UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Knihovní systém

Jan Baťha

Bakalářská práce
2011

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2010/2011**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Jan BAŤHA**
Osobní číslo:	**I08012**
Studijní program:	**B2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Knihovní systém**
Zadávající katedra:	**Katedra informačních technologií**

Zásady pro vypracování:

Úkolem bude vytvořit systém pro knihovnu s webovým rozhraním.

Teoretické část:
Cílem teoretické části je zhodnocení současných systémů na trhu, zhodnocení současných technologií a návrh vhodné databáze pro knihovní systém. Dále technicky navrhnout zabezpečení webové aplikace.

Praktická část
Cílem aplikační části bude vytvořit webovou aplikaci s využitím databázových systémů pro knihovní systém s následujícími požadavky. Aplikace musí zvládnout možnost rezervace a prodlužování výpůjček přes internet a automatické rozesílání upomínek a pokut. Aplikace bude obsahovat různé typy čtenářů s možností vlastního pojmenování (např. student, senior, akademický pracovník) a různé možnosti vypůjčení a prodloužení (např. různý počet knih, různá doba vypůjčení) pro různé typy čtenářů.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

***Hugh E. Williams & David Lane - PHP a MySQL: Vytváříme webové databázové aplikace (Computerpress)**
***Michael J. Hernandez - Návrh Databází (Grada)**

Vedoucí bakalářské práce: **RNDr. Josef Rak**
Katedra informačních technologií

Datum zadání bakalářské práce: **17. prosince 2010**

Termín odevzdání bakalářské práce: **13. května 2011**

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2011

Prohlášení autora

Prohlašuji, že jsem tuto práci 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 Kutné Hoře dne 09. 08. 2011 Jan Baťha

Poděkování

Tímto bych chtěl poděkovat panu RNDr. Josefu Rakovi za rady a připomínky, kterými mi pomohl při tvorbě této bakalářské práce.

Anotace

Bakalářská práce je zaměřena na problematiku automatizovaných knihovních systémů. V teoretické části se nachází popis problematiky knihovních systémů, analýza dostupných systémů na trhu, zhodnocení aktuálních technologií pro vývoj webové aplikace, problematika zabezpečení aplikace a návrh pro systém realizovaný v praktické části. V praktické části je vytvořena webová aplikaci umožňující práci v navrženém sytému. Pro práci byl použit databázový systém Mysql, programovacím jazykem byl zvolen jazyk PHP a vzhled je upraven technologií CSS.

Klíčová slova

Knihovní systém, PHP, Mysql, webová aplikace, zabezpečení webové aplikace, CSS

Title

The library system

Annotation

The bachelor thesis is focused on the automatic libraries system´s issues. In the theoretic part the library system´s issues, the avaliable market´s system´s analysis, the evaluation of the present technologies for the web aplications´s development, the provision of the aplication´s issue and the proposal for the system realized in the practical part are described. In the practical part the web aplication enabling the work in the suggested system is created. The database system Mysql was used for the work, PHP language was chosen to be the programming language and the design is modified by the CSS technology.

Keywords

Library system, PHP, Mysql, web application, security of web application, CSS

Obsah

Seznam zkratek

HTML	Hypertext Markup Language
PHP	HyperText Preprocesor
CSS	Cascading Style Sheets
ISIC	International Student Identity Card
OPAC	Online Public Access Catalog
ASP	Active Server Pages
WYSIWYG	What You See Is What You Get

Seznam obrázků

Seznam tabulek

1 Úvod

V dnešní uspěchané době je největším cílem usnadnění práce a automatizace co nejvíce procesů. Tomuto procesu značně napomáhají počítačové sítě, ale i celosvětová síť Internet.

Rozvoj počítačové technologie se značně podepsal i v oblasti knihovnictví. Díky výpočetní technice odpadla značná část manuální práce knihovníků, ale také možnost, práce duplicitní.

Dalším výrazným mezníkem bylo zavedení protokolu pro meziknihovní komunikaci, díky které knihovny mohou nabídnout svým čtenářům o mnoho větší komfort v podobě zajištění knihy z jiné pobočky knihovny.

Cílem praktické části je realizace knihovního systému, který by umožňoval základní funkce knihovního systému. Do systému je zajištěn přístup webovým klientem, který je napsán v jazyce HTML a doplněn o PHP skripty, které zajišťují vše od generování vzhledu až po zapůjčení knihy. Grafický vzhled aplikace je vytvořen pomocí technologie CSS. Všechna databázová data jsou uložena na Mysql serveru.

2 Knihovní systémy

2.1 Definice knihovního systému

Knihovním systémem je nazýván takový software (aplikace), která slouží k automatizaci transakcí probíhajících v knihovnictví. Obvykle se skládá z modulů a struktura cele aplikace je postavena na modelu klient/server. Mezi typické moduly knihovního systému lze zařadit moduly akvizice, katalogizace, výpůjčky, a online katalog (zkráceně OPAC).

V dnešní době jsou využívány tzv. integrované knihovní systémy. V takovém typu knihovního systému jsou ukládaná data sdílena mezi 2 nebo i více procesů. Odpadá tedy tím nutnost stejný údaj ukládat vícekrát pro různá zpracování. Jelikož jsou novodobé knihovní systémy distribuovány modulárně může si každá knihovna vždy zvolit, které moduly pro ni budou užitečné.

Dalším pohledem na dělení knihovních systémů je jejich propojení s ostatními knihovnami. Systém může být buď lokální, nebo kooperativní. Lokální systém využívá pouze knihovna, v které je systém instalován. V případě kooperativního systému jsou data ukládána do centrální databáze, odkud jsou využívána všemi připojenými systémy.

2.2 Knihovní standardy

Vzhledem k integraci knihovních systémů muselo být kvůli integraci zavedeno několik standardů.

2.2.1 Struktura záznamu

Hlavní bylo zavést určitý standard pro strukturu záznamů. V roce 1989 byl přijat mezinárodní formát UNIMARC, který byl používán v zemích, odkud Česká Republika dokumenty přebírala. V nedávné době mezinárodní organizace pro správu knihovních fondů rozhodla, že standard UNIMARC nebude již nadále podporován a tak začal přechod na formát nový, značený jako MARC21. Tento formát je spravován národní knihovnou Kanady. Během přechodného období knihovní systémy zvládly převod záznamů z formátu UNIMARC do formátu MARC21. (Stöcklová, 2006)

2.2.2 Pravidla popisu dokumentů

V České republice se pro popis dokumentů používají Angloamerické katalogizační pravidla (AACR2), která určují obsah jednotlivých polí. V polích by neměly být vyplněny údaje, podle kterých se nebude nikdy vyhledávat a jejich vyplnění by mohlo vést k nepřehlednosti a zkomplikování výměny dokumentů mezi knihovnami. (Stöcklová, 2006)

2.2.3 Standart pro meziknihovní výměnu

Pro komunikaci mezi knihovnami je standardně použit komunikační protokol Z39.50. Díky využití tohoto protokolu lze velmi jednoduše využít sdílenou katalogizaci a je obsažen ve všech předních knihovních systémech používaných nebo vytvořených na území České a Slovenské Republiky. (Stöcklová, 2006)

2.3 Základní moduly

2.3.1 Modul akvizice

Akviziční modul zajišťuje vytváření a průběžnou správu knihovního fondu. V modulu jsou evidovány informace o dodavatelích dokumentů pro knihovnu a náklady na získání jednotlivých dokumentů.

Ačkoliv se jedná o modul, který je obsažen v automatizovaném systému je vždy rozhodující lidský faktor. Na lidském rozhodnutí vždy závisí, co bude do knihovního fondu přidáno, tak aby v knižním fondu nevznikla duplicita nákupem dokumentu, který je již v knihovním fondu obsažen. Dále je potřeba mít neustále přehled o knižních novinkách, sledovat nové informace z nakladatelství. Akviziční modul vkládá do databázového systému prvotní informace již ve chvíli objednání. Objednané dokumenty tedy mohou být již zobrazeny v katalogu jen s určitým popisem (např. objednán).

2.3.2 Modul katalogizace

Katalogizační modul vytváří záznam, který se vztahuje pro jeden určitý dokument z knihovního fondu. Obsahuje přírůstkové číslo (jednoznačný a jedinečný údaj identifikující daný dokument v knihovně), popis dokumentu a další údaje. Záznamy jsou zadávány pomocí vstupů počítače, nejčastěji klávesnicí, lze ale použít i čtečku čárových kódů. Velmi důležité je, aby vkládání údajů do katalogu bylo co nejjednodušší a uživatelsky přívětivé.

V současnosti je velmi důležité aby katalogizační modul umožňoval vkládat i jiné typy dokumentů než tištěné monografie. Mezi další dokumenty, které je potřeba zadávat do katalogu

2.3.3 Modul výpůjček

Modul výpůjční služby se svou funkcí v každém systémů lehce liší. Každý výpůjční modul by měl umět rozlišovat mezi výpůjčkou absenční a prezenční. Na základě zadaných pravidel a knihovního řádu dokáže samostatně vypočítat poplatky jednotlivým čtenářům. Tak aby tyto výpočty fungovaly správně, je třeba zadat výše poplatku za pozdní vrácení knihy, doba, o kterou lze knihu prodloužit, doba na kterou je kniha vypůjčena a počet dokumentů, které si uživatel může vypůjčit. Zadání výpůjčky probíhá buď přímým zadáním z klávesnice, nebo pomocí čtečky čárových kódů. U každé výpůjčky musí být definováno jednoznačné označení dokumentu (tzv. přírůstkové číslo), identifikátor uživatele a datum vypůjčení. Datum musí být doplněno automaticky, ale zůstat musí možnost toto datum ručně pozměnit.

Modul nemá na starosti pouze vypůjčování dokumentů, ale také jejich vracení. Vrácení knihy probíhá podle identifikátoru dokumentu. Prodloužení nebo pouhá rezervace probíhá také podle čísla dokumentu nebo přes identifikátor vypůjčovatele.

2.3.4 Modul OPAC

Velmi důležitá součást knihovních systémů. Modul, který čtenářům, ale i knihovním zaměstnancům umožňuje prostřednictvím internetového prohlížeče zobrazit všechny záznamy v katalogu knihovny. Díky přístupnosti přes webový prohlížeč bývá místo slovního spojení Online Public Access Catalog (OPAC) používáno spojení Internet/Intranet Public Access Catalalog (IPAC). Záznamy lze buď velmi jednoduše procházet, nebo v něm vyhledávat podle zaběhlých zvyklostí ve světe internetu. Vyhledávání by mělo být dostupné ve všech dostupných polích nebo by měly být možnost vyhledávací dotaz specifikovat pouze na pole určitá (například jméno autora, název knihy atd.). V online katalogu by se neměly vyskytovat odborné knihovnické výrazy, tak aby pro uživatele bylo používání aplikace co nejpřívětivější.

Po přihlášení by měl mít uživatel možnost zobrazit informace o jeho čtenářském kontě, rezervovat si dokumenty, o které má zájem, ale i prodlužovat stávající výpůjčky.

(MIKA, 2000)

2.3.5 Porovnání existujících systémů na trhu

Porovnávány budou přední knihovní systémy používané v knihovnách na území České Republiky. Systém bude popsán nejdříve jako celek, následně bude popsána vždy základní struktura systému a funkce jednotlivých základních, ale i volitelných modulů, které se vyznačují určitou zvláštností oproti systémům jiným.

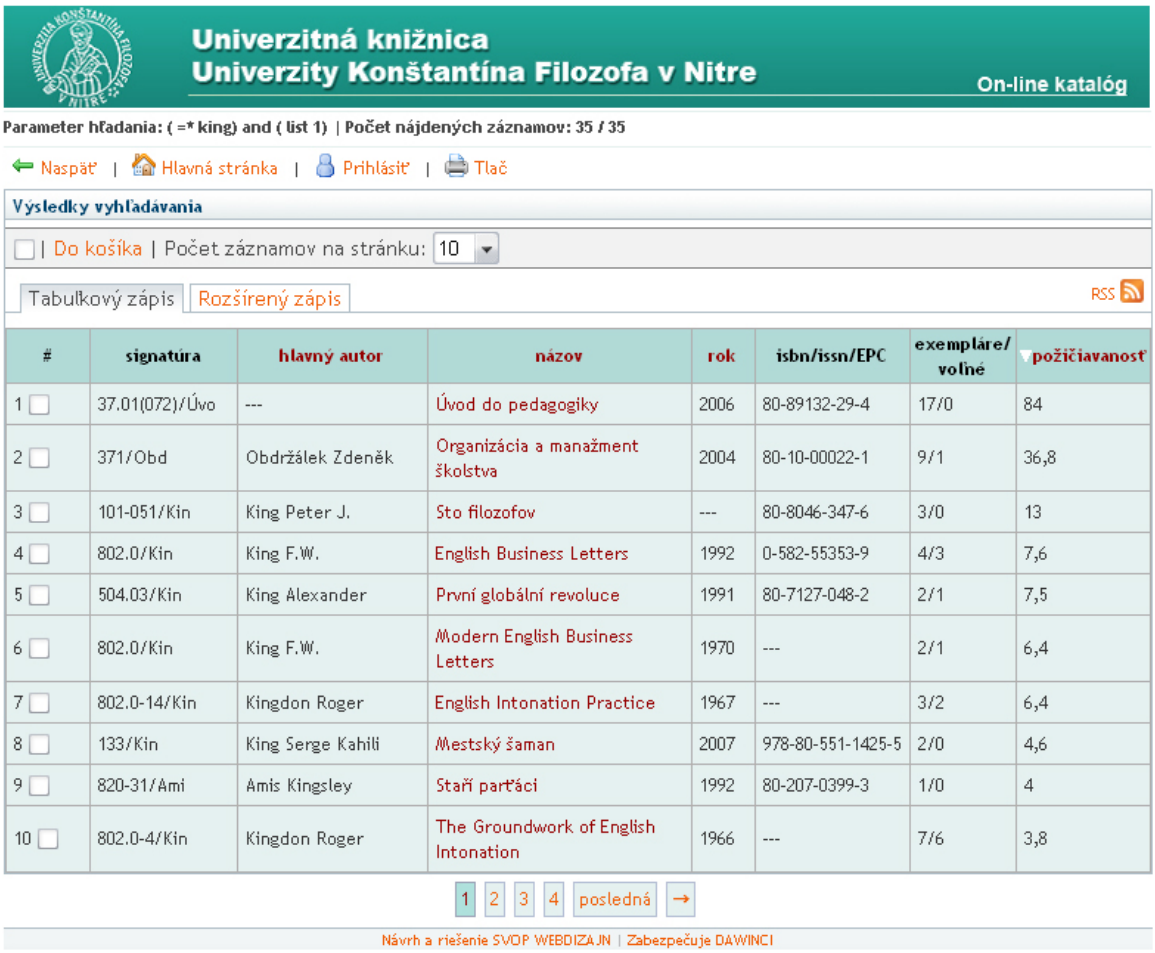
Popisovány budou následující knihovní systémy:

- Dawinci
- Clavius
- KPwin SQL

2.4 Dawinci

Dawinci je automatizovaný knihovní systém vyvíjený od roku 1996 slovenskou firmou SVOP s.r.o. ve spolupráci s několika zkušenými knihovníky. Celý automatizovaný systém se skládá ze serverové části, kde se serverové aplikace starají o logiku dat a všech operací spojenými s jednotlivými údaji.

Koncové aplikace lze rozdělit do jednotlivých modulů: modul administrátor, katalog, modul zajišťující výpůjční služby, online katalog. Model celého systému je tedy klient-server.



#	signatúra	hlavný autor	názov	rok	isbn/issn/EPC	exemplare/ voľné	požičiavanosť
1	37.01(072)/Úvo	---	Úvod do pedagogiky	2006	80-89132-29-4	17/0	84
2	371/Obd	Obdržálek Zdeněk	Organizácia a manažment školstva	2004	80-10-00022-1	9/1	36,8
3	101-051/Kin	King Peter J.	Sto filozofov	---	80-8046-347-6	3/0	13
4	802.0/Kin	King F.W.	English Business Letters	1992	0-582-55353-9	4/3	7,6
5	504.03/Kin	King Alexander	První globální revoluce	1991	80-7127-048-2	2/1	7,5
6	802.0/Kin	King F.W.	Modern English Business Letters	1970	---	2/1	6,4
7	802.0-14/Kin	Kingdon Roger	English Intonation Practice	1967	---	3/2	6,4
8	133/Kin	King Serge Kahili	Mestský šaman	2007	978-80-551-1425-5	2/0	4,6
9	820-31/Ami	Amis Kingsley	Staří parťáci	1992	80-207-0399-3	1/0	4
10	802.0-4/Kin	Kingdon Roger	The Groundwork of English Intonation	1966	---	7/6	3,8

Obrázek 1 Ukázka systému Dawinci (převzato z http://www.dawinci.sk/opac.html)

2.4.1 Modul administrátor

Tento modul slouží k administraci databází a jejich optimalizaci. Jednotlivé funkce by se daly rozdělit podle specifikací do třech různých skupin: funkce pro práci s databázemi, který umožňuje vytvářet nové databáze, upravovat stávající, provádět testování, optimalizaci atd.

Dalším typem funkcí jsou funkce, které umožňují měnit jednotlivé možnosti uživatelů. Samozřejmostí je vytváření uživatelských skupin a specifikace jejich zabezpečení (například přihlášení pomocí uživatelského jména hesla a tím zvýšení

bezpečnosti). Editovat se dají samozřejmě i jednotlivé možnosti uživatelských rolí jako aktivační poplatek, výpůjční doba, maximální počet půjčených knih, výše pokuty za nevrácení knihy v řádném termínu atd.

Funkce pro vlastní nastavení systému. Těmito funkcemi se nastavují základní systémové vlastnosti jako například kdy a jak budou generovány upomínky, jaké jsou výpůjční doby, poplatky, jednotlivé možnosti v oblasti služeb pro čtenáře atd.

2.4.2 Modul katalog

Modul katalog se skládá z dalších podmodulů:

- adresář dodavatelů, který slouží k evidenci dodavatelů a partnerů knihovny
- autority kde jsou budovány nebo přes protokol Z39.50 přenášeny soubory autorit
- evidence publikační činnosti

2.4.3 Modul výpůjček

eviduje čtenáře, čtenářské skupiny, poplatky, výpůjčky a další údaje týkající se výpůjček. Registrace čtenáře probíhá zadáním údajů do formuláře, který má Marc strukturu. Při vkládání je kontrolována duplicita zadaných údajů, ale také jejich správnost. Záznam o čtenáři obsahuje kromě běžných osobních informací také externí identifikátory jako například karty ISIC. Identifikace uživatele nemusí nutně probíhat podle čísla jeho čtenářského průkazu, ale také pomocí již zmiňované karty ISIC či například jednoznačného identifikátoru vypůjčeného dokumentu.

Na kartě uživatele je také vidět informace o případném překročení počtu vypůjčených dokumentů, upozornění na nezaplacení registračního poplatku nebo i informace o pokusu vypůjčit si knihu se speciálním knihovním režimem. Na kartě uživatele jsou mimo základních údajů také uvedeny údaje o výpůjčkách, které jsou barevně odlišeny podle stavu dané výpůjčky (například rezervace, upomínka, prodloužení).

Od dubna roku 2011 přibyla velmi užitečná funkce a to odesílání SMS zpráv jako další z možností pro komunikaci se čtenářem. Výhodou krátkých textových zpráv oproti běžné poště je hlavně jejich velmi nízká cena a rychlost doručení. Nastavení odesílání je velmi flexibilní a měnit lze prakticky cokoliv. Knihovna si může nastavit text pro jednotlivá upozornění, četnost odesílání, odesílání jen do některých zemí či název knihovny místo zobrazení telefonního čísla na straně příjemce. Modul na odesílání SMS je velmi praktický možností odeslat hromadnou zprávu v případě neočekávané situace, kdy je potřeba o situaci informovat všechny čtenáře knihovny.

2.4.4 Modul OPAC

OPAC je služba systému, který umožňuje uživatelům systému prohlížet tituly knihovny, provádět jejich rezervace a v neposlední řadě je vyhledávat. Vyhledávání titulů je maximálně přizpůsobitelné pro potřeby knihovny a je možné vyhledávat podle jednoduchého hesla, podle složeného výrazu, ale i fulltextové vyhledávání. Stejně jako vyhledávání, tak i formát výsledků hledání je maximálně přizpůsobitelný podle potřeb a přání knihovny. Možnosti výstupu jsou: tabulkový, rozšířený, ve standardizovaném Marc formátu (Unimarc, MArc21) nebo úplně volně definovatelný podle přání knihovny. Výsledky hledání lze seřadit podle jména autora, názvu, roku vydání. V detailu vyhledaných děl lze zjistit počet dostupných výtisků k výpůjčce, ale i díla, která byla zapůjčena zároveň s danou knihou.

2.4.5 Modul fakturace

modul, který nemá u konkurenčních AKS ekvivalentní zastoupení, stará se o zpracování a uchování ekonomických údajů

(SVOP, s.r.o. Bratislava, 2009)

2.5 CLAVIUS a LANIUS

Oba tyto systémy jsou tvorbou české firmy LANIUS s.r.o. Firma začala svoje aktivity v oboru knihovnictví již v roce 1992, kdy vznikl systém LANIUS, od roku 1997 začal vývoj nového, vylepšeného systému CLAVIUS, jehož hlavní výhodou oproti LANIUS bylo to, že byl vyvinut pro operační systém Windows. Ovládání je zcela intuitivní a shodné se zvyklostmi ovládání Windows a MS Office. K dispozici je mnoho předdefinovaných šablon vstupních a výstupních formulářů, které lze modifikovat podle přání knihovny. Formuláře lze tvořit taky jednoduchou metodou drag and drop prostřednictvím návrháře. Obsahuje 3 základní moduly, další existující jsou volitelné podle potřeb a přání knihovny.

Obrázek 2 Ukázka systému Clavius (převzato z http://www.hotskolabrno.cz/?stranka=168&nix=navod-pro-vyhledavani-v-on-line-katalogu)

2.5.1 Modul katalogizace

Jedná se o základní modul celého systému, standardně lze do katalogu přidávat monografie, AV media, hudební díla atd. Pro jednotlivé pracovníky lze nadefinovat pouze kategorie a formuláře, které budou pro svůj druh práce potřebovat. Cely proces katalogizace je v souladu norem MARC21/UNIMARC a hlavně AARC2. Ve všech krocích katalogizace je možné zobrazit ISBD dokumentu ve formě katalogového lístku. Pro manipulaci s výtisky je možné využít čárové kód.

2.5.2 Modul výpůjček

Jeho hlavní funkcí je evidence čtenářů, výpůjček a poplatků. Vypůjčení lze zjednodušit použitím čárových kódů. Verze SQL umožňuje zadat výši kreditu u každého čtenáře. Poplatky jsou počítány v závislosti na nastavení informací o daném čtenáři. Oznámení o rezervaci či o případné upomínce lze čtenářům zasílat prostřednictvím e-mailu. Různé informační funkce umožňují knihovně zjistit, kdo měl danou knihu kdy půjčenou, ale také který čtenář měl v libovolné době zpět jakou knihu půjčenou či rezervovanou. Velmi zajímavou statistickou funkcí je žebříček nejžádanějších dokumentů. Všechny výstupy, které lze tisknout lze také uložit do textového souboru či odeslat e-mailem.

2.5.3 Modul OPAC

Je určen především pro pracovníky knihovny, kterým umožňuje vyhledávat podle všech zadávaných údajů. Pro lepší přehlednost hledání vyhledávací formulář obsahuje pouze základní údaje jako například jméno autora, název knihy nebo téma dokumentu. Vyhledávací formulář lze ale také nastavit aby vyhledával ve všech polích databáze či pro využití logických operátorů při zadávání hledaného výrazu. Výsledky hledání lze ještě před zpracováním setřídit podle požadavků. Typickým zpracováním bývá tisk nebo export do souboru.

U vyhledaného záznamu je možné zobrazit kompletní bibliograficky údaj včetně počtu rezervovaných a půjčených výtisků. Pokud je k záznamu připojen multimediální soubor jako například obrázek nebo URL odkaz, lze jej okamžitě zobrazit. Pro umožnění hledání běžným čtenářům slouží modul WWW katalog, který je spuštěn na aplikačním serveru (například Apache, IIS). Cílem toho katalogu je zpřístupnit dokumenty knihovny on-line pro všechny čtenáře knihovny případně pro zaměstnance tak, aby po autentifikaci měli možnost rezervace knih, prodlužování výpůjček nebo rušení rezervací.

2.5.4 Volitelné moduly

Rozšíření o AV-média (katalogizace LP, CD, kazet, CDROM a DVD), Modul Evidence periodik (seriálů) a brožur, Modul Analytický popis článků a ostat. typů dokumentů (mimo AV a periodika).

Modul WWW katalog pro vystavení fondu knihovny (umožňuje i rezervace), Modul Revize knihovního fondu (snadno a rychle provede inventuru fondu knihovny), Modul Ishare Sdílená katalogizace po síti Internet.

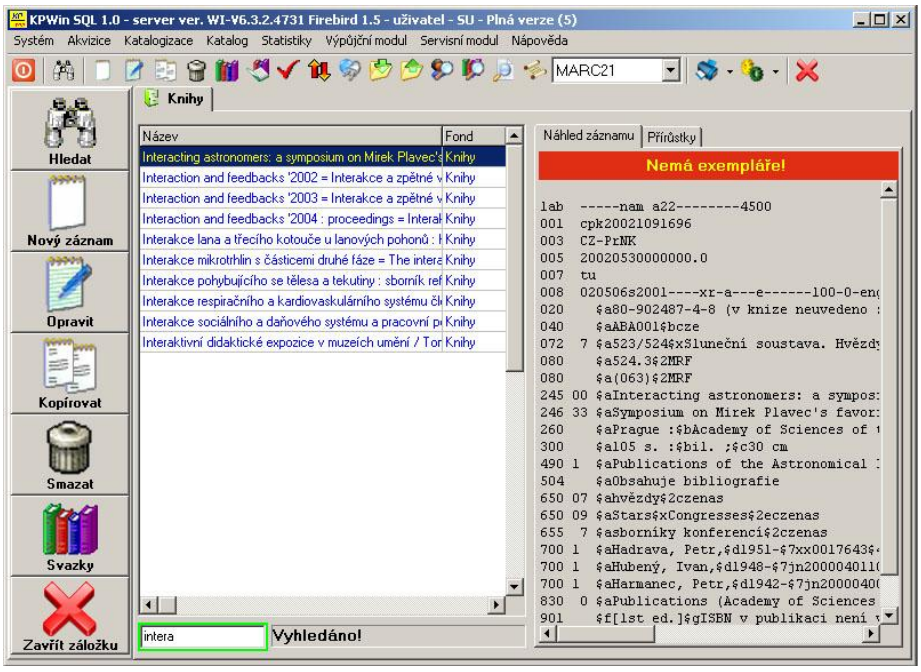
(LANius s.r.o., 2011)

2.6 KpWinSQL

Systém byl představen v roce 2005 českou firmou KP-SYS. Firma se zabývá vývojem knihovních systémů již od roku 1996. První systém distribuovaný touto společností nesl název KP-SYS a byl určen pro systém DOS. Následníkem tohoto systému je systém KpWin. Tento systém byl, jak už název napovídá určen pro práci v operačním systému Windows.

KpWinSQL byl vyvinut jako následník systému KpWin. Je vhodný k nasazení do jakékoliv knihovny jak v síťovém tak lokálním režimu. Serverová část nabízí multiplatformní řešení databázového systému Firebird, který je nabízen zdarma. Variantou k této verzi zdarma, je SQL server Interbase. Od konce roku 2007 je nabízen i Oracle SQL server a od roku 2008 i Microsoft SQL server.

Systém obsahuje všechny základní moduly, které by měl obsahovat moderní integrovaný knihovní systém, vyjma modulu OPAC, který je mezi moduly volitelnými.



Obrázek 3 Ukázka systému KPWin SQL (převzato z http://www.kpsys.cz/kpwinsql/screens.html)

2.6.1 Modul katalogizace

Modul zajišťuje kompletní správu přírůstků. Jednoznačně přiděluje přírůstkové číslo, čárový kód i signaturu. Probíhá v něm i správa seriálů, zobrazit lze historii výpůjček titulu, ale i jednotlivých exemplářů. Díky modulu je možné zjistit podrobnosti o exempláři buď v knihovně lokální, nebo při síťovém provozu i údaje z knihoven regionálních. V modulu je implementováno i základní vyhledávání na základě klíčového slova, změn v záznamech atd. Kombinací těchto vyhledávání lze realizovat i velice složitý vyhledávací dotaz.

Za výhodu lze považovat možnost definování vlastních tiskových sestav. Tisková sestava umožňuje například i automatické vygenerování HTML stránky a další.

2.6.2 Modul akvizice

Stejně jako v konkurenčních systémech tak i zde existuje databáze dodavatelů dokumentů pro knihovnu. Kromě klasického objednávání knih a časopisů je možnost zobrazit i náklady na objednání, průběžný stav rozpočtu.

V modulu dochází i k evidenci účtenek a dokladů o příjmu dokumentu do knihovny. Doklad je buď vytvořen automaticky na základě objednávky, nebo lze využit klienta protokolu Z39.50 pro přenos ze vzdáleného místa.

V dokladu jsou evidovány dodatečné položky jako poštovné a balné. V případě nákupu dokumentů v cizí měně, je částka přepočtena automaticky na požadovanou měnu. Modul je provázán s rozpočty a tak dokáže upozornit v případě, že celkové náklady se blíží k hranici rozpočtu.

2.6.3 Modul výpůjček

Vracení respektive vypůjčení dokumentu je založeno na jednoduchém sejmutí jednoznačné identifikace (čárový kód, přírůstkové číslo). Čtenáři jsou rozděleni do kategorií, pro která platí různá pravidla. Mezi tyto pravidla patří například různé výpůjční doby, různá výše poplatků atd. Ke každé výpůjčce nebo navrácení je možnost tisku potvrzení o této skutečnosti.

Modul eviduje všechny platby a pokuty. V případě problémového čtenáře lze jeho čtenářské konto zablokovat či omezit výpůjčky jen na určité pobočky. Čtenář je identifikován na základě jednoznačného čísla nebo čárového kódu. Pokuty a upomínky jsou rozesílány automaticky k určitému datu. Formát a styl doručení informace o upomínce nebo pokutě lze vytisknout, odeslat e-mailem nebo přes SMS zprávu.

(KP-Sys spol. s.r.o., 2010)

2.7 Návrh vlastního systému

Systém realizovaný v praktické části je v současnosti navrhnut spíše pro menší lokální knihovnu. Vzhledem k této skutečnosti nejsou v systému vůbec obsaženy funkce, které by splňovaly požadavky na modul akvizice a správa periodik. Celý systém je programovaný objektově ve skriptovacím jazyce PHP. Jako server slouží databázový server Mysql. Díky objektovému řešení lze systém kdykoliv a bez větších obtíží rozšířit o další funkce.

Nové dokumenty jsou vkládány do systému z administrátorského rozhraní. Administrátorem je v tomto systému myšlen pracovník knihovny (z pohledu uživatelských rolí je to takový uživatel, který má ID role 1). Pro zadávání nového dokumentu je připraven formulář pro vyplnění základních údajů jako název dokumentu, autor knihy, ilustrátor, jazyk, isbn knihy, počet stran a počet exemplářů, které přidáváme. Posledním polem je usnadněno vkládání v případě, že je do katalogu přidáváno více stejných výtisků. Po vložení je každému výtisku přiřazen jednoznačný identifikátor (přírůstkové číslo). Při vkládání nového titulu, potažmo nového umělce, lze přidat náhled knihy.

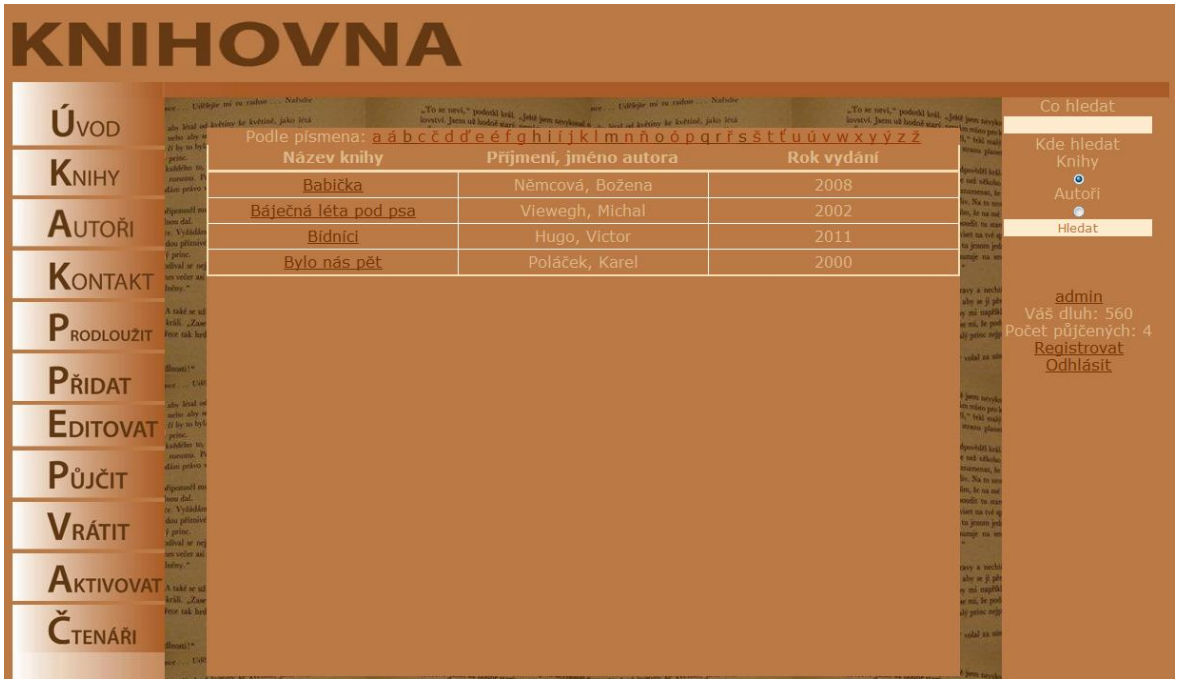
Administrátor má také možnost zadávat do systému nové umělce, kteří mohou být dále zadávání buď do pozice autora, nebo do pozice ilustrátora. Formulář pro toto zadání obsahuje pole pro jméno, příjmení, datum narození a národnost. Ke každému umělci je také možnost nahrát jeho portrét.

Pro práci s výpůjčkami a rezervacemi je připraveno mnoho funkcí jako například: vytváření výpůjček, prodlužování výpůjček, vytváření rezervací, rušení rezervací a vracení dokumentů. Vypůjčení knihy je dovoleno pouze uživateli, který má svůj účet aktivován. Tím je zamezeno přístupu k dokumentům uživatelům, bez zaplacení členských poplatků. Aktivace uživatele probíhá pouze z rozhraní administrátora - například po zaplacení členského poplatku. Kniha není také zapůjčena uživateli, který má v knihovně neuhrazenou dlužnou částku - tuto částku vidí každý uživatel po celou dobu jeho přihlášení do webové aplikace. Stejně tak, má uživatel k dispozici údaj o počtu půjčených knih. Takovému uživateli, je ale umožněno vytvořit rezervaci knihy. Vzhledem k tomu, že jediný přístup do systému zajištěn přes webové rozhraní, je rozesílání pokut každý den při první návštěvě webového rozhraní. Jako pokuta je počítán každý započatý týden nad rámec doby výpůjčky.

Výše pokuty je nastavitelná pro každou z uživatelských rolí zvlášť - každý uživatel má nastavenou příslušnost do určité skupiny a podle toho pro něj platí daná pravidla jako výše pokut, doba prodloužení, maximální počet půjčených knih a celková doba vypůjčení jedné knihy. Systém rezervací a výpůjček je postaven tak, že pokud jsou všechny exempláře dané knihy vypůjčeny, tak dojde k vytvoření rezervace. Pokud by chtěl uživatel, který si knihu půjčil jako první výpůjčku prodloužit, je mu to automaticky zakázáno. Ve chvíli kdy je kniha navrácena (nebo propadne rezervace dřívějšího data) je automaticky uživateli, který čeká na knihu nejdéle zaslán e-mail. že požadovanou knihu má již k dispozici.

Přihlášení do webového rozhraní probíhá přes jednoznačné číslo čtenáře, které je vygenerováno při registraci. Registrační údaje jsou také zaslány e-mailem. Přes webové rozhraní lze prohlížet katalog knih z knihovního fondu. V podrobném zobrazení jsou viditelné detaily o titulu a náhled knihy – pokud je na serveru nahrán. U detailu knihy je zobrazen počet výtisků ve fondu celkem, počet vypůjčených výtisků a celkový počet rezervací. Prohlížet lze také seznam všech umělců, kteří jsou zaznamenáni v databázi knihovny, včetně informace o počtu knih v knihovně u kterých je uveden jako autor. Ve webové aplikaci je také implementováno hledání. Hledat lze v názvech v případě hledání knih a v příjmeních (i jménech) autorů. U všech dostupných výpisů je možnost vyfiltrovat knihu, která začíná na vybrané písmeno a v případě umělců je vyhledáváno podle jejich příjmení.

Administrátor má k dispozici všechny funkce k editování umělců, jednotlivých knih, přidávání nových umělců, žánrů. Při editaci lze nahrávat nové náhledy knih, v případě autorů jejich portréty. Definovatelné administrátorem, jsou také jednotlivé uživatelské role. U každé role lze nastavit výše pokuty, maximální počet současně půjčených knih, maximální výpůjční doba. V administrátorském menu je také k dispozici editace všech registrovaných čtenářů. Má také možnost procházet detaily o jednotlivých uživatelích a přidávání novinek na úvodní strnu pomocí WYSIWYG editoru.



Obrázek 4 Ukázka vlastního systému

2.8 Celkové zhodnocení systémů

Všechny srovnávané automatizované knihovní systémy obsahují základní moduly, které by měl obsahovat každý moderní knihovní systém. Pokusím se tedy ve zkratce popsat výhody jednotlivých systémů, kterými se liší od konkurenčních produktů.

2.8.1 Dawinci

Systém Dawinci je typický jeho netypickým rozdělením modulů a hlavně přítomností modulu fakturace, který není v ostatních systémech nijak reprezentována a funkce, které modul fakturace umožňuje, zahrnuje modul akvizice.

2.8.2 Clavius

Za přednost sytému CLAVIUS považuji jeho intuitivní chování a vzhled podle základních zvyklostí nejrozšířenějšího operačního systému, Windows a kancelářského balíku Office. Další výhodou je tvorba vlastních formulářů jednoduchou formou Drag And Drop.

2.8.3 KPwinSQL

U produktu KPwinSQL lze označit za výhodu variantu, kdy je na místě databázového serveru použit server Firebird, který je poskytován zdarma, což se musí projevit na celkové ceně produktu. Tato skutečnost je nejspíše kompenzována tím, že modul OPAC je poskytován za příplatek.

2.8.4 Vlastní systém

Systém z praktické části nemůže být přímo srovnáván se zde zmíněnými systémy, které jsou vyvíjeny několik let profesionály a za pomoci zkušených pracovníků knihoven. Nicméně základní funkce nutné pro základní knihovní transakce v malé knihovně zvládá a pro další vývoj zde uvedu několik vylepšení, které by ho posunuli do jiné kategorie. Jedno z důležitých vylepšení by měla být možnost spravovat ve fondu knihovny i jiné dokumenty než pouze knihy. Dalším nedostatkem oproti profesionálním systémům je absence funkcí, které by měly na starosti evidenci dodavatelů knihovny nebo umožňovaly finanční situaci v knihovně jako například rozpočet knihovny, průběžné součty nákladů a další. Z pohledu standardů by byla užitečná implementace protokolu Z39.50 a ukládání záznamů ve formátu dnes standardního MARC21.

.

3 Návrh databáze pro knihovní systém

Databázová část knihovního systému je realizována Mysql serverem. Tabulky byly navrhovány tak, aby práce systému byla co nejvíce efektivní. Technologie Mysql byla zvolena z důvodu, že je dostupná zdarma a v jazyce PHP je pro ni velmi dobrá podpora. ER diagram lze vzhledem k rozsáhlosti vidět v přílohách.

3.1 Popis nejdůležitějších tabulek

V systému je použito tabulek více, budou proto popsány tabulky, které jsou z pohledu základní funkce systému důležité.

Tabulka users

Uchovává informace o uživatelích registrovaných v systému. Heslo k uživatelskému účtu je uchováváno v podobě md5 hashe. Informace o uživatelské roli, státu a městě je uložena jako příznak, který odkazuje na další tabulku. Jedná o tzv. cizí klíč, kterým je definován vztah k tabulce druhé. Datum konce aktivace je posunut při pokynu administrátora o 365 dní.

Tabulka 1 Tabulka users

Sloupec	Datový typ	Popis
Id	Int(11), (Auto increment)	Jednoznačné id uživatele
Jmeno	Varchar(30)	Jméno uživatele
Prijmeni	Varchar(30)	Příjmení uživatele
Pass	Varchar(30)	Heslo k účtu uživatele
Email	Varchar(30)	e-mail uživatele
Stat	Int(11)	Id státu uživatele (Cizí klíč)
Aktivovan_do	Date	Konec aktivace uživatele
Id_role	Int(11)	Id role uživatele (Cizí klíč)
Id_mesto	Int(11)	Id města uživatele (Cizí klíč)
Ulice	Varchar(30)	Ulice uživatele
Cp	Int(11)	Číslo popisné uživatele

Tabulka knihy

Uchovává informace o knihách v knihovním fondu. Jednoznačným identifikátorem knihy je sloupec id_kniha (v knihovnické terminologii tzv. přírůstkové číslo). Opět je využito cizích klíčů.

Tabulka 2 Tabulka knihy

Sloupec	Datový typ	Popis
Id_kniha	Int(11), (Auto increment)	Jedinečné označení exempláře
Nazev	Varchar(30)	Název knihy
Id_autor	Int(11)	Id autora knihy (Cizí klíč)
Id_ilustrator	Int(11)	Id ilustrátora knihy (Cizí klíč)
Id_nakladatelstvi	Int(11)	Id nakladatelství (Cizí klíč)
Id_jazyk	Int(11)	Id jazyka (Cizí klíč)
Rok	Int(11)	Rok vydání
Pocet_stran	Int(11)	Počet stran knihy
Isbn	Varchar (18)	ISBN knihy
Zanr	Int(11)	ID žánru knihy (Cizí klíč)

Tabulka vypujcky

V tabulce nejsou zaznamenávány jen výpůjčky, ale také rezervace. Důležitým příznakem je sloupec „vraceno“. Ten v případě výpůjčky (id_manipulace = 2) značí, že byla kniha vrácena, v případě rezervace (id manipulace = 1), že rezervace byla zrušena.

Tabulka 3 Tabulka vypujcky

Sloupec	Datový typ	Popis
Id_vypujcka	Int(11), (Autoincrement)	Jednoznačné označení výpůjčky
Id_kniha	Int(11)	Id vypůjčené knihy (Cizí klíč)
Id_user	Int(11)	Id uživatele (Cizí klíč)
Pujcena_od	Datetime	Časový údaj o vytvoření
Pujcena_do	Datetime	Konec rezervace/výpůjčky
Vraceno	Tinyint	Příznak o stavu
Id_manipulace	Int(11)	Příznak druhu záznamu

Tabulka umelci

Uchovává informace o všech umělcích vyskytujících se v záznamech knihovny. Pro určení národnosti autora je využit cizí klíč odkazující na tabulku narodnost.

Tabulka 4 Tabulka umelci

Sloupec	Datový typ	Popis
Id_umelec	Int(11)	Jednoznačné označení umělce
Umelec_jmeno	Varchar(20)	Jméno umělce
Umelec_prijmeni	Varchar(20)	Příjmení umělce
Id_narodnost	Int(11)	Id národnosti umělce (Cizí klíč)
Narozen	Date	Datum narození

Tabulka role

V této tabulce jsou uchovávány všechny důležité informace vztahující se k jednotlivým uživatelským rolím. Údaje o prodloužení knihy, výpůjčce a rezervaci jsou uvedeny ve dnech a tato doba je přičtena ke koncovému datu daného záznamu.

Tabulka 5 Tabulka role

Sloupec	Datový typ	Popis
Id_role	Int(11)	Jednoznačné označení role
Role	Varchar(20)	Název role
Pokuta	Int(11)	Výše pokuty za započatý týden
Prodlouzeni	Int(11)	Doba prodloužení výpůjčky
Rezervace	Int(11)	Doba rezervace knihy
Vypujceni	Int(11)	Doba výpůjčky knihy
Max_vypujcenych	Int(11)	Maximální počet půjčených knih
Max_doba_pujceni	Int(11)	Maximální doba půjčení 1 knihy

4 Současné technologie webových aplikací

4.1 Definice termínu

Definic termínu "webová aplikace" je možné vidět nespočet. Například podle některých definic je za webovou aplikaci označován program, který ke své základní komunikaci využívá protokol HTTP, a data jsou doručována v jazyce HTML. Oproti tomu firma SUN ke své edici J2EE označuje za webové aplikace takové, které jsou psány pro internet psány buď technologií Java bez rozdílu zda-li se jedná o Java pages či o Java servlet nebo o aplikaci psanou nejavovskou technologií jako například v jazyce Perl nebo CGI. (Harbridge, 2010), (Sun Microsystems, Inc, 2002)

Podle první definice musí být výstup aplikace ve formátu HTML, zatímco definice druhá nejspíše záměrné jednotlivé technologie rozděluje na javovské a nejavovské.

V této práci budeme webovou aplikací označovat aplikaci, která je uživatelům zpřístupněna přes počítačovou síť pomocí protokolu HTTP. Model aplikace je server/klient kde v roli serveru je webový server například Apache. Jako klient slouží webový prohlížeč. Prohlížeč zde vystupuje jako takzvaný tenký klient protože sám nezná logiku webové aplikace.

Velkou výhodou webových aplikací je, že při aktualizaci aplikace není potřeba na klientské počítače instalovat novou verzi.

Webová aplikace je často využívaná pro informační systémy organizací a firem, rezervační systémy, internetové obchody, bazary, diskusní fóra atd.

Nejznámějšími současnými technologiemi pro tvorbu webových aplikací jsou technologie html, xhtml, php, asp.net, ajax, java, python, ruby a další. Všechny tyto technologie lze rozdělit ještě podle principu na značkovací jazyky a skriptovací jazyky. Jazyky skriptovací lze rozdělit na klientské a serverové.

Klientské skriptovací jazyky se vyznačují tím, že skript je vykonáván až na straně klienta, kdežto u serverových je skript vykonán serverem. Pro snadnější porovnání jednotlivých technologií zde uvedu hlavní vlastní, výhody a nevýhody u jednotlivých technologií.

4.2 Značkovací jazyky

Jedná se o jazyky, které mají definovány určité značky. Tyto značky potom vysvětlují obsah nebo vzhled časti textu, který je obsahuje.

Prvním vyvinutým značkovacím jazykem byl jazyk GML vyvinutý firmou IBM. Vznikl z důvodu popisovat, uchovávat a zpracovávat různé právní texty. Na tomto základě začala organizace ISO vyvíjet v 80. letech připravovat normu pro značkovací jazyk. Jako první byl ustanoven standart SGML. Jazyk byl velice robustní a flexibilní a tak si ho pro popis dokumentace vybrala americká společnost Boeing Aircraft pro popis dokumentací

svých velkých projektů. Jazyk SGML používalo i americké ministerstvo obrany. Jazyk SGML byl sice velmi silný, ale zároveň velmi složitý, takže nebylo možné vyvinout nástroj, který by pokrýval všechny možnosti jazyka. Na základě SGML vznikly mnohem jednodušší jazyky, které známe už z dnešní praxe a to HTML a XML. Tyto jazyky´nemají tak silné vyjadřovací možnosti jako jejich předchůdce, ale díky jednoduchosti se značně rozšířily.

Značky značkovacích jazyků lze rozdělit na značky procedurální a značky deklarativní. Procedurální značka určuje, jak daný text bude formátován (například tučné písmo Arial). Hlavním zástupcem procedurálních značkovacích jazyků je jazyk HTML, pomocí kterého je značkována velká část www stránek. Oproti značkám procedurálním stojí značky deklarativní. Ty naopak neříkají nic o tom, jak bude daný text vytištěn, ale určují, o jaký typ textu se jedná (například jméno, příjmení). Hlavním zástupcem jazyků deklarativních je jazyk XML. (Vochozka, 2000),

4.2.1 HTML

HTML je programovací jazyk patřící do rodiny značkovacích jazyků. Struktura HTML dokumentu je přesně definována. Dokument je tvořen pomocí značek (tagů), které jsou psány do špičatých závorek.

Tagy se dělí na párové a nepárové. Mezi párové patří například tag <html>, který musí být ukončen tagem </html>. Mezi nepárové tagy patří například
. Pokud prohlížeč některý z tagů nezná, tak ho jednoduše ignoruje, což výrazně zjednodušuje tvorbu www stránek začínajícím tvůrcům, ale znemožňuje možnosti strojového vyhledávání. Uvnitř otevíracího tagu může být uveden jeden či více atributů, z nichž některé jsou z hlediska validity povinné a musí být uvedené včetně jejich hodnoty. Pokud je hodnota atributu shodná s názvem atributu, nemusí se hodnota uvádět. Pořadí elementů je dáno použitým DTD (Doc Type Definiton). V současné době je standardní verzí HTML verze 4.01 nicméně konsorcium W3C již připravuje standard pro HTML5, které přináší spoustu novinek jako práci s multimédii, nové tagy, přístup aplikace offline a další novinky. V současné době je již v některých prohlížečích část novinek implementována a tak je lze nabídnout čtenářům webu, kteří používají nové verze prohlížečů jako například Google Chrome či Safari. V opačném případě uživatelů se staršími prohlížeči lze nabídnout verzi v dnešní době standardní, tedy verze 4.01.

Jednotlivé HTML DTD:

- DTD strict - velmi přísná pravidla, chybí tagy sloužící k úpravě vzhledu dokumentu jako například , celý vzhled je řešen pomocí CSS
- DTD transitional - použití všech zvyklostí jazyka HTML, pouze se nesmí používat rámy (frame / frameset)
- DTD frameset - stejná pravidla jako pro transitional, lze ale použít rámy

(Janovský, 2011)

4.2.2 XHTML

Jedná se o programovací jazyk, který si vzal něco z html i z jazyka XML, který je svými pravidly přísnější nežli jazyk HTML. Pokud budeme psát dokument v jazyce XHTML, máme jistotu, že výsledný dokument bude syntakticky správný jak podle pravidel HTML tak i XML. Ve struktuře dokumentu nesmí chybět deklarace použitého DTD a žádná ze značek definována daným DTD. Stejně jako v HTML jsou u technologie XHTML tagy párové a nepárové. Pokud se jedná o tag nepárový, je tag doplněn lomítkem. Dalším rozdílem oproti HTML je povinnost uvádět hodnotu všech atributů. Vzhledem ke striktnějšímu přístupu XHTML než HTML je povinností psát tagy a jejich atributy malými písmeny a tagy, které prohlížeč nezná, nejsou ignorovány, tudíž musí být odstraněny. Hodnoty atributu na rozdíl od HTML musí být v uvozovkách.

V praxi se používají 3 verze XHTML a to verze 1.0 transitional, 1.0 strict a verze 1.1. Do konce roku 2009 byl vyvíjen nový standart XHTML 2.0. V prosinci roku 2010 ještě skupina vydala modul pro RelaxNG, ale to bylo poslední dílo této skupiny. V den vydání modulu byla skupina, která na vývoji pracovala rozpuštěna, protože konsorcium W3C se rozhodlo zaměřit pouze na vývoj HTML5.

(Snížek, 2005), (W3C, 2010)

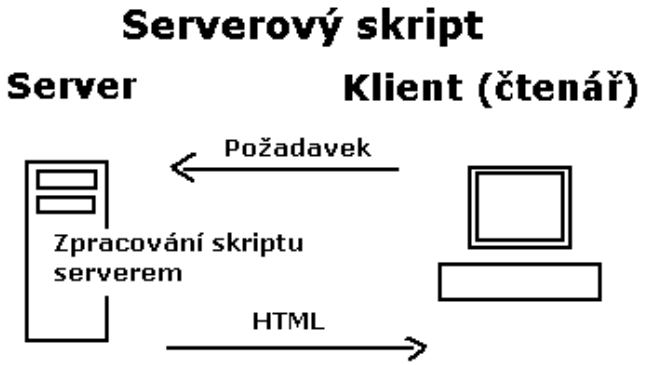
4.2.3 XML

Jazyk, který se v dnešní době používá hlavně ve sféře B2B. Prvotní myšlenka ovšem nebyla taková. Každý XML dokument musí obsahovat jeden kořenový element, který má počáteční tag na začátku dokumentu a koncový na konci. Jazyk nemá žádné definované značky a každý si může vytvořit vlastní podle potřeb. Vzhledem tomu, že XML neobsahuje žádné informace o vzhledu obsahu, lze jej označit za jakýsi předěl mezi databázovou strukturou a textovým dokumentem. Tím, že nejsou definovány žádné značky, je umožněna naprostá flexibilita a značně zjednodušuje výměnu informací bez ohledu na platformy či aplikace. Stejně tak jako jazyk HTML nebo XHTML tak i elementy v jazyce XML mohou mít jeden či více atributu, jejichž hodnoty se uzavírají do uvozovek. Všechny značky jsou párové. Jedinou výjimkou jsou elementy, které nemají žádný obsah, takové tagy ale pak musejí být ukončeny. Dalším pravidlem je zákaz křížení elementů. (Adaptic, s.r.o. - Programování aplikací podle vašich potřeb, 2011)

4.3 Serverové skriptovací jazyky

Zdrojový kód skriptu je prováděn již na straně serveru. Skript má přístup k hardware serveru. Komunikaci s klientem lze zajistit pouze pomocí webových formulářů. Nejčastěji jsou tyto skriptovací jazyky používány u aplikací u kterých se spolupracuje s databázovým serverem. V minulosti serverových technologiím vládly skripty CGI v Perlu. V současnosti jsou předními hráči PHP a ASP. NET. Všechny serverové technologie mají společné, že na serveru musí být nainstalována podpora pro danou technologii. Pomocí serverových technologií lze naprogramovat jednoduché počítadlo přístupu na web, anketu a už vůbec ne elektronický obchod.

Ve všech těchto případech potřebujeme výslednou stránku generovat dynamicky podle zadaných údajů. (Janovský, 2011)



Obrázek 5 Zpracování serverového skriptu (převzato z http://www.jakpsatweb.cz/programovani.html)

4.3.1 PHP

PHP je rekurzivní zkratka slovního spojeni Hypertext preprocessor, původním významem však bylo Personal Home Page. Jedná se o skriptovací technologii, jejíž historie začíná v roce 1995 kdy Rasmus Lerdolf napsal pro své potřeby několik skriptů v jazyce Perl. Potřeba nových funkce vedla k rozvíjení projektu a PHP bylo přepsané do jazyka C. S verzí 3.0, která byla zveřejněna v roce 1998, přibyla podpora objektově orientovaného programování. Verze 4.0 přinesla podporu HTTP sessions a použití lokálních proměnných

V současnosti je aktuální verzí verze 5.3.6, která byla zveřejněna 17. března 2011. Verze 5 přinesla výraznou změnu v OOP práci v php - konstruktory, destruktory, výjimky a další rozšíření. PHP je velmi oblíbenou technologií hlavně kvůli velmi jednoduché syntaxi jazyka. Výhodou je zpracování skriptu již na straně serveru, kdy je klientovi odesíláno již čistá HTML stránka. Vyplývá z toho tedy, že skript nelze jednoduše zcizit na rozdíl od Javasciptu uvedeného přímo ve zdrojovém kódu HTML stránky. Nesmíme také opomenout, že se jedná o technologii, která je zdarma a open-source. Díky licenci open-source mají vývojáři možnost pozměnit si cokoliv podle svých potřeb a přání. Za nevýhodu lze považovat, že kdokoliv má přístup k serveru, tak má možnost vidět zdrojové kódy skriptů.

K vývoji aplikace v jazyce PHP stačí vývojáři jakýkoliv textový editor jako notepad, PsPad, EditPlus a další. Dále pro běh skriptů je třeba mít na webovém serveru nainstalován modul, který se bude starat o spouštění skriptů. Pro PHP je k dispozici spousta různých knihoven, které se starají například o zpracování grafiky, textů, práci se soubory nebo připojení a práci s databázovým serverem. Podporuje také samozřejmě řadu používaných internetových protokolů (http, SMTP, FTP, POP3 a mnoho dalších).

(The PHP Group, 2011), (Žďárek, 2011)

4.3.2 ASP

Zkratka ASP znamená Active Server Pages. Už i z názvu lze tedy poznat, že se jedná o serverovou technologii. Technologie byla vyvinuta firmou Microsoft a už v době vzniku ASP (1996) mělo ASP hodně věcí společných s PHP. Oproti PHP má ale o dost horší přenositelnost mezi platformami. Na rozdíl od PHP, které lze nainstalovat jak na OS Linux, Unix i Windows lze totiž ASP zprovoznit pouze na serverech s operačním systémem Windows. Princip fungování ASP je shodný s PHP, skript je tedy do HTML kódu vkládán pomocí vsuvek. Jednotlivé technologie se liší stylem vkládání – PHP se uzavírá mezi značky <?php a? > kdežto ASP mezi značky <% a %>. Při psaní složitějších strukturovaných skriptů, kde se bude stejná část kódu spouštět vícekrát, se využívá funkcí a procedur. Procedurou je nazývána malá část kódu, který je nejčastěji definován na začátku skriptu a spouštěna může být kdykoliv během běhu skriptu. Špatně napsaná procedura může bohužel velmi zpomalit běh skriptu.

4.3.3 ASP. NET

ASP.net je technologie, která není přímým následníkem technologie ASP, protože původní ASP bylo programováno strukturálně, kdežto ASP. NET využívá frameworku .NET. Tato technologie byla vyvinuta firmou Microsoft. Technologie ASP.net ovšem od technologie PHP již rozdílná je a to zásadně tím, že zdrojový kód aplikace je kompilován.

Zdrojový kód aplikace se kompiluje při prvním spuštění aplikace. Logika aplikace je oddělena od vzhledu stránky. Vzhled stránky je tvořen klasickou definicí pomocí HTML značek a kód tvořící logiku aplikace je připojen v souboru na pozadí. Tento zdrojový kód může být napsán v jakémkoliv jazyce z rodiny. NET. Vývojář, který tedy umí jeden z. NET jazyků se nemusí učit nic moc nového, aby mohl používat technologii ASP.net.

Připojené soubory mohou být psány ve více jazycích, ale nutnou podmínkou je, že daný jazyk musí podporovat tzv. částečné třídy (například Visual basic, C#). Touto podmínkou je ovšem vyloučen jazyk J#. Mírnou nevýhodou této technologie, je provoz Microsoft serveru nebo systému IIS.

4.3.4 Ruby

Tento programovací jazyk se dostal do podvědomí hlavně díky frameworku Ruby on rails. Autor jazyka Yukihiro Matsumoto („Matz“) vycházel z teorie, že programování je tvořivá činnost, která má programátorovi přinášet radost.

Hlavní motivací tedy bylo vytvořit jazyk, který umožní využít co nekomfortnější vyjadřovací prostředky. Neznamená to ale, že by Ruby ztrácelo něco na integraci regulárních výrazů, snadno čitelnou syntaxi a už vůbec, že by jazyk nebyl objektově orientovaný. Matsumoto se odvolává na to, že v různých jazycích různě myslíme. Tento názor vyslovil i Ludwig Wittgenstein když tvrdil, že to co nemůžeme v jazyce vyjádřit, si nemůžeme ani myslet.

Framework Ruby on Rails vychází z principů a myšlenek jazyka Ruby Rails se staly populárními hlavně díky tomu, že zjednodušují práci s Ajaxem a programátorům

umožňují se soustředit na to, co chtějí udělat nežli na to, jak to udělají. Velkou nevýhodou Ruby ovšem je, že nepodporuje UNICODE, takže je nevhodný pro tvorbu aplikací ve více jazykových mutacích.

(Minařík, 2007)

4.3.5 JAVA EE

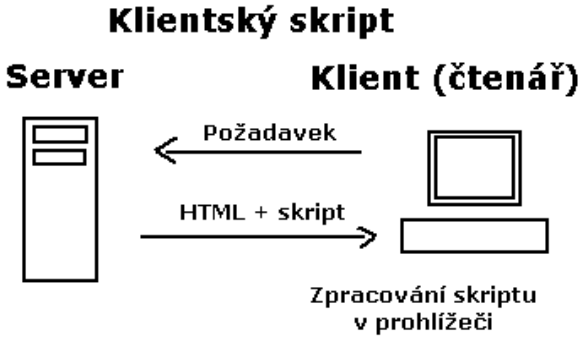
Zkratka EE znamená Enterprise Edition a zahrnuje další metody tvorby webových aplikací: JavaServlets, Java Server Pages a Java Server Faces.

Java nabízí mnoho předností, ať už se jedná o její multiplatformnost či o skvělý objektově orientovaný přístup. I přes to, že je to vyvinutá a prověřená technologie, je mezi běžnými webovými vývojáři velmi málo oblíbená a bývá označována za technologii hodící se pro velké podniky jako například banky. Tato skutečnost je způsobena hlavně tím, že hosting pro tuto technologii je náročnější a z finanční stránky i mnohem nákladnější nežli hosting například pro PHP. Vývoj Javy začal již v roce 1991, kdy se jmenovala Opak. První verze Javy, tedy Java 1.0 spatřila světlo světa v roce 1995. První specifikace Java Server Pages vznikla jako reakce na PHP a ASP v roce 1999. Server, na kterém aplikace běží, za programátora řeší věci jako spojení s databází, řízení uživatelských rolí a jejich práv a další. Jako server může sloužit například software Glassfih.

(Kučera, 2010)

4.4 Klientské skriptovací jazyky

Skript je přenesen spolu s daty stránky ze serveru ke klientovi a je zpracováván až klientem tudíž server nemá zpracování skriptu pod kontrolou. Toto lze považovat z určitého hlediska za výhodu, jelikož nedochází ke znovu načítání stránky a zatěžování serveru. Bezpečnostním rizikem je, že skript má přístup k hardware klientského počítače. Může nastat situace, že na straně klienta nebude možné skript provést. Toto se může stát buď kvůli omezenému výpočetnímu výkonu zařízení spouštěcí daný skript (např. mobilní telefon) nebo jednoduše proto, že není obsažen (např. vyhledávací roboti). Kvůli těmto skutečnostem by na webu využívající skripty měla být možnost zobrazit obsah také pro klienty, kteří nemají možnost skripty provádět. Skripty mohou být prováděny ihned po načtení stránky, ale také jako reakce na určitou situaci, která vznikne během prohlížení.



Obrázek 6 Zpracování klientského serveru (převzato z http://www.jakpsatweb.cz/programovani.html)

4.4.1 Javascript

Často bývá mylně zaměňován s Javou, Java je ovšem samostatný programovací jazyk a s Javascriptem mají pouze podobnou syntaxi protože Javascript vychází právě z Javy. Jedná se o skriptovací jazyk využívající objektového modelu. Jedním z používaných objektů je například Window, který obsahuje všechny instrukce týkající se nastavení okna prohlížeče. Nejdůležitějším objektem je však objekt document, který v sobě zahrnuje všechny objekty vztahující se k danému dokumentu. Jedná se například o odkazy, odstavce atd. Přístup k jednotlivým objektům se provádí pomocí identifikátoru metodou getObjectById. Práce s objekty probíhá přes tečkovou notaci jako v jazyce Java, z kterého Jscript vychází.

Výhoda spočívající ve spouštění skriptů na straně klienta se však velmi rychle může stát nevýhodou ve chvíli, kdy potřebujeme použit složitější funkci. Bohužel na stejnou funkci nemusí všechny prohlížeče reagovat stejně. Další nevýhodou použití je, že potencionální útočník může javascriptu využít. Kvůli této skutečnosti může být na některých počítačích javascript zakázán a tím, vzniká riziko nepoužitelnosti. Pro tuto situaci by na každém webu mělo být řešení, které při nejmenším uživatele upozorní, že pro správné zobrazení bude Javascript potřebovat. Pomocí Javascriptu lze naprogramovat celý program nebo ho využít jen k oživení stránek nějakým zajímavým efektem.

Častým využitím bývá výsuvné menu, kde jsou použity funkce onMouseOver a onMouseOut k zobrazení respektive skrytí určitého obsahu, který slouží jako podmenu. Dalším možným použitím je reagování na stav HTML formulářů před jejich odesláním na serveru. Kontrolovat jednotlivá pole lze například z hlediska jejich vyplnění, případně správnosti tvaru zadaných údajů a následné upozornění uživatele na vzniklé chyby. Javascript lze zapsat buď přímo do webových stránek do míst, kde má reagovat na danou událost nebo do externích souborů, které mají příponu .js.

4.4.2 Ajax

Zkratka Ajax znamená Asynchronous JavaScript and XML. Jak z tohoto slovního spojení vyplývá, jedná se o kombinaci XML, JavaScript, HTTP a (X)HTML. Tato technologie umožňuje za pomocí Javascriptu kontaktování serveru tak, aby nemusela být stránka celá znovu načítána. Data jsou získána ve formátu XML.

V praxi lze vidět Ajax například ve formě našeptávačů ve vyhledávačích, kdy se vám ve vyhledávacím poli průběžně během psaní dotazu mění napovídané výrazy, které by vám mohli usnadnit hledání. Hlavní nevýhodou je nemožnost respektive nefunkčnost tlačítka zpět v prohlížeči. Pokud není tento problém ošetřen například pomocí Javascriptu, případně krkolomným způsobem pomocí <iframe> tak po použití tlačítka Zpět se uživatel dostává na předchozí URL místo uživatelem očekávanou změnu v aplikaci. Nevýhodou je, že při použití například při využití místo klasickým odkazů ztrácíme zpětné odkazy, proto by se dal Ajax přirovnat k Flashi a neměl by se používat pro obsahovou část webu.

(Zralý, 2005), (Vrána, 2005)

5 Zabezpečení webových aplikací

5.1 Cíle útoků

Za útokem může stát hned několik faktorů. Prvním z nich je, že útočník nemá v úmyslu způsobit destruktivní změny, ale například pouze změnit vzhled aplikace čímž dá vývojáři najevo, že v jeho aplikaci je jistá bezpečnostní díra, kterou lze zneužít. Dalším motivem útočníka může být mnohem závaznější myšlenka a to destruktivní změny v aplikaci. Tyto změny mohou být ve smyslu smazání dat z databázového systému či úplné odstavení aplikace čímž se aplikace stává nedostupnou. V neposlední řadě může útočník chtít získat citlivá data o uživatelích systému. Tyto údaje jsou ukládány rovněž v databázovém systému.

5.2 Typy útoků

Popsány budou nejčastější typy útoků na webové aplikace. U každého útoku je vždy uvedeno opatření proti možnosti jeho vzniku.

Popisovány budou útoky:

- XSS
- CSRF
- SQL injection
- PHP injection
- Session hijacking

5.3 XSS (Cross site scripting)

Jedná se o nejběžnější typ útoku, který nastává v případě, kdy jsou odesílána uživatelská data, aniž by předtím byla nějakým stylem ošetřena například šifrováním nebo nahrazením HTML znaků jejich entitním vyjádřením. Tato bezpečnostní chyba umožňuje hackerům spustit škodlivý skript, změnit vzhled webové stránky. Útok probíhá pomocí kódu psanému v jazyce HTML, XHTML nebo jednom ze skriptovacích jazyků – nejčastěji Javascript. Toto ale nemusí být pravidlem a útok může být realizován jakoukoliv jinou technologií jako např. Flash, VBScript a další. Typicky se útočník snaží využít tagu <script>, toto lze monitorovat, ale přes to zůstává způsob jak škodlivý kód do stránky vložit.

Velmi účinnou ochranou je validace kritických částí stránky a to zejména:

- headers
- cookies
- GET a POST požadavky
- databázové dotazy
- formulářová pole
- skrytá pole

Validací v tomto případě myslíme odstranění HTML značek nebo jejich kódování na kombinaci odpovídajících znaků. K validaci existuje několik nástrojů – pro velmi oblíbený skriptovací jazyk PHP je to funkce htmlspecialchars(), které předáme jako parametr vstupní data.

5.3.1 Typy XSS útoků

Lokální

Tento typ chyby nastává ve chvíli, kdy JavaScript v aplikaci přistupuje k parametru URL adresy a dál tuto informaci neobaluje žádnou HTML entitou. Tento útok není prakticky rozdílný od dočasné chyby, tedy v současné době. V době kdy totiž nebyl ještě vydán bezpečnostní balík Service Pack 2 pro operační systém Windows XP tak webový prohlížeč Microsoft Internet Explorer pracoval s klientským skriptem stejně jakoby by byl spouštěn přímo z disku klientského počítače a tak měl útočník možnost spustit skript s privilegii prohlížeče.

Dočasné

Nejčastější případ XSS chyby. Vzniká ve chvíli, kdy webový server používá data přijatá od klienta ihned ke generování dalšího obsahu, aniž by vstupní data byla ošetřena (například odstraněn výskyt HTML značek). Tyto data mohou být použita například jako indikace hledaného řetězce nebo mohou být předvyplněna do hledacího pole pro snadnější úpravu. Toto se jeví jako relativně malé nebezpečí, ale útočník tímto stylem může uživatele donutit kliknout na jeho URL, kterým ovlivní zobrazená data. Tento útok byl v minulosti z pohledu útočníků velmi oblíbený a úspěšný.

Trvalé

Trvalé chyby umožňují uskutečnit nejvíce nebezpečné útoky. Chyba spočívá v neošetřených vstupech tak, že informace jsou rovnou ukládána na server (do databáze, filesystému) jako v případě internetových diskuzí. Jelikož jsou data ukládána na server, může jejich odstranění trvat delší dobu a mohou ovlivnit velké množství uživatelů, proto jsou tyto útoky označovány za trvalé.

XSS proxy

Tento způsob útoků, pomocí kterého lze kompletně ovládnout uživatelův počítač byl představen v roce 2005. Nástroj, který v roce 2005 představil Anton Rager se skládá ze dvou částí. První část je napsána v JavaScriptu, díky XSS zranitelnosti webu je tento škodlivý kód dopraven k uživateli kde se z něj stává klient, který obsahuje různé funkce umožňující útočníkovi ovládat napadený počítač a zároveň komunikovat se serverovou částí škodlivé aplikace. Serverovou část napsanou v Perlu má útočník spuštěn na serveru s veřejnou IP adresou a naslouchá na dvou portech. Jeden z portů je využit pro komunikaci klientů se serverem a druhý k ovládání jejich funkcí.

(Malý J., 2007)

5.4 CSRF

Zkratka CSRF v českém překladu znamená „podvržení požadavku mezi různými stránkami“. Někdy je útok označován také termíny jako XSRF, Cross-Site Reference Forgery, Session Riding nebo Confused Deputy attacks.

Často se stává, že je útok CSRF nesprávně zaměňován s útokem typu XSS. Ve skutečnosti se jedná, ale úplně jiný druh útoku. A pokud je webová aplikace zranitelná útokem XSS, tak je zranitelná i útokem CSRF. K útoku CSRF postačuje jedna jediná možnost útoku XSS. Protože útočník může v tu chvíli emulovat například přihlášení do webové aplikace

. Nejjednodušší strategií je zaútočit pomocí HTML formuláře, který odesílá svá data metodou get. Poté útočníkovi stačí, aby do aplikace přihlášený uživatel navštívil jeho stránky s připraveným elementem, který umí provést kritickou akci (může se jednat o automatické odeslání nesmyslného e-mailu, ale v případě nedostatečného zabezpečení autorizace bankovní transakce i o převod peněz). Mohlo by se zdát, že dostačující ochranou je odesílání formulářů metodou POST. To je ale pravda pouze v případě, že uživatel, přes kterého útok probíhá má ve svém prohlížeči zakázaný JavaScript. Útok přes formuláře s metodou POST totiž probíhají pomocí JavaScriptu, kdy při načtení dokumentu je vytvořen formulář, který má shodné elementy jako originál a zbytek útoku probíhá stejně jako u způsobu předchozího – tedy odeslaní požadavku danému skriptu, který jej zpracuje, jako kdyby ho uživatel odeslal vědomě – vše probíhá ze strany uživatelova klienta.

Někdy je jako řešení tohoto problému uváděno kontrolování http hlavičky referer. Toto řešení, al nemusí být vždy úspěšné, protože referer může zablokovat například proxy server nebo jeho odesílání může být zcela zakázáno.

Jediným efektivním řešením je použití tzv. tokenů. Tokenem je označován skrytý jedinečný kód, který je vygenerován, uložen na serveru a odeslán klientovi. Při odeslání formuláře je připojen i daný Token a server před zpracováním dat porovnává přijatý Token s tokenem uloženým, pokud se neshodují, tak se kritický kód neprovede. Token lze

generovat pro celé přihlášení nebo pro každé načtení formuláře zvlášť. Tento styl ochrany používá například v České Republice velmi oblíbená služba Seznam.cz k přihlašování ke svým e-mailovým účtům.

Největším nebezpečím tohoto útoku je jeho velmi špatná odhalitelnou, že byl útok proveden.

(Ferschmann, 2008)

5.5 SQL Injection

Technika útoku, kterou jsou ohroženy všechny webové aplikace pracující s jakýmkoliv databázovým systémem. Nebezpečí existuje u použití všech databázových systémů bez ohledu na platformu. Tedy riziko je shodné při použití serveru MySQL, Oracle SQL, MS SQL a dalších, protože všechny dotazovací databázové jazyky mají základní logiku stejnou.

Úspěšný útok se projeví na jedné nebo více odvětví:

- Důvěryhodnost – ztráta důvěryhodnosti nastává ve chvíli, kdy se útočník úspěšně připojí k databázi a získá z ní citlivá data
- Autentizace – pokud se útočník do webové aplikace přihlásil bez znalosti uživatelského jména hesla, odpovídá to nevhodným SQL dotazům v přihlašovací části aplikace
- Autorizace – pokud jsou v nabourané databázi uloženy údaje potřebné ke správné autorizaci, tak útočník dostává možnost tyto údaje pozměnit a tak si zvýšit oprávnění k obsluze aplikace
- Integrita – následkem úspěšného útoku může být i narušení integrity dat, tedy jejich správností. Porušení integrity nastává ve chvíli, kdy má útočník možnost měnit nebo mazat záznamy v databázi

Základní zjištění, zdali je web náchylný k SQL injection je velmi jednoduchý. Útočníkovi stačí, aby zkusil do vstupních polí nebo jako parametr do URL adresy zapsat lichý počet apostrofů (tím ukončil podmínku v SQL dotazu). Ve chvíli, kdy je zjištěno, že SQL injection je reálné Pokud web odpoví chybovou hláškou SQL serveru, je zřetelné, že je velká šance provedení útoku.

Pokusy o útok mohou tedy pokračovat. Základním cílem útočníka je přihlásit se do aplikace bez znalosti uživatelského hesla, k tomu mu může pomoci zadání i naprosto jednoduchého spojení ' OR 1=1-- zadaného do pole pro uživatelské jméno. Prvním apostrofem dojde k ukončení podmínky pro vyhledání uživatelského jména (zůstane tedy v dotazu nevyplněné), klíčovým slovem OR si získá možnost druhé varianty podmínky, která bude v jeho dotazu vždy vykonána (1 = 1 bude platit vždy). Díky dvojici pomlček dojde k ignoraci zbytku původního dotazu, jelikož bude označen za komentář.

Pokud útočník dokáže zjistit, jaký databázový systém je používán, dokáže například v případě SQL serveru využít předinstalovanou proceduru, pomocí které si může osah jakékoliv tabulky (tedy klidně i celé databáze) uložit do souboru ve formátu HTML a nahrát ho na zvolený síťový disk, kterému má přístup.

Základem ochrany je stejně jako u útoků typu XSS kontrola vstupů. Kontrola by měla spočívat v kontrole datových typů. Tedy pokud vstupem pole má být číslo, tak jestli se jedná vážně o číslo nebo u vstupních polí jednoduché uvozovky nahrazovat uvozovkami dvojitými. Další ochranný prvek v případě PHP přináší samotné jádro PHP a to tak, že najednou může být vykonán pouze jeden SQL dotaz.

Z pohledu správce databázového systému je velmi účinným přístupem účtu, pod kterým se aplikace přihlašuje do databázového systému nastavit uživatelská práva pouze nezbytně nutná pro provoz aplikace. Takový účet tedy nepotřebuje disponovat právy například k mazání tabulek nebo dokonce celé databáze. V neposlední řadě by aplikace neměla zobrazovat chybové hlášky SQL serveru. Zobrazení těchto hlášek útočníkovi značně zjednodušuje útok.

(Vrána, 2005)

5.6 PHP injection

Jedná se o nepříliš známý typ útoků, nicméně chyby k jeho provedení jsou velmi časté. Chyba je sice častá, ale dopouštějí se jí víceméně pouze začínající PHP programátoři, kteří při svém vývoji nemyslí příliš na bezpečnost. Princip útoku spočívá v zneužití funkce PHP include() nebo jedné z jejich obdob include_once(), require() nebo require_once(). Detaily, ve kterých se jednotlivé funkce liší, jsou pro útočníka zanedbatelné. Pokud dojde k chybě, která umožní PHP injection dostává útočník možnost na server nahrát jakýkoliv vlastní skript a tím pádem získává kontrolu nad celou webovou aplikací.

(()Suprer(), 2006)

5.7 Session hijacking

Vzhledem k tomu, že protokol http je bezstavový je potřeba pro potřeby aplikace – např. kvůli přihlášení uživatele přenášet, respektive uchovávat informace o stavu aplikace. Tento problém je řešen unikátním identifikátorem, takzvaným SESSION_ID. Server si SESSION_ID po přihlášení přiřadí k danému uživateli a kdokoliv tento unikátní řetězec na server odešle, je považován za daného uživatele. Tento identifikátor může být přenášen 2 různými způsoby.

Prvním z nich je přenášení, jako parametr URL kdy je zasílán serverem metodou GET. Tento způsob je velmi nebezpečný a málo používaný.

Druhou variantou je používání cookies. Při použití cookies je SESSION_ID přenášeno v obou směrech v hlavičce http požadavku – respektive odpovědi. Tento způsob je proti krádeži SESSION_ID o mnoho odolnější, ale možnost k útoku se objevit může. Ideálním zabezpečením proti krádeži SESSION_ID je použití kombinace protokolů http a ssl, který se nazývá zkráceně https místo běžnějšího protokolu http. Obměnou krádeže SESSION_ID může být podstrčení pro útočníka známého SESSION_ID, tento identifikátor je po přihlášení uživatele využito útočníkem. Obrana proti této situaci je velmi jednoduchá a spočívá ve vygenerování nového ID těsně před přihlášením uživatele do aplikace.

Poslední varianta získání SESSION_ID útočníkem spočívá v použití http hlavičky referer. Tento způsob bývá zneužit například u diskusních fór, kdy utočník do diskuse vloží odkaz na svůj web s dostatečně zajímavým obsahem. Ve chvíli, kdy uživatel tuto webovou stránku navštíví, tak dostává díky http hlavičce referer možnost přístupu k URL předchozí stránky včetně předchozího SESSION_ID.

(Ferschmann, 2008)

5.8 Zabezpečení aplikace praktické části

Vzhledem k tomu, že aplikace je psána ve skriptovacím jazyce PHP s využitím několika HTML formulářů, tak může nastat několik situací, kdy by potencionální útočník mohl mít možnost získat přístup do databáze či databázová data pozměnit nebo smazat.

5.8.1 Ošetření vstupů

V navržené aplikaci ochrana proti napadení pomocí SQL injection, XSS nebo CSRF spočívá v odstranění nežádoucích znaků tzv. escapováním. Je k tomu použita funkce jazyka PHP mysql_real_escape_string() kde parametrem jsou data z formuláře, který je zpracováván. Pro zlepšení zabezpečení, by bylo vhodné pro účet, pod kterým je připojováno k databázi vytvořit speciální účet s omezenými oprávněními.

(Grudl, 2009)

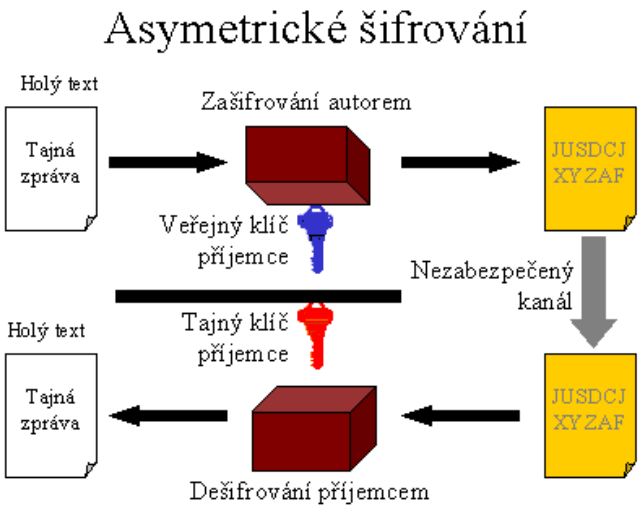
5.8.2 Ošetření proti ukradení session

Ošetření proti ukradení session není v aplikaci nijak implementováno. Nejideálnějším řešením by byl přístup přes protokol https. Https není ve skutečnosti pravý protokol, jedná se o kombinaci dvou protokolů a to protokolu SSL, případně TSL, který je v referenčním modelu vložen mezi transportní a aplikační vrstvu a protokolu http. Protokol SSL poskytuje všem protokolům aplikační vrstvy službu šifrování dat. Pro šifrovaný přenos informací se tedy vůbec nemusí měnit princip aplikačních protokolů a o vše se postará vrstva SSL. Vrstva ssl svojí funkcí řeší šifrování a integritu dat. Vrstva SSL umí sice data zašifrovat a předat je protokolu TCP pro další přenos, neumí je ovšem digitálně podepsat. Podepsání se musí řešit na úrovni aplikační vrstvy.

V tuto chvíli se jistě jeví ještě stále možnost útoku typu „man-in-the-middle“ kdy by mohl útočník podvrhnout svůj web uživateli tak, že by uživatel vyslal požadavek na

server (pro lepší představu budeme uvažovat, že se jedná o server s naší webovou aplikací) o kterém se domnívá, že je pravý. Zde útočník odchytí jeho požadavek, dešifruje ho, analyzuje, zašifruje a odešle na náš pravý server. Náš server tento požadavek přijme, dešifruje a v domnění že požadavek vzešel od pravého klienta, proběhne autorizace. Oznámení o autorizaci server zašifruje a odešle odesílateli (útočník) ten zprávu bez problému rozšifruje, analyzuje, znovu zašifruje a odešle uživateli. V tuto chvíli uživatel nepozná, že jeho komunikace neprobíhá s naším serverem.

Řešením této situace je použití certifikátů. SSL je založené na asymetrickém šifrování. Princip spočívá v existenci 2 různých klíčů. Jeden z klíčů je nazýván veřejným. Uživatel přihlašující se na daný server šifruje data, která odesílá právě tímto klíčem. Server je přijme a pokusí se je rozšifrovat svým soukromým klíčem, pokud data půjdou rozšifrovat, je komunikace v pořádku.



Obrázek 7 Princip asymetrického šifrování
(převzato z http://www.cs.vsb.cz/turecek/vyuka/via/data/diplomky/kacmarcik/index.html)

Bohužel i za této situace se může stát, že se do komunikace vloží třetí strana – to za předpokladu, že si vytvoří vlastní dvojici soukromého a veřejného klíče. Řešení této situace je velmi jednoduché, jelikož každý certifikát vydaný certifikační autoritou je označen stejným principem asymetrické šifry, jako jsem popisoval výše. Každý webový prohlížeč, respektive každý operační systém obsahuje seznam certifikačních autorit a jejich klíčů, pomocí kterých odkazuje na tzv. kořenový certifikát, kterým jsou podepsány všechny certifikáty podepsané danou certifikační autoritou.

V České Republice je certifikační autoritou například Česká Pošta. Vlastník webového serveru si může certifikát vytvořit i sám, v tomto případě bude ale uživatel pokaždé na tuto skutečnost upozorněn. Za vydání certifikátu certifikační autoritou je účtován poplatek.

(Wartha, 2011)

6 Závěr

V teoretické části jsem se zabýval teorií o knihovních systémech, o jejich funkcích a principu. To, že většina knihovních systémů je založena na podobném principu rozdělení funkcí je ukázáno na charakteristice třech nejvíce používaných systémů.

Dále byly představeny moderní technologie tvorby webových aplikací. Nedílnou součástí tvorby webových aplikací je také návrh databáze pro daný systém a návrh zabezpečení aplikace. Tato problematika je v textové části řešena konkrétně pro systém realizovaný v praktické části.

Praktická část byla zaměřena na tvorbu knihovního systému a webovou aplikaci, která systém obsluhuje. Vytvořená aplikace splňuje základní požadavky pro systém tohoto typu. Umožňuje přidávání čtenářů do systému, vytvářet rezervace dokumentů, práci s výpůjčkami. Celý systém je více popsán v textu výše.

K plnohodnotnému využití podle všech požadavků na knihovní systém, by byl ještě potřeba další vývoj aplikace. Vzhledem k využití objektově orientovaného programování bude práce s rozšířením jednodušší. Aplikace neplánuji momentálně dále vyvíjet, pokud by ale byl zájem o její praktické využití, rád bych se k vývoji vrátil.

7 Literatura

Business Control Pty Ltd trading as Simply. Web + Network Design and Development. 1998-2011. Glossary :: Simply - Turning your Online Strategy into Online Action. *Simply - Turning your Online Strategy into Online Action.* [Online] 1998-2011. [Citace: 1. Srpen 2011.] http://www.simply.com.au/glossary.php?alphabet=W.

()Suprer(). 2006. SOOM.cz - PHP Injection. [Online] 11. Červen 2006. [Citace: 2. Srpen 2011.] http://www.soom.cz/index.php?name=usertexts/show&aid=284.

Adaptic, s.r.o. - Programování aplikací podle vašich potřeb. 2011. Adaptic - Co je XML. *Adaptic - internetová řešení podle vašich potřeb.* [Online] 2011. [Citace: 27. Červenec 2011.] http://www.adaptic.cz/znalosti/slovnicek/xml/.

Adaptic, s.r.o. 2011. Adaptic - Programování aplikací. *Adaptic - internetová řešení podle vašich potřeb.* [Online] 2011. [Citace: 12. Srpen 2011.] http://www.adaptic.cz/weby/programovani-aplikaci/.

airdump. WEB Hacking - Útok na web | airdump.cz. *airdump.cz | security wifi hacking cracking exploity.* [Online] [Citace: 1. Srpen 2011.] http://airdump.cz/web-hacking-utok-na-web/.

ČVUT, Fakulta kybernetiky. 2010. *Učební texty k přednáškám X33PTE.* [Učební text] Praha : ČVUT, 2010.

Ferschmann, Petr. 2008. Bezpečnost na webu - přehled útoků na webové aplikace - Zdroják. *Zdroják - tvorba webových stránek a aplikací.* [Online] 10. Listopad 2008. [Citace: 31. Červenec 2011.] http://zdrojak.root.cz/clanky/prehled-utoku-na-webove-aplikace/.

Grudl, David. 2009. Escapování - definitivní příručka » phpFashion. *phpFashion.* [Online] 19. Květen 2009. [Citace: 3. Srpen 2011.] http://phpfashion.com/escapovani-definitivni-prirucka.

Harbridge, Richard. 2010. SharePoint Glossary, Terminology and Acronyms. *Richard Harbridge: Insights.* [Online] 2010. [Citace: 1. Srpen 2011.] http://www.rharbridge.com/?page_id=60.

Janovský, Dušan. 2011. Programování HTML stránek. *Jak psát web, návod na html stránky.* [Online] 7. Červen 2011. [Citace: 31. Červenec 2011.] http://www.jakpsatweb.cz/programovani.html.

Kačmařík, Vojtěch. Výuková podpora předmětu Internetové technologie - Kačmařík Vojtěch. *VŠB | Katedra informatiky FEI VŠB-TUO.* [Online] [Citace: 4. Srpen 2011.] http://www.cs.vsb.cz/turecek/vyuka/via/data/diplomky/kacmarcik/index.html.

KP-Sys spol. s.r.o. 2010. KP-SYS - KpWin SQL - Vlastnosti programu KpWin SQL. [Online] KP-Sys spol. s r.o., 26. Květen 2010. [Citace: 20. Červenec 2011.] http://www.kpsys.cz/kpwinsql/features.html.

Kučera, František. 2010. Java na webovém serveru: první web - Zdroják. *Zdroják - tvorba webových stránek a aplikací.* [Online] 15. Leden 2010. [Citace: 1. Srpen 2011.] http://zdrojak.root.cz/clanky/java-na-webovem-serveru-prvni-web/.

LANius s.r.o. 2011. Aktuality firmy LANius. *Knihovní systémy Clavius a LANius.* [Online] LANius s.r.o., 15. Červen 2011. [Citace: 30. Červenec 2011.] http://www.lanius.cz/.

Makulová, Soňa. 1993. *Automatizácia knižníc: Problémy, východiska, postupy.* Bratislava : STIMUL-Centrum informatických a vzdelávacích služieb, 1993. str. 298. ISBN 80-85977-09-2.

Malý J., Kacálek J. 2007. Zabezpečení webových aplikací I. - klientské skriptovací jazyky. *Access server.* [Online] 15. Srpen 2007. [Citace: 2. Srpen 2011.] http://access.feld.cvut.cz/view.php?cisloclanku=2007090001. ISSN 1214-9675.

Mgr. Osuchowski, Marek. 2009. Výukové materiály - Mgr. Marek Osuchowski. *Výukové materiály - Mgr. Marek Osuchowski.* [Online] 2009. [Citace: 28. Červenec 2011.] http://www.osuchowski.cz/wwwstranky/porovnani.php.

MIKA, Jiří. 2000. Vliv zavádění automatizovaných knihovnických systémů na organizaci a provoz knihovny. *Knihovnická revue.* [Online] 2000. [Citace: 3. Květen 2011.] http://full.nkp.cz/nkkr/Nkkr0001/0001006.html. ISSN 1214-0678.

Minařík, Karel. 2007. Ruby on Rails a revoluce ve vývoji pro web. Část první: Ruby | Karmi is on Rails . *Karmi is on Rails je weblog o Ruby On Rails a web designu, který píše Karel Minařík.* [Online] 17. Červen 2007. [Citace: 30. Červenec 2011.] http://blog.karmi.cz/2007/6/16/co-je-ruby-on-rails-cast-1.html.

Nazghul. Protokol HTTPS - Warez.cz. *Warez.cz - O warezu bez warezu.* [Online] [Citace: 3. Srpen 2011.] http://www.warez.cz/clanky/protokol-https/.

Pomazal, Jiří. 2010. Hrozby pro bezpečnost webových aplikací a serverů. *Ekonomické a informační systémy v praxi.* [Online] Srpen 2010. [Citace: 1. Srpen 2011.] http://www.systemonline.cz/it-security/hrozby-pro-bezpecnost-webovych-aplikaci-a-serveru.htm.

Sládek, Jan. 2010. Webdesignérův průvodce po HTML5 - díl nultý - Zdroják. *Zdroják - tvorba webových stránek a aplikací.* [Online] 25. Květen 2010. [Citace: 2. Srpen 2011.] http://zdrojak.root.cz/clanky/webdesigneruv-pruvodce-po-html5-dil-nulty/.

Snížek, Martin. 2005. AJAX – kde jsou hranice? | snizekweb.cz. *snizekweb.cz | Martin Snížek píše o webu.* [Online] 13. Září 2005. [Citace: 2. Srpen 2011.] http://www.snizekweb.cz/clanky/ajax-kde-jsou-hranice/. ISSN 1802-2103.

—. 2002. XHTML - klientské skripty. *Interval.cz.* [Online] 20. Listopad 2002. [Citace: 28. Červenec 2011.] http://interval.cz/clanky/xhtml-klientske-skripty/.

Stöcklová, Anna. 2006. Automatizace v knihovnách České republiky | Ikaros. *Ikaros.* [Online] 2006. [Citace: 25. Červenec 2011.] http://www.ikaros.cz/automatizace-v-knihovnach-ceske-republiky. ISSN 1212-5075.

Stoyan. [Stoyan's Page] - Web Hacking, PHP Injection. *[Stoyan's Page] - IT security a Programování v C/C++, Delphi, Pascal, PHP skripty, Webdesign a Články nejen z oboru IT.* [Online] [Citace: 31. Červenec 2011.] http://stoyan.cz/hacking-php-injection/.

Sun Microsystems, Inc. 2002. Designing Enterprise Applications with the J2EE Platform, Second Edition . [Online] 2002. [Citace: 1. Srpen 2011.] http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/glossary.html.

SVOP, s.r.o. Bratislava. 2009. Dawinci - Úvodná stránka. *Dawinci.* [Online] SVOP, s.r.o., 2009. [Citace: 28. Červenec 2011.] http://www.dawinci.sk/.

The PHP Group. 2011. PHP: History of PHP and Related Projects - Manual. *PHP: Hypertext Preprocessor.* [Online] 2011. [Citace: 2. Srpen 2011.] http://php.net/history.

Vochozka, Josef. 2000. Značkovací jazyky a XML. *Zpravodaj ÚVT MU.* [Online] 2000. [Citace: 30. Červenec 2011.] http://www.ics.muni.cz/bulletin/articles/201.html. ISSN 1212-0901.

Vrána, Jakub. 2005. AJAX - Root.cz. *>Root.cz - informace nejen ze světa Linuxu.* [Online] 25. Březen 2005. [Citace: 1. Srpen 2011.] http://www.root.cz/clanky/ajax/.

—. 2005. PHP triky - Obrana proti SQL Injection. *PHP triky - Weblog o elegantním programování v PHP pro mírně pokročilé.* [Online] 2. Březen 2005. [Citace: 1. Srpen 2011.] http://php.vrana.cz/obrana-proti-sql-injection.php.

W3C. 2010. W3C XHTML2 Working Group Home Page. *W3C Interaction Domain.* [Online] 17. Prosinec 2010. [Citace: 27. Červenec 2011.] http://www.w3.org/MarkUp/.

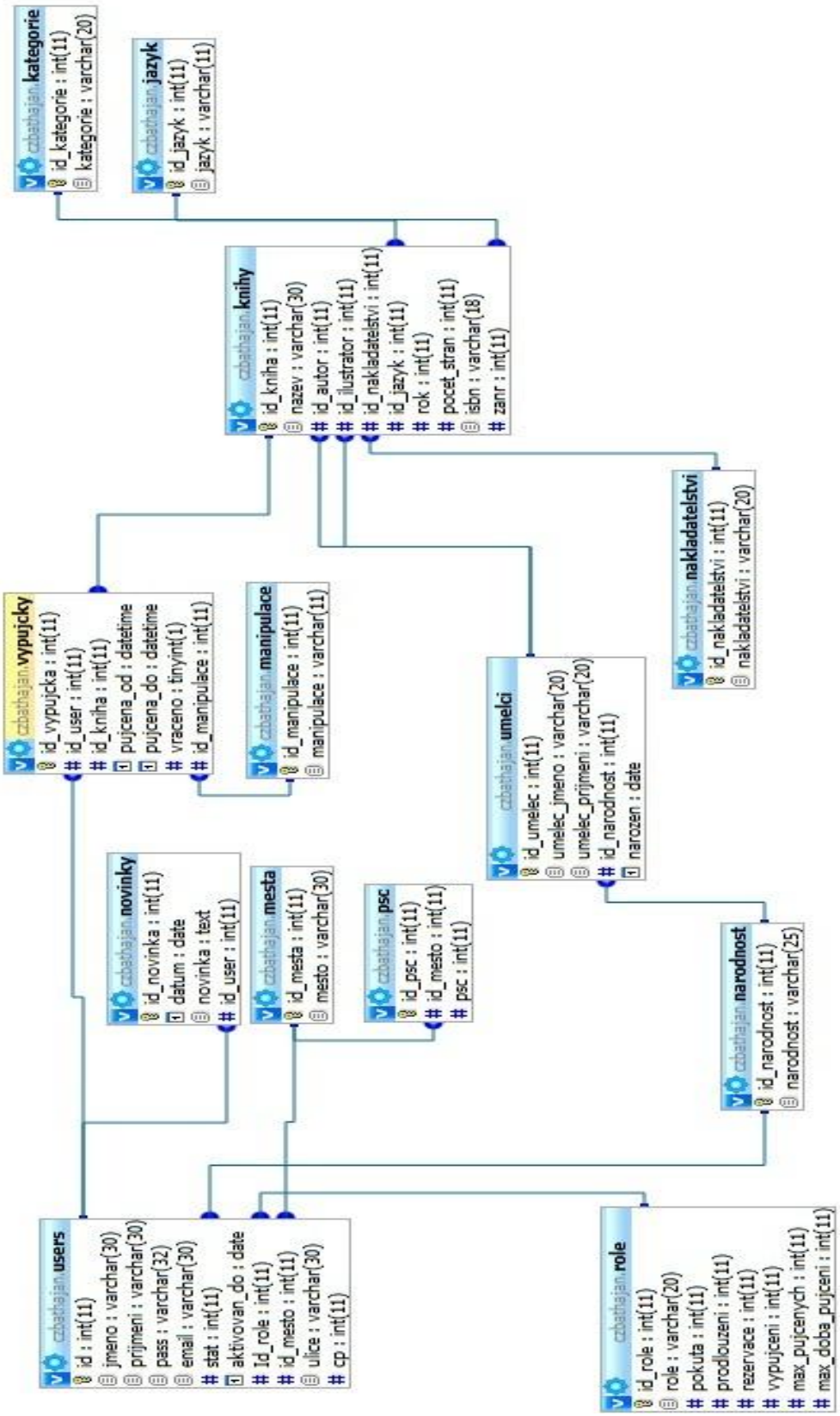
Wartha, Ladislav. 2011. Když HTTPS protokol není bezpečný. *Malé novinky o velkých věcech.* [Online] 28. Březen 2011. [Citace: 3. Srpen 2011.] http://aktuality.firstnet.cz/news/Kdyz-HTTPS-protokol-neni-bezpecny/.

Zralý, Jiří. 2005. AJAX - návod pro začátečníky » digitální Citron. *digitální Citron - Medhiho stránky a galerie.* [Online] 6. Listopad 2005. [Citace: 2. Srpen 2011.] http://citron.blueboard.cz/clanek-239-ajax-navod-pro-zacatecniky.html.

Žďárek, Roman. 2011. Úvod do PHP | PHP. *Tvorba Webu, Tvorba WWW stránek - Návod Zdarma.* [Online] 2011. [Citace: 2. Srpen 2011.] http://www.jakdelatweby.cz/php/uvod-do-php.php.

Seznam příloh

Příloha A – ER diagram



Obrázek 8 ER diagram

Příloha B – Obsah přiloženého CD

CD obsahuje kompletní knihovní systém včetně programátorské a uživatelské příručky. K dispozici jsou i testovací data do databáze.

Aplikační část :

- Soubor bp_bathaJan.rar obsahující zdrojové kódy aplikace

Databázová část:

- Soubor czBathaJan.sql obsahující DDL skript pro vytvoření potřebné databáze
- Soubor bp_bathaJan.sql obsahující DDL skript pro import testovacích dat do Mysql databáze

Programátorský manuál:

- Instalační manuál, UML case diagram, UML activity diagram

Uživatelský manuál:

- Popis obsluhy systému z pohledu uživatele

Obrázky:

- Několik ukázek vzhledu aplikace