -1.7976931348623157E+308 do 1.7976931348623157E+308
<i>Datum a čas</i>	
DATETIME	Datum a čas ve formátu RRRR-MM-DD HH:MM:SS
TIMESTAMP	Detailnější nastavení díky parametru „m“. Pokud nic do sloupce neuložíme, MySQL doplní do sloupce aktuální čas a datum při změně
<i>Řetězce</i>	
VARCHAR(x)	Délka řetězce „m“ o rozsahu 0 – 255 znaků
TEXT	Délka řetězce maximálně 65535 znaků
ENUM	Pole předem definovaných prvků, maximální délka 65535 znaků
<i>Modifikátory</i>	
AUTO_INCREMENT	Automatické generování unikátní číselné hodnoty v příslušném sloupci označeném jako PRIMARY_KEY
PRIMARY_KEY	Unikátní primární klíč, který reprezentuje každý prvek v tabulce
UNSIGNED	Datový typ bude bez znaménka a bude tím posunut rozsah hodnot
FULLTEXT INDEX	Umožňuje rychlejší hledání v textových polích pomocí příkazů MATCH a AGAINST
NOT NULL	Každá taková buňka musí obsahovat nějakou hodnotu

## 4.2 PHP

Programovacím jazykem zvoleným při tvorbě informačního systému je PHP. PHP je na rozdíl třeba od HTML, které zpracovává webový prohlížeč, prováděno na straně serveru. Vývoj PHP tedy vedl k rozvoji dynamických webových aplikací, jelikož PHP obstarává další obsluhu a komunikaci s databázovým serverem.

PHP bylo původně označení pro Personal Home Page, tedy osobní domácí stránky. Vše začalo v roce 1994, kdy byla napsána binární část Common Gateway Interface (CGI) v programovacím jazyku C. Tuto prvotní část napsal Dánský/Grónský Programátor Rasmus Lerdorf. Lerdorf zpočátku vytvořil tyto nástroje pro osobní domácí stránky (Personal Home Page) za účelem možné záměny s malou skupinou skriptů v Perlu, které chtěl používat pro údržbu osobní domovské stránky. Nástroje měly zajistit běh úloh jako například zobrazení resumé (obsahu) a zaznamenávání návštěvnosti jeho stránek. Tento binární kód ještě tentýž

rok skloubil s jiným programem, který sám napsal. Po spojení s Form Interpreter tak vznikla kombinace PHP/FI, která měla mnohem větší funkčnost. PHP/FI obsahovala širokou implementaci pro programovací jazyk C a navíc tato verze mohla komunikovat s databázemi, což umožnilo tvorbu prvních jednoduchých dynamických webových aplikací. Lerdorf veřejně vydal PHP 8. června 1995, aby mohl najít co nejvíce chyb a tak zdokonalil kód. Tato verze byla pojmenována jako PHP verze 2 a již měla základní funkčnost jako má dnešní PHP. To zahrnuje například proměnné ve stylu Perlu, zpracování formulářů a možnost vložit HTML kód. Syntaxe byla obdobná jako u Perlu, ale byla omezenější, jednodušší a méně konzistentní. [2]

Od té doby bylo vydáno několik dalších verzí PHP až po současnou aktuální verzi 5.2.9.

#### 4.2.1 Charakteristika jazyka

PHP prošlo za svou existenci výrazným vývojem. Jeho syntaxe je velmi podobná klasickým jazykům založeným na typologii C jazyka.

- Jazyk PHP je dynamicky typový, tzn. že datový typ proměnné se určí v okamžiku přiřazení hodnoty. [2]
- Kvůli tomu má PHP dva operátory porovnání - '==' a '==='; při použití prvního dochází před porovnáním ke konverzi, při použití druhého je výraz pravdivý jen když jsou oba dva operandy stejného typu. [2]
- Pole se dají indexovat číselnými indexy (jako v jazyce C), nebo mohou fungovat jako hash-mapa. Stejně pole může obsahovat oba typy indexů. [2]
- Pole jsou heterogenní (stejně pole může obsahovat prvky různých typů) a řídká. [2]
- Řetězce lze uzavírat do uvozovek (při vyhodnocení se provede nahrazení proměnných uvnitř) nebo do apostrofů (nahrazuje se jen escape sekvence \). [2]

Další nesmírnou výhodou PHP je bohatá nabídka vestavěných funkcí, které výrazně zjednodušují práci. Pokud vám nějaká důležitá funkce chybí, není nic jednoduššího než si ji napsat kdekoliv uprostřed kódu. PHP je tímto způsobem zakládání nových funkcí a proměnných velmi benevolentní.

### **Klasické funkce**

V této kategorii existuje velké množství matematických funkcí, dále těch co dokáží pracovat s řetězci, zakládat a zpracovávat dynamické pole. V knihovnách se nacházejí i funkce pro práci s datem a časem.

### **Funkce pro přístup k souborům**

Pokud potřebujete načíst soubor z vlastního FTP nebo využít načítání ze vzdáleného serveru existuje zde opět několik možností jak toho docílit. Ať už použitím klasického *fopen* nebo novější *file\_get\_contents*. Díky přítomnosti funkce *ftp\_connect* se dá připojit na libovolný FTP server a prohlížet a spravovat jeho soubory. V PHP tudíž lze naprogramovat webový FTP prohlížeč s plnohodnotnými možnostmi jako je vytváření, mazání, editace adresářů a souborů.

### **Knihovna GD**

Pokud je nainstalována v PHP tato knihovna, naskýtají se vám nepřeborné možnosti pro práci s obrázky. Za pomoci této knihovny se dají vykreslovat čáry, obdelníky a další útvary vhodné pro reprezentaci grafů. S knihovnou GD se dnes také setkáte při opisování bezpečnostních kódů v rámci ochrany proti spamu. Grafická knihovna je pouze samotné jádro obsahující důležité funkce pro práci s obrázky. Existuje nespočet dalších open source řešení, které už jsou přímo uzpůsobené pro tvorbu grafů (jgraph) a nebo právě pro ochranu proti spamu (captcha).

### **Správa e-mailu**

PHP se dokáže napojit na e-mailové servery, dokáže tudíž i odesílat e-maily pomocí funkce *mail*. Rozhraní je přitom univerzální, existují funkce jak pro POP3, tak pro IMAP. V českých podmínkách mohou nastat problémy při kódování češtiny, ať už v těle zprávy nebo v samotném předmětu. Proto je dobré se porozhlédnout po dalších open source řešeních, které mají tento artefakt už vyřešen. Dobrou volbou může být PHPmailer, který nabízí hodně konfiguračních možností spolu s funkcí odesílání příloh.

### Připojení k databázi

PHP obsahuje funkce, které se dokáží připojit k většině známých databázových systémů (MySQL, Oracle, PostgreSQL, MSSQL,...). V tomto projektu byla využita databáze MySQL. Práce s databází je poměrně jednoduchá a využívá pro svou funkci několika málo funkcí. *mysql\_pconnect* pro připojení na databázi, *mysql\_select\_db* pro výběr konkrétní databáze, *mysql\_query* pro zaslání dotazu na databázi a konečně *mysql\_fetch\_array* pro zpracování takového dotazu a uložení výsledků do pole.

Abychom ochránili vkládaná data lze využít některých speciálních funkcí, např. *htmlspecialchars*, která převede veškeré „nebezpečné znaky“ (uvozovky, které mohou narušit syntaxi dotazu) na HTML entity. Další velmi často používanou funkcí je *addslashes*, která vytvoří před uvozovkami escapovací znak.

### Další funkce

PHP dokáže pracovat s regulárními výrazy a od verze 4.0.5 byla implementována možnost pracovat s PDF soubory, dále existuje podpora pro zpracování XML souborů.

## 4.3 Současný stav

Pro projekt webového informačního systému byla zvolena osvědčená kombinace programovacího jazyka PHP, databáze MySQL, to vše fungující za přítomnosti webového serveru Apache. Tato variantra programovacích prostředků běží v drtivé většině případů na linuxových serverech. Hlavní výhodou tohoto uceleného řešení je jeho cena, protože jak PHP, tak MySQL jsou vyvíjeny v rámci open source licencí.

Jako alternativa k PHP je dostupný produkt od firmy Microsoftu zvaný ASP.

ASP (Microsoft Active Server Pages) je skriptovací prostředí pro servery, které se může využít pro tvorbu a provozování dynamických, interaktivních webových aplikací. S využitím ASP, můžete kombinovat HTML stránky, skripty, a COM komponenty a vytvářet tak interaktivní webové stránky nebo výkonné webové aplikace, které se snadno vyvíjejí i upravují. [3]

Na správně nakonfigurovaném serveru (většinou je to IIS) se webová stránka s příponou **.asp** před odesláním klientovi, který si ji vyžádal, zpracuje (tj. pošle se výsledek toho, co

vygeneroval interpret ASP kódu, zavolaný s vyžádanou stránkou jako parametrem). Programovací jazyky, které se u ASP používají, jsou **VBScript** a **JScript**. [4]

Nástupcem technologie ASP je ASP.NET, jenž je součástí NET frameworku firmy Microsoft. Obě verze se od sebe liší tím, že ASP.NET je postaven na principech CLR (Common language runtime), díky tomu má programátor možnost použít při realizaci takový jazyk, který podporuje CLR (Visual Basic.NET, Jscript.NET). Skriptovací prostředí rodiny ASP je objektově založené a obsahují několik základních tříd a metod, přičemž není umožněno vytvářet nové.

## **II. PRAKTICKÁ ČÁST**

## 5 NÁVRH INFORMAČNÍHO SYSTÉMU

Tato diplomová práce se zabývá vývojem webového informačního systému (IS) pro efektivní správu podnikových projektů, obsluhu FTP rozhraní a obecným sdílením všech dat mezi uživateli IS.

Předpoklady funkční konfigurace:

- pohodlný přístup na FTP server skrz webové rozhraní prohlížeče
- vytváření poznámek a jejich správa skrz WYSIWYG editor
- vytváření jednoduchých úkolů v kalendáři
- vnitřní zasílání zpráv mezi uživateli
- správa zpracovávaných projektů

### 5.1 Systém rozdělení uživatelů

V IS se bude nacházet několik skupin uživatelů, přičemž, každý uživatel bude mít možnost být součástí libovolné skupiny. Nejvyšší hodnost bude mít administrátor, který bude mít právo zasahovat libovolně do úprav a nastavení celého systému.

Typově se hodnosti dědí. Hodnost 2 (vytváření adresářů a upload souborů na FTP) dědí vlastnosti hodnosti 3. Avšak číselná pozice neurčuje vyšší pravomoce (hodnost 2 nemá pravomoce na manipulaci s projekty). Tímto způsobem lze uživatele přiřazovat do různých skupin a existuje možnost rozšíření práv do budoucna v rámci zakládání dalších uživatelských skupin.

tabulka 3: Rozdělení práv uživatelů

Číslo hodnosti	Popis
1	Administrátoři
2	Vytváření adresářů + upload souborů
3	Upload souborů
4	Manipulace s projekty + manipulace s pracemi
5	Manipulace s pracemi
6	Obyčejný uživatel



## 5.2 Webové rozhraní FTP serveru

V tomto bodě je potřeba vytvořit webovou aplikaci, která se dokáže připojit na FTP server a zobrazí soubory v přehledné adresářové struktuře. Je zde kladen důraz na bezpečnost přístupu k jednotlivých datům. Uživatel s hodností administrátora má právo nastavit sdílení jednotlivých složek a souborů s ostatními uživateli, kteří mají možnost přihlášení do IS.

Uživatelé s vyšší hodností mají možnost vkládat do sdílených adresářů soubory a vytvářet zde další adresáře. Jejich právem je i takové soubory mazat. Uživatelé s nižší hodností mají možnost také vkládat soubory. Ovšem nemají možnost tyto soubory mazat.

Uživatel s hodností administrátora pak má mít kompletní přehled o vytvořených adresářích a uploadovaných souborech.

## 5.3 Poznámky

Každý uživatel informačního systému bude mít možnost vytvořit si krátkou poznámku, kterou spravuje skrz WYSIWYG editor. Takovými poznámkám lze přiřadit důležitost a bude je možno sdílet s ostatními uživateli systému.

Webový editor poznámek by měl mít možnost uploadu obrázků, přičemž je mohou uživatelé následně vložit do poznámky. Poznámky by zároveň měly sloužit i jako alternativa k psaní odborných textů ve webovém prohlížeči. Cílem takovéto funkčnosti je zprovoznit alternativu k programu Microsoft Word a možnost sdílení a přístupu k takovýmto dokumentům odkudkoliv na světě.

Samozřejmostí je možnost tisku takovýchto dokumentů.

## 5.4 Jednoduché úkoly v kalendáři

Náhradou rozsáhlejšího systému vytváření projektů je i vytváření jednoduchých úkolů (poznámek), které se ukládají do uživatelova kalendáře. Vytvořením takovéto poznámky si bude moci uživatel nechat připomět danou událost zasláním e-mailové zprávy. Úkoly budou moci být označovány jako „dokončené“ nebo „nedokončené“.

Uživatel s hodností administrátora má možnost takovéto jednoduché úkoly přidělovat jednotlivým uživatelům s nižší hodností.

Vše bude následně propojeno s osobním kalendářem uživatele, kde se mu případný nový úkol zobrazí jako nová událost.

### **5.5 Zasílání zpráv mezi uživateli**

Uživatelé mají možnost zasílat si mezi sebou zprávy. Jedná se o formu e-mailového kontaktu skrz IS s tím, že dochází k přehlednému ukládání historie. Zprávy starší určitého data budou automaticky přesunovány do archivu.

V nastavení bude mít každý uživatel možnost nechat si takovéto zprávy přeposílat na e-mailovou adresu.

### **5.6 Správa zpracovávaných projektů**

Důležitou částí každého IS je strukturovaná správa zpracovávaných projektů. Nejinak je tomu i v tomto případě. Systém je rozdělen na hlavní projekty k nimž se budou dát přidávat jednotlivé práce. Každá práce bude mít svého vedoucího, který je zodpovědný za její správné dokončení. Takovéto práce na jednotlivých projektech bude někdy vykonávat více lidí, proto je nutné nastavit možnost sdílení práce, přičemž každý pracovník může získat vlastní hodnocení a případný komentář ke zpracovávanému úkolu.

Po dokončení své práce si každý zaměstnanec ke svému úkolu vyplní v administraci počet odpracovaných hodin a minut a má možnost popsat průběh práce.

Na konci bude mít možnost vygenerovat si počet odpracovaných hodin za jednotlivé časové období.

Uživatelé s hodností administrátora budou mít možnost sledovat rozsáhlou správu projektů, ve které uvidí na čem všem se pracuje a co už je dokončeno, přičemž i časovou náročnost jednotlivých prací. Celý projekt lze ve finále zkalkulovat a zjistit tak jeho finanční náročnost a umožnit vytisknutí případné faktury.

### **5.7 Pracovní plocha**

Po přihlášení bude uživatel systému přesměrován na svou pracovní plochu, na které nalezne poslední aktivity, které byly vytvořeny jím nebo ostatními uživateli a které souvisí s jeho další prací.

Může se zde proto objevit nový úkol z kalendáře, nová vložená nebo sdílená poznámka. Následovat bude informace o zadání práce nebo nového projektu. V pravé liště bude zobrazena poslední komunikace prostřednictvím zasílání zpráv.

Na každé stránce se také bude nacházet globální kalendář ve kterém se budou zobrazovat informace o činnosti a událostech, které souvisí s přihlášeným uživatelem v IS.

## 6 NÁVRH DATABÁZE

Při návrhu databáze byl kladen důraz na minimální redundanci a maximální možnou rozšiřitelnost vlastností jednotlivých tabulek. Pro potřeby projektu jsou tabulky typu MyISAM a kódování českých znaků je nastaveno na cp1250\_czech\_cs. Bylo vytvořeno několik vztahů typu M:N a 1:N pro efektivní správu a aktualizaci vnořených dat.

### 6.1 Tabulka *dirfile\_operace*

V této tabulce se uchovávají veškeré informace o provedených operacích na FTP serveru. Každý upload nového souboru nebo vytvoření nového adresáře je zaznamenán do této tabulky včetně času provedení operace a ip adresy odkud byl vznešen požadavek.

Při vymazání souboru nebo adresáře nedojde k odstranění záznamu z tabulky, ale pouze k úpravě sloupce *aktivni*, který symbolizuje existenci záznamu na FTP serveru. Ve sloupci *typ* je uložena informace, zda-li se jedná o adresář nebo soubor (0 – adresář, 1 – soubor)

tabulka 4: *dirfile\_operace*

Sloupec	Typ
id	int(11)
Uziv_ID	int(11)
nazev	text
typ	tinyint(1)
aktivni	tinyint(1)
datum	datetime
ipadress	varchar(100)

### 6.2 Tabulka *hodnosti*

Tabulka obsahující informace o počtu hodnotí, uživatelských skupin, které charakterizují práva uživatele při práci v IS.

tabulka 5: *hodnosti*

Sloupec	Typ
id	int(11)
nazev	varchar(255)

### 6.3 Tabulka *kalendar\_ukoly*

Veškeré záznamy o úkolech, které si mohou uživatelé vzájemně přiřazovat jsou vedeny v tabulce *kalendar\_ukoly*. Ze schématu je vidět napojení 1:N na tabulku *uzivatele*, která shromažďuje všechny uživatele v systému.

Informace o datumu splnění úkolu je vložena formou čísla typu Integer, který generuje funkce *mktime()* a to z důvodu použití JavaScriptového kalendáře, který tento výstup generuje. Oproti tomu *datum\_splneni* je generován funkcí *NOW()* přímo při ukládání do databáze.

tabulka 6: *kalendar\_ukoly*

Sloupec	Typ
id	int(11)
Uziv_ID	int(11)
Zadal_Uziv_ID	int(11)
Nadpis	varchar(255)
Obsah	text
datum_splneni	int(11)
datum_vlozeni	datetime
upominky_dny	int(11)
splneno	tinyint(1)
aktivni	tinyint(1)

### 6.4 Tabulka *log*

Pro přehled o aktivitě všech uživatelů byla vytvořena tabulka *log*, do které se ukládají informace o novém přihlášení a odhlášení každého uživatele. Tento stav je vyjadřován hodnotou ve sloupci *akce* (0 – odhlášení, 1 – přihlášení)

Vzhledem k předpokládané velké zátěži na tuto tabulku byl zvolen datový typ pro identifikační klíč *bigint*.

tabulka 7: *log*

Sloupec	Typ
Log_ID	bigint(20)
Uziv_ID	int(11)
ipadress	varchar(255)
akce	tinyint(1)
datum_cas	datetime

## 6.5 Tabulka poznamky

V této tabulce se nacházejí všechny poznámky, které si uživatelé vytvořili. Jednotlivé sloupce reprezentují atributy každého záznamu. Ať už se tedy jedná od *Uziv\_ID* (cizí klíč z tabulky *uzivatele*) nebo o sloupec *important* označující důležitost záznamu (0 – nedůležité, 1 - důležité).

tabulka 8: poznamky

Sloupec	Typ
id	int(11)
Uziv_ID	int(11)
nadpis	varchar(255)
text	text
important	tinyint(1)
datum_vlozeni	datetime
ipadress	varchar(100)

## 6.6 Tabulka poznamky\_sdileni

Každá poznámka lze sdílet se všemi uživateli IS. Informace o vazbách sdílení se ukládají v tabulce *poznamky\_sdileni*, která obsahuje vazby 1:N na tabulky *uzivatele* a *poznamky*. Kromě toho je v každé tabulce uložena informace o datumu sdílení poznámky.

tabulka 9: poznamky\_sdileni

Sloupec	Typ
id	int(11)
Poznamka_ID	int(11)
Sdili_Uziv_ID	int(11)
Vlastnik_Uziv_ID	int(11)
datum_sdileni	datetime

## 6.7 Tabulka pristupy\_adr

Právo přístupu na celý FTP server má pouze uživatel s hodnotí administrátora. Tento uživatel také nastavuje kdo bude moci pracovat s jakou složkou na FTP serveru. Nasdílet uživatelům jdou jak jednotlivé soubory a adresáře, tak celá subdoména obsahující data.

V tabulce *pristupy\_adr* tedy nalezneme informace o sdílených souborech a adresářích na FTP serveru. Dalším parametrem je *Uziv\_ID* (uživatelské ID, odkazující do tabulky *uzivatele*) a odkaz na samotný *Adresar* nebo *soubor*.

tabulka 10: pristupy\_adr

Sloupec	Typ
Pristupy_ID	int(11)
Uziv_ID	int(11)
Adresar	text
soubor	text
typ	varchar(50)
datum_vlozeni	datetime

## 6.8 Tabulka projekty

Tabulka *projekty* zahrnuje informace o všech aktivních nebo neaktivních projektech, které kdy byly v IS vytvořeny. Smazání projektu není provedeno vymazáním záznamu z databáze, ale aktivováním atributu *smazano* na hodnotu 1.

tabulka 11: projekty

Sloupec	Typ
id	int(11)
uniqueid	int(11)
nazev	varchar(255)
kategorieid	int(11)
popis	text
datum_zadani	int(11)
datum_dokonceni	int(11)
zadalid	int(11)
dokonceno	tinyint(1)
vedouciid	int(11)
smazano	tinyint(1)
datum_vlozeni	datetime

Položka *uniqueid* reprezentuje unikátní ID projektu, které je pak použito při generování faktur jako symbol odlišující jednotlivé projekty, *kategorieid* zařazuje projekt do své sekce, *datum\_zadani* a *datum\_dokonceni* obsahují integer číslo generované funkcí *mktime()* pro potřeby JavaScriptového kalendáře. Položky *zadalid* a *vedouciid* obsahují ID uživatele z tabulky *uzivatele*.

## 6.9 Tabulka projekty\_hodiny\_lide

Pro potřebu zaznamenávat informace o odpracované práci existuje tabulka *projekty\_hodiny\_lide*. Uživatelé po zvolení požadované práce (*praceid*) vloží do databáze

informaci popisu práce (*popis\_prace*), počtu odpracovaných hodin (*pocet\_hodin*) a minut (*pocet\_minut*).

tabulka 12: projekty\_hodiny\_lide

Sloupec	Typ
id	int(11)
uzivid	int(11)
praceid	int(11)
popis_prace	text
pocet_hodin	tinyint(4)
pocet_minut	tinyint(4)
datum_prace	int(11)
datum_vlozeni	datetime

## 6.10 Tabulka projekty\_kategorie

Tabulka *projekty\_kategorie* uchovává informace o uložených kategoriích, které lze pak přiřadit k projektům.

tabulka 13: kategorie

Sloupec	Typ
id	int(11)
nazev	varchar(255)

## 6.11 Tabulka projekty\_povoleni

Kromě administrátora mají možnost práce s projekty i další uživatelé. Tato pravomoc se ukládá do tabulky *projekty\_povoleni*, která obsahuje atribut *uzivid* (ID uživatele, jenž má právo s projektem manipulovat) a *projektid* (ID projektu, na který má uživatel právo).

tabulka 14: projekty\_povoleni

Sloupec	Typ
id	int(11)
uzivid	int(11)
Projektid	int(11)

## 6.12 Tabulka projekty\_subprace

Tabulka *projekty\_subprace* má podobnou strukturu jako tabulka *projekty*. Jediný výrazný rozdíl je v položce *projektid*, která rozlišuje pro jaký projekt je daná práce zaznamenána.



tabulka 15: projekty\_subprace

Sloupec	Typ
id	int(11)
projektid	int(11)
nazev	varchar(255)
popis	text
datum_zadani	int(11)
datum_dokonceni	int(11)
zadalid	int(11)
dokonceno	tinyint(1)
vedouciid	int(11)
smazano	tinyint(1)
datum_vlozeni	datetime

### 6.13 Tabulka projekty\_subprace\_lide

Informace o tom, který uživatel pracuje na konkrétním projektu se nachází v tabulce *projekty\_subprace\_lide*. Vedoucí pracovník vytvoří práci a zadává ji konkrétnímu uživateli, přičemž má možnost doplnit konkrétní popis práce (*popis\_prace*) a na konci uvést i hodnocení, vyjadřující spokojenost s odvedenou prací (*hodnoceni*).

tabulka 16: projekty\_subprace\_lide

Sloupec	Typ
id	int(11)
uzivid	int(11)
praceid	int(11)
popis_prace	text
hodnoceni	varchar(10)
smazano	tinyint(1)
datum_vlozeni	datetime

### 6.14 Tabulka uzivatele

Uživatelé IS se nacházejí v tabulce *uzivatele*, kde každý sloupec obsahuje charakteristickou vlastnost každého uživatele. Případné další informace o uživateli se dají rozšířit přidáním dalších položek (sloupců) do tabulky.

tabulka 17: uzivatele

Sloupec	Typ
Uziv_ID	int(11)
jmeno	varchar(255)
popis	text
ipadress	varchar(255)
datum_prihlaseni	datetime
Hodnost_ID	int(11)
status	tinyint(1)
uriname	varchar(255)
mail	varchar(255)
jazyk	varchar(10)
mailpoznanky	tinyint(1)
mailftp	tinyint(1)
mailzpravy	tinyint(1)
mailkalendar	tinyint(1)
mzda	smallint(6)
login	varchar(50)
heslo	varchar(100)

### 6.15 Tabulka uzivatele\_prava

Každý uživatel může být v libovolné uživatelské skupině. Tyto skupiny reprezentuje tabulka *hodnosti*. Do tabulky *uzivatele\_prava* se tedy ukládají záznamy o přiřazení uživatelů k jednotlivým hodnostem (uživatelským skupinám).

tabulka 18: uzivatele\_prava

Sloupec	Typ
id	int(11)
uzivid	int(11)
hodnostid	int(11)

### 6.16 Tabulka zpravy

Uživatelé mají možnost si mezi sebou zasílat krátké zprávy. Všechny tyto zprávy jsou ukládány do tabulky *zpravy*. Aby bylo odlišeno, která zpráva je odpovědí na kořenovou zprávu, existuje zde sloupec *parentid*, který ukládá ID hodnotu kořenové zprávy. Sloupce *Odpoved\_from*, *Adresat\_ID* a *Odesilatel\_ID* pak odkazují do tabulky *uzivatele*. Důležitou položkou je také položka *precteno*, která symbolizuje přečtení přijaté zprávy (0 – nepřečteno, 1 – přečteno).

tabulka 19: zprávy

<b>Sloupec</b>	<b>Typ</b>
id	int(11)
parentid	int(11)
Odpoved_from	int(11)
Adresat_ID	int(11)
Odesilatel_ID	int(11)
predmet	varchar(255)
text	text
precteno	tinyint(1)
datum_odeslani	datetime

## 7 TVORBA A VLASTNOSTI INFORMAČNÍHO SYSTÉMU

Vstup do informačního systému (IS) je umožněn pouze registrovaným uživatelům. Registraci provádí administrátor a svým zaměstnancům následně sdělí přihlašovací údaje, které si pak může v nastavení každý uživatel libovolně změnit.

Celá práce v IS je založena na aktivitě session proměnných, které se nastaví po úspěšném přihlášení. V těchto globálních ukazatelech se ukládá informace unikátním ID (`$_SESSION["Uziv_ID"]`) přihlášeného uživatele a aktivitě přihlášení. Díky tomuto unikátnímu symbolu získává uživatel práva přístupu k jednotlivým položkám. Před vstupem do jednotlivých sekcí je provedena kontrola ověření hodnosti uživatele a jeho zařazení v uživatelské skupině, která má přístupová práva pro vstup do požadované části IS.



Obrázek 13: uživatelské menu

### 7.1 Rozhraní FTP serveru

Na FTP server mohou přistupovat všichni uživatelé, ale pouze hodnosti (uživatelské skupiny) 1, 2 a 3 je opravňují k akcím, které lze ve webovém rozhraní provádět. Největší pravomoce má samozřejmě administrátor (hodnost č. 1), jemuž přísluší právo vytváření a mazání adresářů, upload a mazání nových souborů. Jeho nejdůležitější funkcí je však sdílení adresářů a souborů. Ostatní uživatelé mají totiž obecně přístup pouze tam, kam jim administrátor přístup povolí. Pokud jim povolí vstup do adresáře, souvisí s tím i právo vstupu do podadresářů. V případě, že administrátor povolí uživateli zobrazit soubor, tak se uživateli zobrazí v daném adresáři pouze jeden jediný soubor. Všechny informace o sdílení se ukládají do tabulky *pristupy\_adr*.

Jelikož jsou všechny informace předávány prostřednictvím `$_GET` proměnných v adresním řádku, bylo potřeba řešit ochranu proti vstupu do nepovolených adresářů. Proto je před vstupem do každého adresáře zkontrolován záznam v tabulce *pristupy\_adr* a pouze v případě shody s požadovaným prvkem získá uživatel právo přístupu k adresáři nebo souboru.



Obrázek 14: Sdílení souborů z FTP rozhraní

Nižším stupněm hodnosti při přístupu do webového rozhraní FTP serveru je uživatel s hodností 2, který má právo vytvářet adresáře a uploadovat soubory. Rozdíl oproti administrátorovi je ten, že tento uživatel může mazat pouze to co si sám vytvoří a má přístup pouze tam, kam mu povolí vstoupit administrátor. Poslední hodnost s číslem 3 dává uživateli právo uploadovat soubory, přičemž mu náleží právo tyto soubory i mazat. Veškeré informace o vytvořených (resp. smazaných) adresářích a uploadovaných (resp. smazaných) souborech se ukládají v tabulce *dir\_file\_operace*. Administrátor systému tak má přehled o tom, jak uživatelé manipulují s přiděleným datovým prostorem.

Ostatní uživatelé systému mají na webový FTP server přístup, ale pouze pro potřebu stažení dat, která jim nasdílel administrátor.

### 7.1.1 Zpracování funkcí webového rozhraní FTP serveru

Při prvním přihlášení dojde k připojení na FTP server pomocí funkce *ftp\_connect* a přihlášení s využitím *ftp\_login*. Jelikož jsou informace o souborech a adresářích předávány přes adresní řádek je nutné takovýto odkaz ošetřit funkcí *urlencode*, která zajistí efektivní překódování speciálních znaků, které se mohou objevit v názvech souborů, pro přenos v adresním řádku.

Další důležité funkce:

- *ftp\_mkdir* – vytvoří na FTP nový adresář
- *ftp\_rmdir* – smaže adresář z FTP serveru
- *ftp\_put* – uloží na FTP server soubor
- *ftp\_delete* – smaže z FTP serveru soubor

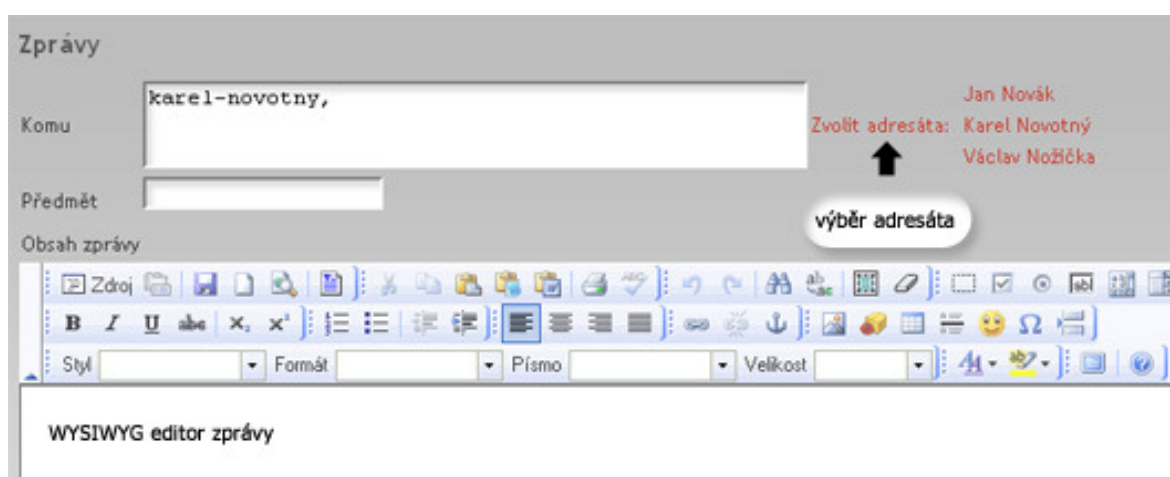
Pro vzdálený přístup k souborům na FTP server by mohla být využita funkce *fopen*, avšak možnosti hostingových služeb použití této funkce zakazují proto bylo nakonec za dostačující zvoleno volání vzdáleného souboru přes funkci *header*.

## 7.2 Zasílání zpráv mezi uživateli

Pro interní komunikaci mezi uživateli v rámci IS slouží sekce „Zprávy“. Každá vytvořená zpráva je dvouúrovňová. Pokud uživatel odpoví na přijatou zprávu, je jeho odpověď přiřazena do bloku, který se odvíjí od hlavní zprávy a tento ucelený celek sdílí od hlavní zprávy její ID. Tato položka se nazývá *parentid*. Díky tomuto systému lze jednoduše rozpoznat, zda-li se jedná o novou zprávu nebo o reakci na již stávající příspěvek.

Všechny zprávy se ukládají do tabulky *zpravy*, ve které dostanou přiděleno unikátní ID a uživatelské ID pro příjemce a odesílatele zprávy. Nepřečtenou zprávu v databázi reprezentuje ukazatel *precteno*, u kterého dojde k aktualizaci v okamžiku prvního zobrazení zprávy.

Příjemce zprávy může být kterýkoliv uživatel IS. Jeho výběr je proveden tlačítkem „Zvolit adresáta“ a následným zobrazením nabídky uživatelů, jimž lze zprávu zaslat. Jelikož pole je samodoplňovací, tak lze zprávu poslat více uživatelům IS najednou. Tělo zprávy lze pak upravit do libovolného tvaru díky přítomnosti FCK editoru. Nová zpráva se pak uživateli zobrazí na pracovní ploše, případně je mu zaslán informační e-mail o této skutečnosti.



Obrázek 15: Zasílání nových zpráv

### 7.3 Kalendář a úkoly

V sekci „Kalendář“ nalezneme rozhraní pro zadávání jednoduchých úkolů, jejichž aktivita je zobrazována v kalendáři. Každý uživatel IS má možnost si vytvořit úkol, který může přiřadit libovolnému registrovanému zaměstnanci v IS, včetně sebe samotného. Při vytváření takovéto informace je zadavateli nabídnuta možnost upozornění na provedení úkolu zasláním včasné e-mailové zprávy.

Všechny úkoly jsou ukládány do tabulky *kalendar\_ukoly* a jejich rozlišením je kromě samozřejmého unikátního ID, také ID zadavatele a ID uživatele jemuž úkol přísluší.

V hlavním kalendáři se následně zobrazují úkoly jenž si uživatel IS vytvořil nebo mu byly přiřazeny. V IS se můžeme setkat s dvěma typy kalendářů. Ten první je postavený na bázi JavaScriptu a slouží k naplnění polí formulářů, která slouží jako informace o požadovaném datumu. Výstup tohoto kalendáře je generován funkcí *mktime()*, takže v databázi se nachází Integer hodnota, která počítá sekundy od 1. 1. 1970 do zadaného datumu. Druhý typ kalendáře (hlavní kalendář) je postaven na PHP a v jeho nitru se skrývá schopnost zpracovat data přijaté z databáze a zobrazit je na požadovaném místě (prvku kalendáře). Tento kalendář využívá možnosti Ajaxu a při přepínání mezi měsíci nebo roky se aktualizuje pouze část stránky s kalendářem.



Po	Út	St	Čt	Pá	So	Ne
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Obrázek 16: Zobrazení nového úkolu v kalendáři a jeho popis po najetí kurzoru myši

## 7.4 Systém poznámek

Jednoduché vložení textu, který lze pak formátovat ve WYSIWYG editoru naplňuje sekce „Poznámky“. Při vložení nové poznámky se nabízí kromě textového pole i vložení nadpisu, který má však pouze orientační smysl. Každé poznámce lze následně zvýraznit důležitost aktualizací stavu „Důležité“. Každou poznámku lze samozřejmě jednoduše vytisknout, upravit nebo smazat.

Aby se nemusely objemné informace předávat zbytečně ve zprávách mezi uživateli, tak je možno každou poznámku sdílet s každým, kdo je registrován v IS. Takováto poznámka se pak zobrazí na pracovní ploše jako sdílená. Informace o sdílení jsou ukládány v tabulce *poznamky\_sdileni*, která slouží jako M:N propojení s tabulkami *poznamky* a *uzivatele*.



Obrázek 17: Možnosti práce s poznámkami

## 7.5 Realizace sekce „Projekty“

V této sekci se nachází správa projektů a jim příslušných prací, které byly zadány do IS systému. Vysvětlení propojenosti jednotlivých částí bude nejlepší na ilustrativním příkladu: Firma dostane za úkol sestříhat dokumentární pořad. Vedoucí pracovník tedy zadá do systému nový projekt s názvem „dokumentární pořad“. Při zakládání je projektu přiřazena kategorie, popis, datum zadání a případně datum dokončení projektu. Zvolen je také uživatel z IS, který se stane vedoucím projektu a jsou přidány základní informace o zákazníkovi pro budoucí vytvoření fakturačních údajů za odvedenou práci.



Zakladatel a vedoucí projektu získávají automaticky právo přístupu a právo zakládání prací na tomto projektu. Další uživatelé s podobnými pravomocemi mohou být přidáni do seznamu lidí, kteří mají právo manipulovat s pracemi na projektu „dokumentární pořad“ (shromažďuje je tabulka *projekty\_povoleni*). Nyní přichází na řadu přidání práce, která se má na projektu provést. U každého založeného projektu k tomu slouží speciální tlačítko, po jehož aktivaci se zobrazí formulářové pole v němž uživatel vyplní požadované údaje pro založení nové práce. Na našem ilustrativním příkladě to může být např. „střih úvodní sekvence“. Po zadání údajů jako název, popis, datum zadání, datum ukončení a vedoucí práce se nová práce zobrazí v seznamu pod projektem „dokumentární pořad“. U každé práce se opět zobrazuje několik tlačítek, kterými ji lze editovat, mazat nebo uzavírat. Důležité je ale tlačítko „Zadat práci“ po jehož aktivaci se zobrazí seznam lidí, kteří na dané práci pracují.



Obrázek 18: Zobrazení, editace, zadání nové práce k projektu

Tímto způsobem lze pak přiřadit kterémukoliv uživateli IS tuto konkrétní práci s tím, že ji můžeme dále konkretizovat vyplněním formuláře pro popis práce. Informace o zadané práci se uživateli zobrazí jako nová aktualita na pracovní ploše, zároveň o ní může být informován e-mailem.

Každý projekt a každá práce může mít zadáno časové rozmezí v němž má být úkol proveden. Tato informace je zobrazována v globálním kalendáři a na splnění úkolu je vedoucí pracovník upozorňován e-mailovou adresou. Projekt a práce je automaticky

uzavřena v okamžiku uplynutí data dokončení, případně jej lze uzamknout nebo znovu odemknout manuálně.



Obrázek 19: Popis funkce tlačítek u každého projektu

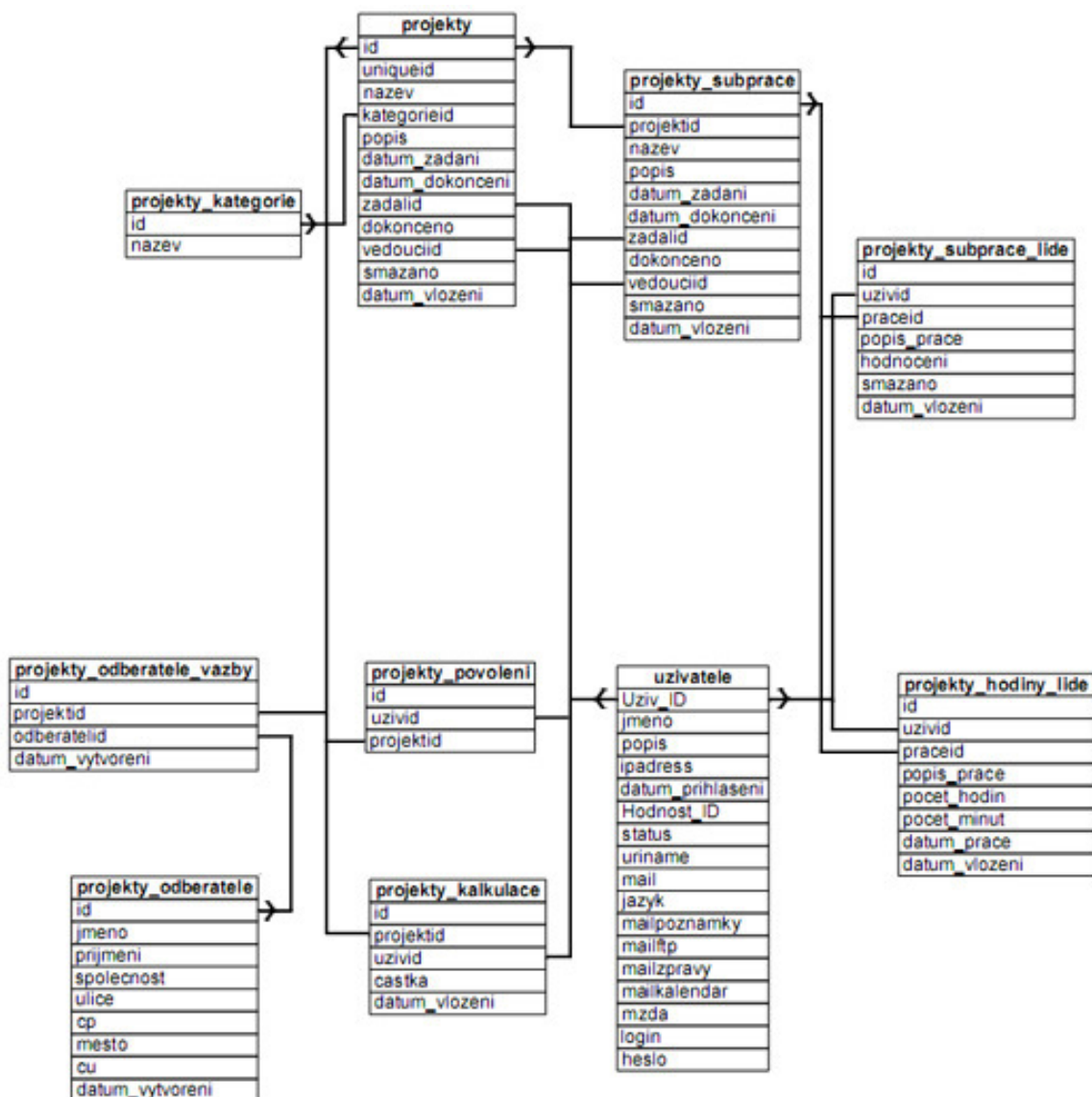
Po uzavření projektu je možné provést jeho kalkulaci, při které jsou spočítány náklady spojené s pracemi na konkrétním projektu, případně přičtení dalších externích cenových parametrů. Tyto údaje spolu s parametry o odběrateli jsou pak následně využity při tisku fakturačních údajů projektu po kliknutí na ikonku symbolizující tisk projektu.

Při vstupu do sekce „Projekty“ evidujeme kromě administrátora další dva typy uživatelů jenž mají práva s projekty (resp. s pracemi) manipulovat. Hodnost 4 umožňuje kompletní editaci a správu projektů a prací. Její nižší varianta (hodnost 5) dává pravomoci pouze ke správě prací, které mají projekty přiřazeny.

Se sekcí „Projekty“ úzce souvisí sekce „Moje práce“, kde si každý uživatel zadává odpracované hodiny a minuty k práci, která je mu přiřazena. Kromě časových údajů je zde nutné vložit datum práce a případný popis práce. Tímto způsobem dochází pak ve finále ke sčítání odpracovaných hodin na konkrétním projektu. Uživatel si pak může zobrazovat všechny odpracované hodiny v kalendáři a třídit je podle požadovaných parametrů (datum, typ práce).

Všechny data, která jsou sekcí „Projekty“ vytvářena jsou zpracovávána do devíti tabulek v databázi a jejich propojení je navrženo jak pro efektivní správu tak pro případnou

aktualizaci a přidání nových funkcí. Systém editace projektů využívá pro své zobrazovací a editovací potřeby výhod Ajaxu a uživatel tak musí obnovovat celou stránku pouze v případech zadávání nových informací do formulářových polí, editace a mazání projektu nebo prací.



Obrázek 20: Relační vztahy tabulek souvisejících s projekty

## 7.6 Realizace pracovní plochy

Po přihlášení je uživatel přesměrován na pracovní plochu. Na této úvodní stránce se nacházejí veškeré aktuality, se kterými má uživatel něco společného. V pravém bloku se mu kromě globálního kalendáře zobrazuje několik posledních zpráv, které uživatel zaslal nebo přijal.

V levé části úvodní stránky se pak nachází v tabulkovém zobrazení seznam novinek. Lze zde tedy nalézt poslední vložené poznámky, nejnovější poznámky, které s uživatelem někdo sdílí. Dále informace o přiřazení k projektu, zadání nové práce, sdílení adresářového prostoru nebo souboru administrátorem.

Zobrazení nejnovějších informací na úvodní stránce zpracovává interní funkce *GenerujPracovniPlochu()*, jejíž úkolem je projít celou databází a zaznamenat do globálního pole nejnovější změny stavů, příslušících pro přihlášeného uživatele.

## 7.7 Práce v informačním systému

Příklad jednoduché práce v IS:

Uživatel Jan Novák se přihlásí pod svým uživatelským jménem a heslem do hlavní sekce IS. Jeho pravomoce v systému jsou omezené na základní práci.

Na pracovní ploše (úvodní stránce systému) se zobrazí nejnovější akce, které byly přiřazeny přihlášenému uživateli. V ilustračním příkladě je to sdílení nového adresáře a zadání nové práce. Podrobnosti o těchto událostech získá uživatel po rozkliknutí odkazu „Podrobnosti“.



Obrázek 21: Úvodní stránka, nové přiřazené akce

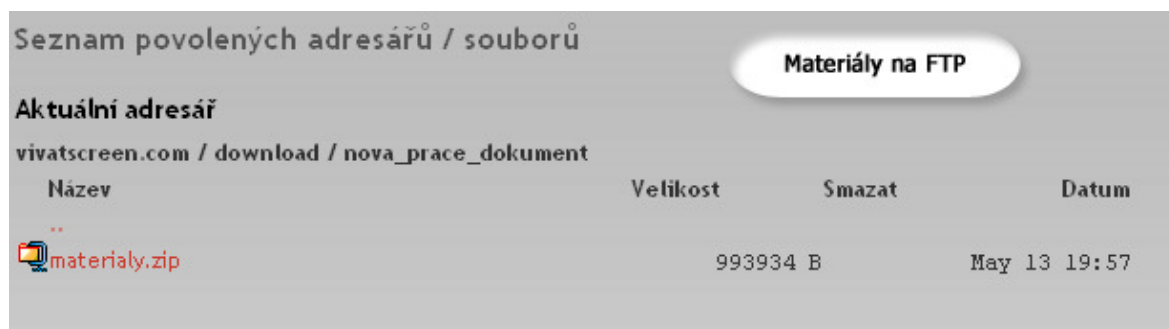
V pravé části úvodní stránky se zobrazují poslední přijaté zprávy a kalendář, který ukazuje aktuální akce schématicky členěné podle data konání. Jan Novák se tedy po rozkliknutí poslední zprávy dozví, že mu byla zadána nová práce. Tento systém je jednoduchý způsob upozornění uživatele a specifikování dalších podrobností k zadané práci.

V kalendáři si pak díky Ajaxu může jednoduše zjistit, jaké úkoly ho čekají v nejbližších týdnech nebo měsících. Po najetí na příslušný aktivní den se mu zobrazí informace o událostech, které byly v daný den vytvořeny.



Obrázek 22: Úvodní stránka, poslední zprávy a kalendář

Jan Novák se na úvodní stránce dozvěděl, že mu byl přiřazen adresní prostor na FTP serveru. Jelikož však nemá práva jakékoliv úpravy souborů nebo vytváření nových adresářů tak je mu umožněno do sdíleného adresáře pouze nahlédnout, případně stáhnout soubory v něm umístěné. Jednotlivé typy souborů jsou pro přehlednější orientaci v adresářové struktuře odlišeny vlastní ikonou. V pravé části se rovněž nachází informace o vzniku a velikosti souboru.



Obrázek 23: Materiály na FTP serveru

Uživatel má nyní všechny podklady k tomu, aby mohl začít pracovat. Ví o zadané práci, ta mu je přidělena ručně v systému a má soubory, které při práci bude potřebovat. Po dokončení práce nastává okamžik, kdy uživatel zadá do systému informace o odpracovaném čase. V sekci „Moje práce“ se nachází položka „Přidat odpracované hodiny“. Po jejím rozkliknutí se zobrazí formulář, ve kterém je nutno zvolit druh práce a následně zadat informace o odpracovaném počtu hodin a minut. Za pomoci JavaScriptového kalendáře si uživatel vybere datum, ke kterému chce informace o odpracovaném čase přiřadit. Nakonec má i možnost připsat vlastní popis o práci, kterou za daný den vykonal.

Na hlavní stránce sekce „Moje práce“ se zobrazují informace o odpracovaných časech a jednotlivých pracích. Uživatel má možnost si třídit informace podle konkrétního rozmezí data, podle kalendářního měsíce a podle druhu práce, na které pracoval. Veškeré informace o odpracované práci se navíc zobrazují i v samostatném kalendáři, jehož Ajaxová obsluha dovoluje zobrazit informace tříděním podle kalendářních dnů a měsíců. Každá vložená odpracovaná práce se dá upravit, popř. smazat a v dolní části tabulky se zobrazuje součet všech odpracovaných hodin, v případě zadané mzdy i částka, která bude uživateli vyplacena.

**Moje práce**

**Přidání informací o odpracované práci**

Zvolte práci

Zvolte datum práce

Počet hodin  (Celá čísla)

Počet minut  (Celá čísla)

**Odpracovaná práce**

Popis práce

**Uložit údaje**

Obrázek 24: Zadání informací o odpracovaném času

Z pohledu klasického uživatele se základními právy, ale možnosti využití IS nekončí. Nadále má možnost komunikovat s dalšími uživateli IS prostřednictvím sekce „Zprávy“, vytvářet vlastní úkoly v sekci „Kalendář“, přidávat si poznámky skrz WYSIWYG editor v sekci „Poznámky“.



Hlavním posláním tohoto uživatele je ale získat skrz IS informaci o zadané práci a následně si zezvidovat počet odpracovaných hodin. Díky propojenosti systému je pak možno ke každému projektu udělat kalkulační součet, který bude zobrazovat čas zaměstnanců strávený na podpůrných pracech v projektu.

Uživatelé s vyšší hodnotí si pak mohou procházet statistiky a informace o konkrétních projektech, zakládat nové práce, nové projekty, přiřazovat adresní prostor FTP serveru a kompletně spravovat IS. Možnosti práce v IS tedy rostou s uživatelskými právy.

The screenshot shows a web interface for managing work records. At the top, there is a calendar for May 2009, with the 15th highlighted in red. Below the calendar are three sorting sections:

- Třídít podle datumu:** Includes input fields for 'Od' (05-13-2009) and 'Do' (05-13-2009), and a 'Třídít' button.
- Třídít podle měsíce:** Includes dropdown menus for 'Měsíc' (Květen) and 'Rok' (2009), and a 'Třídít' button.
- Třídít podle práce:** Includes a dropdown menu for 'Zvolte práci' (Střih úvodní sekvence) and a 'Třídít' button.

On the right side, there is a button labeled 'Třídění práce'. Below the sorting sections is a table with the following data:

Datum práce	Práce	Počet hodin	Počet minut	Nastavení
15.5.2009	Střih úvodní sekvence	8	0	 
<b>Celkem</b>		<b>8</b>	<b>0</b>	
<b>K vyplaceni</b>		<b>Nezadána mzda</b>		

At the bottom of the table, there are navigation arrows: << 1 >>

Obrázek 25: Třídění informací o odpracované práci

## 7.8 Používání informačního systému

Při mnoha úkonech v IS je využit Ajax. Skripty se na pozadí stránky odesílají na webový server a jejich výstup se zobrazí do příslušného objektu s unikátním identifikátorem ID. O zpracování většiny Ajaxových požadavků se stará JavaScriptová třída Prototype (<http://www.prototypejs.org/>) s jejíž vestavěnými funkcemi lze kromě obsluhy skriptů na pozadí manipulovat i s objekty ve struktuře DOM.



```
//zobrazí obsah adresáře
function ZobrazAdresar(idecko) {

    var url = 'included/vypis-ftp.php';
    var pars = 'id='+idecko+'';

    //volání třídy, která zpracuje požadavek na pozadí
    var myAjax = new Ajax.Updater(
        'seznamadresare', //ID objektu, kde se vykreslí data
        url, //odkaz na skript PHP, který zpracuje data
        {method:'get', //metoda předávání parametrů
          parameters:pars} //parametry pro PHP skript
    );
    return false;
}
```

Obrázek 26: Volání Ajaxového skriptu

```
<a href="javascript:ZobrazAdresar('4')">Zobrazit adresář</a>
<div id="seznamadresare"></div>
```

Obrázek 27: HTML kód pro volání Ajaxového skriptu

Výše uvedený HTML kód volá po kliknutí na odkaz funkci JavaScriptu *ZobrazAdresar()*, která vytvoří objekt *myAjax* a s jeho pomocí předá data PHP skriptu a nakonec je vykreslí do požadovaného objektu, identifikovaného unikátním ID, který se nachází na HTML stránce

Třída Prototype není jedinou knihovnou, založenou na bázi JavaScriptu, která je implementována do každé stránky IS. Dále jsou použity JavaScriptové knihovny pro generování kalendáře, pro opravu zobrazení PNG souborů v Internet Exploreru 6 a knihovna qTip pro zobrazení titulků u libovolného HTML prvku.

JavaScript je využíván i v použitém WYSIWYG editoru FCK, jehož nastavení je upraveno pro potřeby IS.

Díky výše zmíněným použitým skriptům je tedy nezbytně nutné, aby byla při práci v IS zapnuta v internetovém prohlížeči podpora JavaScriptu, protože jeho deaktivací se většina důležitých funkcí stává bezvýznamná.

IS má podporu v nejpoužívanějších webových prohlížečích. Jeho funkce byla otestována v těchto verzích prohlížečů:

- Mozilla Firefox 3
- Internet Explorer 6, 7, 8
- Opera 9

System je připraven na přechod k vícejazyčnému režimu. Proto jsou veškeré textové informace uloženy v globálním poli *\$lang*, na které se odkazují příslušné funkce. Jelikož při kombinaci používání Ajaxových skriptů a české diakritiky může vznikat problém se správným zobrazením háčeků a čárek, je každý takovýto textový výstup ošetřen překódováním o které se stará funkce *Cestina()*, vracející správně zobrazené znaky s využitím PHP funkce *iconv*.

## ZÁVĚR

V rámci diplomové práce byl vytvořen webový portál firmy pro sdílení dat, který může být také označován jako podnikový webový informační systém (IS).

Mezi hlavní cíle patřila implementace rozhraní webového FTP serveru, ke kterému by měl přístup pouze uživatel s administrátorskými právy a jeho pravomocí by bylo určovat, které soubory a adresáře mohou vidět ostatní uživatelé. Těmto lidem lze pak přiřadit nižší hodnotu dávající jim právo vytvářet adresáře a uploadovat soubory.

Druhým důležitým prvkem bylo vytvoření správy projektů na nichž firma pracuje. K těmto projektům lze následně vytvářet práce a přiřazovat je k uživatelům IS. Po dokončení projektu dochází k jeho uzavření a kalkulaci nebo případnému tisku faktury za provedenou činnost. Uživatelé si pak vedou statistiku odpracovaných hodin na jednotlivých pracích a ve výsledku tak mají přesný přehled o tom, kdy a jak dlouho pracovali.

V rámci bezpečnosti se nachází v systému několik prvků, které při každé operaci kontrolují práva přístupu uživatele k jednotlivým sekcím IS. Tyto práva jsou specifikována rozdělením na uživatelské skupiny, kde se nacházejí uživatelé s právy administrátora, právem vytváření a uploadu souborů na FTP server, možností kompletní správy prací a projektů a obyčejní uživatelé IS. Tyto práva přiřazuje uživatelům administrátor systému.

Z hlediska použitelnosti je systém připraven na ostrý testovací provoz. V rámci rozvoje do budoucna se počítá s nazením modulů, které ještě více rozšíří práva a možnosti přihlášených uživatelů a dají systému vyšší rozměr pro zpracování firemních dat skrz internetové rozhraní prohlížeče. Rozšíření se tedy dočká FTP server (upload prostřednictvím rozhraní FLASH), připraví se e-mailové rozhraní pro komunikaci prostřednictvím e-mailových schránek (@nejakyemail.cz). Správa projektů se dočká napojení na FTP server, kdy bude ke konkrétnímu projektu (např. stříh dokumentárního pořadu) možno přiřadit odkaz na ukázkový materiál, který se bude nacházet přímo na FTP serveru. Pro zodpovězení dotazů spojených s odbornými znalostmi některých zaměstnanců bude vytvořen základní Helpdesk, který bude odesílat dotazy na jednotlivé firemní sekce (např. stříh, režie, produkce) a jejich zodpovězené otázky se budou zobrazovat ve výpisu Helpdesku.

## ZÁVĚR V ANGLIČTINĚ

In this diploma thesis, a web based data sharing server, or a business web information system was created.

One of main points of this project was the implementation of a web based interface to access the FTP which will be open only for users with administration rights and they will then be able to set which files and directories will be visible to specified users. These users can be defined and given lower than administration rights which will grant them rights such as the ability to create directories or upload files.

Another part of the information system is a business project management tool. It is able to create independent tasks for specific users. After finishing a specific project, chief of the project can create a project clearance or print invoice for all finished tasks. Users have to fill their working hours spent on specific tasks and then they can sort it by specific attributes to see a detailed summary of how long and on what were they working.

There are many components that check if the user has rights to access the given area or section of the system after every operation to ensure absolute security, mainly of the web based interface for accessing the FTP server. Users are arranged into groups, each having specific rights - a group of administrators, users with rights to create folders and upload files, users with rights to manage tasks and projects and ordinary users. Only administrator can assign different rights to a user.

The information system is now fully operational and ready for testing. There are many possibilities to improve the whole system in the future via plug-in modules which can extend user rights, create different ways to log in and expand the systems ability to share data through a web browser. So the FTP server and file upload can be extended to use the FLASH technology, a new section to manage e-mail box through a web browser will be added (@someemail.com), business project management will be more connected with the FTP server with the possibility to add specific file to a specific project. Absolutely new section will be the helpdesk with frequently asked questions with complete set of tools to edit them. Every question will be send to a specific business department and the answer will be saved in database for further usage.

**SEZNAM POUŽITÉ LITERATURY**

- [1] C. ZAKAS, Nicholas , MCPEAK, Jeremy, FAWCETT, Joe. *Ajax - profesionálně*. [s.l.] : [s.n.], 2007. 672 s.
- [2] PHP. *Wikipedie* [online]. 2009 [cit. 2009-04-09]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/PHP>
- [3] BLAHOUT, Michal. Jemný úvod do ASP. *Brinkster.com* [online]. 2008 [cit. 2009-04-09]. Dostupný z WWW: <<http://www12.brinkster.com/mibla/>>.
- [4] Active Server Pages. *Wikipedie* [online]. 2009 [cit. 2009-04-09]. Dostupný z WWW: <[http://cs.wikipedia.org/wiki/Active\\_Server\\_Pages](http://cs.wikipedia.org/wiki/Active_Server_Pages)>.
- [5] JEDLIČKA, Karel, ORÁLEK, Jakub. Prostorové rozhraní informačního systému malé obce řešené v Open Source Software. *Geoinformatics.fsv.cvut.cz* [online]. 2006 [cit. 2009-04-10]. Dostupný z WWW: <[http://geoinformatics.fsv.cvut.cz/gwiki/Prostorov%C3%A9\\_rozhran%C3%AD\\_informa%C4%8Dn%C3%ADho\\_syst%C3%A9mu\\_mal%C3%A9\\_obce\\_%C5%99e%C5%A1en%C3%A9\\_v\\_Open\\_Source\\_Software](http://geoinformatics.fsv.cvut.cz/gwiki/Prostorov%C3%A9_rozhran%C3%AD_informa%C4%8Dn%C3%ADho_syst%C3%A9mu_mal%C3%A9_obce_%C5%99e%C5%A1en%C3%A9_v_Open_Source_Software)>.
- [6] Srovnání RS Joomla Drupal Wordpress. *Redakcni-systemy.com* [online]. 2009 [cit. 2009-04-10]. Dostupný z WWW: <<http://www.redakcni-systemy.com/index.php/vseobecne/clanky/26-srovnani-rs-joomla-drupal-wrdpress>>.
- [7] MySQL manuál. *Gene.cz* [online]. 2005 [cit. 2009-05-04]. Dostupný z WWW: <<http://mm.gene.cz/>>.
- [8] DARIE, Christian, et al. *AJAX a PHP - tvoříme interaktivní webové aplikace profesionálně*. [s.l.] : [s.n.], 2006. 320 s. ISBN 80-86815-47-1.
- [9] E. DAVIS, Michele , A. PHILLIPS, Jon. *Learning PHP & MySQL, Second Edition*. [s.l.] : [s.n.], 2007. 428 s. ISBN 0-596-51401-8.
- [10] PEDRO, Bruno, RODRIGUES, Vitor. *PHP and Smarty on Large-Scale Web Development*. [s.l.] : [s.n.], 2007. 36 s. ISBN 0-596-51379-8.
- [11] TATROE, Kevin, LERDORF, Rasmus , MACINTYRE, Peter. *Programming PHP, Second Edition*. [s.l.] : [s.n.], 2006. 540 s. ISBN 0-596-00681-0.

- [12] PROKOPOVÁ, Zdenka. *Databázové systémy MySQL+PHP* . [s.l.] : [s.n.], 2006. 126 s. ISBN 80-7318-486-9.
- [13] GILMORE W., Jason. *Velká kniha PHP a MySQL 5 : kompendium znalostí pro začátečníky i profesionály*. [s.l.] : [s.n.], 2007. 864 s. ISBN 80-86815-53-6.
- [14] BASL, Josef, BLAŽÍČEK, Roman. *Podnikové informační systémy - Podnik v informační společnosti – 2. výrazně přepracované a rozšířené vydání*. [s.l.] : [s.n.], 2008. 288 s. ISBN 978-80-2472-279-5.
- [15] E. WILLIAMS, Hugh , LANE, David. *Web Database Applications with PHP, and MySQL* . [s.l.] : [s.n.], 2002. 582 s. ISBN 0-596-00041-3.

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

HTML	HyperText Markup Language, značkovací jazyk pro tvorbu WWW stránek
URL	Uniform Resource Locator, definice doménové adresy serveru
XML	eXtensible Markup Language, značkovací jazyk pro správu různých typů dat
PHP	Skriptovací jazyk, používaný k tvorbě WWW stránek
RSS	Really Simple Syndication, formát rodiny XML pro čtení novinek na internetu
XHTML	Extensible Hypertext Markup Language, novější standard HTML jazyka
MySQL	Databázový server, využívající jazyk SQL.
ASP	Active Server Pages, skriptovací jazyk z dílny Microsoftu
FTP	File Transfer Protocol, protokol aplikační vrstvy používaný pro přenos dat
IS	Informační systém
CMS	Content management system, redakční systém pro správu webových stránek
FCK	Editor, používaný při zpracování textových dat před uložením do databáze
WYSIWYG	What You See Is What You Get, typ editoru, který umožňuje upravovat text pomocí HTML bez nutnosti znalosti tohoto jazyka
GNU	Projekt zaměřený na svobodný software
SSL	Secure Sockets Layer, vrstva zabezpečující komunikaci mezi šifrováním a autentizací
UTF	Druh kódování webových stránek
SQL	Standardizovaný dotazovací databázový jazyk
MyISAM	Druh formátu pro uložení dat v MySQL
InnoDB	Druh formátu pro uložení dat v MySQL pro uložení transakcí
CGI	Common Gateway Interface, protokol k propojení WWW aplikací se serverem
Perl	Programovací jazyk vyvinutý v roce 1987

---

GD	Knihovna v PHP obsahující funkce pro manipulaci s obrázky
POP3	Protokol, který zajišťuje připojení k e-mailové schránce
IMAP	Protokol, který zajišťuje připojení k e-mailové schránce na základě trvalého spojení
COM	Component Object Model, způsob komunikace softwarových komponent
IIS	Webový server (Alternativa k Apachi)
ID	Unikátní identifikátor
DOM	Document Object Model, objektově orientovaný způsob reprezentace XML nebo HTML dokumentu



**SEZNAM OBRÁZKŮ**

Obrázek 1: Schéma klasického dotazování prostřednictvím webového prohlížeče [1] .....	13
Obrázek 2: Schéma provedení dotazu za pomoci AJAXU [1] .....	14
Obrázek 3: Jednoduchý příklad použití skriptu šablonovacího systému .....	18
Obrázek 4: HTML šablona pro SMARTY .....	19
Obrázek 5: PHP kód volání šablony pomocí SMARTY .....	19
Obrázek 6: Konečné zobrazení šablony po zpracování pomocí SMARTY .....	19
Obrázek 7: Použití cyklů u SMARTY .....	20
Obrázek 8: Příklad přepsání proměnné u HTMLtmpl .....	21
Obrázek 9: Použití cyklu v šablonovacím systému HTMLtmpl .....	21
Obrázek 10: Schématické znázornění vrstev informačního systému založeném na klientské části formou webového prohlížeče .....	23
Obrázek 11: Ukázka vztahu tabulek 1:N .....	32
Obrázek 12: Ukázka vztahu M:N .....	32
Obrázek 13: uživatelské menu .....	52
Obrázek 14: Sdílení souborů z FTP rozhraní .....	53
Obrázek 15: Zasílání nových zpráv .....	54
Obrázek 16: Zobrazení nového úkolu v kalendáři a jeho popis po najetí kurzoru myši .....	55
Obrázek 17: Možnosti práce s poznámkami .....	56
Obrázek 18: Zobrazení, editace, zadání nové práce k projektu .....	57
Obrázek 19: Popis funkce tlačítek u každého projektu .....	58
Obrázek 20: Relační vztahy tabulek souvisejících s projekty .....	59
Obrázek 21: Úvodní stránka, nové přiřazené akce .....	60
Obrázek 22: Úvodní stránka, poslední zprávy a kalendář .....	61
Obrázek 23: Materiály na FTP serveru .....	62
Obrázek 24: Zadání informací o odpracovaném času .....	63
Obrázek 25: Třídění informací o odpracované práci .....	64
Obrázek 26: Volání Ajaxového skriptu .....	65
Obrázek 27: HTML kód pro volání Ajaxového skriptu .....	65

**SEZNAM TABULEK**

tabulka 1: Srovnání funkcí redakčních systémů Drupal, Joomla, Wordpress [6].....	27
tabulka 2: Výpis vybraných datových typů databáze MySQL [7] .....	34
tabulka 3: Rozdělení práv uživatelů.....	40
tabulka 4: dirfile_operace .....	44
tabulka 5: hodnoti.....	44
tabulka 6: kalendar_ukoly.....	45
tabulka 7: log .....	45
tabulka 8: poznamky .....	46
tabulka 9: poznamky_sdileni .....	46
tabulka 10: pristupy_adr .....	47
tabulka 11: projekty .....	47
tabulka 12: projekty_hodiny_lide .....	48
tabulka 13: kategorie.....	48
tabulka 14: projekty_povoleni .....	48
tabulka 15: projekty_subprace .....	49
tabulka 16: projekty_subprace_lide .....	49
tabulka 17: uzivatele .....	50
tabulka 18: uzivatele_prava .....	50
tabulka 19: zpravy.....	51

## SEZNAM PŘÍLOH

PI: Disk CD s diplomovou prací.