

UNIVERZITA PARDUBICE  
Fakulta elektrotechniky a informatiky

Informační systém knihovny

Jiří Ulbrich

Bakalářská práce

2010

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2009/2010

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jiří ULBRICH**  
Osobní číslo: **I07974**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Informační systém knihovny**  
Zadávající katedra: **Katedra informačních technologií**

### Z á s a d y p r o v y p r a c o v á n í :

Teoretická část bude obsahovat zdůraznění významu informačního systému knihovny pro potřeby knihovny a čtenářů. Součástí teoretické části bude popis stávajících systémů, vzájemné porovnání a výsledná analýza vlastností jednotlivých systémů. Implementační část bude postavena na prezentaci návrhu informačního systému knihovny a vytvoření konkrétního informačního systému knihovny.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

- \*DRUSKA, P. CSS a XHTML - tvorba dokonalých webových stránek krok za krokem. Grada 2006.
- \*PÍSEK, S. HTML a XHTML - začínáme programovat. Grada 2003.
- \*ŠIMŮNEK, M. SQL kompletní kapesní průvodce. Praha: Grada Publishing, 1999.
- \*<http://www.w3.org/Style/CSS/>.
- \*<http://www.w3.org/TR/html4/>.
- \*<http://www.w3.org/TR/xhtml1/>.
- \*SATRAPA, P. Web Design. Neokortex s.r.o. 1997.
- \*VÁCLAVEK, P. Javascript - Hotová řešení. Brno: Computer Press, 2003.
- \*LACKO, L. Oracle, správa, programování a použití databázového systému. Praha: Computer Press, 2002.
- \*LONEY, K., THERIAULT, M. Mistrovství v Oracle. Praha, Computer Press, 2002.
- \*<http://interval.cz/>.
- \*<http://www.jakpsatweb.cz/>.

Vedoucí bakalářské práce:

**Ing. Miloslav Macháček**

Katedra informačních technologií

Datum zadání bakalářské práce: **15. ledna 2010**

Termín odevzdání bakalářské práce: **14. května 2010**



prof. Ing. Simeon Karamazov, Dr.  
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.  
vedoucí katedry

V Pardubicích dne 31. března 2010

## **Prohlášení autora**

Prohlašuji, že jsem tuto bakalářskou práci na téma Informační systém knihovny vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury, který tvoří přílohu této práce.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 10. 5. 2010

Jiří Ulbrich

## **Poděkování**

Touto formou bych chtěl poděkovat panu ing. Miloslavu Macháčkovi za ochotu vézt mě při vypracovávání mé bakalářské práce, za mnoho rad, věcných připomínek a návrhů k mé práci.

Dále bych chtěl poděkovat paní Mgr. Ivě Procháskové, která mi poskytla praktické a cenné informace a materiály ke knihovním systémům a dopomohla mi blíže se seznámit s problematikou knihovních systémů.

## **Anotace**

Bakalářská práce na téma Informační systém knihovny se ve své teoretické části zabývá analýzou a zhodnocením problematiky stávajících knihovních systémů. V praktické části této bakalářské práce je vytvořen informační systém na úrovni obecní knihovny, pomocí kterého je umožněno správcům a knihovníkům spravovat knihovní fond a veškerou potřebnou agendu. Čtenářům je umožněn přístup ke knihovnímu fondu a všechny standardní knihovní služby jako např. výpůjčky, rezervace nebo vyhledávání. To vše pomocí sítě internet.

## **Klíčová slova**

knihovní systém, webová aplikace, WWW, HTTP, PHP, MySQL, Dibi, jQuery

## **Title**

Informative library system

## **Annotacion**

Bachelor thesis on the theme Library information system, in the theoretical part deals with the analysis and appreciation of the problems existing library systems. The practical part of this thesis is created an information system on the level of the local library with which is possible for administrators and librarians who manage the library collection and all necessary agenda. Readers are allowed to access to library collections and all standard library services such as loans, reservations or search. All using the Internet.

## **Keywords**

library system, web application, WWW, HTTP, PHP, MySQL, Dibi, jQuery

## Obsah

SEZNAM OBRÁZKŮ.....	9
SEZNAM TABULEK .....	9
SEZNAM ZKRATEK.....	10
<b>1 ÚVOD .....</b>	<b>11</b>
<b>2 PROBLEMATIKA WEBOVÝCH APLIKACÍ .....</b>	<b>12</b>
2.1 APLIKAČNÍ SOFTWARE PRO VÝVOJ WEBOVÝCH APLIKACÍ.....	12
2.1.1 Služba WWW (World Wide Web) .....	12
2.1.2 HTTP protokol (Hypertext Transfer Protocol) .....	12
2.1.3 Metody HTTP .....	13
2.1.4 Stavové kódy.....	13
2.1.5 HTTP server.....	14
2.1.6 HTTP klient .....	14
2.2 TECHNOLOGIE A JAZYKY PRO VÝVOJ WEBOVÝCH APLIKACÍ .....	15
2.2.1 Značkovací jazyky .....	15
2.2.2 Skriptovací jazyky.....	16
2.2.3 Kaskádové styly.....	17
2.2.4 Jazyk SQL a databázové systémy .....	17
2.3 WEBOVÁ APLIKACE.....	18
2.4 BEZPEČNOST WEBOVÉ APLIKACE.....	19
<b>3 PROBLEMATIKA KNIHOVNÍCH SYSTÉMŮ .....</b>	<b>20</b>
3.1 KNIHOVNÍ SYSTÉMY OBECNĚ .....	20
3.2 OBVYKLÉ MODULY KNIHOVNÍCH SYSTÉMŮ .....	21
3.2.1 Akviziční modul .....	21
3.2.2 Katalogizační modul .....	21
3.2.3 Výpůjční modul.....	21
3.2.4 Modul OPAC.....	21
3.2.5 Správa periodik .....	22
3.2.6 Z39.50 klient server.....	22
3.2.7 Modul pro správu uživatelů.....	22
3.3 PLATNÉ STANDARDY KNIHOVNÍCH SYSTÉMŮ .....	22
3.4 ANALÝZA DOSTUPNÝCH ŘEŠENÍ NA TRHU V OBLASTI KNIHOVNÍCH SYSTÉMŮ .....	23
3.4.1 DAIMON .....	23
3.4.2 Knihovní systém Clavius .....	23
3.4.3 ALEPH .....	24
3.4.4 KpWin SQL.....	25
3.5 ZHODNOCENÍ.....	25
<b>4 VÝVOJ APLIKACE KNIHOVNÍHO SYSTÉMU .....</b>	<b>26</b>
4.1 ANALÝZA A STANOVENÍ CÍLŮ APLIKACE .....	26
4.2 RICH PICTURE .....	26

4.3	TABULKA ROZDĚLENÍ ROLÍ A JEJICH PRÁV .....	27
4.4	ADRESÁŘOVÁ STRUKTURA .....	28
4.5	LAYOUT INFORMAČNÍHO SYSTÉMU .....	29
4.6	TECHNOLOGIE A JAZYKY POUŽITÉ K VÝVOJI.....	30
4.6.1	<i>Jazyky a platformy použité k vývoji aplikace</i> .....	30
4.6.2	<i>Databázová vrstva Dibi</i> .....	30
4.6.3	<i>jQuery knihovna</i> .....	31
4.6.4	<i>PHPmailer třída</i> .....	33
4.7	ER-DIAGRAM .....	34
4.8	POPIS TABULEK DATABÁZE .....	35
4.8.1	<i>Tabulka uživatelů</i> .....	35
4.8.2	<i>Tabulka aktualit</i> .....	35
4.8.3	<i>Tabulka rolí</i> .....	35
4.8.4	<i>Tabulka cronu</i> .....	35
4.8.5	<i>Tabulka výpůjček</i> .....	36
4.8.6	<i>Tabulka rezervací</i> .....	36
4.8.7	<i>Tabulka statusů výpůjček</i> .....	36
4.8.8	<i>Tabulka statusů rezervací</i> .....	36
4.8.9	<i>Tabulka exemplářů</i> .....	37
4.8.10	<i>Tabulka statusů exemplářů</i> .....	37
4.8.11	<i>Tabulka autorů</i> .....	37
4.8.12	<i>Tabulka žánrů</i> .....	37
4.8.13	<i>Tabulka titulů</i> .....	37
4.8.14	<i>Tabulka mezinárodní desetinného třídění</i> .....	38
4.8.15	<i>Tabulka statusů titulů</i> .....	38
4.9	PROGRAMY A OPERAČNÍ SYSTÉMY POUŽITÉ PŘI VÝVOJI .....	38
4.9.1	<i>Programové vybavení</i> .....	38
<b>5</b>	<b>UŽIVATELSKÁ PŘÍRUČKA A POPIS APLIKACE .....</b>	<b>40</b>
5.1	NEPŘIHLÁŠENÝ UŽIVATEL .....	40
5.2	ČTENÁŘSKÝ REŽIM.....	41
5.3	KNIHOVNICKÝ A ADMINISTRÁTORSKÝ REŽIM.....	42
<b>6</b>	<b>INSTALAČNÍ PŘÍRUČKA .....</b>	<b>44</b>
<b>7</b>	<b>ZÁVĚR.....</b>	<b>45</b>
	<b>POUŽITÁ LITERATURA A ZDROJE .....</b>	<b>46</b>
	<b>PŘÍLOHA A – UKÁZKA NĚKTERÝCH ZDROJOVÝCH KÓDŮ APLIKACE .....</b>	<b>48</b>



## Seznam obrázků

Obrázek 1 - Struktura HTTP zprávy (ČEGAN, 2010).....	13
Obrázek 2 - Procentuální poměry využití webových serverů k dubnu 2010 .....	14
Obrázek 3 - Procentuální podíly webových prohlížečů k březnu 2010 .....	15
Obrázek 4 - Schéma webové aplikace.....	19
Obrázek 5 - Obvyklé schéma knihovního systému .....	20
Obrázek 6 - Rich Picture .....	27
Obrázek 7 - Adresářová struktura aplikace .....	28
Obrázek 8 - layout aplikace informačního systému .....	29
Obrázek 9 - ER diagram aplikace knihovního systému .....	34
Obrázek 10 - konkrétní stránka titulu.....	40
Obrázek 11 - formulář pro vyhledávání titulů bez CAPTCHA kontroly.....	41
Obrázek 12 - stránka uživatelského profilu.....	41
Obrázek 13 - administrace titulů v systému .....	42
Obrázek 14 - formulář pro přidání titulu .....	43

## Seznam tabulek

Tabulka 1 – Rozdělení rolí a práv .....	28
---	----

## Seznam zkratek

<b>AACR2R</b>	Anglo-American Cataloguing Rules, 2nd Revised Edition (Anglo-americká katalogizační pravidla)
<b>AJAX</b>	Asynchronous JavaScript and XML (Asynchronní JavaScript a XML)
<b>ASP</b>	Active Server Pages
<b>CAPTCHA</b>	Completely Automated Public Turing test to tell Computers and Humans Apart (Plně automatický obecný Turingův test pro rozlišení počítačů a lidí)
<b>CERN</b>	Conseil Européen pour la Recherche Nucléaire (Evropská rada pro jaderný výzkum)
<b>CSS</b>	Cascading Style Sheets (Tabulky Kaskádových stylů)
<b>DOM</b>	Document Object Model (Objektový model dokumentu)
<b>ERD</b>	Entity-Relationship Diagram (Diagram vztahů mezi entitami)
<b>HTML</b>	HyperText Markup Language (Hypertextový značkovací jazyk)
<b>HTTP</b>	HyperText Transfer Protocol (Protokol pro výměnu hypertextových dokumentů)
<b>ISBD</b>	International Standard Bibliographic Description (Mezinárodní standardní bibliografický popis)
<b>ISBN</b>	International Standard Book Number (Mezinárodní standardní číslo knihy)
<b>ISO</b>	International Organization for Standardization (Mezinárodní organizace pro normalizaci)
<b>MARC21</b>	Machine-Readable Cataloging formats
<b>MDT</b>	Mezinárodní desetinné třídění
<b>MIME</b>	Multipurpose Internet Mail Extension (Víceúčelové rozšíření pošty)
<b>MSDN</b>	Microsoft Developer Network
<b>OPAC</b>	Open Public Access Catalogues (Veřejně přístupný katalog)
<b>PHP</b>	Personal Hypertext Preprocessor (Skriptovací jazyk pro tvorbu webových stránek)
<b>RFC</b>	Request for comments (Žádost o komentáře)
<b>RM OSI</b>	Referention model Open Systems Interconnection (Referenční model propojení otevřených systémů)
<b>SEO</b>	Search Engine Optimization (optimalizace pro vyhledávače)
<b>SGML</b>	Standard Generalized Markup Language (Univerzální značkovací metajazyk)
<b>SMTP</b>	Simple Mail Transfer Protocol (Protokol pro přenos zpráv elektronické pošty)
<b>SQL</b>	Structured Query Language (Strukturovaný dotazovací jazyk)
<b>SSL</b>	Secure Sockets Layer, SSL (Vrstva bezpečných sonetů, stará se o zabezpečení)
<b>TCP</b>	Transmission Control Protocol
<b>TLS</b>	Transport Layer Security (Předchůdce SSL)
<b>W3C</b>	The World Wide Web Consortium (Mezinárodní konsorcium, jehož členové společně s veřejností vyvíjejí webové standardy)
<b>WWW</b>	World Wide Web (Celosvětová pavučina)
<b>XHTML</b>	eXtensible Hypertext Markup Language (Rozšiřitelný hypertextový značkovací jazyk)
<b>XML</b>	eXtensible Markup Language (Rozšiřitelný značkovací jazyk)

# 1 Úvod

Po důkladném rozhodování při volbě vhodného tématu mojí bakalářské práce jsem se rozhodl pro webovou aplikaci knihovního systému. K rozhodnutí mě vedla má dosavadní zkušenost v oblasti návrhu webových aplikací a možnost pracovat v týmu zkušenějších webových programátorů, od kterých se lze hodně naučit.

Původně byla aplikace knihovního systému zamýšlena k praktickému využití v jedné z obecních knihoven. I když později ze záměru sešlo, je systém plně funkční a použitelný na úrovni obecní knihovny.

V teoretické části této bakalářské práce se zabývám obecně návrhy webových aplikací, technologiemi používanými k vývoji aplikací, databázovými systémy atd. V druhé půlce teoretické části jsou popsány a analyzovány některé stávající knihovní systémy a jejich funkce.

V praktické části je popsán samotný vývoj aplikace od návrhu systému až po zhodnocení výsledků mé práce. Čtenář se seznámí kompletně s databázovým návrhem knihovního systému. Součástí jsou i obrázky aplikace a ukázky důležitých částí kódu.

## 2 Problematika webových aplikací

### 2.1 Aplikační software pro vývoj webových aplikací

#### 2.1.1 Služba WWW (World Wide Web)

V překladu *WWW* znamená celosvětová pavučina. Vznik služby *WWW* se datuje k roku 1989, kdy ve výzkumném středisku v *CERNu*<sup>1</sup> v Ženevě ve Švýcarsku byly realizovány dvěma výzkumnými pracovníky projekty návrhů hypertextových systémů pracujících v prostředí internet. V roce 1990 výzkumný pracovník Tim Berners-Lee uveřejnil návrh tohoto systému pod názvem World Wide Web. Naprogramoval první webový server, klient a první webové stránky. Oficiální vznik systému je až v roce 1991.

Služba *WWW* je celosvětová síť tvořená množstvím hypertextových dokumentů, pro kterou je typické vyhledávání, prohlížení nebo sdílení těchto dokumentů. Služba je dostupná prohlížeči pomocí *HTTP* protokolu, o kterém bude pojednávat následující podkapitola. (ČEGAN, 2010)

#### 2.1.2 HTTP protokol (Hypertext Transfer Protocol)

*HTTP* protokol je sada pravidel dodržovaných mezi komunikujícími stranami (klient-server) tak, aby si tyto strany navzájem rozuměly. Jedná se o protokol aplikační vrstvy v modelu *RM OSI*. Protokol například definuje typ, množství dat, náležitosti požadavku, odpovědi atd. Současná verze protokolu je 1.1 a je popsána v *RFC*<sup>2</sup> 2616. Původně byl určen pro přenos pouze hypertextových dokumentů ve formátu *HTML*. V současnosti se používá pro přenos jakýchkoli typů souborů nebo objektů. Tento přenos souborů umožňuje rozšíření *MIME*<sup>3</sup>, které bylo dříve používané pouze pro emaily. *HTTP* je bezstavový protokol, tzn., že neudrhuje stav relace a každý požadavek považuje za unikátní. Vývoj protokolu *HTTP* je možné rozdělit do třech verzí. První verze protokolu *HTTP* je z roku 1991 a je to 0.9. Jednalo se o velice jednoduchý protokol, který nerozlišoval druh zasílaných dat. Obsahoval pouze jednu metodu, a to metodu *GET*. Následující verze je 1.0, který je doplněn o *HTTP* hlavičky, byly přidány nové metody a rozšíření o standard *MIME* pro přenos dalších typů souborů. Poslední verze je 1.1. Je rozšířena o trvalé spojení, tj. dokáže zpracovat více požadavků v rámci jednoho spojení. Podporuje virtuální hostitele s jednou IP adresou a umožňuje využívání *cachování*<sup>4</sup>. Protokol *HTTP* pracuje na principu požadavku a odpovědi. Klient zasílá požadavky webovému serveru prostřednictvím webového prohlížeče, který na ně odpovídá. Základní jednotkou komunikace pomocí protokolu *HTTP* je zpráva. Zprávy jsou bloky dat, nejčastěji strukturované sekvence bajtů. Zprávy jsou uvozeny meta-informacemi, které nám popisují obsah zprávy. Syntaxe *HTTP* zprávy je rozdělena do tří částí. První částí je Start line, pro požadavek říká „*co chci udělat*“, pro odpověď říká „*co se stalo*“. Dále následují hlavičky. Poslední částí je tělo zprávy, které například obsahuje *HTML* kód stránky.

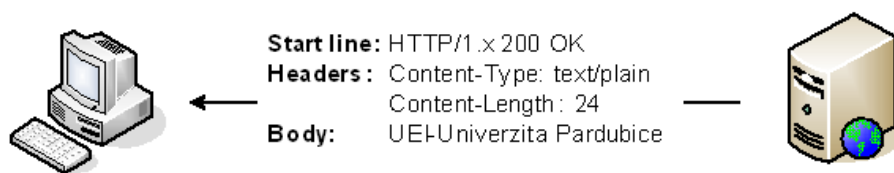
<sup>1</sup> Conseil Européen pour la Recherche Nucléaire (Evropská rada pro jaderný výzkum)

<sup>2</sup> Více na <http://www.faqs.org/rfcs/rfc2616.html>

<sup>3</sup> Více na <http://cs.wikipedia.org/wiki/MIME>

<sup>4</sup> Cachování dat je zjednodušeně uchovávání dat na pozdější použití. Je to elegantní metoda, pomocí které můžete zrychlit běh svých webových stránek.

Strukturu zprávy je možné lépe pochopit z následujícího obrázku, převzatého z přednášek předmětu *Správa web serveru* vyučovaného na Univerzitě v Pardubicích v mém studijním oboru.



Obrázek 1 - Struktura HTTP zprávy (ČEGAN, 2010)

### 2.1.3 Metody HTTP

Metoda určuje druh služby, kterou klient od serveru požaduje. Metoda se uvádí velkými písmeny. Server nemusí vždy všechny metody podporovat. Při dotazu nepodporovanou metodou pak vrací chybové hlášení.

V protokolu *HTTP* jsou následující metody:

- GET - nejběžnější metoda, používá se pro žádost o zaslání určitých dat,
- HEAD - stejná jako metoda GET s rozdílem, že se omezuje pouze na zaslání hlavičky *HTTP* protokolu, používá se pro zjištění typu požadovaného objektu (bez nutnosti získání dat), ověření existence požadovaného objektu (viz stavový kód odpovědi), testování, zdali byl objekt modifikován,
- POST - slouží k zaslání dat na server spolu s požadavkem, zasílaná data nejčastěji slouží jako vstupní data, která jsou předána procesu, který je zpracuje,
- OPTIONS - slouží k dotazu na možnou komunikaci se serverem na konkrétní URL, umožňuje klientovi určit možnosti a omezení spojené se zdroji a schopností serveru, pokud dotaz na URL je ve tvaru \*, pak se jedná o dotaz na server jako celek,
- PUT - slouží k zaslání a zapsání dat na server, data jsou zapsána na požadavkem specifikovanou URL, zapsání dat do souboru na server provádí přímo server nikoli externí aplikace,
- DELETE - slouží k požadavku na smazání určitého objektu na serveru, který je specifikován URL adresou,
- TRACE - slouží k analýze způsobu připojení, požadavek k serveru může putovat skrz proxy, firewall, gateway a další aplikace - každý z nich má možnost modifikovat *HTTP* hlavičku, metoda TRACE umožňuje klientovi vidět, jak vypadá *HTTP* hlavička, když jeho požadavek přijme webový server. (JAKEL, 2002)

### 2.1.4 Stavové kódy

Stavový kód je číslo, ke kterému přísluší stavové hlášení, které je zasláno klientovi spolu s odpovědí. Číslo vyjadřuje, do jaké míry se podařilo splnit klientův požadavek. Uvedu jeden z nejnámějších stavových kódů, a to *200 OK*. Tento kód nese informaci, že operace proběhla bez chyby a požadavek byl úspěšně zpracován.

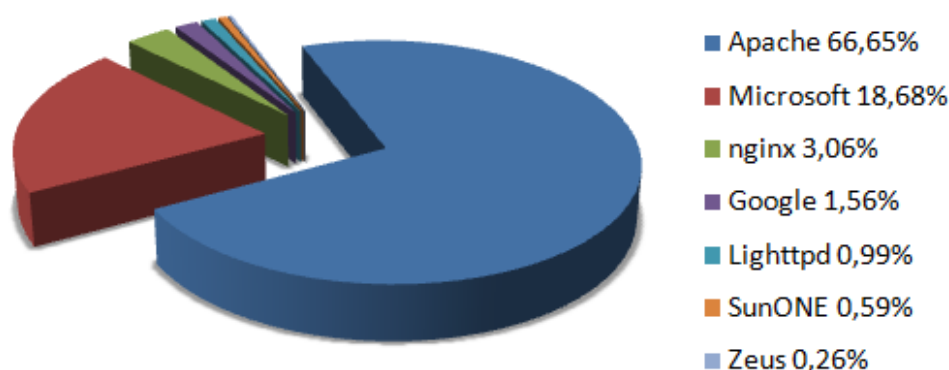
V *HTTP* je pět základních tříd stavových kódů:

- 100 až 199 – informační rozsah,
- 200 až 299 – úspěšný klientský požadavek,
- 300 až 399 – přesměrování,
- 400 až 499 – nekompletní klientský požadavek,
- 500 až 599 – chyba serveru. (JAKEL, 2002)

### 2.1.5 HTTP server

*HTTP* server neboli webový server je vlastně softwarová aplikace běžící jako úloha (démon) na skutečném hardwarovém serveru. Tato aplikace zpracovává a odpovídá na požadavky pomocí protokolu *HTTP*. Typicky naslouchá na *TCP* portu 80 nebo pro *HTTP* přes *SSL/TLS* na portu 443. Mezi nejznámější webové servery patří bezpochyby Apache Server od firmy Apache Software Foundation<sup>5</sup>. Existují i další webové servery od firem Microsoft nebo Sun a dalších.

Poměry využití webových serverů podle news.netcraft.com k dubnu 2010 ukazuje následující graf v obrázku číslo 2. (NETCRAFT, 2010)



Obrázek 2 - Procentuální poměry využití webových serverů k dubnu 2010

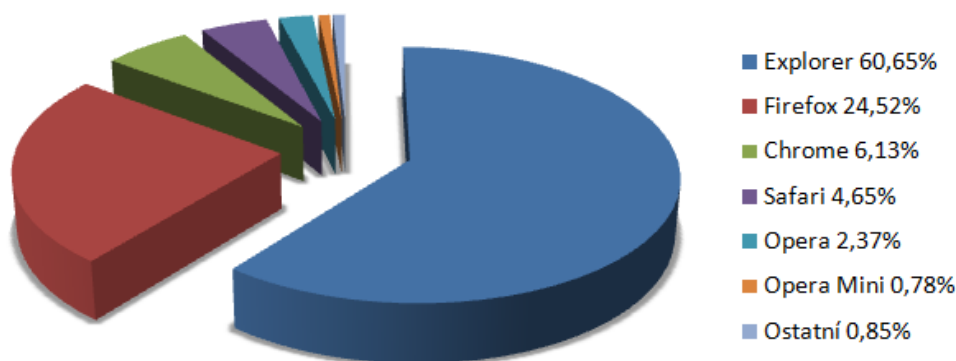
### 2.1.6 HTTP klient

Standardním klientem pro službu *HTTP* je tzv. webový prohlížeč (web browser). Webový prohlížeč je program, který odesílá požadavky webovému serveru a zobrazuje webové stránky vrácené webovým serverem. Jedná se o tzv. tenkého klienta, který kromě zmíněné prezentační funkce umí provádět i například skripty napsané ve skriptovacích programovacích jazycích jako je *JavaScript*<sup>6</sup>.

Tržní podíly webových prohlížečů podle maxiorel.cz k březnu 2010 ukazuje následující graf v obrázku číslo 3. (HÁJEK, 2010)

<sup>5</sup> Více na <http://www.apache.org/>

<sup>6</sup> Více na <http://www.jakpsatweb.cz/javascript/>



Obrázek 3 - Procentuální podíly webových prohlížečů k březnu 2010

## 2.2 Technologie a jazyky pro vývoj webových aplikací

### 2.2.1 Značkovací jazyky

Pro sestavení stránky se používá kód, který je tvořen tzv. značkami. Tento kód určuje, jak bude stránka v prohlížeči vypadat. Na vývoji a standardizaci se podílejí jednak vývojáři nejznámějších webových prohlížečů, tak i konsorcium *W3C*<sup>7</sup>, které vzniklo za účelem nutnosti standardizace na poli značkovacích jazyků. *W3C* je rovněž autorem známého validátoru<sup>8</sup> webových stránek. Validita je jednou ze známek kvalitních stránek. Mezi nejznámějšími značkovacími jazyky patří *HTML*, *XHTML* a *XML*.

#### HTML

Hyper Text Markup Language v překladu znamená hypertextový značkovací jazyk. Byl založen v roce 1990 spolu s protokolem *HTTP*, jak již bylo psáno v předchozí podkapitole. Jedná se o značkovací jazyk založený na *SGML* (Standard Generalized Markup Language)<sup>9</sup>. V roce 1999 byla vydána poslední verze 4.01, ve které se objevily tehdy ještě neznámé možnosti, jako například podpora kaskádových stylů, o kterých bude níže ještě psáno, nebo vkládání multimediálních objektů. Nástupcem jazyka *HTML* je jazyk *XHTML*. (SCHURMAN, 2000)

#### XHTML

*XHTML* je zdokonalená verze *HTML*. Písmeno *X* na začátku znamená eXtensible, což v překladu znamená rozšiřitelný. Ve skutečnosti jde ale spíše o zúžení jazyka *HTML*, vynechány jsou některé nepotřebné definice. Některé rozdíly *XHTML* oproti *HTML*:

- tagy a atributy jsou malými písmeny,
- nepárové tagy končí lomítkem,
- párové tagy jsou párové povinně,
- všechny atributy musejí mít hodnotu,
- dokument požaduje správný doctype. (SNÍŽEK, 2002)

<sup>7</sup> Více na <http://www.w3.org/>

<sup>8</sup> Validátor je služba nebo nástroj, který slouží ke kontrole správnosti kódu *HTML*, např. <http://validator.w3.org/>

<sup>9</sup> *SGML* je universální značkovací jazyk určený k tvorbě dalších jazyků jako svých podmnožin.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html;
  charset=UTF-8" />
  <title>Insert title here</title>
</head>
<body>
</body>
</html>

```

**Syntaxe 1 - deklarece XHTML dokumentu**

## XML

Opravdovou budoucností s největšími možnostmi se může pyšnit a určitě pyšní jazyk *XML*, který byl vyvinut konsorciem *W3C*. Je vhodný pro snadnou tvorbu jiných značkových jazyků. Umožňuje například tvorbu vlastní značek. Jazyk je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů, u kterých popisuje strukturu z hlediska věcného obsahu. *XML* se tedy nezabývá se vzhledem, i když vzhled může být definován pomocí kaskádových stylů. Další možností zpracování je transformace do jiného typu dokumentu, nebo do jiné aplikace *XML*. Jako příklad mohu uvést miniaplikace pro Google<sup>10</sup>, které hojně jazyk *XML* využívají. (KOČÍ, 2000)

### 2.2.2 Skriptovací jazyky

Skriptovací jazyky mají své nedílné použití v tak zvaných dynamicky generovaných stránkách. Skriptovací jazyky je možné rozdělit do dvou skupin, podle toho zda se skript vykonává na straně serveru nebo na straně klienta.

## JavaScript

Pokud mluvíme o skriptech na straně klienta, tak se jedná o skripty, které reagují na nějaké podněty uživatele. Mezi tyto typy skripty řadíme hlavně *JavaScript*. Jedná se o objektově orientovaný jazyk. Podněty, na které reagují v tomto příkladě Java skripty, nazýváme typicky událostmi. Tyto události mohou být například *onclick*<sup>11</sup> nebo *onchange*<sup>12</sup>, ale i další. Jsou definovány v požadovaných elementech, které tvoří stránku. Pokud je vykonána tato událost, je zavolána příslušná javaskriptová funkce, která vykoná požadovanou akci. Pro používání *JavaScriptu* je nutné mít v prohlížeči zapnutou jeho podporu. To v některých případech není možné. Proto pokud vytváříme webové aplikace, snažíme se, aby jejich funkčnost nebyla omezená v případě vypnutého *JavaScriptu* v prohlížeči.

<sup>10</sup> Asi nejznámější nyní už nejenom vyhledávač, ale i spousta dalších služeb na [www.google.cz](http://www.google.cz).

<sup>11</sup> Událost *onclick* je zaznamenána v případě kliknutí na element, kde je definována. Po kliknutí se vykoná požadovaná akce. Nejčastěji nějaká funkce, která vyvolá nějakou vizuální změnu bez znovunačtení stránky.

<sup>12</sup> Událost *onchange* se liší pouze způsobem zaznamenání akce. Využívá při změně obsahu v rolovacím menu.



## PHP

V případě skriptů na straně serveru, je odeslán nejprve požadavek od klienta. Na serveru jsou pak uloženy skripty, které tento požadavek zpracují a vrátí výsledek. Zjednodušeně můžeme říci, že klientovy je zaslána odpověď v podobě *HTML* kódu pro sestavení stránky. Jedná se zde o výše popisovaný model klient - server. Mezi skriptovací jazyky na straně serveru, které jsou nejvíce využívány, řadíme hlavně *PHP* nebo konkurenční *ASP*<sup>13</sup>. *PHP* je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktury jazyka *HTML* či *XHTML*, což lze využít při tvorbě webových aplikací. Syntaxe jazyka je odvozena z jazyka *C* a *Java*. Jazyk je velmi oblíbený i pro začátečníky, kdy začátečník je ve velmi krátké době schopen zvládat i složitější úlohy. Nejnovější verze jazyka *PHP 5* jsou již vyspělé objektově orientované jazyky pro tvorbu webových aplikací.

### 2.2.3 Kaskádové styly

Kaskádové styly (Cascading Style Sheets, proto zkratka *CSS*) jsou jazyk pro popis způsobu zobrazení stránek napsaných v jazycích *HTML*, *XHTML* nebo *XML*. Tento jazyk byl opět navržen konsorciem *W3C*. Jazyk byl vydán v několika verzích. Současná používaná revize je 2.1, pracuje se na dokončení revize 3. Hlavním smyslem *CSS* je oddělení implementace vzhledu dokumentu od implementace struktury a obsahu. Původně to měl umožnit už pouze jazyk *HTML*, ale v jeho vývoji došlo k dříve neočekávaným okolnostem, které to nakonec neumožnily. Vliv měl i konkurenční boj výrobců webových prohlížečů. (Kaskádové styly, 2010) (BŘÍZA, 2004)

Dnes si již bez kaskádových stylů návrh webových stránek nedokážeme představit. Nejenže ulehčují práci, ale bez nich by některé vzhledové vlastnosti prostě nešly ani vytvořit. Ulehčení práce spočívá například v tom, že si nadefinujeme určitý styl jako třídu. Tuto třídu pak použijeme u elementů v *HTML* dokumentu, u kterých chceme mít definovaný tento styl. Čili styl nadefinujeme jednou, ale použijeme ho vícekrát. Při použití *CSS* se můžeme potýkat s odlišnými zobrazeními v různých prohlížečích. I když se situace již zlepšuje, někdy se to může stát. Ale i proti tomu se dá bránit buď podmínkami přímo v kaskádových stylech, nebo pouhým načítáním jiných souborů se styly pro každý prohlížeč.

### 2.2.4 Jazyk SQL a databázové systémy

Jazyk *SQL* je databázový programovací jazyk. Jazyk *SQL* má více úloh. Pokud o něm mluvíme jako o programovacím jazyku, máme na mysli například začlenění kódu jazyka *SQL* do jiných aplikací. Pokud o něm hovoříme jako o interaktivním dotazovacím jazyku, jedná se o použití pro uživatele při získávání nebo zadávání vlastních dat z nebo do databáze. Pokud o něm mluvíme jako o administračním databázovém jazyku, jedná se o použití při vytváření struktury konkrétní databáze. První standardizace byla v roce 1986 a jazyk dostal název *SQL86*. V dalších letech byly postupně odstraňovány nedostatky a v roce 1992 vznikl nový

---

<sup>13</sup> ASP (Active Server Pages) je technologie vyvinutá společností Microsoft. Jedná se o druhou nejvíce využívanější platformu pro tvorbu dynamických webových stránek.

standard *SQL92* (zkráceně *SQL2*). Zatím nejnovějším standardem je *SQL3* z roku 1999, který reaguje na potřeby nejmodernějších databází s objektovými prvky. V následujících podtitulech se seznámíme s nejpoužívanějšími databázovými systémy.

## MySQL

*MySQL* je multiplatformní databázový systém. Komunikace s ním probíhá pomocí jazyka *SQL*. Je k dispozici jak pod bezplatnou licenci, tak pod komerční placenou licenci. Pro svou snadnou implementovatelnost, výkon a především díky tomu, že se jedná o volně šiřitelný software, má vysoký podíl na v současné době používaných databázích. Velmi oblíbená a často nasazovaná je kombinace Linux, *MySQL*, *PHP* a Apache jako základní software webového serveru („technologie LAMP“). *MySQL* bylo od počátku optimalizováno především na rychlost, a to i za cenu některých zjednodušení. Až donedávna nepodporovalo pohledy, trigger, a uložené procedury. Tyto vlastnosti jsou doplňovány teprve v posledních letech. (PONKRÁC, 2004)

## Oracle

Oracle je moderní multiplatformní databázový systém s pokročilými možnostmi zpracování dat a vysokým výkonem. Oficiální název databázové platformy je Oracle Database. Aktuální verze je Oracle Database 11g. Tento systém podporuje standardní relační dotazovací jazyk *SQL92*, programovací jazyk *PL/SQL* (procedure language) rozšiřující možnosti vlastního původního databázového jazyka. V tomto rozšiřujícím jazyce je možné tvořit procedury, uživatelské funkce, programové balíky nebo trigger. Jedná se o špičku na poli databázových systémů. (Oracle, 2010)

## Postgre SQL

*PostgreSQL* je relační databázový systém podporující *SQL92* a *SQL99* normu. Podporuje všechny obvyklé databázové funkce jako např. komplexní dotazy, cizí klíče, trigger, pohledy, transakce, vlastní datové typy, agregační funkce nebo procedury. Z hlediska programátora je výborná možnost přístupu z mnoha jazyků C/C++ (knihovny libpq a libpq++), scriptovacích jazyků Perl (pgsql\_perl5), Python (PyGreSQL), ale i třeba javy (jdbc) a *PHP*. Ve srovnání s *MySQL* vyniká především spoustu vymožeností, které *MySQL* nezvládá. Naproti tomu *MySQL* je nepoměrně rychlejší a na menší webové aplikace více, než dostatečná. (OLŠAVSKÝ, 2004)

## 2.3 Webová aplikace

Webové aplikace jsou softwarové aplikace poskytované uživatelům prostřednictvím sítě internet nebo vnitropodnikové sítě intranet. Mezi největší výhody webových aplikací patří možnost rozšiřitelnosti. Webové aplikace pomocí protokolu *HTTP* přijímají požadavky. Tyto požadavky následně zpracují a vytvoří například *HTML* výstup. *WWW* stránky jsou tedy dynamicky generovány. Webové prohlížeče, na kterých lze naprogramované aplikace zobrazit a provozovat, jsou dnes již na všech ať už podnikových nebo domácích počítačích. Kromě rozšiřitelnosti je i další výhodou to, že se pro funkčnost aplikace nemusí instalovat další software do cílového počítače. V minulosti, kdy ještě nebyly možnosti vyvíjet složité

a vyspělé webové aplikace, byly aplikace vyvíjeny jako například 32bitové aplikace určené pro osobní počítače s nutností instalovat tyto aplikace jako software. Každá aktualizace s sebou nesla nutnost aktualizovat každý cílový počítač, kde se nacházela zastaralá verze těchto software. Pokud jde o webovou aplikaci, tak je každá aktualizace prováděna jen jednou, a to na straně serveru bez nutnosti aktualizace v cílových počítačích. Z toho vyplývá, že se jedná i o velmi levné řešení. (Webová aplikace, 2010)

Webové aplikace jsou používány pro řešení mnoha informačních systémů. Pro příklad uvedu známé systémy na vysoké úrovni, se kterými se jistě každý již setkal. Jsou to například různá CMS řešení blogovacích a redakčních systémů, velmi dobře známé fórum *PHPBB*<sup>14</sup> atd.



Obrázek 4 - Schéma webové aplikace

Architektura webových aplikací je založena na modelu klient-server. Na obrázku číslo 4 je vidět obvyklý třívrstvý model webové aplikace. V první prezenční formě je počítač s webovým prohlížečem a přístupem na internet nebo intranet. Střední tzv. logickou vrstvou je software pro dynamické generování *WWW* stránek (například *PHP* nebo *ASP*). Poslední třetí tzv. datovou vrstvou je databázový server, kde jsou ukládána data v tabulkách. Další možností je ukládat data v souborech.

## 2.4 Bezpečnost webové aplikace

Velmi důležité je, aby aplikace byla bezpečná a chráněna proti tzv. útokům. Zde je krátký přehled nejčastějších typů útoků na webové aplikace:

- *SQL* injection - využívá špatného ošetření vstupu od uživatele při sestavování dotazů,
- krádeže session - případ kdy je získán identifikátor session a následně předstíráno korektní přihlášení,
- Cross-site scripting - útočník dokáže do stránek podstrčit svůj vlastní nebezpečný javascriptový kód,
- Cross-site Request Forgery (*CSRF*) - pracuje na bázi neočekávaného požadavku pro vykonání určité akce v aplikaci; využívá akce uživatelů; kteří jsou k ní již v okamžiku útoku přihlášení (FRESCHMANN, 2008)

<sup>14</sup> Více na <http://www.phpbb.cz/>

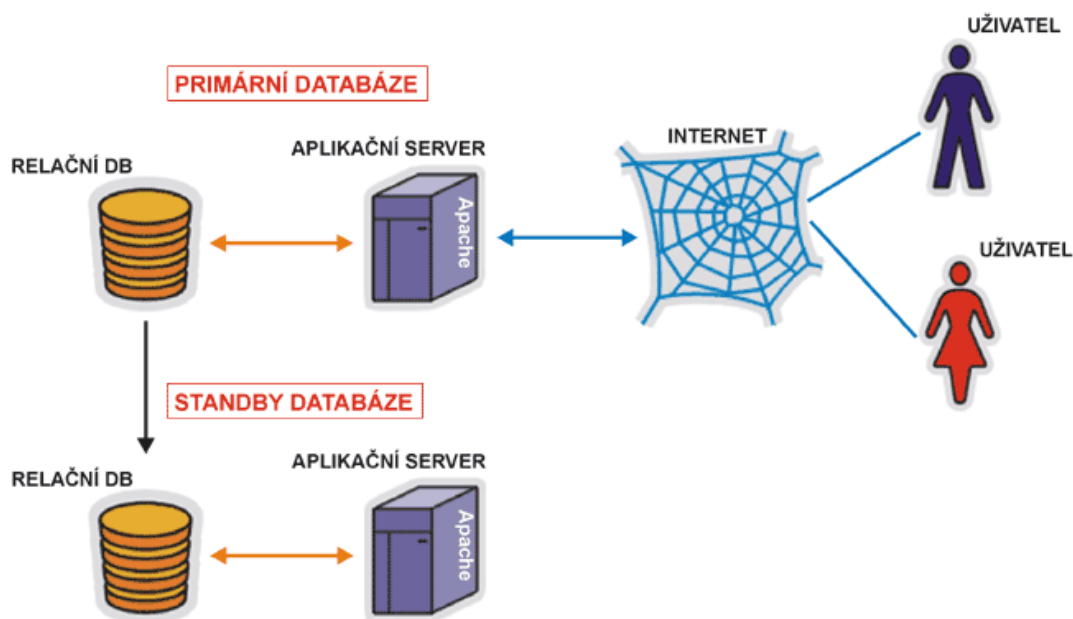
### 3 Problematika knihovních systémů

#### 3.1 Knihovní systémy obecně

Knihovní systémy jsou velmi složité a propracované aplikace. Poskytují mnohem lepší kontrolu nad veškerou agendou spojenou s knihovnou. Knihovníkům pomáhají efektivně zpracovávat úkoly spojené s knihovním fondem, zaslouží se o lepší kontrolu nad výpůjčkami a rezervacemi. Čtenářům umožňují pomocí sítě internet například online přístup:

- ke knihovnímu fondu,
- ke svému účtu v knihovně,
- ke svým výpůjčkám a rezervacím,
- k dalším službám knihovny.

Na obrázku níže je vidět, že knihovník přistupuje k databázi knihovního fondu přímo například přes nějakou 32 bitovou aplikaci pro systém Windows. Touto aplikací se spravuje veškerá administrátorská a knihovní činnost. Naopak čtenář přistupuje ke svému účtu ale i k fondu knihovny přes webové rozhraní, které v základní podobě generuje a aktualizuje právě 32 bitová aplikace samotného knihovního systému. Vždy je generován je minimálně online katalog knihovny.



Obrázek 5 - Obvyklé schéma knihovního systému

## 3.2 Obvyklé moduly knihovních systémů

Protože jsem měl možnost blíže se seznámit na Univerzitě v Pardubicích s knihovním systémem *DAIMON*<sup>15</sup>, budu obvyklé moduly knihovních systémů popisovat právě na tomto systému.

### 3.2.1 Akviziční modul

Akviziční modul slouží pro evidenci a správu:

- titulů, které chce knihovna získat do svého fondu,
- dodavatelů,
- administrativních dokumentů (faktury, objednávky, reklamace),
- rozpočtů knihovny.
- s dodavatelem podporuje písemný kontakt nebo komunikaci prostřednictvím elektronické pošty
- záznam v tomto modulu slouží jako základ bibliografického záznamu.

### 3.2.2 Katalogizační modul

Katalogizační modul umožňuje spravovat tituly v knihovním systému. Umožňuje zpracovávat bibliografické záznamy pro všechny druhy dokumentů, a to i pro elektronické dokumenty. Pokud jde o druh dokumentu, jsou v tomto modulu nabízeny standardní šablony pro katalogizaci titulů. Například elektronický dokument nebude mít při vkládání vlastnost „tip vazby“ atd. Dále tento modul umožňuje provádět vyhledávání. To může být jednoduché nebo složitější s použitím například booleovských operátorů. Samozřejmostí je i fulltextové vyhledávání. Další důležitou funkcí, které tento modul nabízí je, možnost exportu nebo importu vybraných dat. V rámci tohoto modulu je zajišťována a zaznamenávána kompletní historie při manipulaci se záznamem (vlození, editace, kdo a kdy, atd.)

### 3.2.3 Výpůjční modul

Výpůjční modul zajišťuje evidenci a správu čtenářů knihovny, výpůjček, rezervací, upomínek, poplatků a pokut čtenářů. Pokud má čtenář vůči knihovně nějaké pohledávky stále nesplacené již po několika upomínkách, je systém schopen zablokovat jeho konto do doby, než čtenář všechny pohledávky vyrovná. V rámci tohoto modulu lze tisknout různá potvrzení o výpůjčkách nebo platbách. Komunikace je zajištěna pomocí elektronické pošty i prostřednictvím SMS na mobilní telefon čtenáře.

### 3.2.4 Modul OPAC

Díky tomuto modulu jsou čtenářům zpřístupněny všechny funkce, které jsou pro ně určeny. Přístup je zajištěn pomocí webové aplikace ať už v síti internet nebo intranet. Přístup je rozdělen do dvou kategorií:

---

<sup>15</sup> Více na <http://www.sefira.cz/>

- volný přístup pro širokou veřejnost,
- registrovaný přístup pro registrované a přihlášené uživatele.

V rámci volného přístupu lze využít například stejně jako v katalogizačním modulu možnosti jednoduchého nebo složitějšího vyhledávání. Funkce vyhledávání se příliš neliší v těchto modulech. Po nalezení výsledků návštěvník může zobrazit různé dativy titulu a jeho exemplářů. V rámci registrovaného přístupu si uživatel zobrazuje stav svého konta, stav výpůjček a rezervací, může měnit některé detaily svého účtu např. osobní údaje atd.

### 3.2.5 Správa periodik

S periodiky je v systému nakládáno stejně jako s ostatními tituly, ale v některých částech jsou odlišnosti. Jsou vyžadovány speciální operace, které tento modul má na starosti. Jsou to například:

- evidence předplatného,
- evidence dodaných čísel,
- urgency nedodaných čísel.

### 3.2.6 Z39.50 klient server

Díky protokolu Z39.50 je knihovna schopna komunikovat s ostatními vzdálenými knihovnami, s jejich čtenáři nebo i s dalšími institucemi. Tyto subjekty na „druhé straně“ ovšem také musejí podporovat tento protokol. Samozřejmostí je i opačný přístup knihoven a institucí.

### 3.2.7 Modul pro správu uživatelů

Tento modul umožňuje do systému přidávat nové uživatele nebo editovat jejich stávající údaje nebo práva. Vkládání mohou být administrátoři, knihovníci a čtenáři.

## 3.3 Platné standardy knihovních systémů

- norma *ISO2709* - formát pro výměnu bibliografických informací na magnetické pásce,
- UNIMARC, UNIMARC/Autority (ohled je brán také na formát *MARC21*) – formát pro výměnu bibliografických a autoritních údajů ve strojově čitelné podobě,
- *XML* - interní formát pro ukládání záznamů,
- AACR2R - angloamerická katalogizační pravidla,
- doporučení *ISBD* – mezinárodní standard pro bibliografický popis jednotlivých druhů dokumentů,
- protokol Z39.50 – standardizovaný protokol pro vyhledávání a přejímání dat,
- UNICODE - mezinárodní standard pro kódování znaků umožňující podporu vícejazyčných textů a konzistenci při výměně dat,
- CODE39 - formát čárových kódů.

## 3.4 Analýza dostupných řešení na trhu v oblasti knihovních systémů

### 3.4.1 DAIMON

Protože už jsem začal popisovat nejčastější knihovní moduly na systému *DAIMON*, začnu s představením právě tohoto knihovního systému. Tento systém vyvinula česká společnost *SEFIRA*. Tato společnost je nyní už velmi úspěšná nejen na poli informačních systémů různých typů, ale i hotových IT či business řešení. Její informační systémy jsou založeny na databázi *ORACLE*, což patří k hlavním výhodám těchto systémů. Systém *DAIMON* patří ke špičce knihovních systémů v České Republice, i když je v současnosti již nahrazován vyspělejšími a novějšími zahraničními systémy.

Mezi hlavní přednosti patří:

- využití nejmodernějších technologií, například databáze *ORACLE*,
- úplná podpora standardů,
- protokol Z39.50 (klient, server),
- možnost zpracování všech druhů dokumentů na všech úrovních,
- vysoká míra nastavitelnosti dle potřeb uživatele,
- nezávislost na platformě (MS Windows, Linux),
- architektura klient/server,
- ukládání dat do centrální relační databáze,
- zajištění konzistence a ochrany dat,
- výkonnost při zpracovávání velkého objemu dat,
- efektivní a rychlé vyhledávání (strukturované i fulltextové - vč. ohýbání slov),
- možnost správy více knihoven (poboček) v rámci jednoho systému,
- perspektiva dalšího vývoje. (JURÍK, 2002)

### 3.4.2 Knihovní systém Clavius

Dalším knihovním systémem, se kterým jsem měl možnost se seznámit, je systém *CLAVIUS*. Jak již bylo v předchozí kapitole v části poděkování uvedeno, byla mi zapůjčena demoverze tohoto knihovního systému od samotných vývojářů. Tato demoverze nesla jistá omezení, a to v počtu dostupných modulů tohoto systému. Nejdůležitější moduly, které byly dostupné v mé demoverzi:

- katalogizace knih, map, hudebnin,
- vyhledávání a rešerše pro knihovníky,
- výpůjční protokol,
- evidence periodik, kompletů a brožur.

Tento systém vyvinula firma *LANIUS*. Její předchozí knihovní systém pro OS *DOS* nesl stejný název. Mohlo by se zdát, že systém *CLAVIUS* je pouze zdokonalená verze pro systémy *WINDOWS*. Firma se však proti této domněnce vymezuje a zdůrazňuje, že

system CLAVIUS je odlišný, využívá modernější technologie a odstraňuje nedostatky, které měl jejich prvotní knihovní systém.

Přednosti a charakteristiky systému:

- neomezená délka vstupních polí všech údajů,
  - ukládání všech druhů dokumentů do jedné společné databáze,
  - připravenost na změny pravidel,
  - podpora Evropské měny Euro,
  - několik desítek předdefinovaných šablon vstupních/výstupních formulářů s možností úpravy metodou drag and drop pomocí návrháře,
  - široké možnosti vyhledávání všech druhů dokumentů a tvorby rešerší,
  - využití technologie čárkových kódů ve všech modulech při manipulaci s dokumenty,
  - možnost zaslání písemných oznámení pomocí elektronické pošty přímo ze systému,
  - WWW modul pro ON-LINE prezentaci dat v síti Internet nebo na vlastním Intranetu,
  - podpora všech hlavních knihovnických standardů.
- (Clavius – základní informace, 2001)

### 3.4.3 ALEPH

Tento knihovní systém patří mezi špičku. Dá se říct, že to je nejmodernější a nejnovější systém. Vyvinula jej izraelská společnost EX LIIBRIS. Systém je převážně určen pro největší knihovny. Lze jej použít i pro menší knihovny. Nabízí flexibilní volbu modulů a tím umožňuje zvolit si systém „na míru“.

Mezi další jeho hlavní výhody patří:

- komplexnost - jeden integrovaný systém obsluhuje všechny aspekty Vaší knihovny, jak z pohledu zaměstnanců, tak z pohledu čtenářů, WWW a grafický OPAC, výpůjční proces, katalogizaci, správu seriálů, akvizici, meziknihovní službu a správu dat, včetně statistických analýz,
- otevřenost - rozhraní, propojitelné s jinými systémy a databázemi, nabízí široké možnosti sdílení zdrojů,
- systém není omezený velikostí databáze,
- rozšiřitelnost - horizontálně a vertikálně strukturovaná architektura Vám umožní bez námahy rozšiřovat jak hardware, tak software,
- spolehlivost - několik generací systému ALEPH, vyvíjených po dvě desetiletí, si získalo spokojené zákazníky ve více než 500 instalacích ve 40 zemích,
- snadná konverze dat - stovky knihoven využily rychlou a spolehlivou konverzi dat při přechodu z nejrůznějších datových formátů a systémů do systému ALEPH,
- lokalizace - systém má schopnosti komunikace ve 20 jazycích, zápisu různých znakových sad a levo- i pravostranného zápisu údajů. (Aleph – hlavní rysy, 2010)



### 3.4.4 KpWin SQL

Další významný integrovaný knihovní systém, který zde budu, popisovat se jmenuje *KpWinSQL*. Je to poměrně nový a moderní systém. Vyvinula jej firma KP-SYS spol. s r.o. se sídlem v Pardubicích. Je vhodný pro všechny typy knihoven díky své vysoké možnosti konfigurovatelnosti a flexibilitě. Systém je postaven na databázi Firebird nebo na databázi Borland Interbase. V současné době se pracuje na rozšíření o databáze *Oracle* a Microsoft *SQL* server. Aplikace je určena pro systémy Windows. Server je schopný běžet na více operačních systémech, například: Linux, Unix, Solaris, FreeBSD(FB), HP-UX(FB), MacOS X(FB).

Další vlastnosti:

- architektura - jednouživatelské verze využívají vložený *SQL* server (pouze FB), více uživatelské verze vyžadují nainstalovaný server FB nebo IB. Není problém, aby byl nainstalovaný na jednom PC klient i server,
  - Z39.50 Client - integrovaný již v základní verzi, automatické použití Z39.50 při zakládání záznamů,
  - libovolný formát dat - v základním nastavení formát *MARC21*, čísla a názvy polí jsou plně definovatelné,
  - správa více poboček v centrální databázi - všechny verze mohou pracovat ve více pobočkovém, nebo regionálním režimu,
  - tiskové sestavy - program obsahuje vizuální návrhář tiskových sestav,
  - moduly - program obsahuje online *WWW* katalog, katalogizaci, výpůjční modul i akvizici, k dispozici jsou i moduly pro spolupráci s externími databázemi/hesláři (PSH, MeSH),
  - přehlednost – intuitivnost a systém záložek uživateli usnadňuje přehled v programu,
  - možnosti nastavení - program umožňuje pomocí parametrů v ini souboru, databázi a registrech nastavit program přesně dle vašich potřeb.
- (KpWin SQL – charakteristika programu, 2010)

### 3.5 Zhodnocení

Jak jsem měl možnost poznat, knihovní systémy jsou velmi složité aplikace vyvíjené v širokém týmu odborníků po delší dobu. Měl jsem možnost se i detailně seznámit se systémy *Daimon* a *Clavius*. Nejvíce jsem se zaměřil na systém *Daimon*, protože v něm mám díky univerzitě účet a mohl jsem tak plně poznat možnosti tohoto systému z hlediska čtenáře. Zpracování online katalogu tohoto systému je dle mého soudu velmi intuitivní a podařené.

V systém *Clavius* jsem měl možnost využít demoverze s určitými omezeními. I tak jsem mohl nahlédnout do 32bitové aplikace tohoto systému a inspirovat se některými důležitými vlastnostmi které by v aplikaci neměly chybět. Byly to převážně různé evidenční údaje titulů nebo exemplářů apod.

## 4 Vývoj aplikace knihovního systému

Tato kapitola bude pojednávat o samotném návrhu a tvorbě konkrétního informačního systému knihovny.

### 4.1 Analýza a stanovení cílů aplikace

Cílem navrhované webové aplikace je vytvoření informačního systému knihovny na úrovni obecní knihovny. Původní záměr bylo vytvořit informační systém pro určitou obecní knihovnu. Z tohoto záměru sice sešlo, avšak cíle projektu se nezměnily.

Systém bude návštěvníkům umožňovat vyhledávat tituly, autory a prohlížet veškerý fond knihovny. Každý titul a autor bude mít v systému svoji jedinečnou přátelskou webovou adresu, důležitou pro *SEO*<sup>16</sup> optimalizaci. Návštěvníkům bude umožněna registrace a vytvoření tak svého osobního účtu v systému. Po aktivaci tohoto účtu knihovníkem nebo administrátorem bude uživateli umožněno využívání veškerých služeb knihovny. Aktivace bude předpokládána po ověření osobních údajů a zaplacení příspěvku na pobočce knihovny.

Uživatel bude moci rezervovat pouze již vypůjčené exempláře titulů, tak jak to je ve většině knihovních systémů popisovaných v této bakalářské práci. Uživatel bude mít svoji osobní stránku, kde bude moci spravovat své rezervace a zobrazovat své výpůjčky. Bude moci editovat informace ve svém profilu nebo měnit svoje přístupové heslo.

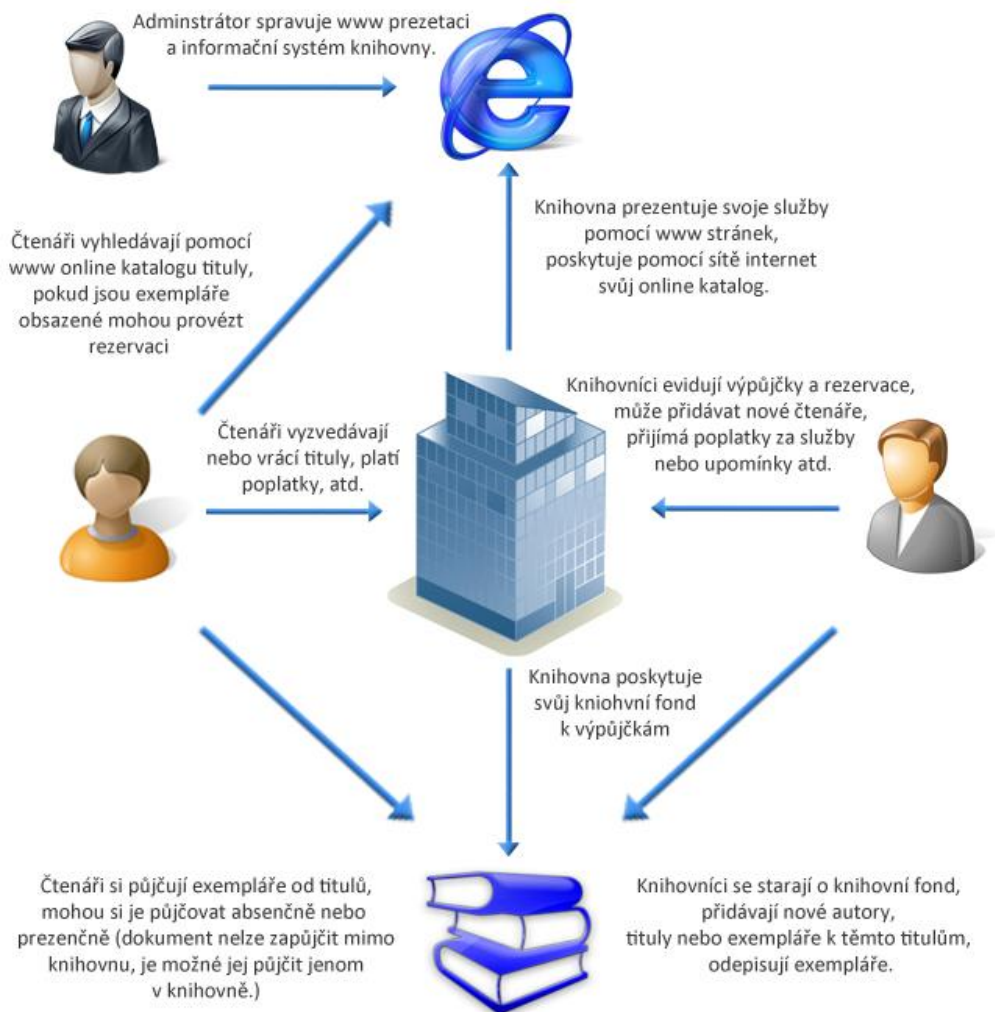
Administrátor systému bude mít největší práva. Bude moci zobrazovat v administraci všechny uživatele. Bude moci vkládat aktuality do systému, které se budou zobrazovat v boxu na stránce. Knihovník bude mít kromě výše uvedených funkcí dostupné veškeré další funkce jako administrátor. Bude moci spravovat výpůjčky, rezervace, tituly a jejich exempláře. Zobrazovat bude moci pouze čtenáře.

### 4.2 Rich Picture

K základní definici a pochopení všech nutných činností knihovny je potřeba si uvědomit co vše má s knihovnou něco společného a kdo všechno využívá služeb knihovny. K tomuto účelu docela dobře poslouží tak zvaný *Rich picture*. Tento náčrt vyjadřuje situace a problémy, které mohou při každodenním využívání knihovny nastat. Je v něm použito jednoduchých diagramů. Jak je patrné z obrázku číslo 6, v aplikaci se uplatní tři uživatelské role. Jsou to administrátor, knihovník a čtenář. Jednotlivá práva a možnosti uživatelských rolí jsou uvedeny v tabulce na rozdělení rolí a jejich práv.

---

<sup>16</sup> *SEO* (Search Engine Optimization, optimalizace pro vyhledávače) je metodika vytváření a upravování webových stránek takovým způsobem, aby jejich forma a obsah byly vhodné pro automatizované zpracování v internetových vyhledávačích.



Obrázek 6 - Rich Picture

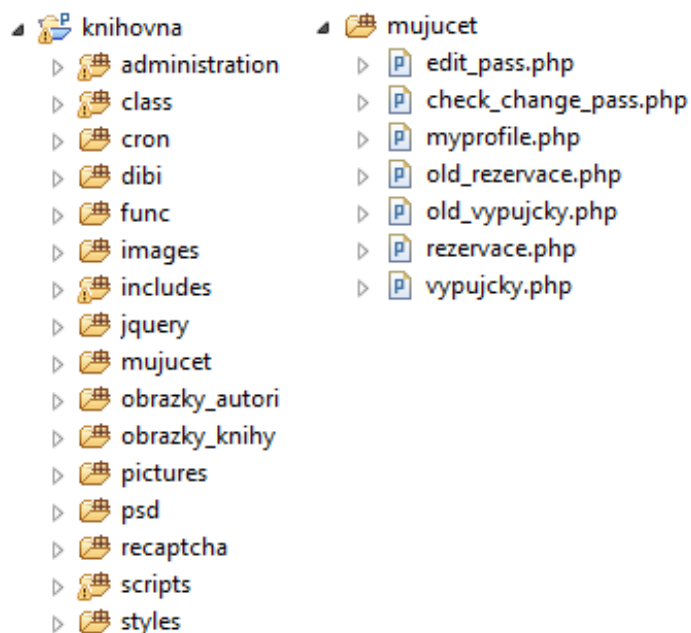
### 4.3 Tabulka rozdělení rolí a jejich práv

Knihovna je koncipována tak, aby také administrátor mohl obsloužit v případě absence knihovníka úkony spojené s knihovním fondem, rezervacemi nebo výpůjčkami. Administrátor může přidávat jakoukoliv osobu a aktuality. Knihovník může přidávat, co se týče osob pouze čtenáře. Pokud provede registraci nový zákazník, a potvrdí ji pomocí emailu, může se přihlásit a má právo si například prohlížet tituly. Nemůže ovšem rezervovat tituly nebo zamlouvat výtisky. Musí vyčkat aktivace od administrátora nebo knihovníka. Tato aktivace je v praxi zpravidla provedena po ověření osobních údajů na pobočce knihovny, popřípadě po zaplacení určité nutné částky, například členského příspěvku nezbytného k využívání služeb knihovny.

**Tabulka 1 – Rozdělení rolí a práv**

typ akce / role	administrátor	knihovnik	čtenář
přidávání osob	ano	pouze čtenáře	žádné
úprava osob	ano	sebe a čtenáře	pouze sebe
přidávání titulů	ano	ano	ne
úprava titulů	ano	ano	ne
přidávání exemplářů	ano	ano	ne
úprava exemplářů	ano	ano	ne
výpis osob	ano	pouze čtenáře	ne
přidávání aktualit	ano	ne	ne
úprava aktualit	ano	ne	ne
přidávání výpůjček	ano	ano	ne
rušení výpůjček	ano	ano	ne
výpis rezervací	ano	ano	pouze svých
výpis výpůjček	ano	ano	pouze svých
rušení rezervací	ano	ano	pouze svých
rušení výpůjček	ano	ano	ne
správa naplánovaných úloh	ano	ne	ne

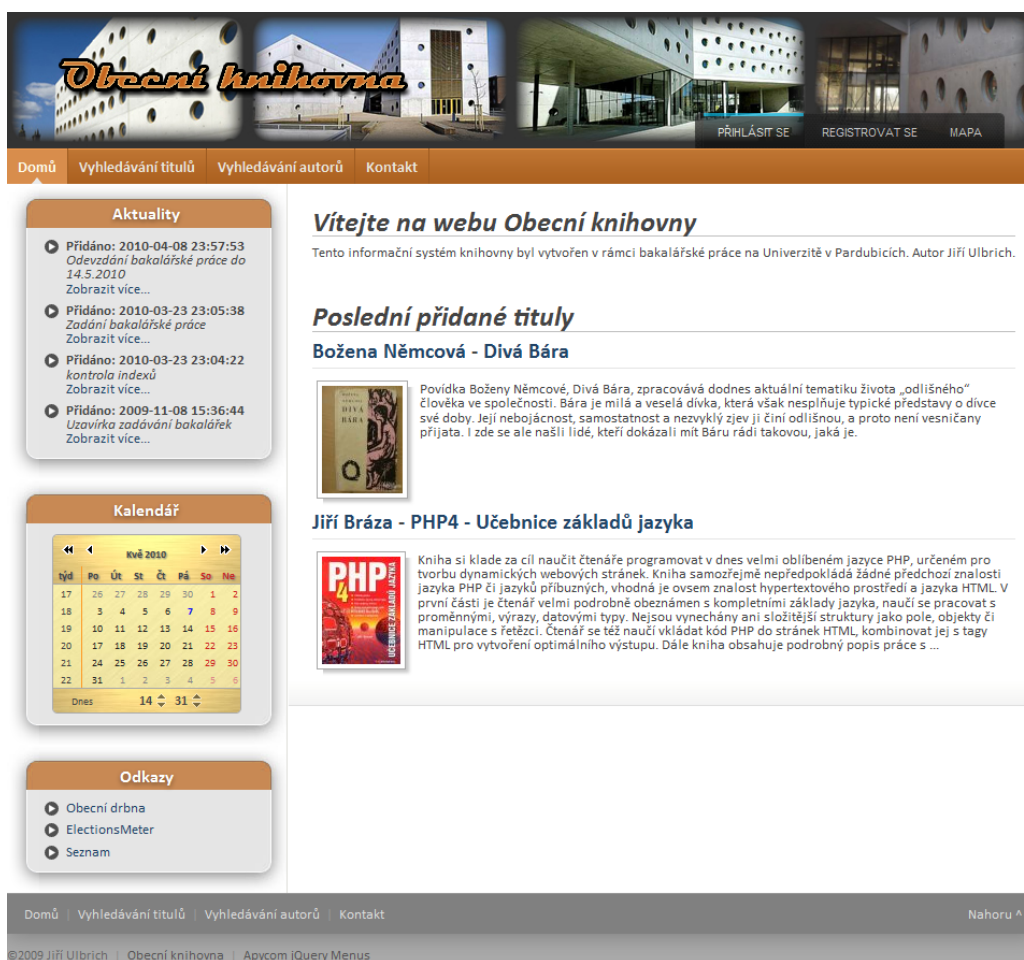
#### 4.4 Adresářová struktura



**Obrázek 7 - Adresářová struktura aplikace**

V adresáři *administration* se nachází soubory, které využívají pouze administrátoři nebo knihovníci systému. Ti jako jediní mohou přidávat a upravovat jakoukoli osobu. Dále jako jediní pracují s aktualitami, které se zobrazují v postranním boxu. V adresáři *mujucet* se nachází soubory, se kterými mohou pracovat pouze přihlášení uživatelé. Každý uživatel má po přihlášení dostupný svůj profil a soubory, které s tímto profilem pracují. Soubory *rezervace.php* a *vypujcky.php* slouží k výpisu pouze vlastních rezervací a výpůjček. Analogicky soubory *old\_rezervace.php* a *old\_vypujcky.php* slouží k výpisu archivu pouze svých rezervací a výpůjček. Knihovna je koncipována tak, aby administrátor i knihovník mohli oba obsluhovat knihovní fond, tzn. aby oba mohli přidávat autory, tituly a k nim exempláře, aby oba mohli vypisovat rezervace a výpůjčky všech čtenářů v systému atd. V adresáři *styles* se nachází soubory se styly. Obrázky pro layout webu jsou v adresářích *images* a *pictures*. V adresáři *scripts* se nachází všechny používané javascripty. Jsou to například jQuery knihovny, editor TinyMCE vkládaný do některých textových polí ve formulářích, skript pro popup okna atd. Mezi další důležité adresáře patří adresář *includes*, kde jsou uloženy soubory, které se často někde vkládají. V adresáři *func* se nacházejí všechny soubory s PHP funkcemi, například na změnu velikosti obrázků atd. Obrázky titulů a autorů jsou ukládány v adresářích *obrazky\_ autori* a *obrazky\_tituly*.

#### 4.5 Layout informačního systému



Obrázek 8 - layout aplikace informačního systému

Layout aplikace je navržen jako dvousloupcový. V levém sloupci se nachází tzv. boxy, které obsahují aktuality v systému, kalendář a odkazy na jiné webové stránky. Pravý sloupec tvoří obsah samotné zvolené stránky. Layout obsahuje obvyklé prvky jako hlavičku, patičku menu nebo submenu. V hlavičce je obrázek s pozadím a obrázek s nápisem *Obecní knihovna*, na který je aplikován efekt vyjíždění pomocí javascriptového řešení *jQuery*. Úvodní stránka také zobrazuje posledních deset přidáných titulů. Tyto tituly jsou postupně měněny efektem s plynulým přechodem vytvořeným opět pomocí *jQuery*. Údaje o těchto posledních deseti přidáných titulech jsou načítány ze souboru, tak aby nezatěžovaly častými dotazy databázi. Tento soubor je vytvářen v pravidelném intervalu třiceti minut pomocí *cronu* a naplánovaných úloh.

## 4.6 Technologie a jazyky použité k vývoji

### 4.6.1 Jazyky a platformy použité k vývoji aplikace

K vývoji jsou použity osvědčené a známé jazyky a platformy, které již byly popsány v úvodních kapitolách o technologiích používaných ke tvorbě webových aplikací. K vytvoření stránky je použit značkovací jazyk *HTML/XHTML*. K úpravě vzhledu jsou použity grafické prvky a kaskádové styly *CSS*. Funkcionalita aplikace je naprogramována ve skriptovacím jazyku *PHP*. K ukládání dat je použita databáze s jazykem *SQL*, konkrétně databázový systém *MySQL*. Kombinace *PHP* a *MySQL* jsem použil především pro svoji rozšířenost na webových úložištích, a to i na bezplatných. Navíc v případě *PHP* se dá hovořit opravdu o povedených manuálových stránkách a celkově vysoké podpoře a velké komunitě. V případě databáze jsem se rozhodl mezi *Oracle* a *MySQL*. *Oracle* má mnohem více funkcí a je profesionálnější než *MySQL*. Ale opět to, že *MySQL* je v nabídce každého hostingu, rozhodlo o jejím využití. Navíc jak ještě bude zmíněno, je použita nadstavba *DIBI*, která velmi zpřimňuje práci s databází a přináší další funkce a vylepšení.

V následujících podkapitolách se tedy již budu věnovat jen speciálním technologiím, které jsem ke tvorbě aplikace knihovního systému použil.

### 4.6.2 Databázová vrstva Dibi

Kvůli přechodu z databázového systému *Oracle* jsem se poohlížel po nadstavbách databáze *MySQL*, které by mohly částečně vyplnit rozdíly a mezery ve funkcích mezi těmito systémy. Nakonec jsem se rozhodl pro databázovou vrstvu *DIBI layer*. Tato vrstva má spoustu výhod a doplňujících funkcí a může tak zpřimnit přechod z *Oracle* databáze. V úvodních kapitolách bylo psáno o bezpečnosti. Jedním z nejdůležitějších bezpečnostních prvků je zabezpečení vstupů do databáze a zamezení tak *SQL injection*. Právě toto za nás řeší *DIBI* automaticky a pokud správně používáme tzv. modifikátory, automaticky za nás escapuje<sup>17</sup> nebezpečné znaky a chrání před nebezpečím *SQL injection*. Modifikátory jsou speciální znaky, které hlídají typy proměnných vstupujících do dotazů za klauzulí *WHERE*. Použití si ukážeme na příkladu:

---

<sup>17</sup> Ochrana nebezpečných znaků z uživatelských vstupů tím, že je před tyto znaky vloženo zpětné lomítko.

```
$res = dibi::query("SELECT [uz_id], [uz_jmeno], [uz_id_role],
[rl_nazev] FROM [\$this->table] JOIN [role] ON [role.rl_id] =
[uzivatele.uz_id_role] WHERE [uz_uziv_jmeno] = %s", \$this->
login_name, " AND [uz_heslo] = %s", trim(\$new_pw), " AND
[uz_overeni] = %i", 1, "%lmt", 1);
```

### Syntaxe 2 - příklad SQL dotazu v DIBI

Na příkladu výše vidíme hned několik modifikátorů. Jsou jimi %s, %i a %lmt. Modifikátor %s říká, že proměnná vstupující do podmínky je typu string, %i říká, že proměnná je typu integer, %lmt je speciální modifikátor určující limit počtu řádků ve výsledku dotazu.

Další přednosti a cíle DIBI:

- maximálně ulehčit práci programátorům,
  - zjednodušit zápis *SQL* příkazů, co to jen půjde,
  - snadný přístup k metodám, i bez globálních proměnných,
  - funkce pro několik rutinních úkonů,
- eliminovat výskyt chyby,
  - přehledný zápis *SQL* příkazů,
  - přenositelnost mezi databázovými systémy,
  - automatická podpora konvencí (escapování / slasho-vání, uvozování identifikátorů),
  - automatické formátování speciálních typů, např. datum, řetězec,
  - sjednocení základních funkcí (připojení k db, vykonání příkazu, získání výsledku),
- a především KISS (Keep It Simple, Stupid),
  - zachovat maximální jednoduchost,
  - raději jeden geniální nápad, než 10.000 hloupých řádků kódu.

(Dibi – Quick Start, 2010)

#### 4.6.3 jQuery knihovna

Rozhraní aplikace jsem se rozhodl vylepšit různým javascripty s vizuálními efekty. Po mnoha pokusech s různými typy javascriptů jednoznačně padla volba na knihovnu jQuery<sup>18</sup>. Jedná se o malý a dnes už velmi rychlý javascriptový framework. Slouží k práci s technologií *AJAX*<sup>19</sup>, k manipulaci s *XHTML* stránkou (*DOM*<sup>20</sup>), k práci s kaskádovými styly a obsahuje různé funkce pro provádění animací. Vyhledávání elementů *DOM* je velmi snadné, stejně tak jejich modifikace nebo vytváření nových elementů. Mezi hlavní výhody patří malá velikost a možnost pracovat s komprimovanými knihovnami, podpora všech prohlížečů a velké množství materiálů a návodů na internetu, se kterými je práce s touto knihovnou velmi

<sup>18</sup> Více informací a příkladů k nahlédnutí na <http://jqueryui.com/>

<sup>19</sup> Kombinace technologií používaná při tvorbě webových aplikací. Výsledkem je měnění obsahu částí stránek bez znovunačtení celé stránky.

<sup>20</sup> Document Object Model (Objektový model dokumentu)

jednoduchá a efektivní. Moje osobní zkušenost je taková, že díky této knihovně jsem byl schopen vytvořit spoustu pěkných efektů, které bych jinak sám naprogramoval velmi těžce a složitě.

Práci s jQuery si ukážeme na následujících jednoduchých převzatých příkladech:

- ✓ na tomto příkladu je vidět použití CSS selektorů k vyhledání všech odkazů,

```
var $odkazy = jQuery("a");
```

**Syntaxe 3 - použití CSS selektorů v jQuery**

- ✓ všechny odkazy můžeme zařadit do třídy *test*,

```
jQuery("a").addClass("test");
```

**Syntaxe 4 - přiřazení třídy elementu v jQuery**

- ✓ následující kód zařídí zobrazení zprávy při kliknutí na odkaz v seznamu,

```
var pozdrav = fiction() {  
    alert("Hallo World!");  
    return false;  
};  
jQuery("ul a").click(pozdrav);
```

**Syntaxe 5 - zobrazení zprávy pomocí jQuery**

- ✓ následující kód přidá za každý odstavec *div* se zvýrazněným nápisem. (ŠERÝ, 2007)

```
var $odkazy = jQuery("a");  
jQuery("p").after("<div><em>Hello World!</em></div>");
```

**Syntaxe 6 - přidání objektu za vybrané elementy pomocí jQuery**

V aplikaci je využita tato knihovna k zobrazování efektů. Konkrétně vyjždění informačního okénka v pravém horním rohu obrazovky, k efektu nadpisu v hlavičce, k přepínání stránek s výpůjčkami a rezervacemi ve vlastním profilu přihlášeného uživatele bez znovunačtení stránky, k zablíkání tlačítka exemplářů na stránce konkrétního titulu a tlačítka titulů na stránce konkrétního autora, ke změně posledních přidanych titulů na úvodní stránce bez znovunačtení stránky. Hlavní knihovnu jQuery můžeme do svých stránek načítat například z Google Api a minimalizovat tak datový přenos na svém serveru.

```
<script src="http://www.google.com/jsapi"  
type="text/javascript"></script>  
<script type="text/javascript" charset="utf-8">  
google.load("jquery", "1.4.2");</script>
```

**Syntaxe 7 - načtení hlavní knihovny jQuery z Googlu**



#### 4.6.4 PHPmailer třída

Jelikož v aplikaci se používá kontaktního formuláře a rozesílání upomínek pomocí emailů, bylo potřeba vyřešit technicky pomocí skriptovacího jazyka *PHP* rozesílání emailových zpráv. Posloužila k tomu volně dostupná třída *PHPmailer*<sup>21</sup>. Třída *PHPmailer* slouží k posílání emailů prostřednictvím *PHP*. Jedná se o lepší způsob než užití vestavěné funkce *mail()*. Výhodou je, že není nutná instalace poštovního serveru na počítač, kde webový server běží. Další výhodou je, že lze posílat i zprávy na servery vyžadující *SMTP*<sup>22</sup> autentizaci. Použití a nastavení této třídy je velmi jednoduché a ukážu jej na příkladu:

```
$mail = new PHPMailer();
$mail->IsSMTP();
$mail->Host = "66.147.240.150";
$mail->SMTPAuth = true;
$mail->Username = "noreply@easycheq.com";
$mail->Password = "mojeheslo";
$mail->From = $_POST['od'];
$mail->FromName = $odesilatel;
$mail->AddAddress($_POST['od']);
$mail->IsHTML(true);
$mail->WordWrap = 50;
$mail->Subject = "Dotaz z kontaktního formuláře";
$mail->Body = $telozpravy;
$mail->CharSet = "utf-8";
if(!$mail->Send()) echo 'Error: '.$mail->ErrorInfo;
```

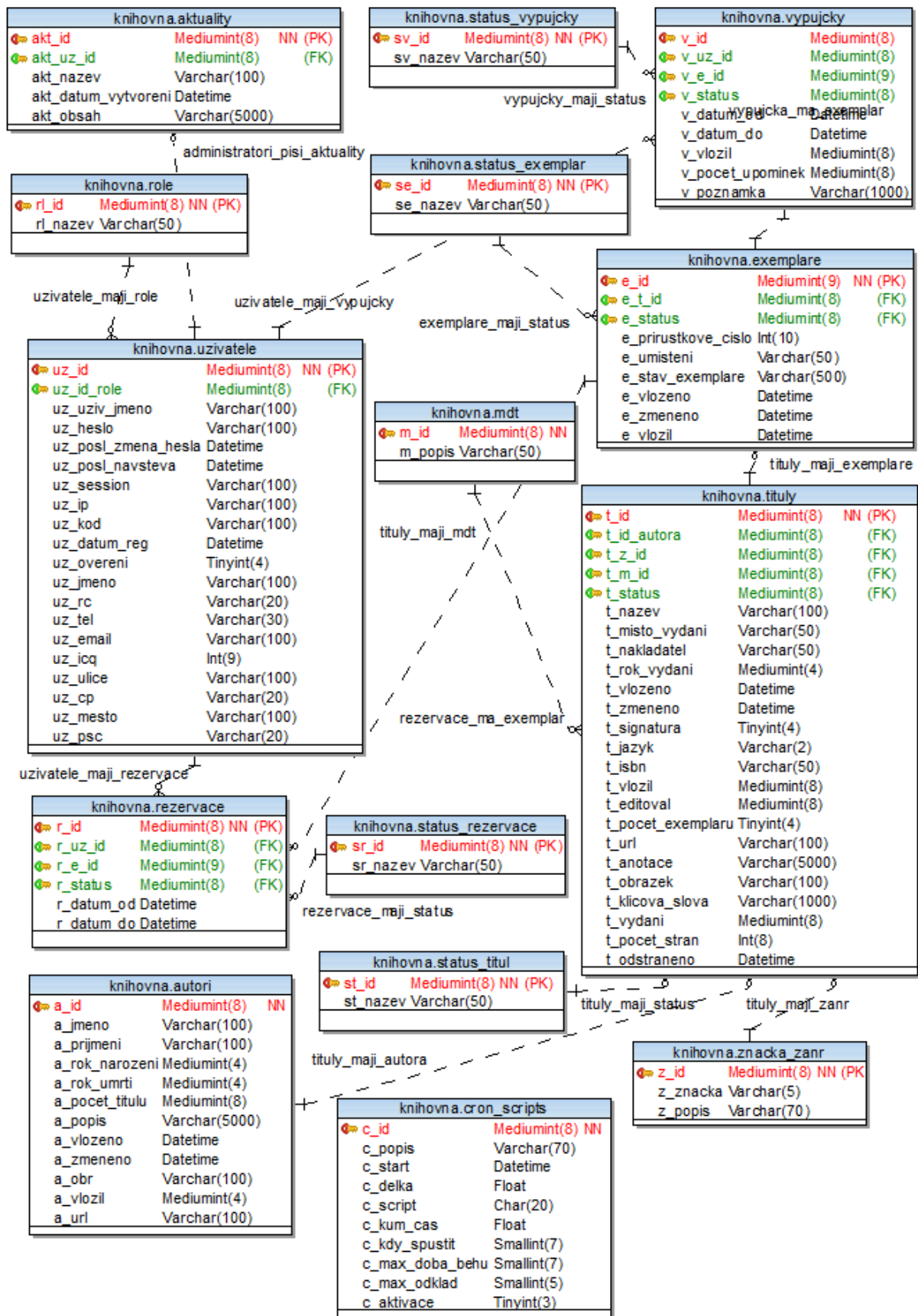
**Syntaxe 8 - syntaxe nastavení PHPmaileru**

První řádek nám vytvoří novou instanci třídy s názvem *\$mail*. Přes tuto instanci se budeme odkazovat na ostatní metody této třídy. Druhý řádek říká, že chceme odesílat pomocí *SMTP* serveru. Následující čtyři řádky souvisí s nastavením *SMTP* serveru. Nejprve se nastaví adresa pro tento server, zapne se autentizace pomocí *SMTPAuth* a zadá se uživatelské jméno a heslo pro tento server. Na dalších řádcích již následuje nastavení samotné emailové zprávy. V položce *From* nastavím emailovou adresu odesílatele, v položce *FromName* nastavíme jméno odesílatele. Příjemce zprávy se nastaví pomocí metody *AddAddress*, která má dva parametry. První parametr je povinný a je to emailová adresa příjemce, druhý parametr je nepovinný a je určený pro jméno příjemce. Následuje nastavení samotného obsahu zprávy. Pomocí metody *IsHTML* nastavíme v tomto případě, že tělo zprávy má být ve formátu *HTML*. Položka *WordWrap* určuje pořadí znaku, za kterým dojde k zalomení řádku, doporučená hodnota je 50. Položka *Subject* nastavuje předmět zprávy a položka *Body* nastavuje samotný obsah zprávy. V našem případě je obsah zprávy uložen v proměnné. Nyní zbývá již jenom pomocí položky *CharSet* nastavit požadované kódování textu a zprávu odeslat pomocí metody *send()*. V případě neúspěchu je zobrazena chybová hláška pomocí položky *ErrorInfo*. (ŠERÝ, 2007)

<sup>21</sup> Třidu *PHPmailer* je možné bezplatně stáhnout na této adrese <http://phpmailer.worxware.com>

<sup>22</sup> Simple Mail Transfer Protocol (Protokol pro přenos zpráv elektronické pošty)

## 4.7 ER-diagram



Obrázek 9 - ER diagram aplikace knihovního systému

## 4.8 Popis tabulek databáze

### 4.8.1 Tabulka uživatelů

V tabulce *knihovna.uzivatele* jsou uchovávány všechny informace o uživateli. Sloupce s adresou svou přítomností zde sice porušují normální formu, nicméně se nepředpokládá, že tabulka uživatelů naroste do takových rozměrů, aby se vyplatilo mít další tabulky pro všechna města, ulice a PSČ. Datový typ proměnné pro sloupec *uz\_tel* je Varchar, kdyby uživatel chtěl uvést u telefonu předvolbu v závorce apod. Nebezpečné znaky jsou chráněny jak na straně databáze pomocí automatického escapování, tak na straně PHP pomocí různých regulárních výrazů. Pro vyšší rychlost dotazů je na cizím klíči *uz\_id\_role* vytvořen index. Na primárním klíči se generuje index automaticky při návrhu databáze. Primární klíč se generuje pomocí MySQL funkce *auto\_increment*<sup>23</sup>. Ve sloupci *uz\_session* je uchováván hash kód, který slouží ke správné autentifikaci a je jedním z bezpečnostních prvků zamezujícím tzv. krádežím session.

### 4.8.2 Tabulka aktualit

Tabulka *knihovna.aktuality* slouží k uchovávání aktualit, které jsou pak pro všechny osoby zobrazovány na všech stránkách v levém sloupci. Zobrazováno je implicitně pět posledních vložených aktualit. Vkládat a upravovat aktuality může pouze administrátor systému. Ve sloupci se zobrazí datum přidání s názvem aktuality a odkaz na vyskakovací okno s obsahem vložené aktuality. Pro vyšší rychlost dotazů je na cizím klíči *akt\_uz\_id* vytvořen index. Na primárním klíči se generuje index automaticky při návrhu databáze.

### 4.8.3 Tabulka rolí

V tabulce *knihovna.role* jsou definovány role, kterých může uživatel v systému nabývat. Jeden řádek z tabulky rolí odpovídá více řádkům z tabulky osob. Naopak jeden řádek z tabulky osob odpovídá právě jednomu řádku z tabulky rolí. Z toho vyplývá že, pokud by například administrátor chtěl využívat knihovnických služeb, musel by si vytvořit další účet jako čtenář. Při registraci z úvodní stránky je rolovací menu vynecháno a automaticky se vkládá do tabulky uživatelů do sloupce *uz\_id\_role* hodnota tři což odpovídá názvu role čtenář.

### 4.8.4 Tabulka cronu

V tabulce *cron\_scripts* jsou ukládány skripty a informace o nich, které se spouští z naplánovaných úloh v systému Windows nebo z cronu na linuxových systémech. Tato tabulka nemá žádné relace, slouží pouze k načítání a uchovávání informací o skriptech jako jsou například kdy skript spouštět, maximální doba běhu, délka posledního běhu, kumulovaný čas atd.

---

<sup>23</sup> Pomocí této funkce je automaticky navyšována hodnota sloupce v novém řádku o 1. Používá se typicky pro navyšování hodnoty ID v tabulce.

#### 4.8.5 Tabulka výpůjček

V tabulce *knihovna.vypujcky* jsou evidovány všechny výpůjčky v systému. Jednomu řádku z tabulky výpůjček odpovídá jeden řádek z tabulky uživatelů. Opačně je tomu jinak a jednomu řádku z tabulky uživatelů může odpovídat více řádků z tabulky výpůjček. Čili jedná se v tomto případě o vztah 1:N. Vztah mezi tabulkou výpůjček a tabulkou exemplářů je 1:1. Čili jednomu řádku z tabulky výpůjček odpovídá jeden řádek z tabulky exemplářů a je tomu vždy i naopak. Pro vyšší rychlost dotazů je na cizím klíči *v\_e\_id* vytvořen index, obdobně je tomu u cizího klíče *v\_uz\_id* a *v\_status*. Na primárním klíči se generuje index automaticky při návrhu databáze. Výpůjčky může vytvářet knihovník i administrátor. Při vkládání výpůjčky si vybere čtenáře a požadovaný exemplář titulu a zadá délku výpůjčky.

#### 4.8.6 Tabulka rezervací

V tabulce *knihovna.rezervace* jsou evidovány všechny rezervace v systému. Jednomu řádku z tabulky rezervací odpovídá jeden řádek z tabulky uživatelů. Opačně je tomu jinak a jednomu řádku z tabulky uživatelů může odpovídat více řádků z tabulky rezervací. Jedná se v tomto případě o vztah 1:N. Vztah mezi tabulkou rezervací a tabulkou exemplářů je 1:1. Čili jednomu řádku z tabulky rezervací odpovídá jeden řádek z tabulky exemplářů a je tomu vždy i naopak. Pro vyšší rychlost dotazů je na cizím klíči *r\_e\_id* vytvořen index, obdobně je tomu u cizího klíče *r\_uz\_id* a *r\_status*. Na primárním klíči se generuje index automaticky při návrhu databáze. Primární klíč se generuje pomocí *MySQL* funkce *auto\_increment*. Rezervovat tituly může každý aktivovaný čtenář. Kombinací podmínek a *SQL* dotazů jsou vytvořena omezení, aby si čtenář nemohl rezervovat stejný titul a aby maximální počet rezervovaných titulů nemohl přesáhnout tři kusy.

#### 4.8.7 Tabulka statusů výpůjček

V tabulce *knihovna.status\_vypujcky* jsou definovány statusy, které může výpůjčka v systému nabývat. Jsou to „*aktivní platná výpůjčka, vypršená platnost výpůjčky a odeslána upomínka, zrušená výpůjčka*“. Jeden řádek z tabulky statusů výpůjček odpovídá více řádkům z tabulky výpůjček. Naopak jeden řádek z tabulky výpůjček odpovídá právě jednomu řádku z tabulky statusů výpůjček. Jedná se v tomto případě o vztah 1:N.

#### 4.8.8 Tabulka statusů rezervací

V tabulce *knihovna.status\_rezervace* jsou definovány statusy, které může rezervace v systému nabývat. Jsou to „*čeká na exemplář, odeslán email a hlídán exemplář, neplatná rezervace*“ Relace a vztah mezi tabulkou rezervací a tabulkou statusů rezervací je stejný jako v předchozím případě u tabulky statusů výpůjček.

#### 4.8.9 Tabulka exemplářů

V tabulce *knihovna.exemplare* jsou uchovávány všechny informace o konkrétních exemplářích daného titulu. Pro vyšší rychlost dotazů je na cizím klíči *e\_t\_id* vytvořen index. Stejně tak je vytvořen i na sloupci *e\_status*. Na primárním klíči se generuje index automaticky při návrhu databáze. Primární klíč se generuje pomocí *MySQL* funkce *auto\_increment*. Vkládat exempláře může knihovník i administrátor. Při vkládání exempláře si z rolovacího menu vybere příslušný titul. Pokud není titul v systému, musí se nejprve vložit a až po té se může vložit požadovaný exemplář.

#### 4.8.10 Tabulka statusů exemplářů

V tabulce *knihovna.status\_exemplar* jsou definovány statusy, které může exemplář v systému nabývat. Jsou to „*existující, odepsaný*“. Jeden řádek z tabulky statusů exemplářů odpovídá více řádkům z tabulky exemplářů. Naopak jeden řádek z tabulky exemplářů odpovídá právě jednomu řádku z tabulky statusů exemplářů. Jedná se v tomto případě o vztah 1:N.

#### 4.8.11 Tabulka autorů

Tabulka *knihovna.autori* slouží k uchovávání informací o autorech. Autory mohou přidávat jak administrátoři, tak i knihovníci. Primární klíč se generuje pomocí *MySQL* funkce *auto\_increment*. Sloupec *a\_pocet\_titulu* je udržován jednak pomocí triggerů při vkládání, tak i pomocí *PHP* skriptu, který je kontroluje a přepočítává počty titulů. Tento skript je spouštěn pomocí vlastního modulu *cronu*, kdy v naplánované úloze nebo v samotném *cronu* v případě unixového systému je pouze jeden soubor, který se stará o spouštění dalších požadovaných *PHP* skriptů.

#### 4.8.12 Tabulka žánrů

V tabulce *knihovna.znacka\_zanr* jsou definovány žánry, kterých může titul v systému nabývat. Jsou to „*beletrie, naučná pro dospělé, mládež beletrie, mládež naučná*“. Jeden řádek z tabulky žánrů odpovídá více řádkům z tabulky titulů. Naopak jeden řádek z tabulky titulů odpovídá právě jednomu řádku z tabulky žánrů. Jedná se tedy v tomto případě o vztah 1:N.

#### 4.8.13 Tabulka titulů

V tabulce *knihovna.tituly* jsou uchovávány všechny informace o titulech. Pro vyšší rychlost dotazů jsou na všech cizích klíčích vytvořeny indexy. U sloupce *t\_isbn* je použit datový typ *Varchar2*. I když se jedná o číselné údaje, jsou v nich i znaky jako čárka nebo pomlčka. Nebezpečné znaky jsou chráněny jak na straně databáze pomocí automatického escapování<sup>24</sup>, tak na straně *PHP* pomocí různých regulárních výrazů. Na primárním klíči se generuje index automaticky při návrhu databáze. Primární klíč se generuje pomocí *MySQL* funkce *auto\_increment*. Vkládat tituly může knihovník i administrátor. Při vkládání titulu si z rolovacího menu vybere příslušný žánr pro vkládaný titul a jeho autora. Pokud není autor

---

24

v systému, musí se nejprve vložit autor titulu a až po té se může vložit požadovaný titul. Na tabulku titulů jsou navázány tabulky autorů, žánrů, kódů *MDT* a statusů titulu. Všechny vztahy jsou 1:N. Například autor má v systému více titulů, ale jeden titul má vždy jednoho autora.<sup>25</sup> Sloupec *t\_pocet\_exemplaru* je stejně jako sloupec *a\_pocet\_titulu* v tabulce autorů udržován pomocí triggerů při vkládání nebo pomocí *PHP* skriptu, který kontroluje a přepočítává počty exemplářů.

#### 4.8.14 Tabulka mezinárodní desetinného třídění

V tabulce *knihovna.mdt* jsou definovány značky mezinárodního desetinného třídění. Jsou to „0 - fylozofie; 1 - náboženství, 2 - teologie; 3 - společenské vědy, statistika, politika, ekonomika; 4 - není obsazena; 5 - matematika, přírodní vědy; 6 - užití vědy, lékařství, technika; 7 - umění, umělecký průmysl, fotografie, film, hudba; 8 - jazykověda, krásná literatura, literární věda; 9 - zeměpis, biografie, dějiny“. *MDT*<sup>26</sup> je nastavováno pouze u naučné literatury jak pro děti tak pro dospělé. Pokud se jedná o beletrii, je výchozí nastavení čtyři pomocí defaultní hodnoty sloupce v tabulce titulů. Jeden řádek z tabulky mezinárodního desetinného třídění odpovídá více řádkům z tabulky titulů. Naopak jeden řádek z tabulky titulů odpovídá právě jednomu řádku z tabulky mezinárodního desetinného třídění. Jedná se tedy v tomto případě o vztah 1:N.

#### 4.8.15 Tabulka statusů titulů

V tabulce *knihovna.status\_titul* jsou definovány statusy, které může titul v systému nabývat. Jsou to „*existující, odstraněný*“. Jeden řádek z tabulky statusů titulů odpovídá více řádkům z tabulky titulů. Naopak jeden řádek z tabulky titulů odpovídá právě jednomu řádku z tabulky statusů titulů. Jedná se v tomto případě o vztah 1:N.

## 4.9 Programy a operační systémy použité při vývoji

Při vývoji mé aplikace jsem používal programové vybavení, které Vám nyní popíšu. Z finančních důvodů jsem používal programy, které jsou zcela zdarma, nebo programy, které nám poskytuje univerzita zdarma k nekomerčním účelům. Například v projektu firmy Microsoft *MSDN*<sup>27</sup>.

### 4.9.1 Programové vybavení

K samotné editaci *PHP*, *HTML*, javascriptových souborů a souborů s kaskádovými styly jsem používal program *PSPad*<sup>28</sup> ve verzi 4.5.5. Tento freewarový editor splňuje všechny mé požadavky. Je velmi rychlý a má mnoho funkcí. Mezi hlavní výhody patří například zvýrazňování syntaxe u poměrně hodně jazyků, které tento program zná. Dále pak například automatické formátování u *HTML* a *CSS* jazyků.

---

<sup>25</sup> Systém je nastaven tak, že pokud má titul více autorů, je uváděn pouze první.

<sup>26</sup> *MDT* – mezinárodní desetinné třídění.

<sup>27</sup> Více na <http://www.upce.cz/fei/spoluprace/microsoft.html>.

<sup>28</sup> Více na <http://www.pspad.com/cz/>.

Další program, který jsem využil při návrhu databáze, se jmenuje *Toad Data Modeler*<sup>29</sup>. Tento program není zcela zdarma. Nicméně nám byla poskytnuta verze ke studijním účelům, která má jistá omezení. Například v maximálním počtu tabulek, které se v návrhu mohou objevit.

K vytváření a editaci grafických prvků stránek jsem využil buďto program *Gimp*<sup>30</sup>, který je zcela zdarma. Dále pak program *Adobe Photoshop*, který nám rovněž poskytla univerzita v akademické verzi pro nekomerční využití. Tento program patří mezi špičkové nástroje při vytváření designu webových stránek.

Ke zprovoznění webového a databázového serveru jsem nejprve využil balíček *Wampp*. Později jsem přešel na balíček *Xampp*<sup>31</sup> ve verzi 1.7.1 který obsahuje ještě *FTP* server a má více funkcí oproti *Wamppu*. Pro přístup k databázi je v tomto balíčku zprovozněno webové rozhraní *phpmyadmin*.

Vývoj aplikace probíhal na operačních systémech *Windows 7*<sup>32</sup> a *Ubuntu 9.10*<sup>33</sup>. První jmenovaný nám poskytuje opět univerzita v již zmiňovaném programu firmy Microsoft *MSDN*. Druhý operační systém je zcela zdarma. V něm byly používány ekvivalenty výše zmiňovaných programů. Snad kromě *Toad Data Modeleru* lze vše bez problémů v *Ubuntu* nahradit nebo použít verzi pro Linux.

---

<sup>29</sup> Více na [http://www.toadsoft.com/toaddm/toad\\_data\\_modeler.htm](http://www.toadsoft.com/toaddm/toad_data_modeler.htm).

<sup>30</sup> Více na <http://www.gimp.cz/>.

<sup>31</sup> Více na <http://www.apachefriends.org/en/xampp.html>.

<sup>32</sup> Více na <http://windows.microsoft.com/cs-cz/windows7/products/home?os=nonwin7>.

<sup>33</sup> Více na <http://www.ubuntu.com/>.

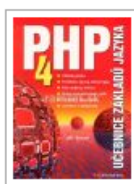
## 5 Uživatelská příručka a popis aplikace

Knihovní systém pracuje ve čtyřech režimech, a to v návštěvnickém, čtenářském, knihovnickém a administrátorském. Než jakýkoliv uživatel bude moci využívat níže popisovaných služeb, musí být řádně registrován a přihlášen.

### 5.1 Nepřihlášený uživatel

Nepřihlášeným uživatelům je k dispozici vyhledávání titulů nebo autorů podle různých kritérií a jejich zobrazování. Každý titul a autor má v systému svoji jedinečnou tzv. přátelskou url adresu. Pěkný tvar této adresy je velmi důležitý pro vyhledávače a obecně je jeden z hlavních prvků tzv. *SEO* optimalizace. Dále je všem uživatelům dostupný jednoduchý kontaktní formulář, kdy po vyplnění jména, emailové adresy a *CAPTCHA* ochrany je možné odeslat zprávu administrátorovi. Každý uživatel si také může zobrazovat detail aktualit z boxu aktualit v levém sloupci. Nesmím zapomenout asi na nejdůležitější možnost pro každého návštěvníka, a tou je možnost registrace v přehledném registračním formuláři se zabezpečovacími prvky, kontrolou *CAPTCHA* a autorizačním procesem pomocí emailové zprávy. Následující obrázek ukazuje konkrétní stránku určitého titulu, bez boxů, hlavičky a patičky. Url adresa tohoto titulu je v tomto případě `<http://...../knizky/php4-ucebnice-zakladu-jazyka-id4>`.

#### Jiří Bráza - PHP4 - Učebnice základů jazyka



*Kniha si klade za cíl naučit čtenáře programovat v dnes velmi oblíbeném jazyce PHP, určeném pro tvorbu dynamických webových stránek. Kniha samozřejmě nepředpokládá žádné předchozí znalosti jazyka PHP či jazyků příbuzných, vhodná je ovšem znalost hypertextového prostředí a jazyka HTML. V první části je čtenář velmi podrobně obeznámen s kompletními základy jazyka, naučí se pracovat s proměnnými, výrazy, datovými typy. Nejsou vynechány ani složitější struktury jako pole, objekty či manipulace s řetězci. Čtenář se též naučí vkládat kód PHP do stránek HTML, kombinovat jej s tagy HTML pro vytvoření optimálního výstupu. Dále kniha obsahuje podrobný popis práce s funkcemi včetně vytváření vlastních funkcí a jejich používání a řízení programu pomocí podmínek, cyklů a dalších konstrukcí. Na těchto základních vlastnostech potom staví další kapitoly, které se zabývají konkrétními problémy a aplikacemi, jež jsou v PHP nejčastěji řešeny. Velmi podrobně je zpracována obsluha formulářů včetně výkladu samotné tvorby formulářů v HTML tak, aby byl každý čtenář schopen vytvářet profesionálně vypadající a profesionálně se chovající formuláře. Jednou z nejdůležitějších a nejčastěji používaných oblastí využití PHP je interakce s databázemi. Díky tomu je možné vytvářet rychlé, spolehlivé a bezpečné aplikace manipulující s daty. PHP pro databáze nabízí velmi dobrou podporu a umožňuje s nimi snadno pracovat. Čtenář se v kapitole o databázích naučí zakládat vhodně navržené databáze a tabulky, upravovat jejich strukturu a samozřejmě manipulovat s obsahem databáze (přidávání záznamů, úprava záznamů, vyhledávání, mazání), a to jak prostřednictvím PHP, tak přímo či s využitím dalších nástrojů. Dále jsou v knize popsány ještě další oblasti aplikace PHP, jako například práce s elektronickou poštou, práce se soubory, vytváření, úprava a manipulace s dokumenty XML či úprava grafických dokumentů. Nechybí samozřejmě ani podrobný popis veškerého potřebného softwaru nutného pro vývoj aplikací v PHP a jejich provoz v prostředí WWW či popis instalace a konfigurace těchto aplikací.*

- Nakladatel: Grada Publishing a.s.
- Místo vydání: Praha
- Rok vydání: 2002
- ISBN: 80-247-0442-0
- Signatura: 84367
- Ilustroval:

- Exempláře      Detail URI

Přírůstkové číslo	Umístění	Status	Datum od	Datum do	Počet rezervací	Akce
5	423652542	volný			0	reservation ▶

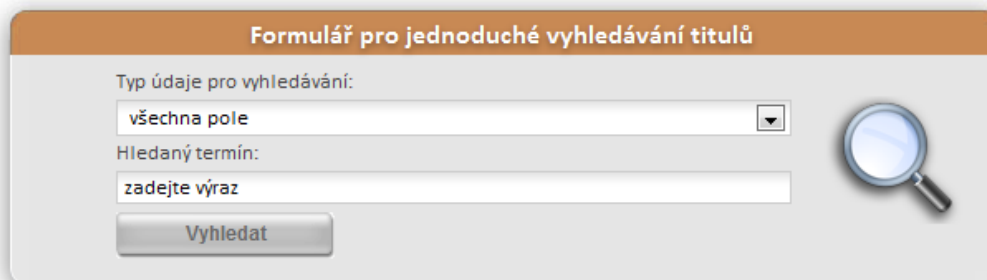
Záznamů: 0 - 1 z celkem 1 | << | předchozí | následující | >> |

Obrázek 10 - konkrétní stránka titulu



## 5.2 Čtenářský režim

Jako přihlášený uživatel má čtenář přístupné všechny funkce, které má i návštěvník. Avšak formuláře v těchto funkcích již nejsou chráněny *CAPTCHA* kontrolou.



Obrázek 11 - formulář pro vyhledávání titulů bez CAPTCHA kontroly

Každý přihlášený uživatel si v systému může zobrazit svůj profil, v něm lze sledovat například své aktuální nebo již uběhnuté rezervace a výpůjčky. Stavů výpůjček jsou tři, a to *aktivní platná výpůjčka*, *vypršená platnost výpůjčky*, *odeslána upomínka* a *zrušená výpůjčka*. Stavů rezervací jsou také tři, a to *čeká na exemplář*, *odeslán email*, *hlídán exemplář* a *neplatná rezervace*. Čtenář po nalezení požadovaného titulu a zobrazení exemplářů k tomuto titulu může rezervovat pouze již vypůjčené exempláře. Systém je momentálně nastaven tak, že dohromady může mít čtenář pouze tři rezervace a tři výpůjčky. Rezervace je platná po dobu sedmi dní. O její aktuálnost a zaslání emailu při uvolnění exempláře se stará skript běžící v cronu nebo naplánovaných úlohách. Čtenář kromě svých aktuálních rezervací a výpůjček může zobrazit i historii svých rezervací a výpůjček. V profilu jsou k dispozici další běžné funkce jako například editace informací ve svém profilu nebo změna hesla.

### Uživatelský profil

Nacházíte se na stránce svého uživatelského profilu.

- Jméno: Jiří Ulbrich
- Přihlašovací jméno: juke
- Registrován: před 6 měsíci (2009-11-08 11:22:15)
- ICQ: 150591394
- Rodné číslo: 831011/3141
- Ulice: Blahoutova 622
- Město: Pardubice
- PSČ: 53012

Odhlásit

Editovat profil

Změnit heslo

Výpůjčky	Rezervace	Archiv výpůjček	Archiv rezervací									
<table><thead><tr><th>Název titulu</th><th>Od kdy</th><th>Do kdy</th></tr></thead><tbody><tr><td>V zámku a podzámčí</td><td>2010-03-25 22:07:20</td><td>2010-03-30 22:07:20</td></tr><tr><td>Babička</td><td>2010-03-26 09:43:26</td><td>2010-03-31 09:43:26</td></tr></tbody></table>				Název titulu	Od kdy	Do kdy	V zámku a podzámčí	2010-03-25 22:07:20	2010-03-30 22:07:20	Babička	2010-03-26 09:43:26	2010-03-31 09:43:26
Název titulu	Od kdy	Do kdy										
V zámku a podzámčí	2010-03-25 22:07:20	2010-03-30 22:07:20										
Babička	2010-03-26 09:43:26	2010-03-31 09:43:26										
Záznamů: 0 - 2 z celkem 2   <<   předchozí   následující   >>												

Obrázek 12 - stránka uživatelského profilu

### 5.3 Knihovnický a administrátorský režim

Jak už bylo zmíněno dříve, knihovna je koncipována tak, že funkce pro knihovníka a administrátora jsou jen minimálně rozdílné, aby oba mohli spravovat funkce knihovního fondu, výpůjčního a rezervačního modulu. Po úspěšném přihlášení administrátora nebo knihovníka se v menu objeví možnost *administrace*. V tomto menu je další submenu, kde se administrátorovi oproti knihovníkovi navíc zobrazuje možnost aktualit a cronu. Knihovníkovi jsou tyto funkce nepřístupné, nemůže tedy spravovat aktuality a zobrazovat informace o skriptech v cronu. Aktuality jsou zobrazovány v levém sloupci v boxu aktualit. Zobrazováno je posledních pět vložených aktualit. Administrátor systému oproti knihovníkovi má možnost spravovat nebo přidávat všech typy osob v systému, knihovník může spravovat nebo přidávat pouze čtenáře.

Tituly

ID	Název	Autor	Rok	Exempláře	Vloženo	Změněno	Signatura	ISBN	Akce
1	<a href="#">Babička</a>	Božena Němcová	1995	2 +	2010-01-20	2010-03-22	12424	80-901199-6-4	
2	<a href="#">V zámku a podzámčí</a>	Božena Němcová	2002	1 +	2010-01-20	2010-02-26	14265	80-86612-02-3	
3	<a href="#">Divoká</a>			1 +	2010-01-20	2010-02-26	43258	80-86132-62-5	
4	<a href="#">PHP4 - Učebnice</a>			1 +	2010-01-20	2010-02-26	84367	80-247-0442-0	
7	<a href="#">PHP4 - Prakt</a>			0 +	2010-01-20	2010-02-26	32532	80-247-0441-2	
8	<a href="#">Psoh</a>			0 +	2010-03-17		4326436542	80-7050-318-1	
10	<a href="#">Petroleio</a>			0 +	2010-03-22	2010-04-01	4326346245	80-901223-8-8	
13	<a href="#">Vinnetou I. - It</a>			0 +	2010-04-13		3426567546	80-7264-086-0	
14	<a href="#">Dynamické HTML v akci</a>	Eric M. Schurman	2000	1 +	2010-04-28		34654094320	80-7226-401-X	

Záznamů: 0 - 9 z celkem 9 | << | předchozí | následující | >> |

Obrázek 13 - administrace titulů v systému

Společné funkce pro knihovníka a administrátora jsou evidence autorů, titulů, exemplářů, výpůjček a rezervací. Formulář pro přidání autora obsahuje povinné údaje jméno, příjmení a obrázek autora. Nepovinné jsou rok narození, rok úmrtí a popis autora. Správa autorů zahrnuje i editaci již přidávaných autorů. Při zobrazení autorů v administraci je při najetí na odkaz na jedinečnou url adresu autora zobrazováno tooltip okno s malým obrázkem a zkráceným popisem autora. Stejně rozhraní je použito i na tituly, liší se pouze vstupními hodnotami. Povinné údaje pro přidání titulu jsou název titulu, autor, místo vydání, rok vydání, signatura, *ISBN*<sup>34</sup>, jazyk, vydání, počet stran, žánr, *MDT* a obrázek titulu. Nepovinný údaj je anotace. Při zobrazení titulů je stejně jako u autorů zobrazováno při najetí myši na odkaz na titul tooltip okno s malým obrázkem a zkrácenou anotací titulu. Dále je ve sloupci *exempláře* zobrazován počet exemplářů formou odkazu na stránku s exempláři daného titulu. Vedle tohoto odkazu je zobrazován další odkaz formou obrázku, který po najetí myši mění barvu. Odkaz vede na formulář pro přidání exempláře k danému titulu. Po kliknutí na tento odkaz je v tomto formuláři předvolen titul a je zamknut select s výběrem titulu. Další povinný údaj ve formuláři pro přidání exempláře je umístění, nepovinný údaj je popis exempláře. Na formulář pro přidání exempláře je možnost se dostat i ze submenu administrace s tím že, select na výběr titulu je odemčený a umožňuje vybrat požadovaný titul, ke kterému má být vložen nový exemplář. Formulář pro vytvoření výpůjčky je dostupný ze submenu

<sup>34</sup> International Standard Book Number (Mezinárodní standardní číslo knihy)

administrace. Mezi jeho povinné údaje patří titul, konkrétní exemplář, uživatel a délka výpůjčky. Délka výpůjčky může být na čtrnáct dní nebo jeden měsíc. Jako nepovinný údaj může být uvedena poznámka o výpůjčce. Mohou být zobrazeny aktuální výpůjčky všech uživatelů nebo archiv všech zrušených výpůjček. Pokud čtenář přinese vrátit vypůjčený exemplář, knihovník najde požadovanou výpůjčku a zruší ji, výpůjčka je pak přesunuta do archivu výpůjček. Rezervace jsou na tom obdobně, jednak může být rezervace zrušena a přesunuta do archivu rezervací v režii knihovníka nebo administrátora, o druhý způsob se starají cron nebo naplánované úlohy a jejich skripty.

**Formulář pro přidání titulu**

Název titulu: \*

Autor: \*

- vyberte autora -

Nakladatelství: \*

Místo vydání: \*

Rok vydání: \*

Signatura: \*

ISBN: \*

Jazyk: \*

Vydání: \*

Počet stran: \*

Klíčová slova (oddělená čárkou): \*

Žánr: \*

- vyberte žánr -

Mezinárodní značení MDT u naučné literatury:

4 - není obsazena


Obrázek: \*

Procházet..

Anotace:

**B** *I* U ABC | ↺ ↻ 📄 | ☰ ☱

Vložit titul



Obrázek 14 - formulář pro přidání titulu

## 6 Instalační příručka

V této kapitole bude popsán postup instalace systému na webový server. Knihovní systém vyžaduje server s podporou *PHP 5* a databázi *MySQL*. Databázový server musí podporovat tvorbu tzv. triggerů. Nejdříve je nutné vytvořit samotnou databázi pro knihovnu. Po otevření webového rozhraní *phpmyadmin* nám stačí vytvořit databázi s kódováním *utf8*. Do této databáze naimportujeme všechny tabulky a základní data ze souboru *knihovna.sql* přiloženém na instalačním mediu.

Po úspěšném vytvoření databáze nakopírujeme všechny soubory aplikace ze složky *knihovna* do adresáře nastaveného jako *document root*. Tento adresář má většinou název *www* nebo *public\_html*. Nicméně adresář je možné v případě vlastního serveru změnit pomocí konfiguračního souboru na jakýkoliv jiný. Je nutné nakopírovat úspěšně všechny soubory a neměnit adresářovou strukturu. Nyní již zbývá nastavit v souboru *includes/connect.php* přihlašovací údaje k databázi. Nastavení provedeme editací tohoto souboru například pomocí programu *PSPad*. Způsob nastavení tohoto souboru znázorňuje následující ukázka kódu.

```
$domain name http = "http://$ SERVER[SERVER NAME]";
include_once dirname(__FILE__).'/../includes/login_class.php';
session_start();
session_set_cookie_params(3600, '/', '127.0.0.1');

require_once dirname(__FILE__).'/../dibi/dibi.php';
dibi::connect(array(
    'driver' => 'mysql',          // typ použité databáze
    'host'   => 'localhost',     // server databáze
    'username' => 'root',        // uživatel
    'password' => 'jirka',       // heslo
    'database' => 'knihovna',    // jméno databáze
));

dibi::query("SET NAMES 'utf8'");
```

Syntaxe 9 – nastavení připojení k databázi pomocí DIBI

V tomto souboru nastavíme požadované údaje pro připojení k databázi podle komentářů. Jak již bylo v předchozích kapitolách zmíněno, je použita databázová nadstavba *DIBI*. Postupně nastavím typ použité databáze, server databáze většinou udávaný pomocí *IP*<sup>35</sup> adresy, jménu a heslo uživatele pro přístup do databáze. Jako poslední položku uvedeme název vytvořené databáze, v našem případě *knihovna*. Dále je nutné nastavit server pro držení cookies, jedná se o čtvrtý a pátý řádek. V některých případech je nutné pozměnit i cestu v proměnné *\$domain\_name\_http*, například pokud máme všechny soubory knihovny umístěny v podadresáři od adresáře v *document root*.

Po úspěšném zvládnutí výše uvedených kroků je aplikace funkční a k dispozici k používání. K přihlášení jako administrátor použijeme jména „*admin*“ a hesla „*adminadmin*“.

## 7 Závěr

Cílem této bakalářské práce bylo popsat a analyzovat stávající knihovní systémy. Na základě nabytých informací pak vytvořit konkrétní informační systému knihovny, který by umožňoval minimálně správu knihovního fondu, čtenářů, rezervací, výpůjček a umožňoval pohodlně vyhledávat tituly a údaje o nich. To vše prostřednictvím sítě internet.

Podarilo se mi vytvořit funkční informační systém na úrovni obecní knihovny. Knihovní systémy jsou velmi propracované a složité aplikace, které jsou vyvíjeny týmem odborníků po dobu i několik let, z tohoto důvodu systém nedosahuje profesionality popisovaných informačních systémů v této bakalářské práci. Nicméně je systém plně funkční a použitelný například pro nějakou menší školní nebo obecní knihovnu, která nemá dostatečné finance na nákup profesionálního softwaru.

Při vývoji bylo použito moderních prvků a javascriptových řešení, jak bylo popsáno v kapitole *Technologie a jazyky použité k vývoji*. Lze vyzdvihnout také například systém kontroly rezervací a výpůjček. Pomocí naplánovaných úloh jsou hlídány datумы rezervací a výpůjček a odesílány příslušné emaily uživatelům. Například v případě výpůjčky je to email s upomínkou o vypršení výpůjčky a nutnosti navrácení vypůjčeného exempláře, v případě rezervací je to informační email o dostupnosti rezervovaného exempláře. Naplánované úlohy slouží i například k vytváření datových souborů, konkrétně k vytváření souboru s daty o posledních deseti přidaných titulech. Tento soubor je načítán do úvodní stránky, nedochází tak k zatěžování databáze. Další funkcí, kterou mohu zmínit, je systém archivu rezervací a výpůjček. Každý uživatel si ve svém profilu může zobrazit jak aktuální rezervace a výpůjčky tak i již uběhlé rezervace nebo výpůjčky. Vše je názorně vidět v *Obrázku 26 - stránka uživatelského profilu*.

V budoucnu bude aplikace doladována a rozšiřována. Mezi hlavní cíle rozšíření patří vytvoření modulu pro import dat z jiných knihoven. Pokud jde o technologii pro tvorbu webových aplikací, je dalším cílem využívat více ajaxové technologie nebo využití některého profesionálního frameworku pro tvorbu aplikací.

## Použitá literatura a zdroje

**ČEGAN, Lukáš. 2010.** *Návrh a tvorba WWW stránek - Přednáška 1.* Pardubice, Česká Republika. Univerzita Pardubice.

**JAKEL, Milan. 2002.** *Stavové kódy a hlášení v odpovědi protokolu HTTP* [online] 29. 05. 2002 [Citace: 20.04.2010]. Dostupné z: <<http://interval.cz/clanky/stavove-kody-a-hlaseni-v-odpovedi-protokolu-http/>>.

**NETCRAFT, 2010.** *April 2010 Web Server Survey* [online] [Citace: 01.05.2010]. Dostupné z: <[http://news.netcraft.com/archives/web\\_server\\_survey.html](http://news.netcraft.com/archives/web_server_survey.html)>.

**HÁJEK, Petr. 2010.** *Tržní podíly internetových prohlížečů za březen 2010* [online] 02. 04. 2010 [Citace: 14.04.2010]. Dostupné z: <<http://www.maxiorel.cz/trzni-podily-internetovych-prohlizecu-za-brezen-2010>>.

**SCHURMAN, Eric. 2000.** *Dynamické HTML v akci* 1.vyd. Praha: Computer Press, 2000. 417 s. ISBN 80-7226-401-X

**SNÍŽEK, Martin. 2002.** *XHTML – základní pojmy a prvky* [online] 31.07.2002 [Citace: 24.04.2010]. Dostupné z: <[http://cs.wikipedia.org/wiki/Extensible\\_HyperText\\_Markup\\_Language](http://cs.wikipedia.org/wiki/Extensible_HyperText_Markup_Language)>.

**KOČÍ, Michal. 2000.** *Co je XML?* [online] 21.02.2000 [Citace: 24.04.2010]. Dostupné z: <[http://cs.wikipedia.org/wiki/Extensible\\_Markup\\_Language](http://cs.wikipedia.org/wiki/Extensible_Markup_Language)>.

**Kaskádové styly.** Wikipedie otevřená encyklopedie [online] 2010 [Citace: 26.04.2010]. Dostupné z: <[http://cs.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://cs.wikipedia.org/wiki/Cascading_Style_Sheets)>.

**BŘÍZA, Petr. 2004.** *Tvorba layoutu webu – teoretický úvod* [online] 03.03.2004 [Citace: 26.04.2010]. Dostupné z: <<http://interval.cz/clanky/tvorba-layoutu-webu-teoreticky-uvod/>>.

**PONKRÁČ, Miroslav. 2004.** *PHP – 25. díl – úvod do databází* [online] 29.07.2004 [Citace: 28.04.2010]. Dostupné z: <<http://www.zive.cz/clanky/php---25-dil--uvod-do-databazi/sc-3-a-118655/default.aspx>>.

**Oracle.** Wikipedie otevřená encyklopedie [online] 2010 [Citace: 28.04.2010]. Dostupné z: <<http://cs.wikipedia.org/wiki/Oracle>>.

**OLŠAVSKÝ, Marek. 2004.** *PostgreSQL (2) – Proč PostgreSQL, data a relace* [online] 01.09.2004 [Citace: 29.04.2010]. Dostupné z: <[http://www.linuxsoft.cz/article.php?id\\_article=354](http://www.linuxsoft.cz/article.php?id_article=354)>.

**Webová aplikace.** Wikipedie otevřená encyklopedie [online] 2010 [Citace: 26.04.2010]. Dostupné z: <[http://cs.wikipedia.org/wiki/Webová\\_aplikace](http://cs.wikipedia.org/wiki/Webová_aplikace)>.

**FRESCHMANN, Petr. 2008.** *Bezpečnost na webu - přehled útoků na webové aplikace* [online] 10.11.2008 [Citace: 29.04.2010]. Dostupné z: <<http://zdrojak.root.cz/clanky/prehled-utoku-na-webove-aplikace/>>.

**JURÍK, Marián. 2002.** *Knihovní systém DAIMON – řešení pro automatizaci služeb moderní knihovny* [online] 2002 [Citace: 16.04.2010]. Dostupné z: <<http://www.ikaros.cz/knihovni-system-daimon-reseni-pro-automatizaci-sluzeb-moderni-knihovny>>.

**Clavius – základní informace.** [online] 2010 [Citace: 16.04.2010]. Dostupné z: <<http://www.clavius.cz/frcla.htm?info.htm&informace>>.

**Aleph – hlavní rysy.** [online] 2010 [Citace: 16.04.2010]. Dostupné z: <<http://aleph.cuni.cz/ALEPH-12.html>>.

**KpWin SQL – charakteristika programu.** [online] 2010 [Citace: 16.04.2010]. Dostupné z: <<http://www.kpsys.cz/kpwinsql/>>.

**Dibi – Quick Start.** [online] 2010 [Citace: 26.04.2010]. Dostupné z: <<http://dibiphp.com/cs/quick-start>>.

**ŠERÝ, Richard. 2007.** *Javascript s jQuery – lehký úvod* [online] 2007 [Citace: 18.04.2010]. Dostupné z: <<http://interval.cz/clanky/javascript-s-jquery-lehky-uvod/>>.

## Příloha A – Ukázka některých zdrojových kódů aplikace

### Ukázka a popis části kódu třídy *login* v souboru *login\_class.php*

```
class login { // začátek třídy login
    var $login_name;
    var $login_pw;
    var $id_user;
    var $user_name;
    var $is_logged;
    var $session_login_string;
    var $ip;
    var $last_time;
    var $checktimelimit;
    var $table;
    var $zprava;
    var $domena;
    var $prava;

    // konstruktor třídy obsahující inicializaci některých proměnných,
    // konstruktor je v PHP funkce se stejným názvem jako má třída
    function login() {
        if(isset($_POST["login_name"])) $this->login_name=$_POST["login_name"];
        if(isset($_POST["login_pw"])) $this->login_pw=$_POST["login_pw"];
        $this->table = "uzivatele";
        $this->ip = $_SERVER["REMOTE_ADDR"];
        $this->domena = 'http://'.$_SERVER['SERVER_NAME'].'/books_mysql';

        // délka časového limitu v sekundách od posledního přístupu
        $this->checktimelimit = 3600;
    }

    // přihlášení/odmítnutí uživatele
    function first_login() {
        if (strlen($this->login_name) > 1) {
            $new_pw = md5($this->login_pw);
            $res = dibi::query("SELECT [uz_id], [uz_jmeno], [uz_id_role],
[rl_nazev] FROM [$this->table] JOIN [role] ON [role.rl_id] =
[uzivatele.uz_id_role] WHERE [uz_uziv_jmeno] = %s", $this->login_name, "
AND [uz_heslo] = %s", trim($new_pw), " AND [uz_overeni] = %i", 1, "%lmt",
1);
            if ($res) $count = count($res);
            if (isset($count) and ($count == 1)) {
                $row = $res->fetch();
                $this->session_login_string = md5(uniqid(rand()));
                dibi::query("UPDATE [$this->table] SET [uz_session] = %s", $this-
>session_login_string, ", [uz_ip] = %s", $this->ip, " WHERE [uz_uziv_jmeno]
= %s", $this->login_name, " AND [uz_heslo] = %s", $new_pw, "%lmt ", 1);
                $_SESSION["login_name"] = $this->login_name;
                $_SESSION["password"] = $new_pw;
                $_SESSION["user_name"] = $row['uz_jmeno'];
                $_SESSION["nazev_role"] = $row['rl_nazev'];
                $_SESSION["user_id"] = $row['uz_id'];
                $_SESSION["role"] = $row['uz_id_role'];
                $_SESSION["popup_login"] = 1;
                $this->load();
                return 1;
            } else {
                $this->zprava = "Špatné jméno nebo heslo.";
            }
        }
    }
}
```



```

        $_SESSION['bad_login'] = $_SESSION['bad_login'] + 1;
        return 0;
    }
} else {
    $this->zprava = "Musí být vyplněny všechny údaje.";
    $_SESSION['bad_login'] = $_SESSION['bad_login'] + 1;
    return 0;
}
}

// odhlášení uživatele
function logout() {
    dibi::query("UPDATE [$this->table] SET [uz_session] = %s",
md5(uniqid(rand())), " WHERE [uz_session] = %s", $this->
session_login_string, " AND [uz_ip] = %s", $this->ip, " %lmt ", 1);
    $result = mysql_query($query);
    $this->session_login_string = md5(uniqid(rand()));
    $this->login_name = md5(uniqid(rand()));
    session_unset();
    session_destroy();
}

```

## Část kódu vyhledávání, konkrétně logika sestavení dotazu k hledání

```

if (!isset($_SESSION['autori'])) { // jsou hledány tituly
    $order = array('t_id' => 'asc');
    $query[] = "SELECT tituly.t_id, tituly.t_nazev, tituly.t_obrazek,
CONCAT(autori.a_jmeno, ' ', autori.a_prijmeni) AS autor,
tituly.t_rok_vydani, tituly.t_signatura, tituly.t_isbn,
tituly.t_pocet_exemplaru, tituly.t_url, tituly.t_annotace FROM tituly JOIN
autori ON autori.a_id = tituly.t_id_autora";

    switch ($_SESSION['typ']) { // switch na typ vyhledání
        case 0: // všechna pole
            array_push($query, "WHERE ([t_nazev] LIKE %s", "%$_SESSION[vyraz]%",
") OR ([a_jmeno] LIKE %s", "%$_SESSION[vyraz]%", " OR [a_prijmeni] LIKE
%s", "%$_SESSION[vyraz]%", ") OR ([t_klicova_slova] LIKE %s",
"%$_SESSION[vyraz]%", ") OR ([t_isbn] LIKE %s", "%$_SESSION[vyraz]%", ")");
            break;
        case 1:
            array_push($query, "WHERE [t_nazev] LIKE %s", "%$_SESSION[vyraz]%",);
            break;
        case 2:
            array_push($query, "WHERE [a_jmeno] LIKE %s", "%$_SESSION[vyraz]%", "
OR [a_prijmeni] LIKE %s", "%$_SESSION[vyraz]%",);
            break;
        case 3:
            array_push($query, "WHERE [t_klicova_slova] LIKE %s",
"%$_SESSION[vyraz]%",);
            break;
        case 4:
            array_push($query, "WHERE [t_isbn] LIKE %s", "%$_SESSION[vyraz]%",);
            break;
    }

    if(isset($_GET['by']) and ($_GET['by'] == "nazev")) { // řazení
        $order = array('t_nazev' => $_GET['sort']);
        if($_GET['sort'] == "desc") $jak = "asc"; else $jak = "desc";
    }

    if(isset($_GET['by']) and ($_GET['by'] == "autor")) {

```

```

    $order = array('autor' => $_GET['sort']);
    if($_GET['sort'] == "desc") $jak = "asc"; else $jak = "desc";
}
if(isset($_GET['by']) and ($_GET['by'] == "vydano")) {
    $order = array('t_rok_vydani' => $_GET['sort']);
    if($_GET['sort'] == "desc") $jak = "asc"; else $jak = "desc";
}
if(isset($_GET['by']) and ($_GET['by'] == "exemplare")) {
    $order = array('t_pocet_exemplaru' => $_GET['sort']);
    if($_GET['sort'] == "desc") $jak = "asc"; else $jak = "desc";
}
if(isset($_GET['by']) and ($_GET['by'] == "sign")) {
    $order = array('t_signatura' => $_GET['sort']);
    if($_GET['sort'] == "desc") $jak = "asc"; else $jak = "desc";
}
if(isset($_GET['by']) and ($_GET['by'] == "isbn")) {
    $order = array('t_isbn' => $_GET['sort']);
    if($_GET['sort'] == "desc") $jak = "asc"; else $jak = "desc";
}

// dokončení sestavení dotazu
array_push($query, "ORDER BY %by %lmt %ofs", $order, $limit,
$_GET['side']);
$res_tituly = dibi::query($query);
$celkem_vypisu_dotazu = count($res_tituly);
} else { // jsou hledání autoři, u nich není switch na výběr typu hledání
$order = array('a_id' => 'asc');

if(isset($_GET['by']) and ($_GET['by'] == "autor")) { // řazení
    $order = array('autor' => $_GET['sort']);
    if($_GET['sort'] == "desc") $jak = "asc"; else $jak = "desc";
}
if(isset($_GET['by']) and ($_GET['by'] == "umrti")) {
    $order = array('a_rok_umrti' => $_GET['sort']);
    if($_GET['sort'] == "desc") $jak = "asc"; else $jak = "desc";
}
if(isset($_GET['by']) and ($_GET['by'] == "narozeni")) {
    $order = array('a_rok_narozeni' => $_GET['sort']);
    if($_GET['sort'] == "desc") $jak = "asc"; else $jak = "desc";
}
if(isset($_GET['by']) and ($_GET['by'] == "tituly")) {
    $order = array('a_pocet_titulu' => $_GET['sort']);
    if($_GET['sort'] == "desc") $jak = "asc"; else $jak = "desc";
}

// dokončení sestavení dotazu
$res_autori = dibi::query("SELECT a_id, CONCAT(a_jmeno, ' ', a_prijmeni)
AS autor, a_rok_narozeni, a_rok_umrti, a_pocet_titulu, a_popis, a_obr,
a_url FROM autori WHERE a_jmeno LIKE %s", $_SESSION['vyraz'], " OR
a_prijmeni LIKE %s", $_SESSION['vyraz'], " ORDER BY %by %lmt %ofs", $order,
$limit, $_GET['side']);
$celkem_vypisu_dotazu = count($res_autori);
}

```