

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Knihovní systém pro obec Hostomice pod Brdy
Petr Šimek

Bakalářská práce

2009

University of Pardubice
Faculty of Electrical Engineering and Informatics

Library system for Hostomice pod Brdy
Petr Šimek

Bachelor work
2009

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Katedra informačních technologií
Akademický rok: 2008/2009

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Petr ŠIMEK**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**

Název tématu: **Knihovní systém pro obec Hostomice pod Brdy**

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

V teoretické části bakalářské práce bude představena a zhodnocena problematika systémů spravujících knihovnickou agendu. Budou představeny dostupné knihovní systémy a následně zhodnocena jejich funkčnost. V aplikační části práce bude vytvořen knihovní systém pro obec Hostomice pod Brdy. Aplikace bude umožňovat veškerou správu každodenní knihovnické práce včetně on-line objednávání a rezervace knih.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

Marianne Hauser, Tobias Hauser, Christian Wenz, "HTML a CSS - Velká kniha řešení", ISBN: 80-251-1117-2

Luke Welling, Laura Thomson, "MySQL -Průvodce základy databázového systému", ISBN: 80-251-0671-3

Lukáš Krejčí, "PHP-Kapesní přehled", ISBN: 80-251-0808-2

Vedoucí bakalářské práce:

Ing. Lukáš Čegan

Katedra informačních technologií

Datum zadání bakalářské práce: **15. ledna 2009**

Termín odevzdání bakalářské práce: **15. května 2009**



doc. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Lukáš Čegan
vedoucí katedry

V Pardubicích dne 31. března 2009

PROHLÁŠENÍ

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

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. 2009

Petr Šimek

PODĚKOVÁNÍ

Touto formou bych chtěl poděkovat panu ing. Lukáši Čeganovi za mnoho rad, věcných připomínek a návrhů k bakalářské práci. Zároveň bych chtěl poděkovat slečně ing. Blance Jankovské, která mi pomohla se blíže seznámit s problematikou knihovních systémů.

ANOTACE

V této bakalářské práci je v teoretické části zhodnocena problematika knihovních systémů. Součástí bakalářské práce je vyvinutá webová aplikace – knihovní systém pro obec Hostomice pod Brdy. Tato aplikace si klade za cíl umožnit knihovníkům spravovat bázi knihovny a poskytovat její záznamy čtenářům prostřednictvím sítě Internet.

KLÍČOVÁ SLOVA

knihovní systémy, webové aplikace, PHP, SQL

TITLE

Library system for city Hostomice pod Brdy

ANNOTATION

The theoretic part of this bachelor thesis deals with problems of library systems. Then the work involves developing of web application which implements library system for municipality Hostomice pod Brdy. The aim of this application is to allow the librarians to manage library resources and to provide its entries to readers via Internet.

KEYWORDS

library systems, web applications, PHP, SQL

OBSAH

1	Úvod.....	12
2	Webová aplikace	13
2.1	Moderní webová aplikace	14
2.2	Kritéria úspěšné webové aplikace	14
3	Programové vybavení.....	16
3.1	Vývojové prostředí.....	16
3.2	Nástroje pro tvorbu aplikací.....	16
3.3	Nástroje pro provoz aplikací na lokální stanici.....	16
4	Jazyky a technologie	17
4.1	Značkovací jazyky	17
4.1.1	Popis značkovacího jazyka HTML	18
4.2	Kaskádové styly	18
4.3	Grafické prvky	19
4.4	Skriptovací jazyky.....	19
4.5	Databázové systémy.....	20
5	Problematika knihovních systémů	22
6	Základní moduly knihovních systémů	23
6.1	Akviziční modul.....	23
6.2	Katalogizační modul	23
6.3	Výpůjční modul.....	23
6.4	OPAC	24
6.5	Z39.50 klient, server	24
6.6	Správa periodik	24
6.7	Správa uživatelů	24
7	Popis vybraných knihovních systémů	25
8	Analýza, specifikace cílů	26
8.1	Sběr informací o knihovních systémech	26
9	Návrh aplikace	28
9.1	Analýza požadavků systému	28
9.2	Relační schéma.....	29
9.3	Návrh vzhledu aplikace.....	30
10	Vývoj aplikace	31
10.1	Prostředky použité při vývoji.....	31
10.1.1	Nástroje	31
10.1.2	Jazyky.....	32
10.2	Databáze	32
10.2.1	Migrace dat.....	33
10.3	Logické členění aplikace.....	34
10.4	Uživatelské rozhraní.....	35
10.4.1	Vzhled aplikace.....	36
11	Popis aplikace.....	37
11.1	Funkce pro knihovníka.....	37
11.2	Funkce pro čtenáře	39
11.3	Ostatní funkce systému	39
12	Běžný provoz aplikace	40
13	Požadavky aplikace.....	41

13.1	Technické požadavky na provoz aplikace.....	41
13.2	Požadavky na uživatele	42
13.3	Instalace aplikace	42
14	Testování a optimalizace.....	43
14.1	Zkušební verze knihovního systému.....	43
15	Možná vylepšení	43
16	Závěr	44
17	Použitá literatura	45
18	Seznam příloh.....	47

SEZNAM OBRÁZKŮ

Obrázek č.1: Schéma webové aplikace.....	13
Obrázek č.2: Vývoj značkovacích jazyků.....	17
Obrázek č.3: Princip zpracování skriptů.....	20
Obrázek č.4: Schéma klasického knihovního systému (systém Daimon).....	22
Obrázek č.5: Rich picture knihovního systému.....	28
Obrázek č.6: Návrh vzhledu aplikace.....	30
Obrázek č.7: Náhled na hotovou webovou aplikaci.....	31
Obrázek č.8: Migrace dat.....	33
Obrázek č.9: Rozmístění prvků webové aplikace.....	36
Obrázek č.10: Srovnání bezplatných hostingů.....	41

SEZNAM ZKRATEK

- AV – Audio – Video, obraz zvuk
- ASP – Active Server Pages, aktivní serverové stránky
- CD – Compact Disc, kompaktní disk
- CSV – Comma-separated values, hodnoty oddělené čárkami
- DOS – Disc Operating System, diskový operační systém
- E-mail – Electronic Mail, elektronická pošta
- GIF – Graphic Interchange Format
- GPRS – General Packet Radio Service, rádiový přenos datových paketů
- HTML – Hyper Text Markup Language, odkazový značkovací jazyk
- ISAM – Indexed Sequential Access Method, indexová sekvenční přístupová metoda
- ISBN – International Standard Book Number, mezinárodní standardní číslo knihy
- JPEG – Joint Photographic Expert Group
- PC – Personal Computer, osobní počítač
- PDA – Personal Digital Assistant, osobní digitální pomocník
- PHP – Personal Home Page, osobní domácí stránka
- PNG – Portable Graphic Network
- SQL – Structured Query Language, strukturovaný dotazovací jazyk

1 ÚVOD

Obecní knihovna v *Hostomicích pod Brdy* eviduje v současné době tisíce *jednotek knihovního fondu (dále jen titulů)* pomocí tabulkového procesoru Excel. Tento přístup je výhodný v tom, že při umístění titulu na webové stránky knihovny¹ mají čtenáři přístup k informacím o titulech. Zásadní problém je v tom, že ne každý čtenář vlastní program Excel a uložení dat v tabulce není nijak optimalizováno pro vyhledávání jednotlivých titulů. Současně tato forma prezentace titulů knihovny nenabízí dostatečný komfort čtenářům při zjišťování potřebných informací o titulech.

Rozhodl jsem se, že vytvořím webovou aplikaci, která by tyto nedostatky řešila. Chtěl jsem vytvořit takový informační systém, který by pomohl knihovníkovi při každodenní správě titulů knihovny a zároveň nabídl komfort čerpání informací o titulech čtenáři z jakéhokoliv osobního počítače připojeného k celosvětové síti Internet.

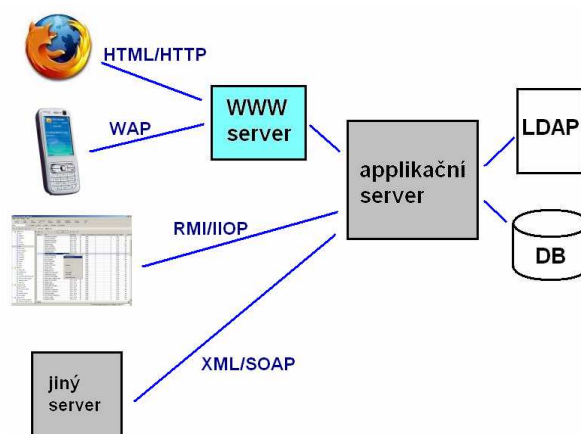
V teoretické části této bakalářské práce se zabývám obecně webovými aplikacemi. Dále pak nástroji a technologiemi pomocí nichž lze webové aplikace vyvíjet. Čtenář této bakalářské práce se dále seznámí s problematikou knihovních systémů a získá základní přehled o dnešních knihovních systémech. V praktické části jsou podrobně popsány jednotlivé fáze procesu tvorby knihovního systému a to od jeho návrhu až po zhodnocení výsledků práce.

¹ <http://www.knihovnahostomice.cz>

2 WEBOVÁ APLIKACE

Webová aplikace je informační systém, který využívá ke svému provozu buď síť Internet, nebo podnikovou síť intranet. Jak vyplývá z *obrázku č.1* uvedeného níže, tak k aplikaci přistupuje koncový uživatel prostřednictvím internetového prohlížeče v osobním počítači, případně PDA. Jeho požadavek dále zpracují servery, které komunikují s případnou databází a výsledek jeho požadavku se uživateli zobrazí na monitoru jeho počítače.

Řešení informačního systému pomocí webové aplikace umístěné na webovém serveru je výhodné v tom, že k systému má přístup každý, kdo má přístup k síti Internet. Další výhodou je, že se jedná o levné řešení, protože systém vyžaduje pouze webový prostor, kde je aplikace umístěna a počítač připojený na Internet ze kterého se aplikace ovládá. Za předpokladu dobře navržené a validní aplikace lze aplikaci spouštět na všech podporovaných počítačových platformách s internetovým prohlížečem a připojením k Internetu (PC, PDA). Oproti těmto výhodám klade veřejný přístup k aplikaci další nároky na programátora aplikace a to především na to, aby zajistil její bezpečnost. Cílem útoku jsou především data uložená v databázi. Napadnout aplikaci může kdokoliv a existuje mnoho technik napadení. Mezi nejčastější útoky patří útok na databázi pomocí SQL Injection¹ nebo Session stealing² [1].



Obrázek č.1: Schéma webové aplikace³

¹ Technika napadení databáze

² Technika zcizení globální proměnné

³ Zdroj: http://kore.fi.muni.cz:5080/wiki/index.php/Webov%C3%A9_aplikace

2.1 Moderní webová aplikace

Moderní webová aplikace se skládá z těchto logicky oddělených částí:

- uživatelský interface
- logika aplikace,
- datová vrstva.

Uživatelský interface, neboli uživatelské rozhraní je část aplikace se kterou uživatel přijde do styku nejdříve, proto na ni vždy kladu největší důraz. Logická vrstva řídí provoz aplikace a využívá datovou základnu, která jí poskytuje data. Díky oddělení těchto vrstev je webová aplikace rychlejší, je možné ji flexibilně upravovat, dobře se pracuje s daty atd. Každou z těchto vrstev lze vytvářet pomocí jiného vývojového nástroje.

2.2 Kritéria úspěšné webové aplikace

Aby aplikace uspěla v dnešním konkurenčním prostředí, je potřeba dodržet několik zásad pro úspěšnou aplikaci.

První zásadou je, že *aplikace musí být bezpečná*. Při každém uživatelském vstupu se musí vstup zkontrolovat a vymezit možné hodnoty, které nenaruší provoz aplikace. V případě chybného vstupu aplikace uživateli zobrazí chybovou hlášku, případně ho přesměruje na jinou stránku aplikace. Přehled útoků na aplikaci:

- injection a SQL Injection,
- spouštění nebo načtení souboru,
- krádež session,
- cross site Scripting,
- cross site Request Forgery¹ [2].

Další kritériem úspěšné aplikace je to, že musí být *optimalizovaná na přenos dat* v pomalém prostředí, čímž i v současné době v některých případech Internet je. V současné době se začínají lidé připojovat k Internetu pomocí mobilních zařízení, která většinou využívají k přenosu dat technologii GPRS s pomalým datovým přenosem. Někteří uživatelé se zase mohou připojovat k Internetu pomocí vytáčeného tele-

¹ Více na <http://zdrojak.root.cz/clanky/prehled-utoku-na-webove-aplikace>

fonického připojení jenž rovněž přenáší data poměrně pomalu. V případě, že není aplikace optimalizovaná na přenos dat v pomalém prostředí, tak uživatel čeká na výsledek své akce v aplikaci dlouho a aplikaci raději ukončí, případně s ní přestane později pracovat.

Neméně důležitými aspekty úspěšné webové aplikace je její *snadné ovládání a příjemný vzhled* optimalizovaný pro zobrazení ve všech nejpoužívanějších internetových prohlížečích. Zároveň by měli vývojáři počítat s tím, že někteří uživatelé jsou nějakým způsobem hendikepovaní a měli by umožnit i těmto uživatelům aplikaci využívat.

Pro případné budoucí vývojáře aplikace je důležitá *přehlednost aplikace*. Přehled a systém musí být v adresářích a souborech ze kterých se aplikace skládá. Dále pak v databázi, která musí být konzistentní a musí vykazovat alespoň třetí normální formu. Zdrojový kód aplikace by měl být čitelný a dobře strukturovaný.

3 PROGRAMOVÉ VYBAVENÍ

3.1 Vývojové prostředí

Vývojová prostředí jsou specializované programy, které slouží programátorům při vývoji jejich programů. Vždy obsahují editor kódu, kompilátor a případně interpret. Například pro vývoj programů v programovacím jazyce PHP lze využít Zend Studio¹, v programovacím jazyce Java vývojové prostředí JDeveloper² [15].

3.2 Nástroje pro tvorbu aplikací

Pro editaci značkovacího a skriptovacího jazyka se nemusí žádné nástroje nijak instalovat, protože stačí textový editor operačního systému Windows. Doporučuji ovšem zvolit některý editor specializovaný na editaci kódu těchto jazyků, který barevně odliší syntaxi. V kódu se poté lépe orientuje.

Grafické editory slouží k editaci obrázků, lze je rozdělit na editory bitmapové a rastrové. Mezi nejznámější rastrové editory patří editor Adobe Photoshop³ a Gimp⁴, nejznámější vektorový editor je pravděpodobně program CorelDraw⁵. Vektorové editory se používají především při vytváření loga, schémat a diagramů. Bitmapové editory se zase používají při zpracování digitálních fotografií a ilustrací [16], [17].

Dalším nástrojem, který je užitečný, je specializovaný nástroj pro návrh databáze. Nejznámější je pravděpodobně Toad Data Modeler⁶.

3.3 Nástroje pro provoz aplikací na lokální stanici

Pro podporu skriptovacího jazyka PHP a databáze MySQL a pro spouštění skriptů na lokálním počítači je výhodné využít některý z balíčků. Například balíček Xampp.⁷ Tento balíček se nainstaluje jako Windows aplikace a po restartování systému vytvoří z osobního počítače server sloužící ke spouštění PHP skriptů.

¹ <http://www.zend.com/products/studio/>

² <http://www.oracle.com/technology/products/jdev/index.html>

³ <http://www.adobe.com/cz/products/photoshop/family/>

⁴ <http://www.gimp.cz/>

⁵ <http://apps.corel.com/int/cz/index.html>

⁶ <http://www.casestudio.com/csy/>

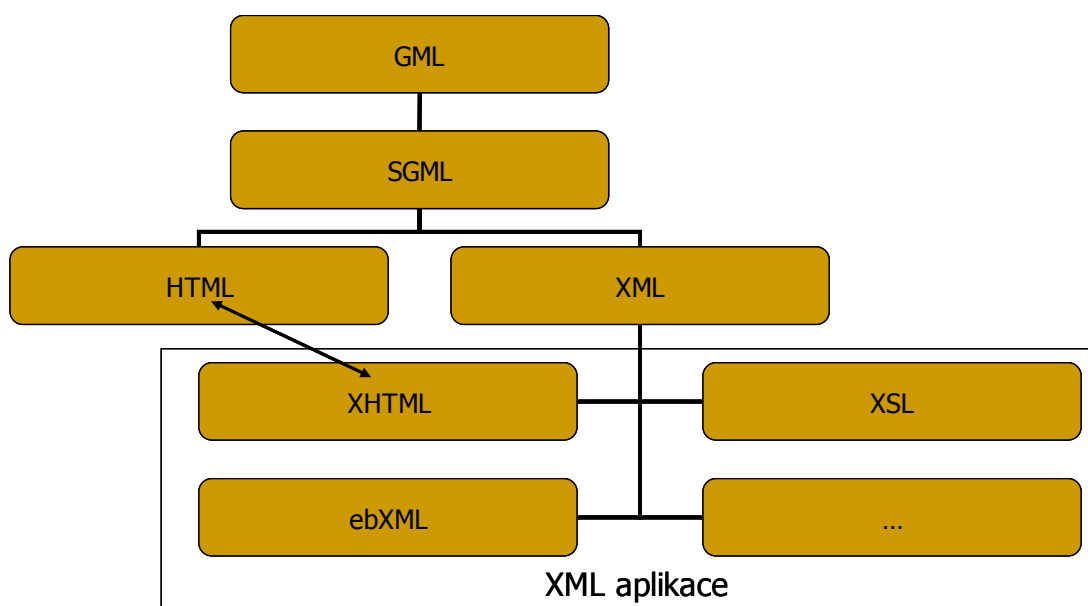
⁷ <http://www.apachefriends.org/en/xampp.html>

4 JAZYKY A TECHNOLOGIE

Webové aplikace je možné vyvíjet prostřednictvím několika různých jazyků a technologií. Jsou to značkovací jazyk, skriptovací a programovací jazyk a databáze. Nepovinné, ale důležité pro výsledný efekt aplikace jsou kaskádové styly a grafické prvky vytvořené v nějakém grafickém editoru.

4.1 Značkovací jazyky

Zdrojový kód každé webové stránky je tvořen tzv. značkami, které předepisují jak bude stránka vypadat v internetovém prohlížeči. Na vývoji značkovacích jazyků se podílejí především vývojáři internetových prohlížečů – hlavně Microsoft a Netscape. *W3C (WWW Consortium)*¹ vzniklo za účelem sjednocení značek značkovacího jazyka pomocí kterých je webová stránka tvořena a schvaluje konečnou podobu značkovacího jazyka. Značkovací jazyky mají několik verzí, které se liší v různých úpravách jimiž značkovací jazyk prošel v důsledku svého vývoje. *Validitu*² webových stránek je možné zkontrolovat pomocí validátoru³ [13].



Obrázek č.2: Vývoj značkovacích jazyků,
zdroj: https://akela.mendelu.cz/~malo/wa/tyden_2/prezentace/WA_XML.ppt

¹ <http://www.w3.org/>

² Webová stránka odpovídá platným standardům podle W3C

³ <http://validator.w3.org/>

Často používané značkovací jazyky v dnešní době jsou:

- HTML,
- XML,
- XHTML.

4.1.1 Popis značkovacího jazyka HTML

Jazyk HTML je tvořen skupinou značek a případně jejich atributy. Do těchto značek se vkládá text dokumentu a společně tak tvoří celek, který lze nazvat internetová stránka. Značky se dělí na párové a nepárové. Párové značky dále dělíme na otevírací a zavírací. Jak jsem již uvedl značky mohou obsahovat doplňující atributy, které upřesňují jejich vlastnosti [4]. Příklad zdrojového kódu jednoduché webové stránky obsahující základní značky jazyka HTML je uveden v následující syntaxi.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Hlavička</title>
  </head>
  <body>
    <h1>Nadpis stránky</h1>
    <br>
    <p font color="red">Odstavec</p>
    Nenaformátovaný text
  </body>
</html>
```

Syntaxe č.1: Syntaxe značkovacího jazyka HTML

Význam značek uvedených v *Syntaxi č.1*:

- html – deklarace stránky,
- head – hlavička stránky,
- body – tělo stránky,
- h1 – nadpis,
- br – nový řádek,
- p – odstavec, v tomto případě červeného písma.

4.2 Kaskádové styly

Kaskádové styly vznikly mimo jiné za účelem oddělení samotného textu webové stránky od jejího vzhledu. Dříve se vzhled stránky určoval atributy značek

značkovacího jazyka HTML, v současné době se ale od tohoto přístupu upustilo a moderní webové stránky mají vzhled formátovaný pomocí kaskádových stylů.

Základní výhody formátování stránek pomocí kaskádových stylů jsou:

- Rychlejší načítání stránek.
- Zdrojový kód webové stránky je přehlednější, protože je oproštěn od formátování.
- Vzhled všech stránek je možné upravit na jednom místě – v souboru kaskádových stylů.
- Prvky webové stránky je možné umístit jinak než jak jsou definované ve zdrojovém kódu [5].

4.3 Grafické prvky

Grafické prvky slouží k oživení webové stránky a k zdůraznění některých důležitých částí stránky na které chceme návštěvníka stránek upozornit. Dříve byly webové stránky pouze nositeli textové informace a obrázky sloužily pouze jako doplňková informace pro text. V současné době při takovém množství stránek, které Internet nabízí, je potřeba svoji stránku odlišit od ostatních. Hlavní „zbraní“ v boji s konkurencí je zejména odlišný a originální vzhled tvořen obrázky a webovou grafikou. Obrázky tedy nejsou již jen nositeli informace, ale doplňují design webové stránky.

Grafické prvky je nejvhodnější umístit na stránky ve formátu *JPEG*, *GIF*, nebo *PNG*. Každý z těchto formátů je uzpůsoben k jinému použití. Formát *GIF* se používá u jednoduchých obrázků, které mají málo barevných přechodů. *PNG* se používá tehdy, když chce vývojář zajistit částečnou průhlednost obrázku. *JPEG* se používá při umístitování graficky složitějších obrázků na stránky.

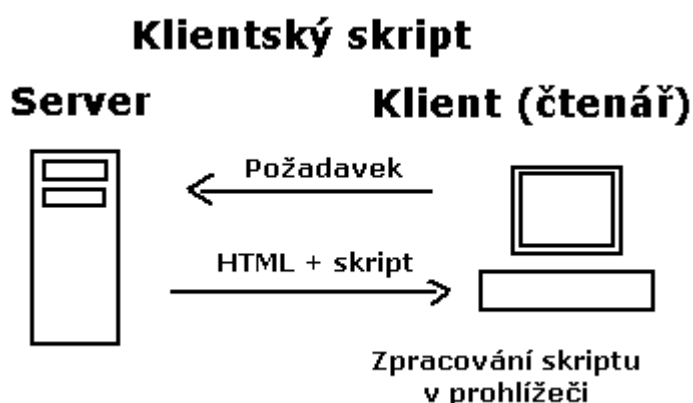
Vždy je ale důležité zvolit ten formát grafických prvků kterým dosáhneme toho co chceme udělat a zároveň zajistíme co nejnížší kompresní poměr obrázku.

4.4 Skriptovací jazyky

Logická část aplikace je tvořena tzv. skripty. Skript je zpracován za běhu interpretem, není potřeba ho nijak dopředu kompilovat. Skriptovací jazyky se dělí na dvě hlavní skupiny. Do první skupiny kde jsou skripty vykonávány na *straně serveru* patří například PHP nebo ASP. Druhou skupinu tvoří jazyky kde jsou skripty

zpracovány na *straně klienta*. Do této skupiny skriptovacích jazyků patří například JavaScript.

Rozdíl mezi skripty, které zpracovává server a klient, je znázorněn na *obrázku č.3*. V případě skriptovacího jazyka, který zpracovává skripty na straně serveru, je nejdříve odeslán požadavek na server, kde se skript zpracuje a zpět se vrací výsledek skriptu klientovi (čtenáři). Pokud je spuštěn skript, který je zpracován na straně klienta, je skript proveden přímo na lokální stanici.



*Obrázek č 3: Princip zpracování skriptů,
zdroj: <http://www.jakpsatweb.cz/javascript/javascript-uvod.html>*

Mezi nejpoužívanější skriptovací jazyky v současné době patří jazyk PHP. Skriptovací jazyk PHP je určený pro programování dynamických stránek. Skripty PHP se vykonávají na straně serveru a zpět k uživateli se přenášejí zpracovaná data na základě logiky skriptu a případně zadaných uživatelských vstupů. Jazyk PHP kvalitně podporuje databáze SQL [6].

```
<?PHP
    echo "Žiju!";
?>
```

Syntaxe č.2: Syntaxe skriptovacího jazyka

4.5 Databázové systémy

Databáze je základní součástí aplikace. Pokud je špatně navržena datová základna, aplikace bude s největší pravděpodobností vykazovat při provozu chyby. Databáze musí vyhovovat integritním omezením. *Integritní omezení* omezuje vkládané hodnoty na určitý datový typ, počet znaků. Může omezovat hodnoty tak, že budou unikátní aj. Speciálním případem integrity dat je referenční integrita. Referenční

integrita nám zajišťuje to, že řádce v jedné tabulce odpovídá řádka popřípadě řádky v jiné tabulce. Pokud se tato podmínka nedodrží a vymaže se, nebo se změní hodnota primárního klíče, vznikne nekonzistence dat a řádky nebude možné již spojit.

Nejvýznamnější databázové systémy jsou:

- MySQL,
- Oracle,
- PostgreSQL,
- Microsoft SQL Server.

Jedním z nejpoužívanější databázových systémů je v současné době *MySQL*. Databázový systém MySQL je dostupný buď v placené nebo neplacené verzi. V současné době se často používá k provozu webových aplikací kombinace jazyka PHP, MySQL a softwarového webového severu Apache. MySQL je optimalizován především na rychlost. Z tohoto důvodu donedávna nepodporoval některé důležité funkce (triggery, procedury) databázových systémů [14].

Často používané formáty tabulek v databázích SQL jsou dva. Je to formát *MyISAM* a *InnoDB*. Formát *MyISAM* sice nepodporuje cizí klíče a transakce, ale zato je mnohem rychlejší při častých dotazech na výběr dat z databáze než typ *InnoDB* [7].

```
SELECT * FROM zam WHERE zam.id='1'
```

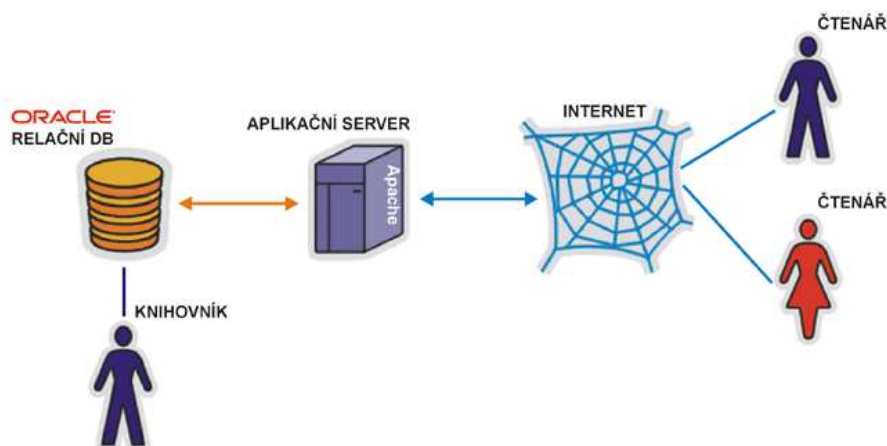
Syntaxe č.3: Syntaxe jazyka SQL

5 PROBLEMATIKA KNIHOVNÍCH SYSTÉMŮ

Knihovní systémy jsou velice komplexní a složité systémy, které mají nahradit evidenci knih a jejich správu ruční formou. Z dlouholetých zkušeností lze poznat, že knihovní systémy jsou užitečné, potřebné a veřejností žádané. Výhody, které správa titulů knihovny pomocí knihovních systémů nabízí:

- Online přístup k titulům,
- lepší využití fondů,
- vylepšení knihovnických služeb,
- poskytnutí dalších výhod za pomoci moderních informačních technologií [8].

Knihovní systém *Clavius*¹, který využívá např. knihovna v okresním městě Příbrami obsahuje několik modulů, které jsou navzájem spolu svázané. Stejný stupeň propracovanosti má i knihovní systém *Daimon*², který je využíván například v knihovně Univerzity Pardubice. Z obrázku č.4, který je uveden níže a popisuje princip práce systému Daimon lze poznat, že knihovník pracuje přímo se systémem pomocí lokální aplikace, která ukládá data do databáze (v tomto případě Oracle³). Aplikace je nainstalována na aplikačním serveru. Čtenáři přistupují k systému pomocí sítě Internet. Tímto způsobem je řešena většina knihovních systémů.



Obrázek č.4: Schéma klasického knihovního systému (systém Daimon)⁴

¹ <http://www.clavius.cz/>

² <http://www.sefira.cz/reseni/daimon/daimon.htm>

³ <http://www.oracle.com/global/cz/index.html>

⁴ Zdroj: http://www.sefira.cz/reseni/daimon/doc/20070118%20DAIMON_popis%20systemu.pdf

6 ZÁKLADNÍ MODULY KNIHOVNÍCH SYSTÉMŮ

Základní moduly profesionálního moderního knihovního systému jsou:

- Akviziční modul,
- Katalogizační modul,
- Výpůjční modul,
- OPAC,
- Z39.50 client,
- Správa periodik,
- Správa uživatelů [9].

6.1 Akviziční modul

Akviziční modul slouží k získávání nových titulů od dodavatelů. Odběratel – tedy knihovna provádí operace spojené s nákupem nových titulů, může měnit a spravovat své dodavatele. Dále si odběratel definuje svůj rozpočet, aby ho nemohl překročit. Knihovní systém sleduje termíny dodání titulů do knihovny [10].

6.2 Katalogizační modul

Katalogizační modul slouží ke správě titulů v databázi. Tituly, které lze obvykle vkládat jsou: monografie¹, AV media, hudebniny, periodika². Mezi těmito tituly lze v systému vytvářet vazby. Pokud monografie obsahuje disk CD, lze přiřadit disk k této monografii aby uživatel knihovního systému věděl, že monografie obsahuje ještě tuto přílohu a lze si ji vypůjčit. Při vkládání nových titulů jsou obvykle předem definované formuláře, které obsahují jen ty atributy, které mají. Monografie například má atribut počet stran, kdežto AV medium nikoliv [10].

6.3 Výpůjční modul

Výpůjční modul obsahuje evidenci čtenářů, jejich výpůjček, upomínek a rezervací, dále případných poplatků nebo pokut čtenářům. Každý čtenář je evidován jako samostatná položka, která obsahuje jeho jméno, příjmení, adresu, telefon, email

¹ Monografie je bibliografický zdroj, který je složen z jedné části nebo konečného počtu částí a zpracovává jedno téma.

² Periodikum je tiskovina která vychází opakovaně

a případné další doplňující údaje. Rezervace je služba, která slouží k rezervaci titulu v případě, že je titul již vypůjčený. V případě, že je titul k dispozici je čtenář, který má na titul rezervaci obvykle upozorněn o přítomnosti titulu prostřednictvím e-mailu. Funkce výpůjčky sleduje aktuální výpůjčky čtenáře. Upomínky zase evidují tituly, které má čtenář ještě půjčené a překročil výpůjční dobu [10].

6.4 OPAC

Díky této službě jsou čtenáři zpřístupněny všechny základní funkce systému, které potřebuje. Přístup pro čtenáře se dělí podle toho zda je registrovaný, nebo není. V případě, že čtenář registrovaný není, tak může tituly vyhledávat, zobrazovat jejich detaily a zjišťovat různé informace o těchto titulech. Pokud se jedná o registrovaného čtenáře, tak může navíc například spravovat své výpůjčky, objednávat, rezervovat a prodlužovat výpůjčky. Dále může sledovat své konto, měnit své osobní údaje a nastavení jeho www katalogu [10].

6.5 Z39.50 klient, server

Pomocí protokolu Z39.50 je možné komunikovat s jinou knihovnou. Knihovna může ukládat do své databáze autoritní a bibliografické záznamy z jiné knihovny. Zároveň je možné, aby ostatní knihovny přistupovali do knihovny a ukládali si záznamy, které má knihovna k dispozici [9].

6.6 Správa periodik

Periodika jsou zvláštní typy dokumentů. Z toho důvodu se většinou evidují jiným systémem než ostatní tituly. Periodika se především liší (například od monografií) jejich pravidelným publikováním a tudíž velkým počtem kusů, které je nutné evidovat [9].

6.7 Správa uživatelů

Modul Správa uživatelů umožňuje správci systému vkládat a spravovat nové uživatele a to jak čtenáře, tak knihovníky. Dále má umožněno přidělovat knihovníkům přístupová práva [9].

7 POPIS VYBRANÝCH KNIHOVNÍCH SYSTÉMŮ

Úspěšných knihovních systémů na českém trhu není mnoho. Je to z důvodů vysoké specializace a nároků kladených na dlouholetý vývoj.

Mezi nejznámější knihovní systémy patří knihovní systém *Clavius* pracující v prostředí Windows. Tento systém je nepřímým následníkem systému *LANius* od stejnojmenné firmy, který pracuje v prostředí DOS. Firma LANius ale upozorňuje, že systém *Clavius* není odvozený ze systému *LANius*. Knihovní systém *Clavius* nabízí neomezenou délku vstupních polí. Do systému lze vkládat všechny typy titulů. Knihovní systém pracuje v prostředí Windows, je intuitivní a tudíž uživatel je produktivní při práci s tímto systémem [10].

Dalším významným knihovním systémem je systém *Daimon*. Tento systém vytvořila společnost Sefira spol. s r. o., která se zaměřuje především na tvorbu specializovaných informačních systémů. Systém *Daimon* je určený pro velké a středně velké knihovny, přičemž se specializuje na vysokoškolské knihovny. Výhodou systému *Daimon* je využívání databáze Oracle, která je špičkou v problematice databázových systémů [9].

Firma Kosmotron s.r.o vyvinula systém *Rapid Library*. Tento systém slouží k administraci titulů různých typů v síťovém prostředí. Jedná se o moderní systém, který nabízí flexibilitu, modulárnost, otevřenost, spolehlivost. *Rapid Library* je schopen pokrýt všechny požadavky na moderní knihovní systém. Jádro systému se rovněž skládá z modulů knihovních systémů, které jsem uvedl výše. Systém je dodáván ve verzi, která obsahuje základní moduly, ale tyto moduly lze rozšířit [11].

Další knihovní systém o kterém bych se chtěl zmínit je knihovní systém *DAWINCI* firmy ASP a.s. Tento knihovní systém se objevil na českém trhu v roce 2001. Kromě všech základních modulů nabízí systém ještě doplňkové služby např. full textové vyhledávání, redakční systém NEMESIS pro správu samotných webových stránek [12].

Izraelská firma Ex Libris vyvinula knihovní systém *ALEPH*. Systém *ALEPH* patří mezi nejmodernější knihovní systémy, je možné ho využít pro jakkoliv velkou knihovnu. Tento systém využívá Národní knihovny ČR a chce ho začít používat v blízké budoucnosti i knihovna Univerzity Pardubice [18].

8 ANALÝZA, SPECIFIKACE CÍLŮ

Aplikace si klade za cíl nahradit stávající evidenci knih v městské knihovně obce *Hostomice pod Brdy*, kde jsou v současné době tituly evidovány pomocí programu Excel. Je nutné zajistit pomocí nové aplikace běžnou správu a evidenci knih a zároveň nabídnout čtenářům možnost zjišťovat informace o dostupných titulech v knihovně.

Aplikace se bude skládat ze dvou základních částí a to z redakčního systému a knihovního systému. Redakční systém není součástí této bakalářské práce, ale je nezbytnou součástí realizace samotného knihovního systému. Knihovní systém je tedy nadstavbou redakčního systému, redakční systém nebude v této bakalářské práci podrobněji popisován. Tyto dvě části tvoří jeden celek, který bude samozřejmě kompletně předán knihovně.

Aplikaci budou ovládat dva typy uživatelů. Bude to správce respektive knihovník a čtenáři. Oba dva typy uživatelů budou moci ovládat systém pomocí stejného uživatelského rozhraní. Knihovní systém bude řešen jako model klient-server a bude umístěn na Internetu a to na některém z českých neplacených webových hostingů.

Z časových důvodů neplánuji to, že aplikace bude obsahovat všechny funkce, které mají ostatní knihovní systémy, ale bude možné aplikaci o tyto funkce rozšířit.

8.1 Sběr informací o knihovních systémech

Pro úspěšný vývoj samotné aplikace jsem se rozhodl nejdříve zjistit jak jsou realizovány současné knihovní systémy. Z toho důvodu jsem navštívil městskou knihovnu v Hořovicích a v Příbrami, kde mi techničtí pracovníci poskytli cenné informace. Následně jsem zjišťoval jak je knihovní systém realizován v knihovně Univerzity Pardubice, kde mi poskytla cenné informace slečna ing. Jankovská.

Nejdříve jsem zjišťoval informace o tom jaké typy titulů a jaké údaje se u nich evidují. Většina knihoven rozlišuje dokumenty na *knihy, periodika a různé druhy médií*. Každý druh dokumentu má kromě společných atributů i některé atributy odlišné od ostatních typů dokumentů. Důležité společné atributy jsou zejména ty číselné. V knihovně je evidován každý exemplář pomocí tzv. *přírůstkového čísla*. Toto číslo je jedinečné a v knihovně se pomocí něho například zjišťuje přírůstek

exemplářů v knihovně a to tak, že se odečte přírůstkové číslo zjištěné z posledního sčítání od přírůstkového čísla posledního exempláře. Dále slouží toto číslo k jednoznačné identifikaci každého exempláře v knihovně. Čtenáři vyhledávají tituly v každé větší knihovně pomocí *mezinárodního desetinného třídění* a pomocí tzv. *signatury*, která upřesňuje polohu titulu v regálu. Zde jsem ale zjistil, že se systémy značení v některých knihovnách liší. Některé menší knihovny například evidují pouze dvě čísla a to přírůstkové číslo a signaturu. Tento rozdíl značení v knihovnách je z toho důvodu, že některé knihovny jsou malé a nepotřebují používat složitější systém značení, protože čtenář knihu bez problému najde.

Různé druhy dokumentů mají některé atributy odlišné. U knih se například eviduje číslo ISBN a počet stran. V případě periodik nikoliv.

Následně jsem zjišťoval jaké funkce mají současné knihovní systémy. Zjistil jsem, že většina knihoven využívá systémy, které obsahují *funkce*, jež jsou popsány v *kapitole 6*. Seznámil jsem se rovněž s běžným provozem systémů jak z pohledu knihovníka, tak z pohledu čtenáře. Po těchto zkušenostech jsem navštívil knihovnu v Hostomicích pod Brdy a s paní Lisou, která je místní knihovnicí, jsem daný problém zkonzultoval a navrhl jí řešení knihovního systému.

9 NÁVRH APLIKACE

Nejdříve bylo nutné definovat interakci knihovního systému s jeho okolím. K zachycení těchto vazeb lze využít nákres, který se nazývá „Rich picture“. Tento nákres je vytvářen vývojářem aplikace na základě zjištěných informací o tom jak systém bude komunikovat s jeho okolím. Po této části návrhu jsem zjišťoval jaké entity mají být v systému zachyceny a jaké mají mít atributy a vzájemné vztahy. Dále bylo nutné zjistit jaké základní funkce má aplikace poskytovat a jaké má představy o vzhledu aplikace zákazník. Na základě těchto získaných informací jsem mohl začít navrhovat vzhled, databázi, logiku a další součásti aplikace.

9.1 Analýza požadavků systému

Jak je zřejmé z obrázku č.5, tak do systému zasahují tři typy osob: *administrátor, knihovník a čtenáři*. Nejvyšší práva má administrátor, nižší knihovník a nejnižší čtenář, přičemž uživatel s vyššími právy má přístup k funkcím uživatele s právy nižšími. Administrátor spravuje aplikaci, knihovník se stará o správu titulů a čtenáři mohou ze systému získávat potřebné informace o titulech. Předpokládá se, že knihovna bude spolupracovat s některou jinou knihovnou a budou si vzájemně vyměňovat záznamy. Tato komunikace bude pravděpodobně uskutečněná pomocí standardizovaných datových souborů.



Obrázek č.5: Rich picture knihovního systému

9.2 Relační schéma

Jak vyplývá z přílohy A, tak hlavními tabulkami jsou tabulka *publikace*, *knihy*, *periodika* a *exemplare_publicaci*. Společné atributy všech titulů jsou uvedené v tabulce *publikace* a atributy, které se týkají každého typu titulu (kniha, medium, periodikum) jsou uvedeny v tabulkách: *knihy*, *media*, *periodika*. Tímto je vymezená možnost použití databáze, respektive aplikace na tyto tři druhy titulů. Tabulka *publikace* je propojena s tabulkami: *jazyky*, *typ* a *nakladatelstvi*. Tyto tabulky mají cizí klíče právě v tabulce *publikace*.

Další významnou částí relačního schéma jsou tabulky *autori* a *publikace_autoru*, kde jsou zachyceni autoři jednotlivých titulů. Pomocí tabulky *publikace_autoru* je realizován vztah m:n¹, který je pro zachycení skutečných možností nezbytný.

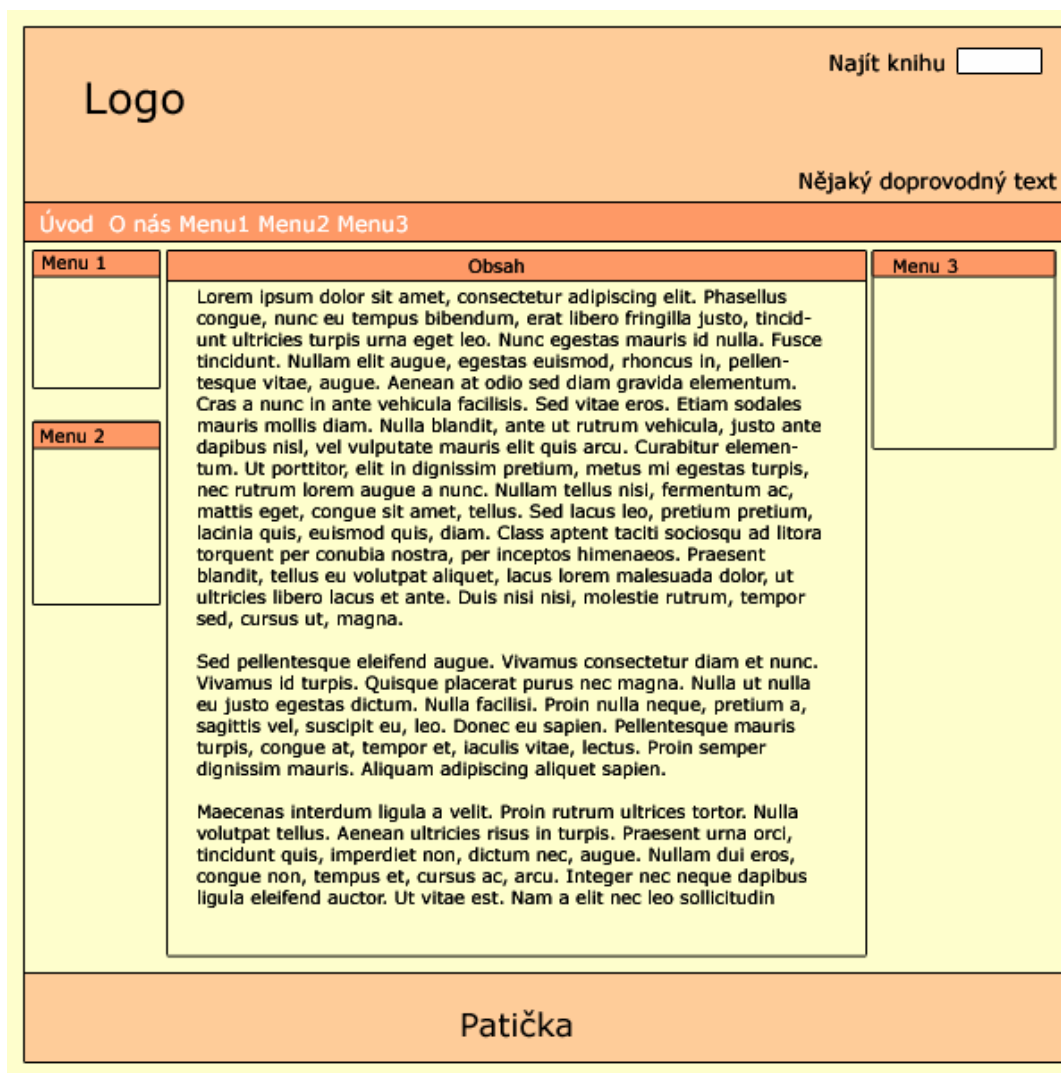
Důležitou částí relačního schéma databáze je tabulka *exemplare_publicaci*, která umožňuje evidovat každý fyzický výtisk jako samostatnou položku a zároveň dává možnost ji pomocí tabulek *vypujcky_osob* a *rezervace_osob* půjčovat, popřípadě rezervovat příslušené osobě.

V knihovních systémech je jedním z nejdůležitějších atributů tzv. **přírůstkové číslo**. V případě tohoto modelu se jedná o sloupec *id_exemplare_publicace*. Dále jsou zajímavé sloupce *signatura* z tabulky *publikace* a sloupec *exemplar* z tabulky *exemplare*, které dohromady tvoří unikátní číslo tzv. **signaturu** podle které lze titul najít v příslušném regále.

¹ Vztah m:n – Např. jeden autor může napsat více knih a zároveň jedna kniha může být naspána více autory

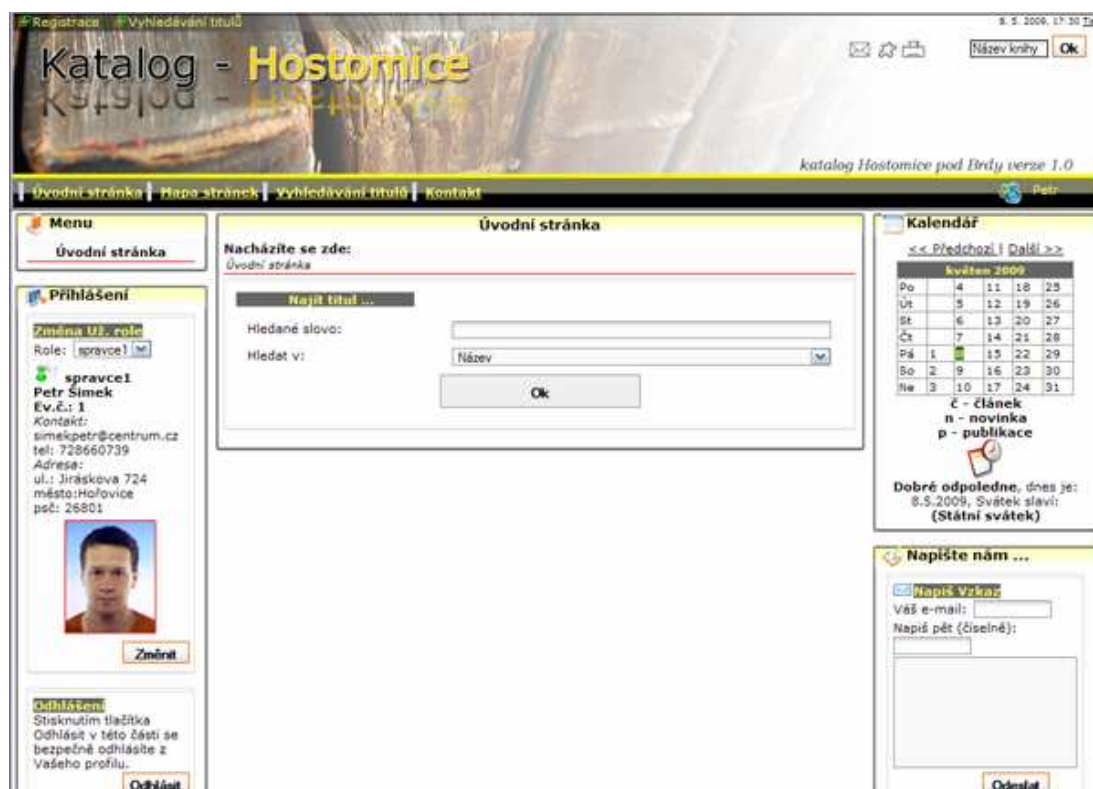
9.3 Návrh vzhledu aplikace

Před samotným ztvárněním aplikace bylo nutné navrhnout její vzhled. Zvolil jsem klasické *třísloupcové rozložení* vzhledem k tomu, že aplikace bude sloužit i jako katalog a pravděpodobně i internetové stránky. Hlavičku aplikace jsem navrhl tak, že v ní je umístěn nápis, který jasně uživateli říká o jakou aplikaci se jedná. Dále je zde vyhledávací políčko formuláře pro rychlé hledání titulů podle jejich názvů. Ve spodní části hlavičky je hlavní menu s odkazy první úrovně. S tímto hlavním menu souvisí vedlejší menu (v *obrázku č.6 Menu1*), které doplňuje hlavní menu a jsou v něm odkazy na ostatní stránky aplikace nižších úrovní. Dále je vzhled aplikace tvořen levými a pravými panely a patičkou. Ve středu aplikace je umístěna hlavní zobrazovací část aplikace společně s výpisem, který poskytuje uživateli stránek informaci o tom na jaké stránce se nachází.



Obrázek č.6: Návrh vzhledu aplikace

10 VÝVOJ APLIKACE



Obrázek č.7: Náhled na hotovou webovou aplikaci

10.1 Prostředky použité při vývoji

Jak jsem již dříve uvedl, je nutné aplikaci vyvinout pomocí programových nástrojů. Já jsem z finančních důvodů volil programy, které jsou buď zcela zadarmo, nebo jsou zdarma pro akademické účely. Ve skupině *Nástroje* jsou zařazeny aplikace a ve skupině *Jazyky* jsou popsány ostatní prostředky, které jsem použil při vývoji.

10.1.1 Nástroje

Veškeré zdrojové kódy jsem editoval pomocí programu *PSPad*¹. Tento nástroj je zdarma, umožňuje zvýraznění syntaxe a je v češtině. Dále jsem použil program *Toad Data Modeler*², který umožňuje navrhovat databáze. Program *Toad Data Modeler* zdarma není, ale firma *Quest Software* nabízí zkušební verzi, která uživatele omezí pouze ve vytváření určitého maximálního počtu tabulek. Grafické prvky jsem editoval pomocí programu *Adobe Photoshop* ve verzi ve které je pro akademické

¹ <http://www.pspad.com/cz/>

² http://www.toadsoft.com/toaddm/toad_data_modeler.htm

účely k použití zdarma. Program Adobe Photoshop je velmi dobrý nástroj pro tvorbu grafických prvků internetových stránek a v současné době patří k nejlepším programům tohoto zaměření. Pro běh databáze a spouštění skriptů jazyka PHP na lokálním počítači jsem si nainstaloval a nakonfiguroval program *Xampp*¹. Xampp je balík služeb, který obsahuje Apache, MySQL, PHP, prostředí pro ovládání databáze PHPMyAdmin a jiné služby, které jsem ovšem nevyužil.

10.1.2 Jazyky

Vzhled aplikace je tvořen kódováním webové stránky v jazyce *HTML* spolu s kaskádovými styly a grafickými prvky. Logická část je naprogramovaná v jazyce *PHP* a data jsou uložena v databázi *SQL*.

Značkovací jazyk *HTML* jsem zvolil z toho důvodu, že je specifikován již dlouhou dobu a chyby které se v něm vyskytují se podařilo programátorům nahradit jiným řešením, které je dostupné na Internetu. Kaskádové styly doplňují význam značek jazyka *HTML* a je výhodné je využít z důvodu případných pozdějších úprav vzhledu aplikace. Zároveň je možné pomocí kaskádových stylů definovat takový vzhled, který by pomocí jazyka *HTML* definovat buď nešlo, nebo šlo velice obtížně.

Kombinaci skriptovacího jazyka *PHP* a databáze *SQL* jsem zvolil protože tato kombinace je zdarma a webový server na který jsem aplikaci umístil tuto kombinaci technologií podporuje.

10.2 Databáze

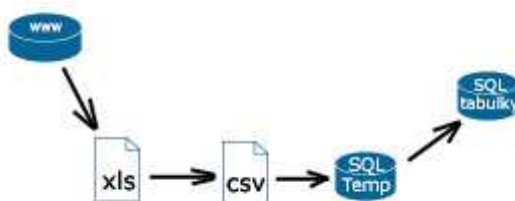
Při realizaci databáze jsem zvažoval použití dvou typů tabulek a to *myISAM* a *innnoDB*. Nakonec jsem zvolil *MyISAM* z toho důvodu, že jsou dotazy na databázi prováděny rychleji. Kódování řetězců znaků jsem volil *utf8_czech_ci*. U sloupců tabulek jsou nastavená různá *integritní omezení*, například: primární klíče, nulové hodnoty, unikátní hodnoty, datové typy, velikost datových typů atp.

Celý systém, tedy redakční systém a knihovní systém obsahuje 38 tabulek a zabírá v současné době 4,4 Mb.

¹ <http://www.apachefriends.org/en/xampp.html>

10.2.1 Migrace dat

Před samotným naprogramováním aplikace jsem musel nejdříve data migrovat ze souboru ve formátu XLS do databáze SQL. Nejdříve jsem si uložil soubor ze stránek Městské knihovny Hostomice¹, který je v datovém formátu **XLS** a obsahuje veškeré tituly knihovny na svůj počítač. Soubor jsem si vhodně upravil a uložil ve formátu **CSV**, soubor jsme pojmenovali *temp_imp.csv*. Následně jsem mohl přistoupit k importu dat do databáze **SQL** a to do jedné pracovní tabulky, kterou jsem nazval *temp*. V této tabulce jsou všechny sloupce, které jsou v souboru *temp_imp.csv*. Dále jsem mohl již v navržené databázi pomocí vhodných databázových dotazů kopírovat data do připravených tabulek z tabulky *temp*. Jak jsem ale zjistil, tak ne všechna data byla ve správném tvaru. Musel jsem tudíž aplikovat několik dotazů na databázi abych dosáhl kýženého stavu a databáze nevykazovala chyby.



Obrázek č.8: Migrace dat

Jednou z chyb ve zdrojových datech bylo například to, že bylo chybně uloženo příjmení autora, protože bylo celé velkými písmeny. Musel jsem tedy nejdříve převést všechna písmena sloupce *prijmeni* na malá písmena a následně realizovat dotaz uvedený v *Syntaxi č.4* pro převedení prvního malého písmena zpět na velké písmeno.

¹ <http://knihovnahostomice.cz/katalog.html>

Níže je uvedeno několik použitých dotazů při migraci dat.

```
update autori set autori.prijmeni=CONCAT( LEFT(autori.prijmeni,0) ,
upper (SUBSTRING(autori.prijmeni,1,1)) ,
SUBSTRING(autori.prijmeni,2) )
```

Syntaxe č.4: Změna prvního malého písmena na velké písmeno

```
insert into nakladatelstvi(nakladatelstvi) select distinct
nakladatelstvi from temp
```

Syntaxe č.5: Import dat do tabulky nakladatelstvi

```
insert into druhy_knih(druh) select distinct druh from temp
```

Syntaxe č.6: Import dat do tabulky knihy

```
insert into publikace(id_publicace, nazev, cena, mdt, id_typ,
id_jazyka, id_nakladatelstvi)
select distinct temp.id, temp.nazev, temp.cena,
temp.mdt, typ.id_typ, jazyky.id_jazyka,
nakladatelstvi.id_nakladatelstvi
from temp, typ, jazyky, nakladatelstvi
where temp.typ=typ.typ
and temp.jazyk=jazyky.jazyk
and temp.nakladatelstvi=nakladatelstvi.nakladatelstvi
```

Syntaxe č.7: Import dat do tabulky publikace

```
insert into publikace_autoru (id_atora, id_publicace)
select autori.id_atora, publikace.id_publicace
from publikace, autori, temp
where publikace.id_publicace=temp.id
and temp.prijmeni=autori.prijmeni
and temp.jmeno=autori.jmeno
```

Syntaxe č.8: Import dat do tabulky publikace_autoru

V případě, že dotaz neobsahuje všechny sloupce tabulek, jsou hodnoty buď automaticky generovány funkcí *auto_increment* nebo je doplní systémem podle výchozí hodnoty. Celá databáze je značně datově objemná. Z toho důvodu jsem musel v některých případech dotazy přeformulovat, abych dosáhl výsledku v reálném časovém horizontu.

10.3 Logické členění aplikace

Aplikaci je možné ovládat prostřednictvím čtyř základních typů účtů (*administrátor, správce, registrovaný uživatel – čtenář a neregistrovaný uživatel*). Uživatel

nemůže zobrazovat funkce, které vlastní správce nebo administrátor. V opačném případě funkce zobrazit lze. Každá osoba může mít různé typy těchto účtů, minimálně však jeden. Toto rozložení funkcí je předpokládám dostatečné, protože v systému je plánováno, že *administrátor* bude spravovat systém, *správce* bude ovládat jen některé funkce týkající se samotné evidence titulů a *uživatel* (registrovaný i neregistrovaný) bude především zjišťovat informace, které vložil správce systému. Rozdíl mezi uživatelem, který je zaregistrovaný a uživatelem neregistrovaným je v tom, že registrovaný uživatel má k dispozici některé speciální funkce systému.

10.4 Uživatelské rozhraní

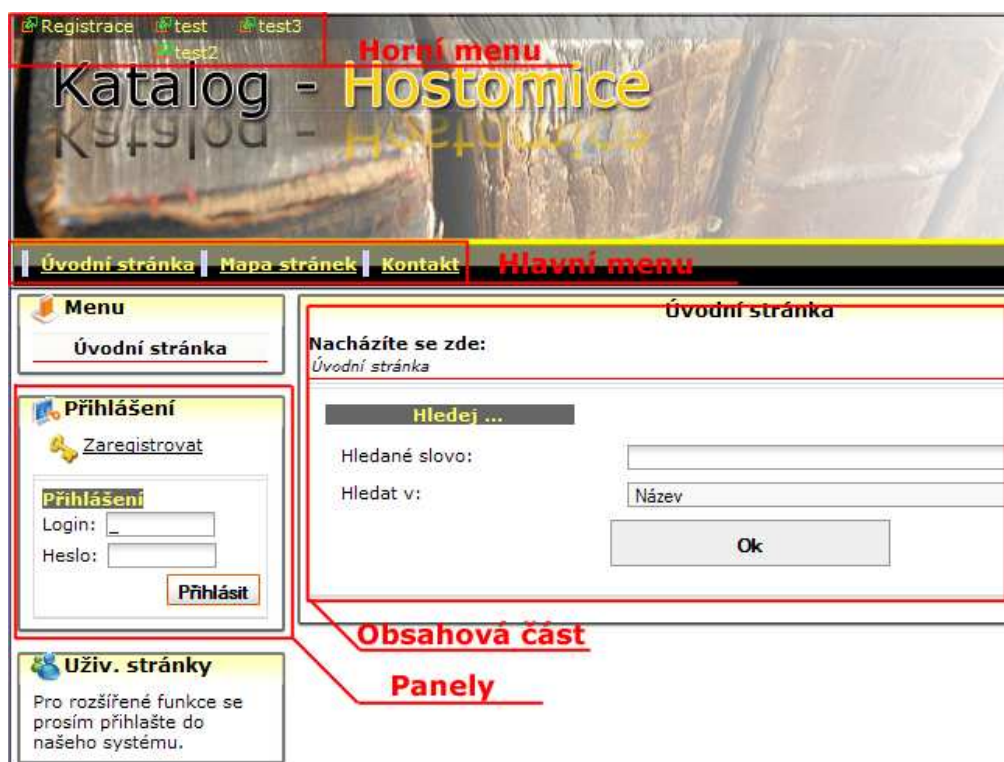
Uživatelské rozhraní jsem navrhoval s cílem *maximalizovat přehlednost a uživatelský komfort* při ovládání a to z jakékoliv uživatelské úrovně. Pokusil jsem se navrhnout takový vzhled, kde by uživatel ihned věděl kde je hlavní menu, vedlejší menu, nebo ihned viděl funkce, které mu jsou přístupné po přihlášení. Využil jsem mnoho doprovodných ikon pro zdůraznění důležitých částí systému a zpříjemnění práce se systémem. Barvy jsou spíše světlejší a na celé stránce převažuje bílá barva.

Každá funkce systému kde jsou některé volby, obsahuje ve spodní části stránky legendu s vysvětlivkami. V případě, že uživatel vykoná některou nepřipustnou akci, je mu zobrazeno varování a akce kterou vyvolal je přerušena.

Pro využití speciálních funkcí systému, které jsou popsány v *kapitole 11* je potřeba, aby se uživatel nejdříve zaregistroval a následně přihlásil, nebo mu byl zřízen účet správcem systému. Uživatel se vždy přihlašuje s nejvyššími právy, které mu byly přiděleny. V případě, že má uživatel více typů účtu, může mezi nimi libovolně přepínat.

10.4.1 Vzhled aplikace

Aplikace se skládá z několika logicky oddělených částí. Jsou to *Hlavní menu*, *Horní menu*, *Postranní panely* a *Hlavní obsahová část*. *Hlavní menu* a *Horní menu* slouží k pohybu uživatele v aplikaci. *Postranní panely* obsahují různé prvky, mimo jiné obsahují taktéž navigaci, která je svázaná s *Hlavním menu*. V *Postranních panelech* jsou zobrazeny i ostatní funkce redakčního a knihovního systému jako například kalendář s vloženými tituly, pole pro rychlé napsání vzkazu správci systému, počítačlo návštěv, anketa atp.



Obrázek č.9: Rozmístění prvků webové aplikace

11 POPIS APLIKACE

V této kapitole podrobně popíše funkce systému. Přehled funkcí je rozdělený podle toho kdo je využívá. Knihovní systém obsahuje tyto funkce:

- Funkce pro knihovníka,
 - Půjčit,
 - Vrátit,
 - Evidence výpůjček,
 - Evidence upomínek,
 - Přehled rezervací,
 - Přidat publikaci,
 - Přidat exemplář,
 - Smazat publikaci,
 - Správa uživatelů,
- Funkce pro čtenáře,
 - Změna hesla,
 - Změna osobních údajů,
 - Moje výpůjčky,
 - Moje rezervace,
 - Upomínky,
- Ostatní funkce systému,
 - Rychlé vyhledávání titulů,
 - Pokročilé vyhledávání titulů,
 - Detailní zobrazení titulu.

11.1 Funkce pro knihovníka

V případě, že knihovník zvolí funkci *Půjčit*, která se nachází v záložce *Správa výpůjček*, je vyzván k tomu, aby zadal přírůstkové číslo titulu. Po zadání unikátního přírůstkového čísla titulu zvolí komu se titul půjčuje a do kdy ho bude mít vypůjčen. Pro dokončení procesu vypůjčení titulu zbývá stisknout tlačítko vypůjčit. V případě, že je titul k dispozici v knihovně, je titul čtenáři zapůjčen. V opačném případě aplikace vypíše informaci „Kniha je již půjčená“. Algoritmus je popsán v příloze B2.

S touto funkcí souvisí funkce *Vrátit* (viz příloha B1). Nejdříve je nutné vložit evidenční číslo uživatele, které mu bylo automaticky přiděleno systémem při jeho registraci a následně je systémem vypsán seznam jeho výpůjček. Po kliknutí na ikonu ve sloupci vrátit v příslušném řádku je titul vrácen. Pro ukončení vrácení výpůjček čtenáře slouží tlačítko *Ukončit*. V případě, že si jiný čtenář rezervoval titul který se vrací, je mu po vrácení titulu o který žádal, automaticky odeslán e-mail s informací o tom, že je titul přístupný k vypůjčení.

Upomínky všech uživatelů může knihovník sledovat pomocí funkce *Evidence upomínek*. Upomínky jsou tituly, které čtenář nevrátil v den data vrácení. Knihovník má možnost si omezit výpis upomínek na upomínky vybraného čtenáře a zároveň si je seřadit podle některých kritérií jako jsou například přírůstkové číslo nebo datum vrácení. Všechny čtenáře je možné hromadně informovat o nevrácení jejich výpůjček prostřednictvím jednorázového odeslání výzvy k vrácení výpůjček, kde jsou uvedeny jejich vypůjčené tituly. Knihovník může v systému sledovat i tituly, které jsou *vypůjčené* nebo je na ně podaná *rezervace*.

Titul se přidává pomocí funkce *Přidat publikaci*. Nejdříve se zvolí typ titulu a následně se vyplní ostatní pole. Společná pole všech titulů jsou vyznačená černě. Pole ve kterých se typy titulů liší jsou vyznačené jinými barvami. V případě, že je potřeba přidat nový exemplář, lze použít funkci *Přidat exemplář*. Při použití této funkce se vloží přírůstkové číslo titulu a veškerá data se zkopírují.

K odebrání titulu z databáze slouží funkce *Smazat publikaci*. Vzhledem k velkému počtu titulů je možné výpis filtrovat podle kritérií, která dále omezují výpis seznamu titulů.

Dále má knihovník možnost spravovat uživatelské účty. Může uživatelům odeprít přístup do systému, případně jejich účet smazat. Další funkce jsou již spojené s redakčním systémem, který není náplní této bakalářské práce

11.2 Funkce pro čtenáře

Čtenář má k dispozici řadu různých funkcí systému, které bezprostředně souvisí s jeho osobou. Může si změnit své osobní údaje nebo heslo. Dále může sledovat tituly, které si vypůjčil nebo rezervoval¹ (algoritmus rezervace je popsán v *příloze B3*). Čtenář má k dispozici i funkci na sledování jeho upomínek².

11.3 Ostatní funkce systému

Aby čtenáři mohli tituly snadno vyhledávat, systém obsahuje pole „*rychlé hledání*“ (viz *příloha B4*), které je umístěné v pravém horním rohu aplikace. Do tohoto vstupního pole čtenář zadává slovo, které je obsaženo v názvu titulu jenž hledá. Pro *podobnější hledání* je možné využít formulář umístěný na úvodní stránce, který disponuje výběrem v jakých sloupcích databáze se bude slovo vyhledávat.

Po vyhledání titulu systém zobrazí čtenáři tituly které našel a čtenář si může jednotlivé tituly detailně zobrazit. V detailu každého titulu jsou zobrazeny veškeré dostupné informace. V případě že je titul vypůjčený, je čtenáři přístupné tlačítko pomocí kterého lze titul rezervovat.

Kalendář knihovního systému vypisuje tituly, které byly vloženy do databáze knihovny.

¹ Rezervuje se titul, který je vypůjčen a tudíž není přístupný v knihovně

² Upomínka je titul který, čtenář nevrátí včas.

12 BĚŽNÝ PROVOZ APLIKACE

V této kapitole se zabývám popisem provozu aplikace za normálních okolností a to jak z pohledu knihovníka, tak z pohledu čtenáře.

Nejdříve *knihovník vloží* do databáze nové tituly, které se následně vypíší v kalendáři systému. Tímto způsobem je čtenář je informován o nových titulech přístupných v knihovně. Čtenář si nejprve vyhledá v systému tituly, které si chce zapůjčit. V případě, že je titul již vypůjčený, je možné provést rezervaci vypůjčeného titulu. V opačném případě si titul může jít do knihovny čtenář vypůjčit.

V knihovně si titul najde čtenář podle signatury a mezinárodního desetinného třídění nebo podle přírůstkového čísla. Tituly, které si chce čtenář zapůjčit předá knihovníkovi. Knihovník následně zadá do systémů přírůstkové číslo titulu a titul čtenáři *zapůjčí*.

Čtenář si může po přihlášení do systému své výpůjčky kontrolovat a zjišťovat kdy je nutné je vrátit. V případě, že výpůjčky nevrátí, tak je mu systémem automaticky zaslána upomínka. Jestliže na upomínky nereaguje a titul nevrátí, má knihovník možnost zablokovat čtenáři účet a systém nedovolí čtenáři se přihlásit. Jestliže se čtenář rozhodne titulu vrátit se zpožděním, tak si knihovník může zjistit o kolik dní překročil termín vrácení knihy a podle toho vyvodit patřičné důsledky.

V případě *vracení výpůjček* zadává knihovník do systému evidenční číslo uživatele a následně je mu vypsán seznam čtenářových výpůjček i s upozorněním na překročení nejzazšího termínu vrácení výpůjček.

13 POŽADAVKY APLIKACE

13.1 Technické požadavky na provoz aplikace

Pro správný provoz aplikace je nutné, aby na webovém serveru byly nahrány všechny soubory ze kterých se aplikace skládá. Dále je potřeba, aby byla založena databáze a v konfiguračním souboru k ní byly správně nastavené přihlašovací údaje. Webový server na kterém bude aplikace umístěna musí zároveň podporovat spouštěcí skriptů PHP a provoz databáze SQL.

Při výběru webového hostingu jsem vybíral hosting podle kritérií uvedených v obrázku č.10. Po uvážení jsem zvolil hosting na serveru *Internet Centrum* a to na internetové adrese *www.kataloghostomice.ic.cz*. Tento hosting nabízí v současné době k dispozici 250 Mb webového prostoru, podporuje PHP5, MySQL, Sqlite a Postgresql. Jediné omezení tohoto hostingu je reklamní pruh v horní části stránky a nezaručená stálá dostupnost internetových služeb.

Jméno	Prostor[MB]	II	III	Mail	FTP	PHP	.NET	MySQL	PgSQL	Reklama	Komerční	Adult	Dostupnost
ASP2	25	✗	✓	✗	✓	✗	✓	✗	✗	✗	✓	✗	99.852
ASPone	40	✗	✓	✗	✓	✗	✓	✗	✗	✗	✗	✗	100.0
ASPweb	30	✗	✓	✓	✓	✗	✓	✗	✗	✗	✓	✗	99.66
AtlasWeb	15	✗	✓	✗	✓	✓	✗	✗	✗	✓	✓	✗	99.901
Domeny.as	150	✓	✓	✗	✓	✓	✗	✗	✗	✗	✓	✗	99.931
HostingZdarma	1000	✓	✗	✓	✓	✓	✗	✓	✗	✗	✓	✗	99.94
HostujZdarma	500	✓	✗	✓	✓	✓	✗	✓	✗	✗	✓	✗	99.963
Hostuju	1000	✗	✓	✗	✓	✓	✗	✓	✗	✓	✓	✓	99.917
Hy.cz	300	✗	✓	✓	✓	✓	✗	✓	✗	✓	✓	✗	98.015
Internet Centrum	250	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	99.73
LTweb	100	✓	✗	✓	✓	✓	✗	✓	✗	✓	✗	✗	99.992
N-Host	2500	✗	✓	✓	✓	✓	✗	✓	✗	✓	✓	✗	99.936
PHP5	200	✗	✓	✗	✓	✓	✗	✓	✗	✓	✓	✗	99.66
Pipni	1500	✓	✗	✓	✓	✓	✗	✓	✓	✗	✓	✗	96.019
Porno-Hosting	150	✗	✓	✗	✓	✓	✗	✗	✗	✗	✓	✓	99.912
Profitux	300	✗	✓	✓	✓	✓	✗	✓	✗	✓	✗	✗	99.888
QSH	100	✓	✓	✗	✓	✓	✓	✗	✗	✗	✓	✗	99.829
WebZdarma	500	✗	✓	✓	✓	✓	✗	✓	✗	✓	✓	✗	99.401

Obrázek č.10: Srovnání bezplatných hostingů
zdroj: <http://freehostingy.ic.cz/>

13.2 Požadavky na uživatele

Z hlediska uživatele je nutné mít počítač, který je připojený k síti Internet a internetový prohlížeč.

13.3 Instalace aplikace

Nejdříve je nutné importovat databázi ze souboru *databáze katalog.txt* pomocí webové aplikace phpMyAdmin. Po otevření textového souboru *databáze katalog.txt* jsou na začátku tohoto souboru informace o tom kdy a na jakém serveru byla databáze vygenerovaná. Dále jsou v tomto souboru příkazy na vytvoření tabulek a na jejich naplnění daty.

Po úspěšném vytvoření databáze je potřeba změnit přihlašovací údaje v konfiguračním souboru na přihlašovací údaje serveru kde bude aplikace provozována. Posledním krokem je zkopírování všech souborů na webový server, případně lokální stanici na které je nainstalovaný nástroj jenž ze stanice vytvoří webový server na kterém lze provozovat databázi SQL a spouštět skripty jazyka PHP.

14 TESTOVÁNÍ A OPTIMALIZACE

Hotovou aplikaci je nutné otestovat pomocí různých metod testování. Nejdříve je nutné vyzkoušet zda se aplikace zobrazuje ve všech známých prohlížečích korektně. Dále je vhodné požádat osobu ze svého okolí, zda by mohla se systémem pracovat. V případě, že má osoba při práci se systémem nějaké nejasnosti, je nutné tuto část aplikace upravit.

Následně je možné systém otestovat na to jak zpracovává *nestandardní vstupy* a zda je dostatečně zabezpečená proti různým *technikám útoků*. Zde musím ale podotknout, že zcela otestovat systém z tohoto hlediska nikdy nelze. Chyby v aplikaci je možné minimalizovat následnými úpravami aplikace v průběhu testování.

Finální otestování aplikace jsem provedl po uložení aplikace na webový server, kde jsem mohl zhodnotit jak je aplikace schopná pracovat v „reálném prostředí“.

14.1 Zkušební verze knihovního systému

Zkušební verze knihovního systému je přístupná na internetové adrese *www.demo-hostomice.ic.cz*. Uživatelské jméno účtu správce systému je *demo*, heslo je *demoupce*. Pomocí této zkušební verze systému je možné systém vyzkoušet a zhodnotit jeho funkčnost.

15 MOŽNÁ VYLEPŠENÍ

V systému není řešen akviziční modul, který zajišťuje komunikaci s dodavateli titulů a nákup titulů. Dále není v současné době zpracován systém na výměnu záznamů s ostatními knihovnami. Zároveň by bylo možné určitým způsobem vylepšit správu periodik a více ji specifikovat a diferencovat od správy ostatních druhů titulů.

Z časových důvodů nejsou naprogramovány některé funkce, které například umožňují knihovníkovi přidat nakladatelství nebo originální jazyk titulu.

Je možné uvažovat o tom, že by systém byl podpořen technologií čárových kódů. Tento systém by ovšem vyžadoval větší finanční investici. Pokud by se knihovna rozhodla investovat, tak bych spíše volil investici v podobě nákupu placeného hostingu, který by zaručil rychlejší a bezproblémovější provoz aplikace.

16 ZÁVĚR

Cílem této bakalářské práce bylo seznámit se s knihovními systémy a na základě zjištěných poznatků realizovat aplikaci, která by umožňovala každodenní správu titulů v knihovně a zároveň umožnila čtenářům zjišťování informací o titulech prostřednictvím Internetu.

Zjistil jsem, že knihovní systémy jsou rozsáhlé a poměrně hodně specifické informační systémy, které jsou vyvíjeny mnoho let týmem odborníků. Z toho důvodu jsem nedosáhl úrovně těchto profesionálních systémů. Zpracování knihovního systému na profesionální úrovni ovšem nebylo cílem této bakalářské práce.

Aplikaci se mi podařilo vytvořit a umístit ji na Internet. Aplikace je přístupná na internetové adrese *www.kataloghostomice.ic.cz*, popřípadě je možné si ověřit její funkčnost na internetové adrese *www.demo-hostomice.ic.cz*.

Přínos této bakalářské práce je v tom, že se mi podařilo vyvinout aplikaci, která uspokojuje požadavky na knihovní systém menší knihovny. Typická knihovna, která by mohla aplikaci využívat je ta, která se rozhodla evidovat tituly elektronicky a zároveň má alespoň jeden počítač připojený do sítě Internet. Tato knihovna ovšem nezískala dotaci nebo nemá prostředky na zakoupení drahého profesionálního systému, jehož cena se pohybuje v desítkách až stovkách tisíc českých korun.

Aplikace je plně funkční a obsahuje všechny tituly, které jsem mohl do systému imporovat. Do budoucna by bylo vhodné aplikaci rozšířit o nové funkce, případně stávající funkce upravit dle požadavků knihovny. Tento knihovní systém je možné použít s drobnými úpravami i v jiné knihovně než v knihovně v Hostomicích pod Brdy.

17 POUŽITÁ LITERATURA

- [1] *Webová aplikace* [online]. [2009] [cit. 2009-04-06]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Webov%C3%A1_aplikace>.
- [2] FERSCHMANN, Petr. *Bezpečnost na webu – přehled útoků na webové aplikace* [online]. 2008-2009 [cit. 2009-04-06]. Dostupný z WWW: <<http://zdrojak.root.cz/clanky/prehled-utoku-na-webove-aplikace/>>.
- [3] JANOVSKEÝ, Dušan. *Základy HTML, jak na HTML* [online]. [2002] [cit. 2009-04-06]. Dostupný z WWW: <<http://www.jakpsatweb.cz/zaklady-html.html>>.
- [4] *HyperText Markup Language* [online]. [2009] [cit. 2009-04-06]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/Html>>.
- [5] JANOVSKEÝ, Dušan. *CSS styly* [online]. [2002] [cit. 2009-04-07]. Dostupný z WWW: <<http://www.jakpsatweb.cz/css/css-uvod.html>>.
- [6] *Skript (program)* [online]. [2009-] [cit. 2009-04-07]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Skriptovac%C3%AD_jazyk>.
- [7] *Databáze* [online]. [2009] [cit. 2009-04-07]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/Datab%C3%A1ze>>.
- [8] *Automatizace knihoven* [online]. 2009 [cit. 2009-04-07]. Dostupný z WWW: <<http://www.ics.muni.cz/zpravodaj/articles/425.html>>.
- [9] *Knihovní systém Daimon* [online]. 1995-2008 [cit. 2009-04-08]. Dostupný z WWW: <http://www.sefira.cz/reseni/daimon/doc/20070118%20DAIMON_popis%20systemu.pdf>.
- [10] *Knihovní systémy Clavius a LANius* [online]. [2009] [cit. 2009-04-07]. Dostupný z WWW: <<http://clavius.cz/>>.
- [11] *Cosmotron* [online]. 2002-2005 [cit. 2009-04-08]. Dostupný z WWW: <<http://www.cosmotron.cz/>>.
- [12] *Knihovní systém DAWINCI* [online]. 2005 [cit. 2009-04-08]. Dostupný z WWW: <<http://www.akvs.cz/akp-2005/18-bujalka.pdf>>.
- [13] BROŽA, Petr. *Tvorba WWW stránek: pro úplné začátečníky*. [s.l.] : [s.n.], 2004. 149 s.
- [14] *MySQL* [online]. [2009] [cit. 2009-04-20]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/MySQL>>.

- [15] *Vývojové prostředí* [online]. 2009 [cit. 2009-04-09]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/V%C3%BDvojov%C3%A9_prost%C5%99ed%C3%A9>.
- [16] *Bitmapový grafický editor* [online]. 2009 [cit. 2009-05-09]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Bitmapov%C3%BD_grafick%C3%BD_editor>.
- [17] *Vektorový grafický editor* [online]. 2009 [cit. 2009-05-09]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Vektorov%C3%BD_grafick%C3%BD_editor>.
- [18] *Aleph: nový knihovní systém* [online]. [2009] [cit. 2009-05-11]. Dostupný z WWW: <<http://www.ics.muni.cz/bulletin/articles/263.html>>.

18 SEZNAM PŘÍLOH

A	Relační schéma databáze.....	48
B	Ukázky zdrojových kódů	49
B.1	Část zdrojového kódu funkce Vrátit	49
B.2	Část zdrojového kódu funkce Půjčit	50
B.3	Část zdrojového kódu funkce Rezervace titulu.....	51
B.4	Část zdrojového kódu funkce Rychlé hledání.....	52

B UKÁZKY ZDROJOVÝCH KÓDŮ

V této příloze jsou uvedeny vybrané části zdrojových kódů aplikace. Z technických důvodů není možné zdrojový kód formátovat tak, aby byl tak dobře čitelný jako v souborech ze kterých se skládá vlastní aplikace.

B.1 Část zdrojového kódu funkce Vrátit

```
if (!empty($_POST["ukoncit"]))
    unset($_SESSION["ev_cislo"]);

$zobrazitPrvniFormular = true;

if ( !empty($_GET["id_ex_pub"]) ){

if ( ochrana($_GET["id_ex_pub"],10,"^[0-9]*$",false) ){

if ($vysledek = mysql_query("delete from vypujcky_osob where vypujcky_osob.id_exemplare_publicace=" .antiInjection($_GET["id_ex_pub"])." ")){
zobrazZpravu("i","Publikace vrácena.");

if ($vysledek = mysql_query("select exemplare_publicaci.id_exemplare_publicace,
publicace.nazev, osoba.jmeno, osoba.prijmeni, osoba.email, rezervace_osob.id_rezervace_osoby
from publicace, exemplare_publicaci, rezervace_osob,
osoba where publicace.id_publicace=exemplare_publicaci.id_publicace and exemplare_publicaci.id_exemplare_publicace=rezervace_osob.id_exemplare_publicace
and rezervace_osob.id_osoby=osoba.id_osoby and exemplare_publicaci.id_exemplare_publicace=" .antiInjection($_GET["id_ex_pub"])." ")){

if ( mysql_num_rows($vysledek) == 1 ){
$zaznam = mysql_fetch_array($vysledek);
zobrazZpravu("i","Publikace ".$zaznam["nazev"].", p.č:
".$zaznam["id_exemplare_publicace"]." byla dříve rezerována uživatelem
".$zaznam["jmeno"]." ".$zaznam["prijmeni"].", automaticky se mu odeslal e-mail na
".$zaznam["email"]." o dostupnosti publikace.");

$recipient = $zaznam["email"];
$subject = "dostupnost rezerovaneho titulu v knihovne Hostomice pod Brdy";
$message .= "V knihovně je k dispozici vámi rezervovaný titul
".$zaznam["nazev"].", můžete si ho přijít vypůjčit.\n";
$message .= "Městská Knihovna Hostomice pod Brdy";
$header .= "From: knihovna@hostomice.cz\r\n";
$header .= "Content-type: text/plain; charset=utf-8";
if (@mail($recipient, $subject, $message, $header))
echo "<p>Email se podařilo odeslat.</p>";
else
```

```
echo "<p>Email se nepodařilo odeslat.</p>";
```

```
if ($vysledek = mysql_query("delete from rezervace_osob
where rezerva-
ce_osob.id_rezervace_osoby='".antiInjection($zaznam["id_rezervace_osoby"])."
")){
zobrazZpravu("i", "Rezervace zrušena.");
}
else
    zobrazZpravu("w", "Chyba při komunikaci s databází.");
}
}
else
    zobrazZpravu("w", "Chyba při komunikaci s databází.");
}
else
    zobrazZpravu("w", "Chyba při komunikaci s databází.");
}
else
    zobrazZpravu("w", "Publikaci se nepodařilo vrátit z důvodu špatných vstup-
ních parametrů.");
}
```

B.2 Část zdrojového kódu funkce Půjčit

```
if (!empty($_POST["vypujcit"])){

if ( (ochrana($_POST["id_osoby"],10,"^[0-9]*$",false)) and (ochra-
na($_POST["id_exemplare_publicace"],10,"^[0-9]*$",false)) ){

if ($vysledek = mysql_query("select exemplare_publicaci.id_exemplare_publicace
from exemplare_publicaci, vypujcky_osob where exempla-
re_publicaci.id_exemplare_publicace=vypujcky_osob.id_exemplare_publicace and
exemplare_publicaci.id_exemplare_publicace='".antiInjection
($_POST["id_exemplare_publicace"])."'")){

if ( mysql_num_rows($vysledek) < 1 ){
$zaznam = mysql_fetch_array($vysledek);

if ($vysledek = mysql_query("insert into vypujcky_osob (id_osoby, vypujceno, vra-
tit, id_exemplare_publicace) values
('".antiInjection($_POST["id_osoby"])."', '".antiInjection(Date("Y-m-d
H:i:s"))."', '".antiInjection($_POST["vratit"])."', '".antiInjection($_POST["id_exemplar
e_publicace"])."'"))){
zobrazZpravu("i", "Publikace úspěšně zapůjčena.");
}
else
    zobrazZpravu("w", "Chyba při komunikaci s databází.");
```

```

}
else
    zobrazZpravu("w", "Publikace je již půjčená.");
}
else
    zobrazZpravu("w", "Chyba při komunikaci s databází.");
}
else
    zobrazZpravu("w", "Pokuste se publikaci zapůjčit znovu.");
}

```

B.3 Část zdrojového kódu funkce Rezervace titulu

```

if ( !empty($_POST["rezervovat"]) ){

if ( ochrana($_POST["rezervace_id_publ"],10,"^[0-9]*$",false) ){

if ($vysledek = mysql_query("select exemplare_publicaci.id_exemplare_publicace,
publicace.nazev from publicace, exemplare_publicaci where publika-
ce.id_publicace=exemplare_publicaci.id_publicace and exempla-
re_publicaci.id_exemplare_publicace not in(select rezerva-
ce_osob.id_exemplare_publicace from rezervace_osob) and publika-
ce.id_publicace='".$_antiInjection($_POST["rezervace_id_publ"])."' order by exempla-
re_publicaci.id_exemplare_publicace")){

if ( mysql_num_rows($vysledek) >= 1 )
$szaznam = mysql_fetch_array($vysledek);

$id_ex_publ = $szaznam["id_exemplare_publicace"];
$dat = Date("Y-m-d H:i:s");
$dat2 = date("Y-m-d H:i:s", time()+24*60*60*14);

if ($vysledek = mysql_query("insert into rezervace_osob (id_exemplare_publicace,
id_osoby, rezervovano, konec_rezervace)
('.$id_ex_publ.', '$_SESSION[id]', '$_dat.', '$_dat2.'))) {
zobrazZpravu("i", "Publikace \'"$szaznam[nazev]\' rezervována.");
}
else
    zobrazZpravu("w", "Chyba při komunikaci s databází.");
}
else
zobrazZpravu("i", "Všechny exempláře jsou již rezervované, zkuste to později.");
}
}
else
    zobrazZpravu("w", "Chyba při komunikaci databází.");
}
else

```

```
zobrazZpravu("w", "Rezervaci publikace se nepodařilo provést z důvodu špatných vstupních parametrů.");}
```

B.4 Část zdrojového kódu funkce Rychlé hledání

```
if ($vysledek = mysql_query("select publikace.id_publicace, publikace.signatura, publikace.nazev, autori.jmeno, autori.prijmeni, nakladatelstvi.nakladatelstvi, typ.typ, publikace.mdt from publikace, publikace_autoru, autori, nakladatelstvi, typ where publikace.id_publicace=publikace_autoru.id_publicace and publikace_autoru.id_aura=autori.id_aura and publikace.id_nakladatelstvi=nakladatelstvi.id_nakladatelstvi and publikace.id_typ=typ.id_typ and nazev REGEXP '[:<:~>]' .antiInjection($_POST["base_search"])."[:>:~>]' order by publikace.signatura")){
```

```
if (mysql_num_rows($vysledek) >= 1){ ?>
```

```
<div class="uprostred">
<table class="klasicka">
<caption>Rychlé hledání titulu</caption>
<tr>
<th>MDT</th>
<th>Signatura</th>
<th>Název</th>
<th>Autor</th>
<th>Nakladatelství</th>
<th>Typ</th>
</tr>
```

```
<?
while ($zaznam = mysql_fetch_array($vysledek))
{ ?>
<tr <? if ($sudy) echo "class=lichy" ?> >
<td><? echo $zaznam["mdt"]; ?></td>
<td><? echo $zaznam["signatura"]; ?></td>
<td>
<? echo "<a
href=\"".$_SERVER["PHP_SELF"]."?page=186&detail=".$zaznam["id_publicace"]."\">".$zaznam["nazev"]."</a>"; ?>
</td>
<td><? echo $zaznam["jmeno"]." ".$zaznam["prijmeni"]; ?></td>
<td><? echo $zaznam["nakladatelstvi"]; ?></td>
<td><? echo $zaznam["typ"]; ?></td>
</tr><? $sudy =! $sudy;
} ?>

</table>
</div>
```

```
<?  
}  
else  
    zobrazZpravu("i","Nepodařilo se najít žádnou knihu.");  
}  
else  
zobrazZpravu("w","Chyba při komunikaci s databází.");
```