

Informační systém knihovny

Library Information System

Bc. David Adámek

Diplomová práce
2012



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2011/2012

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. David ADÁMEK**
Osobní číslo: **A09466**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Informační systém knihovny**

Zásady pro vypracování:

1. Analyzujte současné webové stránky.
2. Stanovte základní pravidla pro optimalizaci webu.
3. Navrhněte a vytvořte novou webovou aplikaci s dodržением pravidel pro SEO.
4. Praktikuje optimalizaci a implementujte novou aplikaci do provozu.
5. Provedte testování a vyhodnocení nových webových stránek.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. RIORDAN, R. M. Vytváříme relační databázové aplikace. Praha : Computer Press, 2000. 280 s. ISBN 80-7226-360-9.
2. LECKY-THOMPSON, Ed, Steven D NOWICKI a Tom MYER. Professional PHP 6. Indianapolis, IN: Wiley Pub., 2009, 744 s. ISBN 978-047-0395-097.
3. KUBÍČEK, M. Velký průvodce SEO. Jak dosáhnout nejlepších pozic ve vyhledávačích. Computer Press, Brno, 2008, ISBN: 978-80-251-2195-5.
4. KUBÍČEK, M. 333 tipů a triků pro SEO. Computer Press, Brno, 2008, ISBN: 978-80-251-1833-7.
5. SMIČKA, R. Optimalizace pro vyhledávače ? SEO. Jasmínka, Dubany.
6. CLIFTON, B. Google Analytics ? Podrobný průvodce webovými statistikami. Computer Press, Brno, 2009, ISBN: 978-80-251-2231-0.

Vedoucí diplomové práce:

doc. Ing. Zdenka Prokopová, CSc.

Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce:

24. února 2012

Termín odevzdání diplomové práce:

21. května 2012

Ve Zlíně dne 24. února 2012

prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

ABSTRAKT

Informační systémy knihovny by měly být stejně jako ostatní aplikace neustále vyvíjeny. Tato práce pojednává o vývoji informačního systému knihovny v malé obci, která se rozhodla přejít z „papírové“ formy evidence záznamů na elektronický systém.

Teoretická část popisuje použité webové technologie, jeden konkrétní standard, kterým se v knihovnických systémech realizuje výměna knih, princip fungování ISBN a závěrečná část je zaměřená na metodiku používanou pro optimalizaci webových stránek pro vyhledávače tzv. SEO.

Praktická část se zabývá problematikou implementace formátu MARC 21, optimalizací stránek pomocí metodiky SEO a validací HTML kódu validátorem organizace W3C. Tato část také popisuje novou funkcionalitu knihovny a nasazení systému do testovacího režimu.

Klíčová slova: knihovna, MARC 21, PHP, HTML, MySQL, relační databáze, SEO, XML

ABSTRACT

A Library Information Systems should be constantly developed like the other applications. This thesis is aimed to describe a development of a Library Information System of a small village library, which has decided to leave the paper records system and start using electronic one instead.

The theoretical part describes web technologies, one specific standard, which is used among Library Systems for bibliographical data transfer, principle of ISBN and the final part is about methodology which is used to optimize web sites for search engines.

The Practical part is focused to implementation of MARC 21 format, methodology of SEO web page optimization and HTML code validation using W3C tool. This part also includes new Library functionality and deployment system to the testing mode.

Keywords: Library, MARC 21, PHP, HTML, MySQL, relation database, SEO, XML

Na tomto místě bych chtěl poděkovat vedoucí mé diplomové práce doc. Ing. Zdence Prokopové, CSc., za odbornou pomoc, cenné rady a čas, který mi věnovala. Také bych chtěl poděkovat své ženě, která mě po celou dobu podporovala.

Prohlašuji, že

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

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 PHP	11
1.1 OŠETŘOVÁNÍ VÝJIMEK	11
1.1.1 Úrovně oznamování chyb	11
1.2 JAZYK XML.....	12
1.3 XML v PHP.....	12
1.3.1 Historie.....	12
1.3.2 SimpleXML.....	13
1.3.3 DOM	13
1.4 JMENNÉ PROSTORY	14
1.5 SCHÉMATA	14
2 SQL	16
2.1 RELAČNÍ MODEL.....	16
2.2 DATOVÝ MODEL.....	16
2.2.1 Entity	16
2.2.2 Atributy	17
2.2.3 Domény	17
2.2.4 Vztahy	17
2.3 RELAČNÍ OPERÁTORY	18
2.3.1 Restrikce.....	18
2.3.2 Projekce.....	18
2.3.3 Spojení.....	19
2.3.4 Dělení	19
3 MARC	20
3.1 ROZSAH FORMÁTU	21
3.2 PRVKY BIBLIOGRAFICKÉHO ZÁZNAMU	21
3.2.1 Typy označení proměnných polí.....	23
3.3 MINIMÁLNÍ ZÁZNAM	24
4 ISBN	26
4.1 KONTROLNÍ ČÍSLICE ISBN-13.....	26
5 SEO	28
II PRAKTICKÁ ČÁST	29
6 DATABÁZE	30
6.1 TABULKY	31
7 MARC 21	32
7.1 MARC 21 XML.....	32
7.2 IMPORT SOUBORU XML	32
7.3 POUŽITÉ ADRESÁŘE.....	34
7.3.1 Automatické vytvoření záznamu v číselníku	35
7.3.2 Relace mezi záznamem knihy a MARC 21	36

8	SEO	37
8.1	ANALÝZA.....	37
8.2	NÁPRAVA.....	37
9	NOVÉ FUNKCE KNIHOVNY	39
9.1	VÝPŮJČKY.....	39
9.2	KNIHY.....	39
9.3	HTML5.....	41
10	IMPLEMENTACE	43
	ZÁVĚR	45
	SUMMARY	46
	SEZNAM OBRÁZKŮ	49
	SEZNAM TABULEK	50
	SEZNAM PŘÍLOH	51

ÚVOD

Zájem o čtení knih neustále upadá. V poslední době se tento trend snaží změnit elektronické čtečky knih, ale protože knihovny stále existují, je jednou z možností jak podpořit jejich vyšší návštěvnost, zpřístupnit katalog knih v elektronické podobě na internetu.

Tento cíl mé bakalářské práce se nakonec nepodařilo naplnit, protože v knihovně byl z technických důvodů zrušen počítač, na kterém měl být systém provozován. Od té doby se situace změnila, počítač již v knihovně funguje a to byl také důvod pro obnovení původní myšlenky v této diplomové práci.

Systém měl několik nedokonalostí, jejichž řešení se stalo náplní této práce. Hlavní požadavek na rozšíření byla jednoznačně implementace formátu MARC 21, který slouží knihovnám na celém světě pro výměnu knih ve výměnném fondu, na jehož základě vypůjčují větší knihovny knihy menším knihovnám. V případě knihovny v Horním Břečkově bude výpůjčka knih probíhat z okresní knihovny ve Znojmě.

Práce se zabývá výběrem vhodného formátu pro import souborů výměnného fondu, popisuje řešení problémů, které vznikly při vývoji, a vysvětluje, proč bylo řešení standardu MARC 21 nasazeno v omezeném rozsahu vzhledem ke specifikaci.

Dalším výrazným nedostatkem byla chybějící kontrola ISBN. Tento údaj se používá jako hlavní identifikátor knihy a proto je nezbytné ověřovat jeho pravost při ručním vkládání.

SEO, česky se překládá jako optimalizace pro vyhledávače, je velmi často skloňovaným termínem v dnešní době, kdy už mají weby jistou úroveň kvality a jejich správci hledají další způsoby, jak web více zviditelnit v silné konkurenci. SEO znamená především dodržování existujících pravidel při tvorbě webu, na které ovšem vývojáři často zapomínají. Optimalizované stránky vyhledávače preferují ve vyhledávání konkrétního klíčového slova a přední pozice ve vyhledávači jsou cílem každého vlastníka webové aplikace.

I. TEORETICKÁ ČÁST

1 PHP

Autoři jazyka PHP si jazykovou syntaxi vypůjčili částečně od jiných jazyků, jako jsou C, Perl, Java nebo příkazové prostředí. PHP je skutečně hybridním jazykem, jenž od jiných jazyků přebírá jen ty nejlepší vlastnosti a vytváří tak snadno použitelný a především velmi výkonný skriptovací jazyk. [1]

1.1 Ošetřování výjimek

Počet chyb v aplikaci můžete snížit třeba dodržováním správných programátorských postupů. Mnoho chyb však může vzniknout bez našeho přičinění. Výpadky sítě, plný pevný disk, selhání hardwaru, chyby v jiných komponentách jazyka PHP nebo v programech s nimiž váš skript komunikuje – to vše může způsobit chyby ve vaší aplikaci.

Pokud nemáte pro ošetření chyby připraven žádný vlastní mechanismus, je jádro PHP nastaveno implicitně tak, že upozorní na chybu uživatele. Součástí zobrazené zprávy je rovněž odkaz na stránku manuálu popisující funkci, která chybu způsobila.

Chybové zprávy jsou však v podstatě určeny vám, vývojářům, nikoliv uživatelům vašich aplikací. Uživatelé ocení spíše, bude-li jim problém objasněn tak, aby mu rozuměli, tedy bez odborných termínů. Uživatele nezajímají podrobnosti z dokumentace programovacího jazyka nebo umístění řádku ve zdrojovém kódu.

1.1.1 Úrovně oznamování chyb

Pro všechny integrované funkce jazyka PHP se používá společný mechanismus ošetření chyb. Tento mechanismus implicitně vypíše chybovou zprávu, k níž připojí název zdrojového souboru a číslo problémového řádku.

Chyby způsobené v jádře PHP jsou kategorizovány podle závažnosti. Nejméně závažná jsou oznámení, naopak nejzávažnější jsou kritické chyby.

E_ERROR

E_WARNING

E_PARSE

E_STRICT

E_NOTICE

1.2 Jazyk XML

Extensible Markup Language (XML), česky *rozšiřitelný značkovací jazyk*, je jednoduchý, velmi přizpůsobivý textový formát, který je odvozen od SGML (ISO 8879). Původně byl navrhnout, aby uspokojil potřeby rozsáhlého publikování v elektronické podobě. XML také hraje stále důležitější roli při výměně širokého spektra informací na internetu i jinde. [2]

XML dnes zcela dominuje na poli publikování metainformací. Jedná se například o formáty pro publikování přehledů nových článků, jako jsou RSS či Atom. Protože začleňování sémantiky ve strojově čitelné podobě přímo do webových stránek je stále v plenkách v začátcích mnoho vyhledávačů nabízí vlastní formáty, ve kterých jim můžete předávat informace vylepšující vyhledávání – například Google Sitemap nebo Google Base.

Další využití XML je pro komunikaci a předávání dat. XML se využívá jednak pro výměnu dat mezi backendy jednotlivých aplikací a dále pak v AJAXových aplikacích pro zasílání aktualizací dat do prohlížeče. „Enterprise“ aplikace pak pro samotnou komunikaci nevyužívají prosté XML, ale komplexnější mechanismus webových služeb.

Diverzita koncových zařízení používaných pro přístup k webu se také stále zvyšuje. Webové stránky se už neprohližejí jen z klasického počítače, ale i z chytrých telefonů nebo různých PDA. Mnohé aplikace potřebují uspokojivě řešit možnost kvalitního tisku. Už nepostačí vytvořit aplikaci, která výstupy generuje jen jako HTML. Pro jednotlivá koncová zařízení je potřeba generovat odlišné formáty výstupu a webovou aplikaci je potřeba obohatit o flexibilní prezentační vrstvu. Pro vytvoření takové vrstvy lze využít i jazyk XML a stylové technologie jako XSL. [3]

1.3 XML v PHP

1.3.1 Historie

Podpora XML v PHP 4 byla implementována pomocí různých externích knihoven. Podpora rozhraní SAX byla implementována pomocí staré knihovny Expat, transformační jazyk XLST byl podporován díky knihovně Sablotron (nebo díky knihovně `libxml2`, jež je rozšířením modelu DOM). Objektový model DOM byl implementován pomocí výkonnější knihovny `libxml2`, která je součástí projektu GNOME.

Mnoho různých knihoven práci s dokumenty XML nezlepšilo. Údržba kódu byla velmi obtížná, knihovny často nepodporovaly různé standardy XML, výkon nebyl tak dobrý, jak bychom očekávali, a interoperabilita mezi doplňky pro manipulaci s dokumenty XML prakticky neexistovala.

V PHP5 byly všechny doplňky (extensions) přepsány, aby využívaly vynikající sadu nástrojů pro práci s dokumenty XML `libxml2`. Jde o implementaci standardů XML bohatou na funkce, o implementaci, jejíž správě je věnována velká pozornost. Díky ní se může novou technologií XML pyšnit také jazyk PHP. [1]

1.3.2 SimpleXML

Dokumenty XML mají hierarchickou strukturu tvořenou vnořením jednotlivých elementů. Hierarchické struktury lze v počítači reprezentovat mnoha způsoby, v poslední době je populární modelování pomocí objektů. Knihovna SimpleXML využívá právě tento způsob. Dokument XML načte celý do paměti struktury objektů, jejichž jména odpovídají názvům elementů zpracovávaného dokumentu. Díky tomu se pak velmi jednoduše přistupuje k jednotlivým informacím.

Pro vytvoření struktury objektů z dokumentu XML slouží funkce `simplexml_load_file()`, která jako parametr očekává jméno souboru ke zpracování. Výsledkem je objekt, který zastupuje celý dokument XML.

```
1. $xml = simplexml_load_file("dokument.xml");
```

Podelementy jsou přitom dostupné jako členské proměnné. K elementu *element* se tak dostaneme zápisem:

```
1. $xml->element
```

1.3.3 DOM

Rozhraní DOM je standardní rozhraní pro práci s dokumenty XML definované konsorciem W3C. Rozhraní definuje způsob, jakým se dokument XML mapuje na hierarchii objektů v paměti. Každé části dokumentu, jako je element, atribut, textový uzel a podobně, odpovídá v paměti jeden objekt. Model dokumentu vytvořený pomocí DOM je podobný stromu dokumentu složenému z jednotlivých uzlů. Pomocí metod a vlastností dostupných na každém objektu můžeme zjišťovat druh uzlu, jaký ve stromu dokumentu XML zastupují, jejich název, obsah, seznam objektů reprezentujících dětské uzly, objekt zastupující rodiče uzlu atd.

Dokument je v paměti reprezentován jako objekt, který je instancí třídy `DOMDocument`. Pomocí metody `load()` je možné vytvořit DOM reprezentaci dokumentu XML uloženého v souboru:

1. `$doc = new DomDocument();`
2. `$doc->load("dokument.xml")`

1.4 Jmenné prostory

V některých případech může být velice užitečné v jednom dokumentu používat několik sad značek definovaných v různých DTD. Můžeme v XML vytvářet katalog knih podle nějakého DTD určeného pro tvorbu katalogů. Abychom v tomto katalogu mohly snadno vyhledávat služby zaměřené na prohledávání bibliografických záznamů, můžeme názvy knih, autorů, ISBN a další důležité údaje označit pomocí nějakého bibliografického DTD. Využívání sad již hotových značek má i další výhody. Nemusíme znovu vynalézat kolo, stačí poskládat dohromady již existující věci.

Jmenné prostory (XML namespaces) nabízejí možnost, jak v jednom dokumentu kombinovat více sad značek. Pod pojmem sada značek přitom máme na mysli elementy a atributy. Každá sada značek je jednoznačně definována svojí URI adresou. Pokud chceme v nějakém elementu a jeho podelementech používat elementy patřící do určité sady značek, musíme si o to říct pomocí speciálního atributu `xmlns`.

1.5 Schémata

Počáteční nadšení z toho, že XML dokumenty mohou mít libovolnou strukturu a že v nich lze používat libovolné značky, brzy vystřídá vystřízlivění. V praxi je potřeba vyměňovat si XML zprávy s nějakou předem definovanou podobou, a ne jen nahodile pojmenované směsice elementů XML. A to je právě oblast, kde do hry vstupují jazyky pro popis schématu dokumentu XML. Tyto jazyky umožňují definovat, jaké elementy a atributy můžeme v dokumentu XML používat, jak je lze vzájemně kombinovat a co mohou obsahovat. Schéma dokumentu XML vlastně definuje nový značkovací jazyk, který má syntaxi XML, ale používá námi vytvořené značky. [3]

Ukázka ze schématu *MARC21slim.xsd*, který se používá pro validaci bibliografického záznamu:

1. `<xsd:simpleType name="recordTypeType" id="type.st">`
2. `<xsd:restriction base="xsd:NMTOKEN">`

```
3.         <xsd:enumeration value="Bibliographic"/>
4.         <xsd:enumeration value="Authority"/>
5.         <xsd:enumeration value="Holdings"/>
6.         <xsd:enumeration value="Classification"/>
7.         <xsd:enumeration value="Community"/>
8.     </xsd:restriction>
9. </xsd:simpleType>
10. <xsd:complexType name="leaderFieldType" id="leader.ct">
11.     <xsd:annotation>
12.         <xsd:documentation>MARC21 Leader, 24
13.         bytes</xsd:documentation>
14.     </xsd:annotation>
15.     <xsd:simpleContent>
16.         <xsd:extension base="leaderDataType">
17.             <xsd:attribute name="id" type="idDataType"
18.             use="optional"/>
19.         </xsd:extension>
20.     </xsd:simpleContent>
21. </xsd:complexType>
```

2 SQL

Databázová terminologie je skoro stejně ošemetná, jako pojem „objektově orientované programování“. Slovem „databáze“ můžeme totiž označit prakticky cokoliv – od jedné, jednoduché množiny dat, jako je například telefonní seznam, až po složitou množinu různých nástrojů, jako třeba SQL server – mezi těmito dvěma extrémy se pochopitelně nachází řada dalších možností.

2.1 Relační model

Relační model vychází z jistého souboru základních matematických principů odvozených z teorie množin a predikátové logiky. Definiuje způsob, jakým je možné data reprezentovat (tedy strukturu dat), způsoby jejich ochrany (neboli integritu dat) a dále operace, které můžeme nad daty provádět (manipulace s daty).

Obecně se dá říci, že relační databázové systémy vykazují následující charakteristické vlastnosti:

- Veškerá data se pomyslně dají reprezentovat v pravidelně uspořádaných strukturách s řádky a sloupci, kterým se říká *relace*.
- Všechny hodnoty v databázi jsou *skalární*. To znamená, že v každé konkrétní pozici řádku a sloupce dané relace se nachází jedna (a to přesněji právě jedna) hodnota.
- Operace v databázi se provádějí vždy nad celou relací a jejich výsledkem je opět jiná celá relace, tomuto mechanismu se říká *uzávěr*.

2.2 Datový model

Nejabstraktnější částí celého návrhu databáze je *datový model*; to je totiž myšlenkový (pojmový) popis celého prostoru problému. Datové modely se vyjadřují pomocí entit, atributů, domén (oborů hodnot) a vztahů.

2.2.1 Entity

Podat přesnou definici pojmu entita je dosti obtížné, jeho podstata je ale docela intuitivní a přímo se nabízí: *entita* je cokoliv, o čem v systému potřebujeme uchovávat nějaké informace.

Při zahájení prací na návrhu datového modelu není sestavení prvotního seznamu entit nijak obtížné. Jestliže vy sami nebo váš zákazník hovoříte o příslušném prostoru problému,

používáte jako podstatná jména a slovesa většinu vhodných kandidátů na entity. „Zákazníci kupují výrobky. Zaměstnanci prodávají výrobky. Dodavatelé nám prodávají svoje výrobky.“ Z těchto vět vybereme slova „zákazníci“, „výrobky“, „zaměstnanci“ a „dodavatelé“ – to jsou všechno zcela zřejmě entity.

2.2.2 Atributy

Navrhovaný systém bude o každé entitě zaznamenávat a sledovat určité skutečnosti. Těmto skutečnostem (údajům) se říká atributy dané entity. Pokud náš systém obsahuje například entitu Zákazník, budeme nejspíše potřebovat znát jeho jméno a adresu, a možná také jeho obor podnikání.

Určování atributů daného modelu je sémantický proces. To znamená, že se budeme rozhodovat podle významu dat a podle způsobu jejich využití. Podívejme se opět na jeden běžný příklad – tentokrát to bude adresa. Stačí adresu zákazníka vymodelovat jako jedinou entitu (Adresa), nebo jako množinu několika entit (Ulice, ČísloDomu, Město, Okres, PSČ)? Většina návrhářů použije obecný princip, podle něhož se se strukturovanými daty snáze pracuje, takže každý téměř automaticky rozdělí adresu do množiny atributů.

2.2.3 Domény

Definice domény neboli oboru hodnot popisuje *typ* dat, která daný atribut reprezentuje. Přesněji řečeno, obor hodnot tvoří množina všech přípustných platných hodnot, které smí určitý atribut obsahovat.

Obory hodnot se často zaměňují s datovými typy, toto vyjádření je ale nepřesné. Datový typ je totiž fyzický pojem, zatímco obor hodnot je logický. „Číslo“ je tedy datový typ, zatímco slovo „Věk“ představuje již obor hodnot. Uvedeme si ještě jiný příklad: atributy JménoUlice a Příjmení můžeme oba vyjádřit pomocí textového pole neboli řetězcové hodnoty, přestože se ale zcela zřejmě jedná o dva různé *typy* textového pole; oba atributy tak náleží do různých domén, mají jiný obor hodnot.

2.2.4 Vztahy

Kromě atributů jednotlivých entit musíme v datovém modelu určit také vztahy, definované mezi různými entitami. Na myšlenkové (konceptuální) úrovni jsou *vztahy* jednoduše jistými asociacemi mezi entitami. Z tvrzení „Zákazníci nakupují výrobky“ tak vyplývá

existence jistého vztahu mezi entitami Zákazníci a Výrobky. Entity, zapojené do určitého vztahu se nazývají *účastníci*. Počet účastníků označujeme jako *stupeň* vztahu.

Drtivá většina vztahů je binárních, tedy se dvěma účastníky – jako například vztah „Zákazníci nakupují výrobky“. Běžné jsou i ternární vztahy, tedy vztahy se třemi účastníky. Ze dvou binárních vztahů „Zaměstnanci prodávají výrobky“ a „Zákazníci nakupují výrobky“ implicitně vyplývá ternární vztah, vyjádřený větou „Zaměstnanci prodávají výrobky zákazníkům“.

Vztahy mezi libovolnými entitami se dají rozdělit do třech typů:

- Jedna k jedné – jsou poměrně vzácné a nejčastěji představují vztah mezi entitou určitého nadtypu a podtypu.
- Jedna k více – jsou nejobvyklejší. Jedna faktura může obsahovat položky více výrobků.
- Více k více – zákazník nakupuje mnoho výrobků a o každý výrobek se zajímá mnoho zákazníků.

2.3 Relační operátory

2.3.1 Restrikce

Operátor restrikce (omezení množiny záznamů) vrátí jen ty záznamy, které vyhovují stanovenému výběrovému kritériu. V příkazu SELECT jej implementuje klausule WHERE, kterou ukazuje následující příklad:

```
SELECT * FROM Zaměstnanci WHERE Příjmení = "Davolio";
```

Výběrová kritéria, specifikovaná v klausuli WHERE mohou být libovolně složitá. Jednotlivé logické výrazy spojujeme pomocí operátorů AND a OR. Tyto výrazy se posléze vyhodnocují pro každý jednotlivý záznam v původní množině záznamů, a pokud celý výraz vrátí pro daný záznam hodnotu True, stane se tento záznam součástí výsledné množiny záznamů.

2.3.2 Projekce

Zatímco restrikce neboli omezení vyřízne z výsledné množiny záznamů jistý „horizontální“ výřez, projekce neboli promítání představuje vertikální řez – z původní množiny záznamů vrací pouze vybranou podmnožinu polí.

Jazyk SQL vyjadřuje tuto jednoduchou operaci v sekci <SeznamPolí> příkazu SELECT a do výsledků zahrne jen ta pole, která v této sekci uvedeme.

```
SELECT Příjmení, Jméno, Linka
FROM Zaměstnání
ORDER BY Příjmení, Jméno;
```

Klausule ORDER BY provádí uspořádání dat; v tomto případě se tedy výsledky operace setřídí abecedně podle pole Příjmení a následně podle pole Jméno.

2.3.3 Spojení

Operace spojení (v databázi též spojení tabulek, join) jsou zřejmě nejběžnějším případem relačních operací. Pro relační model jsou zajisté přímo životně důležité – pokud by se data v relacích nemohla podle potřeby zpětně sestavit, nemohli bychom je ani dekomponovat. A přesně toto provádí operátor spojení: propojí množiny záznamů na základě porovnávání jednoho nebo více polí.

Operace spojení se v příkazu SELECT implementují pomocí klausule JOIN.

2.3.4 Dělení

Poslední relační operátor je dělení. Operátor *relačního dělení* (pro rozlišení od běžného matematického dělení) vrátí všechny záznamy z jedné množiny záznamů, jejich hodnoty se shodují se všemi odpovídajícími hodnotami ve druhé množině záznamů. Máme-li například množinu záznamů, která zobrazuje kategorie výrobků dodávaných jednotlivými dodavateli, bude výsledkem relačního dělení seznam dodavatelů, kteří dodávají výrobky ve všech kategoriích.

Tato situace není nikterak neobvyklá, její řešení ale není triviální, protože příkaz SELECT jazyk a SQL operaci relačního dělení přímo nepodporuje. Stejného výsledku jako s relačním dělením můžeme dosáhnout pomocí řady různých postupů. Nejsnazší způsob spočívá v přeformulování původního požadavku.

Protože dotaz „vypiš všechny dodavatele, kteří dodávají výrobky ve všech kategoriích“, se zpracovává obtížně, zkusíme namísto něj vytvořit jiný dotaz: „Vypiš všechny dodavatele, u nich se počet dodávaných kategorií výrobků rovná počtu všech kategorií výrobků“. [4]

3 MARC

MARC je akronymem (používaným knihovní vědou) pro MACHine-Readable Cataloging, česky „Strojově čitelnou katalogizaci“. Standardy MARC sestávají z MARC formátů, které jsou standardy pro reprezentaci a komunikaci bibliografických a příbuzných informací ve strojově čitelné formě. Definují bibliografický datový formát vyvinutý počátkem šedesátých let dvacátého století Henriette Avramovou v Knihovně Kongresu. Poskytují protokol, kterým počítače vyměňují, zpracovávají a interpretují bibliografické informace. Jeho datové elementy vytvářejí základ pro většinu dnes používaných knihovních katalogů.

Struktura záznamu MARC je implementací standardu ISO 2709 známého také jako ANSI/NISO Z39.2. [5]

Pět komunikačních formátů *MARC 21: MARC 21 Format for Bibliographic Data, MARC 21 Format for Authority Data, MARC 21 Format for Holdings Data, MARC 21 Format for Classification Data a MARC 21 Format for Community Information* představuje pět široce rozšířených formátů pro prezentaci a výměnu záznamů bibliografických, autoritních, klasifikačních, záznamů knihovních jednotek a komunitních informací ve strojově čitelné podobě.

Záznam ve formátu MARC se skládá ze tří základních složek: struktury záznamu, označení polí a podpolí a obsahu záznamu. Záznam představuje implementaci mezinárodní normy *Format for information exchange (ANSI/NISO Z39.2)*. Označení polí a podpolí (tj. kódy a konvence, které jsou zavedeny, aby jednoznačně identifikovaly obsah a umožnily manipulaci s údaji) je dáno jednotlivými formáty typu MARC. Obsah datových údajů je zpravidla definován standardy stojícími mimo tento formát. Příkladem takového standardu je *International Standard Bibliographic Description (ISBD)*, *Anglo-American Cataloguing Rules*, *Library of Congress Subject Headings (LCSH)* a další pravidla katalogizace, předmětové hesláře a klasifikační schémata, používaná různými institucemi, které záznam zpracovávají. Obsah některých kódovaných údajů je definován přímo formátem MARC21 (např. návštějí, pole 007, 008 atd.).

Formát *MARC21 Bibliografický formát* definuje kódy a konvence (např. označení (tag) pole, indikátory, označení podpole a kódované údaje), které identifikují prvky údajů v bibliografických záznamech ve formátu MARC.

3.1 Rozsah formátu

Formát *MARC21 Bibliografický formát* je vytvořen tak, aby se v něm daly ukládat bibliografické údaje o tištěných a rukopisných textových dokumentech, počítačových souborech, mapách, hudebních dokumentech, pokračujících zdrojích, vizuálních a smíšených dokumentech. Do bibliografických údajů jsou obvykle zahrnuty názvy, jména, předměty, poznámky, data vydání a údaje fyzického popisu dokumentu. Bibliografický formát obsahuje údaje, týkající se následujících typů dokumentů:

Druhy bibliografických záznamů

Bibliografické záznamy v MARC se od dalších záznamů typu MARC liší specifickými kódy v Návěští/06 (Typ záznamu), které identifikují následující typy bibliografických záznamů:

Textový dokument	Nehudební zvukový záznam
Rukopisný textový dokument	Hudební zvukový záznam
Počítačový soubor/Elektronický zdroj	Projekční médium
Kartografický dokument	Dvojměrná neprojekční grafika
Rukopisný kartografický dokument	Trojrozměrný předmět nebo přírodní objekt
Hudebnina	Souprava/Soubor
Rukopisná hudebnina	Smíšený dokument

3.2 Prvky bibliografického záznamu

Bibliografický záznam MARC se skládá ze tří hlavních složek: návěští, adresáře a polí. Následující informace stručně popisují strukturu záznamu MARC.

- **Návěští** - Datové údaje, které poskytují informace pro zpracování záznamu. Datové údaje obsahují číslíce či kódované údaje a identifikuje je určitá znaková pozice. Návěští má pevnou délku 24 znakových pozic a je prvním polem záznamu MARC.
- **Adresář** - Série datových řetězců tvořených označením (tagem) pole, délkou pole a počáteční pozicí každého pole v rámci záznamu. Každá skupina je tvořena 12 znakovými pozicemi. Nejdříve se uvádějí označení kontrolních polí v pořadí podle vzestupné hodnoty tagu. Potom následují další pole v pořadí podle vzestupné hodnoty první číslice označení (tagu) pole. Pořadí dat uváděných v adresáři nemusí

nutně odpovídat pořadí polí v záznamu. Duplicitní výskyt polí se rozlišuje pouze počáteční pozicí údajů. Adresář je ukončen znakem ukončení pole (ACSII 1 E hex).

- **Proměnná pole** - Údaje bibliografického formátu jsou uspořádány do proměnných polí, identifikovaných tříznakovým číselným označením (tagem) pole, které je uloženo pro každé pole v adresáři. Každé pole končí znakem ukončení pole. Poslední pole v záznamu končí jak znakem ukončení pole, tak znakem ukončení záznamu (ACSII 1 D hex). Existují dva typy proměnných polí:
 - **Proměnná kontrolní pole** - Pole OOX. Tato pole jsou identifikována v adresáři označením (tagem) pole, ale neobsahují indikátory ani označení podpolí. Struktura proměnných kontrolních polí se liší od struktury ostatních polí. Mohou obsahovat jeden údaj nebo skupinu několika kódovaných údajů definovaných pozicí v rámci pole.
 - **Proměnná pole údajů** - Poslední typ údajů definovaný formátem. Kromě toho, že jsou pole identifikována označením v adresáři, mají tato pole definované i dvě pozice indikátorů na začátku každého pole a dvouznakové označení podpole předcházející datovým prvkům uvedeným v poli.

Proměnná pole údajů jsou uspořádána podle hodnoty prvního znaku označení (tagu) pole do bloků, které, až na několik výjimek, identifikují funkci údaje v rámci záznamu. Typ informace zpřesňují další znaky označení pole.

0XX Kontrolní informace, identifikační čísla, kontrolní pole, klasifikační znaky atd.

1XX Hlavní záhlaví

2XX Názvy a údaje o vydání, nakladatelské údaje atd.

3XX Údaje fyzického popisu atd.

4XX Údaje o edici

5XX Poznámky

6XX Věcné selekční údaje

7XX Vedlejší záhlaví s výjimkou předmětů a edic; propojovací pole

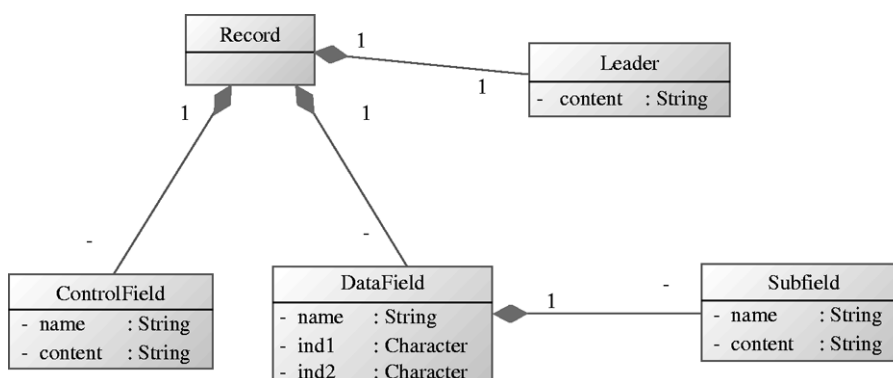
8XX Vedlejší záhlaví pro edici, knihovní jednotky atd.

9XX Pole pro národní využití

3.2.1 Typy označení proměnných polí

Pozice indikátoru - První dvě znakové pozice v proměnném poli, které interpretují nebo zpřesňují informace v poli. Hodnoty indikátoru se interpretují nezávisle na sobě, jejich význam není dán oběma indikátory společně. Hodnotou indikátoru mohou být malé abecední znaky či číslice. Mezera (ASCII SPACE), označovaná i jako prázdná pozice (#), se použije na místě nedefinovaných indikátorů. Na místě definované pozice může mít mezera určitý význam či znamená, že informace není k dispozici.

Označení podpole - Dva znaky, které odlišují v rámci pole údaj, který vyžaduje samostatnou manipulaci. Označení podpole se skládá z delimiteru (ASCII 1 F hex), který v tomto dokumentu nahrazuje znak \$, následovaného identifikátorem podpole. Identifikátorem může být malé abecední písmeno nebo číslice. Identifikátory podpole jsou definovány samostatně pro každé pole, nicméně kde je to možné, existují paralely (např. pole 100, 400 a 600 pro zápis osobních jmen). Kódy podpolí jsou definovány pro účely identifikace, nikoli uspořádání údajů. Pořadí podpolí je zpravidla dáno standardy definujícími obsah zápisu, např. katalogizačními pravidly.



Obrázek 1 – Relační schéma tabulek MARC 21

(zdroj: <http://www.emeraldinsight.com/>)

Příklad bibliografického záznamu ve formátu MARC:

```

Leader/00-23  *****nam##22*****#a#4500
001          <control number>
003          <control number identifier>
005          19920331092212.7
007/00-01    ta
008/00-39    820305s1991####nyu#####001#0#eng##
020          ##$a0845348116 :$c$29.95 (£19.50 U.K.)
020          ##$a0845348205 (pbk.)
040          ##$a[organization code]$c[organization code]
050          14$aPN1992.8.S4$bT47 1991
082          04$a791.45/75/0973$219
100          1#$aTerrace, Vincent,$d1948-
245          10$aFifty years of television :$ba guide to series and pilots, 1937-
              1988 /$cVincent Terrace.
246          1#$a50 years of television
260          ##$aNew York :$bCornwall Books,$cc1991.
300          ##$a864 p. ;$c24 cm.
500          ##$aIncludes index.
650          #0$aTelevision pilot programs$zUnited States$vCatalogs.
650          #0$aTelevision serials$zUnited States$vCatalogs.

```

[6]

3.3 Minimální záznam

Rada pro katalogizační politiku schválila rozsah minimálního záznamu pro tištěné monografie. Definuje povinná čísla *proměnných polí* a *proměnných polí údajů*, které musí obsahovat každý záznam MARC. Dokument je zveřejněn na webovém portálu Národní knihovny České republiky.

Výčet povinných polí bez uvedení vnořených polí:

- 001 Kontrolní číslo
- 003 Identifikátor kontrolního čísla
- 005 Datum posledního zpracování
- 008 Údaje pevné délky

- 020 Mezinárodní standardní číslo knihy (ISBN)
- 040 Zdroj katalogizace
- 041 Kód jazyka
- 044 Kód země vydání
- 072 Kód předmětové kategorie
- 080 Mezinárodní desetinné třídění (MDT)
- 100 Hlavní záhlaví – osobní jméno
- 110 Hlavní záhlaví – jméno korporace
- 111 Hlavní záhlaví – jméno akce
- 245 Údaje o názvu
- 250 Údaje o vydání
- 260 Nakladatelské údaje
- 300 Fyzický popis
- 440 Údaje o edici/vedlejší záhlaví – název
- 490 Údaje o edici
- 5XX Poznámky
- 700 Vedlejší záhlaví – osobní jméno
- 710 Vedlejší záhlaví – jméno korporace
- 711 Vedlejší záhlaví – jméno akce
- 910 Údaje pro souborný katalog

4 ISBN

ISBN je celosvětově unikátní identifikátor knihy, který spravuje a vydává Mezinárodní agentura ISBN v Londýně. Systém mezinárodního standardního číslování knih - ISBN (International Standard Book Numbering) vznikl koncem 60. let ve Velké Británii původně jako národní systém. Postupně se rozšířil do více než 190 zemí. V České republice je zaveden od roku 1989.

V České republice je vrcholným orgánem Národní agentura ISBN pracující v Národní knihovně ČR.

Systém používá třináctimístného čísla ISBN k jednoznačné identifikaci knih (neperiodických publikací). Číslo ISBN slouží především knižnímu obchodu, knihovnám (pro jednoznačnou identifikaci titulu v bibliografických databázích, ale i pro potřeby meziknihovnických služeb) i nakladatelům (pro kontrolu vlastní produkce i pro vedení skladu).

K roku 2011 dosáhl počet nakladatelů používajících systém ISBN čísla 5074. [8]

4.1 Kontrolní číslice ISBN-13

Vypočítává se podle modulu 10. Každá z prvních dvanácti číslic se vynásobí střídavě 1 a 3. Kontrolní číslice je rovna 10 minus zbytek rezultující z vydělení součtu součinů prvních 12 číslic deseti s jedinou výjimkou: je-li výsledkem tohoto výpočtu 10, pak je kontrolní číslicí 0.

Například kontrolní číslice pro ISBN 978-80-252-0070 se vypočítává následujícím způsobem:

Krok 1: Výpočet součtu součinů prvních 12 číslic ISBN (viz následující tabulka).

Tabulka 1 - Výpočet kontrolní číslice ISBN-13

	Prefix	Ident.skup	Ident. vydav.	Ident. titulu	Kontrolní číslice	Součet
ISBN	9 7 8	8 0	2 5 2	0 0 7 0	?	
Váha	1 3 1	3 1	3 1 3	1 3 1 3	-	
Výsledek	9 21 8	24 0	6 5 6	0 0 7 0	-	86

Krok 2: Vydělit součet součinů prvních 12 číslic ISBN vypočtený v kroku 1 deseti (10) a získat zbytek.

$$86 : 10 = 8 \text{ zbytek} = 6$$

Krok 3: Odečíst zbytek vypočtený v kroku 2 od 10. Výsledný rozdíl je kontrolní číslice s jedinou výjimkou: je-li zbytek 0, pak je kontrolní číslice 0 (viz výše).

$$10 - 6 = 4$$

Kontrolní číslice = 4

ISBN = 978-80-252-0070-4

Součet vážených součinů prvních 12 číslic plus kontrolní číslice musí být dělitelný 10 beze zbytku, aby bylo číslo ISBN platné. [9]

5 SEO

SEO je zkratka, která znamená Search Engine Optimization, tedy česky optimalizace pro vyhledávače. SEO jako takové je vlastně obecný postup, jak optimalizovat konkrétní webové stránky pro vyhledávače. Zabývá se optimalizací pro fulltextové vyhledávání, tzn. pro vyhledávání klíčového slova nebo slov. Optimalizací se v této souvislosti myslí vytvoření nebo upravení struktury webových stránek tak, aby vyhledávače zaindigovaly (uložily do své databáze) co nejvíce klíčových slov nebo frází, obsažených na těchto stránkách. Situace na internetu je velmi podobná reálnému konkurenčnímu prostředí. Čím více internet obsahuje webových prezentací, tím je náročnější přimět návštěvníka k návštěvě právě našeho webu. Je lhostejné zda se jedná o komerční web nebo neziskový server, cílem vždy je, aby na tyto stránky přicházeli návštěvníci, pokud možno v co největší míře.

Některé postupy užívané při aplikaci SEO:

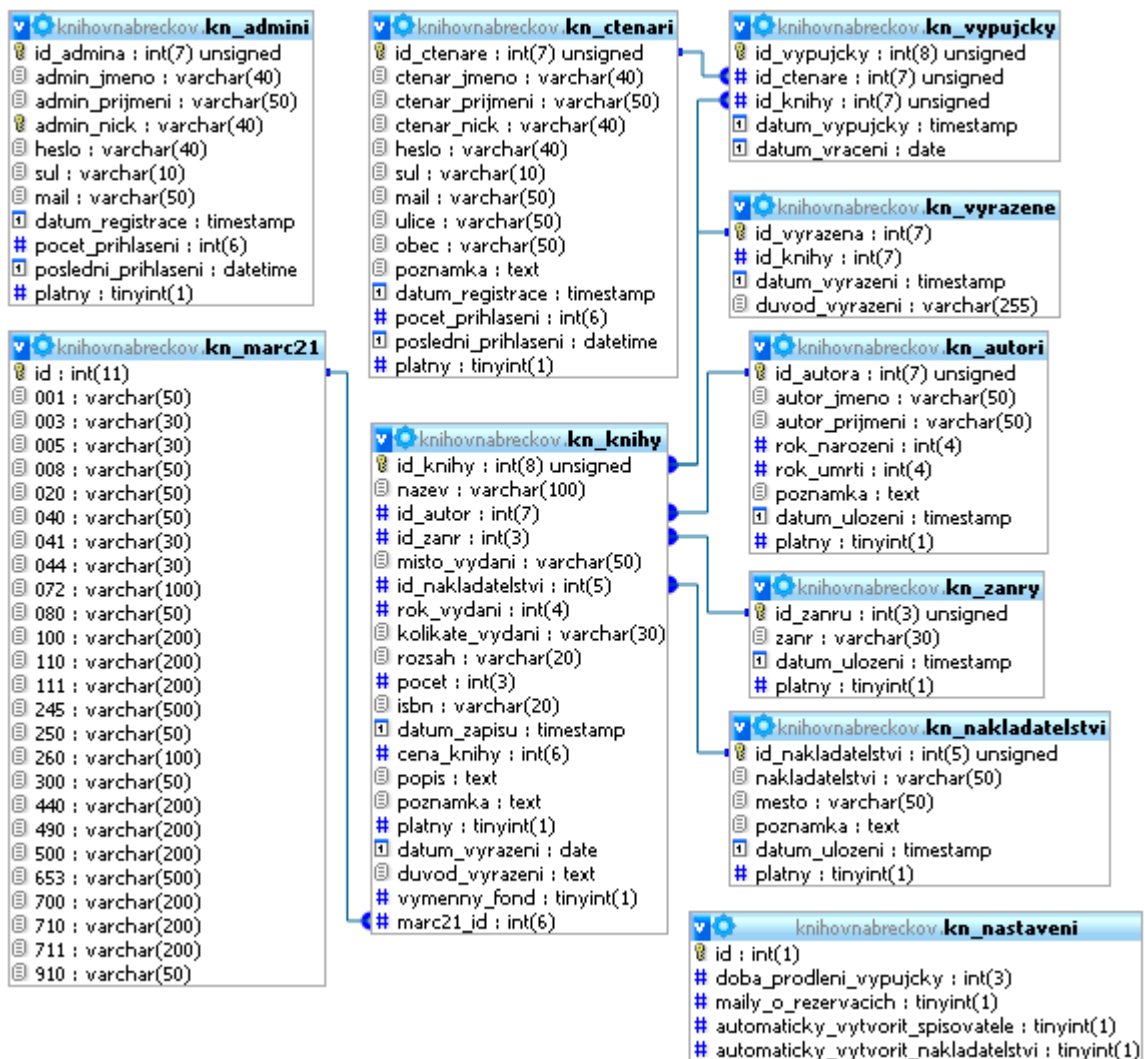
- Stanovení slova a frází, které jsou nás z hlediska obsahu nejdůležitější a na něž chceme web optimalizovat. Musíme vzít v úvahu, že žádné stránky nemohou být optimalizovány na všechna slova.
- Titulek by měl vždy správně reprezentovat název stránky, který odpovídá skutečnosti.
- Nepoužívat rámce (frames), tento styl psaní stránek až na malé opodstatněné výjimky patří minulosti a vyhledávače je nemají rádi.
- Používejme CSS styly pro vzhled webu místo nesmyslných obrázků a javascriptových funkcí.
- Vyhnout se tabulkám, které určují členění webu, naraďme je DIVy. Vyžaduje to mnohem více zkušeností, aby se stránky dobře zobrazovaly ve všech prohlížečích, ale výsledkem je správné rozvržení, které vyhovuje i vyhledávačům.
- Udržovat moderní webdesign a validní kód podle W3C standardů, nejlépe XHTML, který nahrazuje předešlé HTML.
- Nepoužívat neetické metody ke zvýšení návštěvnosti, zpravidla se dočkáme opačného výsledku a vyhledávače (často na základě konkurenční upozornění) naši stránku odstraní ze svého seznamu. [10]

II. PRAKTICKÁ ČÁST

6 DATABÁZE

Databázový model byl rozšířen o tabulku *kn_marc21*, která obsahuje záznamy knih ve formátu MARC21 importované z výměnného fondu. Dále byla rozšířena tabulka *kn_nastaveni*, o nové pole vztahující se k importu záznamů MARC21.

Relační schéma tabulek bylo vytvořeno nástrojem Návrhář, který je součástí phpMyAdmin.



Obrázek 2 - Relační schéma databáze knihovny

6.1 Tabulky

V tabulce *kn_knihy* přibyl cizí klíč *marc21_id*, který odkazuje na primární klíč v tabulce *kn_marc21*, jak je vidět na relačním modelu níže.

Data jsou importována z XML souboru a není předem definováno, jakého jsou datového typu, proto jsou všechny pole v tabulce typu *varchar*. Bibliografické záznamy jsou velmi variabilní a nelze předpokládat jednoznačný datový typ ani u polí typu počet stran nebo rok vydání, protože mohou obsahovat různé upřesňující údaje.

Názvy sloupců jsou pro pojmenovány podle polí, jejichž hodnoty uchovávají. Pole MARC nemají definovanou délku. Jsou zvoleny délky polí, které jsou pro dané pole dostatečně dlouhé.

Tabulka 2 - Tabulka MARC21

Sloupec	Typ	Nulový	Výchozí
id	int(11)	Ne	Žádná
1	varchar(50)	Ano	NULL
3	varchar(30)	Ano	NULL
5	varchar(30)	Ano	NULL
8	varchar(50)	Ano	NULL
20	varchar(50)	Ano	NULL
40	varchar(50)	Ano	NULL
41	varchar(30)	Ano	NULL
44	varchar(30)	Ano	NULL
72	varchar(100)	Ano	NULL
80	varchar(50)	Ano	NULL
100	varchar(200)	Ano	NULL
110	varchar(200)	Ano	NULL
111	varchar(200)	Ano	NULL
245	varchar(500)	Ano	NULL
250	varchar(50)	Ano	NULL
260	varchar(100)	Ano	NULL
300	varchar(50)	Ano	NULL
440	varchar(200)	Ano	NULL
490	varchar(200)	Ano	NULL
500	varchar(200)	Ano	NULL
653	varchar(500)	Ano	NULL
700	varchar(200)	Ano	NULL
710	varchar(200)	Ano	NULL
711	varchar(200)	Ano	NULL
910	varchar(50)	Ano	NULL

7 MARC 21

Při návrhu struktury nové tabulky *kn_marc21* jsem zohlednil potřeby knihovny v malé obci a její rozsah tedy odpovídá minimálnímu záznamu pro tištěné monografie, který schválila Rada pro katalogizační politiku. Rozsah minimálního záznamu je uveden na webovém portálu Národní knihovny České republiky. [11]

7.1 MARC 21 XML

Jako podporovaný formát importovaného souboru jsem zvolil MARC21 XML. Původní návrh formátu MARC definuje textový soubor s pevnou strukturou, která je obtížná na manipulaci zpracování. MARC 21 XML byl definován jako moderní nástupce tohoto formátu a jeho zpracování v prostředí PHP5 je usnadněno ucelenou sadou funkcí.

Pro manipulaci s dokumentem XML se mi nejdříve zdálo vhodné rozhraní SimpleXML. Při pokusu o zpracování jsem ale narazil na omezení, které neumožňovalo pracovat se jmennými prostory, které dokument MARC 21 XML používá. Z tohoto důvodu jsem nakonec použil rozhraní DOM, které nepracuje s XML v PHP natolik přirozeně jako SimpleXML ale podporuje jmenné prostory.

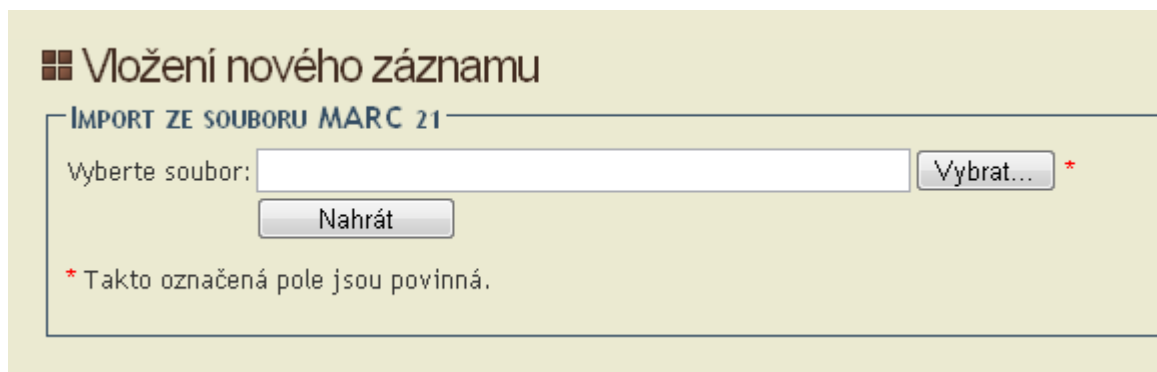
7.2 Import souboru XML

V administrátorském rozhraní přibylo v katalogu knih tlačítko na import souborů MARC 21. Je podporováno importování více záznamů z jednoho souboru.



Obrázek 3 – Nové tlačítko Import souboru MARC 21

Tlačítko přesměruje knihovníka na stránku, kde vybere soubor pro import.



Vložení nového záznamu

IMPORT ZE SOUBORU MARC 21

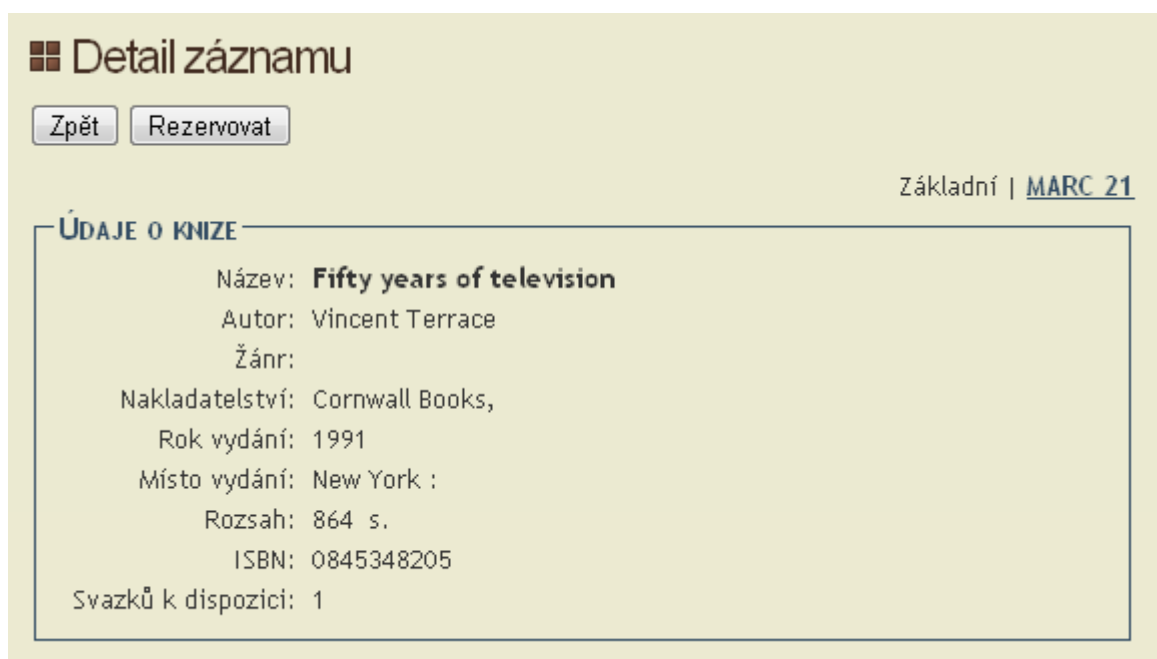
Vyberte soubor: *

* Takto označená pole jsou povinná.

Obrázek 4 – Formulář pro výběr importovaného souboru

Soubory jsou na serveru uloženy do dočasného adresáře, ze kterého jsou po úspěšném importu vymazány. Pokud by se během importu vyskytla chyba a proces nedoběhl do konce, bude soubor vymazán při příštím importu.

U knihy, které v katalogu vznikly importem, se v detailu záznamu zobrazí tlačítka na přepnutí základního zobrazení a zobrazení MARC 21.



Detail záznamu

Základní | [MARC 21](#)

ÚDAJE O KNIZE

Název: **Fifty years of television**
Autor: Vincent Terrace
Žánr:
Nakladatelství: Cornwall Books,
Rok vydání: 1991
Místo vydání: New York :
Rozsah: 864 s.
ISBN: 0845348205
Svazků k dispozici: 1

Obrázek 5 – Detail knihy v základním zobrazení

The screenshot shows a web interface for a library catalog. At the top, there is a title 'Detail záznamu' with a grid icon. Below the title are two buttons: 'Zpět' and 'Rezervovat'. On the right side, there is a link 'Základní' and the text 'MARC 21'. The main content is a box titled 'ÚDAJE O KNIZE' containing MARC 21 data fields:

```

001 12149120
003 Control No.
005 19920331092212.7
008 820305s1991####nyu#####001#0#eng##
020 ##$a0845348205
040 ##$aIEU$cIEU$dN@F$dDLC
041 ##$aeng
100 1#$aTerrace, Vincent,$d1948-
245 10$aFifty years of television :$ba guide to series and pilots, 1937-1988 /
   $cVincent Terrace.
260 ##$aNew York :$bCornwall Books,$cc1991.
300 ##$a864 p. ;$c24 cm.
500 ##$aincludes index.

```

Obrázek 6 – Detail knihy v zobrazení MARC 21

Bylo nutné vyřešit automatické přiřazování přírůstkového čísla, protože knihy zadávané knihovníkem mají přírůstkové číslo zadávané ručně. Bylo zvoleno automatické přiřazení přírůstkového čísla při importu z číselné řady 1 000 000 a výše, ve které nehrozí kolize s číselnou řadou, kterou zadává knihovník ručně.

7.3 Použité adresáře

Funkce, která provádí parsování XML dokumentu je nastavena tak, aby zpracovala jen adresáře (tagy) definované minimálním záznamem. Ostatní adresáře jsou ignorovány. Dokument XML se taktéž nikam nearchivuje, do databáze se uloží jen zpracované adresáře.

Funkce dále vyhodnotí, které adresáře a podpole se použijí pro vytvoření záznamu o knize. Kontrolní adresáře 00X jsou vynechány, protože obsahují všeobecné informace, které se vztahují k záznamu MARC. V následujícím výčtu jsou uvedeny adresáře, ze kterých vznikne nový záznam v tabulce *kn_knihy*.

020 Mezinárodní standardní číslo knihy (ISBN)

\$a – ISBN (včetně vysvětlivky)

100 Hlavní záhlaví – osobní jméno

\$a – osobní jméno

\$d – data související se jménem

245 Údaje o názvu

\$a – název

250 Údaje o vydání

\$a – označení vydání

260 Nakladatelské údaje

\$a – místo vydání

\$b – jméno nakladatele

\$c – datum vydání

300 Fyzický popis

\$a – rozsah

500 Poznámky

\$a – poznámka

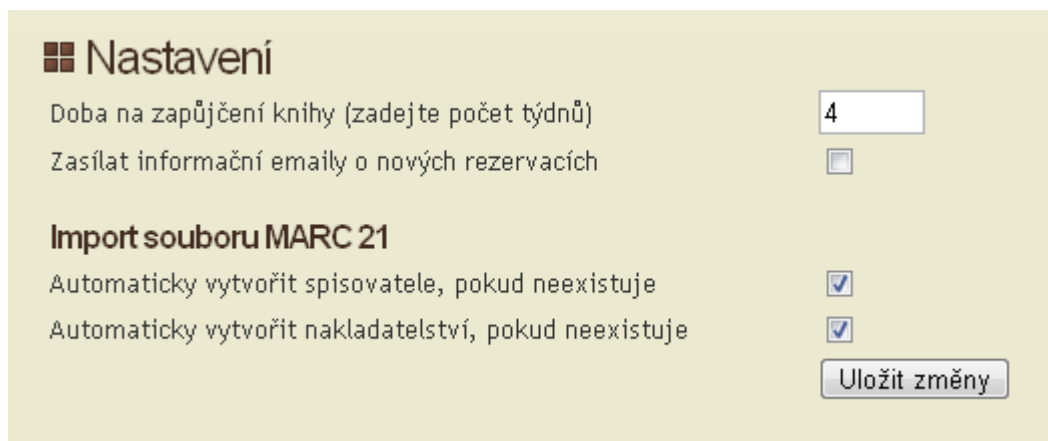
U adresáře 020 a podpoli \$c má být podle normy uvedeno datum dostupnosti (*Terms of availability*). Na tomto místě jsem našel odlišný způsob použití v katalogu Clavius – jednoho z nejrozšířenějších katalogových systémů v České republice, která v poli \$c uvádí cenu za knihu. K využití tohoto pole proto nebylo přistoupeno, aby nedocházelo k nekonzistenci dat.

7.3.1 Automatické vytvoření záznamu v číselníku

Při importu knih z výměnného fondu bude často nastávat situace, kdy spisovatel nebo nakladatelství nebude existovat v číselníku. Pro tento případ je možné při importu tyto záznamy v databázi automaticky vytvořit pokud nebudou nalezeny. Funkce, která vyhodnocuje existenci záznamu v číselníku, ovšem nemůže pokrýt případ, kdy je záznam zapsaný v odlišném tvaru, např. jedno ze jmen je zapsáno pouze prvním písmenem

s tečkou. Aby mohl knihovník zabránit vzniku duplicitních údajů, má možnost tuto funkci v nastavení vypnout. Číselníkové údaje pak musí doplnit ručně.

Duplicitní záznamy poskytují čtenáři stejnou informaci, neumožňují však efektivně vyhledávat záznamy podle parametrů a v databázi nastává redundance dat.



Nastavení

Doba na zapůjčení knihy (zadejte počet týdnů)

Zasílat informační emaily o nových rezervacích

Import souboru MARC 21

Automaticky vytvořit spisovatele, pokud neexistuje

Automaticky vytvořit nakladatelství, pokud neexistuje

Obrázek 7 – Volby pro import souboru MARC 21

7.3.2 Relace mezi záznamem knihy a MARC 21

Importem jednoho záznamu MARC 21 v databázi vzniknou záznamy v tabulkách *kn_knihy* a *kn_marc21*. V tomto bodě jsem narazil na problém, jak vložit id záznamu v tabulce *kn_marc21* do cizího klíče v tabulce *kn_knihy*. Řešením této situace je použít PHP funkci `LAST_INSERT_ID()`, která vrátí ID vytvořené při posledním dotazu do databáze.

Funkci `LAST_INSERT_ID()`, lze použít i ve víceuživatelském prostředí, protože je vázána na aktuálně otevřené spojení s databází. Nemůže tedy nastat situace, kdy by bylo vráceno ID jiné transakce.

8 SEO

8.1 Analýza

Při tvorbě stránek byl kladen důraz na dodržování obecných zásad SEO. Přesto byly při důkladnější analýze stavu nalezeny některé nedostatky:

- Titulek okna nezobrazoval na všech stránkách vhodný popis
- Kód nebyl validní podle W3C

Na webu organizace W3C, která vyvíjí webové standardy, je dispozici validátor, který zobrazí k nevalidním částem kódu popis s návodem, jak chybě předejít. Díky tomuto nástroji je snadné chyby v kódu opravit.

Errors found while checking this document as HTML 4.01 Strict!	
Result:	8 Errors, 1 warning(s)
Source:	<pre><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"> <html> <head><title>Povídky z jedné kapsy - Knihovna Horní Břečkov</title> <meta http-equiv='Content-Language' content='cs'> <meta http-equiv='Content-Type' content='text/html; charset=utf-8'> <meta name='description' content='Knihovna Horní Břečkov'><link rel='stylesheet' type='text/css' href='http://localhost/knihovna/stylesheet.css'> <link rel='stylesheet' type='text/css' href='http://localhost/knihovna/knihovna.css'></head></pre>
Encoding:	utf-8 <input type="text" value="(detect automatically)"/>
Doctype:	HTML 4.01 Strict <input type="text" value="(detect automatically)"/>
Root Element:	HTML

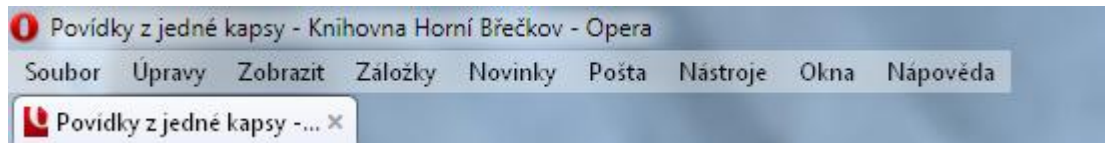
Obrázek 8 – Validátor W3C

8.2 Náprava

Titulek okna

Text za pomlčkou byl změněn z *Online knihovna – Horní Břečkov* na *Knihovna Horní Břečkov*. Někteří odborníci na SEO doporučují název stránek/firmy z titulku úplně odstranit. Domnívám se, že v tomto případě je vhodné název *Knihovna Horní Břečkov* zachovat, protože se jedná o místní název a pro vyhledávače je to relevantní údaj.

Titulek byl také upraven v případě zobrazení detailu knihy. Nyní se nově zobrazuje na začátku titulku název knihy.



Obrázek 9 – Zobrazení názvu knihy v titulku

Validace kódu

Po odstranění nevalidních tagů a konstrukcí kódu, je stránka schválena jako validní podle standardu HTML 4.01 Strict. Zobrazené varování se týká způsobu validace. Tento obrázek byl pořízen po validaci zdrojového kódu vloženého do validátoru jako text, který byl zkopírován přes schránku. V tomto případě validátor neumí správně pracovat se znakovou sadou definovanou na stránce a výsledkem varování, které není způsobené chybným zápisem kódu.

This document was successfully checked as HTML 4.01 Strict!	
Result:	Passed, 1 warning(s)
Source :	<pre><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"> <html> <head><title>Úvod - Knihovna Horní Břečkov</title> <meta http-equiv='Content-Language' content='cs'> <meta http-equiv='Content-Type' content='text/html; charset=utf-8'> <meta name='description' content='Knihovna Horní Břečkov'><link rel='stylesheet' type='text/css' href='http://localhost/knihovna/stylesheet.css'> <link rel='stylesheet' type='text/css' href='http://localhost/knihovna/knihovna.css'> <script type='text/javascript'></pre>
Encoding :	utf-8 utf-8 (Unicode, worldwide) ▼
Doctype :	HTML 4.01 Strict (detect automatically) ▼
Root Element:	HTML

Obrázek 10 - Úspěšná validace stránky

9 NOVÉ FUNKCE KNIHOVNY

9.1 Výpůjčky

Výpůjčky čtenáře

Administrátor měl v předchozí verzi systému možnost zobrazit pouze všechny výpůjčky, s možností seřazovat tabulku podle sloupců. Nyní přibyla možnost filtrovat výsledky pomocí formuláře Zobrazit výpůjčky na detailu čtenáře.



Obrázek 11 – Nové tlačítko Zobrazit výpůjčky

9.2 Knihy

Kontrola ISBN

Původní systém měl jeden větší nedostatek a tím byla chybějící funkce, která by kontrolovala správnost zadaného ISBN. Systém nyní umí validovat jak starší verzi ISBN-10, tak novější ISBN-13, která je identická s kódem EAN-13. Ověření ISBN-13 se týká prefixu 978, který se používá v současnosti, a je připraveno i na prefix 979, se kterým se počítá do budoucna, až bude vyčerpána kapacita prefixu 978.

Obrázky obalů knih

Návrh systému také počítal s možností vkládat ke knihám obrázky obalů. Tato funkcionality by v reálném prostředí vyžadovala poměrně velké technické a časové zatížení knihovníka. Stahování obrázků z internetu je problémové kvůli autorským právům

a skenování knih přímo v knihovně, jejich úprava a vkládání ke každé knize zvlášť by pravděpodobně nebylo v časových možnostech knihovníka. Tato funkcionality tedy nakonec nebyla v původním systému implementována.

Řešení přinesla webová služba ObálkyKnih.cz, která vznikla v roce 2008 díky projektu Moravské zemské knihovny. Na webu služby jsou shromažďovány obaly knih z různých zdrojů, a ty je následně možné za určitých podmínek zobrazovat na webech knihovních informačních systémů.

Připojení k webové službě systému ObálkyKnih.cz probíhá přes poskytnuté rozhraní API. Na službu je odeslán požadavek na zobrazení obálky s parametrem ISBN a v kladném případě je vrácen odkaz na obrázek obálky.

Detail záznamu

[Zpět](#) [Rezervovat](#)

ÚDAJE O KNIZE

Název:	Evoluční výpočetní techniky - principy a aplikace
Autor:	Ivan Zelinka
Žánr:	Všeobecnosti
Nakladatelství:	BEN - technická literatura
Rok vydání:	2009
Místo vydání:	Praha
Kolikáté vydání:	1. české vyd.
Rozsah:	536 s.
ISBN:	80-7300-218-3
Svazků k dispozici:	1

Evoluční výpočetní techniky
Principy a aplikace

Ivan Zelinka - Zuzana Oplátková
Miloš Šeda - Pavel Ošmera
František Věleň

BEN

Obrázek 12 – Detail knihy s obrázkem obalu

Odkaz na Wikipedii

Pro větší komfort čtenářů, bylo upraveno zobrazení detailu knihy. Nyní se u autora zobrazuje odkaz vedoucí na stránky české Wikipedie. Čtenář tedy může v případě zájmu jedním kliknutím zjistit o spisovateli více informací.

Nově je také možné kliknout na název autora a zobrazit od něj všechny knihy v katalogu.



Obrázek 13 - Odkaz na Wikipedii se stránkou o spisovateli

9.3 HTML5

HTML formuláře byly upraveny pomocí nových formulářových atributů, dostupných ve specifikaci HTML5. Uživatel je upozorněn ještě před odesláním formuláře např. na pole, která je nutné vyplnit nebo na špatně vyplněnou emailovou adresu.

Validace formulářů na serverové části je zachována z důvodu zpětné kompatibility, protože ne všechny prohlížeče podporují novou specifikaci.

Povinné formulářové pole typu `input='text'`, do kterého zadáváme jméno čtenáře, může vypadat např. takto:

```
1. <input type='text' name='ctenar_jmeno' required='required'>
```

Do tagu `type`, můžeme místo hodnoty `text` zadat hodnotu `email` a prohlížeč při odeslání formuláře automaticky zkontroluje, zdali je vyplněna emailová adresa ve správném tvaru:

```
1. <input type='email' name='mail' size='30'>
```

Vložení nového záznamu

NOVÝ UŽIVATEL

Pořadové číslo: * (Nejnižší dostupné čtenářské číslo: 7)

Jméno: *

Příjmení: *

Přihlašovací jméno: *

Heslo: *

Heslo znovu: *

E-mail:

Ulice a č. p.:

Obec:

Poznámka:

* Takto označená pole jsou povinná.

Toto je povinně vyplňované pole

Obrázek 14 – Validace formulářového pole atributem „required“

10 IMPLEMENTACE

Portál knihovny bude provozován na doméně třetího řádu, na oficiální internetové prezentaci Horního Břečkova na adrese **<http://knihovna.hornibreckov.cz>**.

V době dokončení této práce je systém v testovacím režimu. Po nahrání souborů na FTP, se ukázalo, že server pracuje odlišně s kořenovou složkou, než jak je tomu u vývojového prostředí na lokálním počítači a bylo potřeba upravit konstantu, která je používána jako absolutní cesta k souborům.

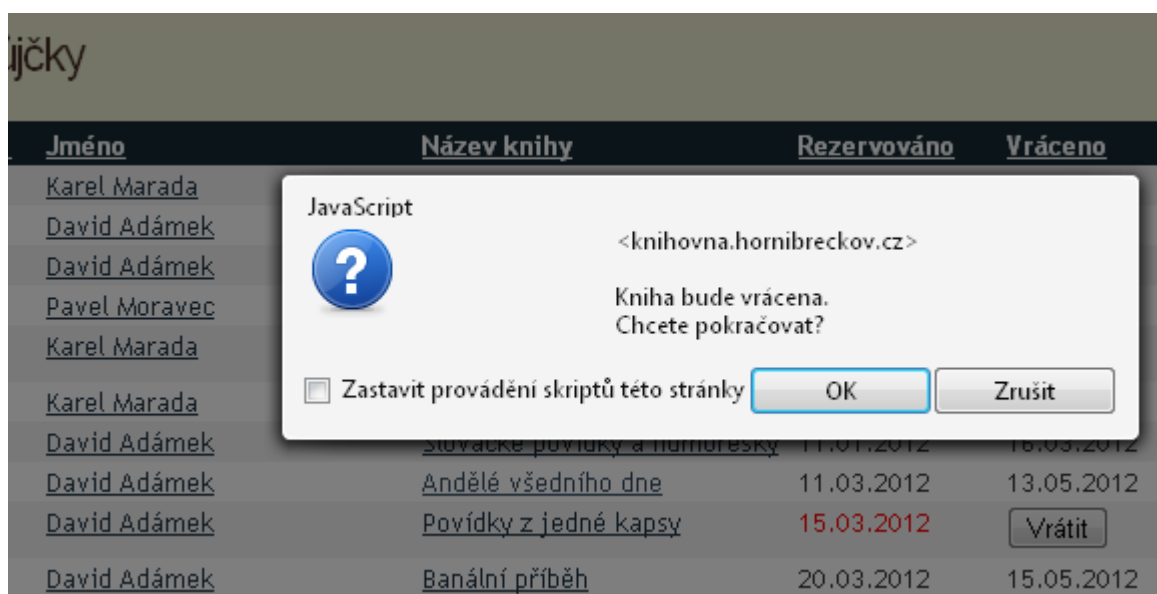
```
1.  if ($_SERVER['SERVER_NAME'] == "localhost")
2.    define("PATH", "http://".$_SERVER['SERVER_NAME']."/knihovna/");
3.  else
4.    define("PATH", "http://".$_SERVER['SERVER_NAME']."/");
```

Tato podmínka zjistí ze superglobální proměnné `$_SERVER['SERVER_NAME']` název běžícího serveru a podle toho rozhodne, která definice se použije. Tím je automaticky zajištěna funkčnost aplikace jak ve vývojovém prostředí, tak na ostrém serveru.

Při testovacím provozu se zatím vznikla jedna připomínka na zlepšení a jedena procesní chyba.

1. Při vrácení knihy by měl být knihovník dotázán, zda chce opravdu knihu vrátit, aby nedocházelo k nechtěnému vrácení knihy.
2. Proces rezervací by měl být rozdělen do dvou fází: **rezervace, vypůjčeno**.

Úprava, která zobrazí dotaz před vrácením knihy, byla přidána v průběhu testovacího provozu.



Obrázek 15 - Potvrzení před vrácením knihy

ZÁVĚR

Hlavním cílem této práce bylo rozšířit funkcionalitu obecní knihovny v Horním Břečkově. Knihovnik má nyní nástroj na jednoduchý import knih do výměnného fondu. To by mohlo prospět k častějšímu používání výměnného fondu a rychlejší obměně knihovní bibliotéky. Podpora formátu MARC 21 v jeho XML variantě zaručuje dlouhou podporu ze strany ostatních knihoven, protože formát MARC 21 je celosvětově nejpoužívanější protokol, mezi knihovnickými informačními systémy.

Při vkládání nové knihy je nyní kontrolována pravost ISBN, a to jak v jeho aktuálním formátu ISBN-13 tak i v jeho starší verzi ISBN-10.

Pro čtenáře bude největším přínosem implementace webové služby ObálkyKnih.cz. Tím je zaručeno, že se u knih bude v mnoha případech zobrazovat i obálka a později bude možné připojit i další informace. Mezi další změny patří větší možnosti filtrování katalogu knih a přímý odkaz na stránku o spisovateli na Wikipedii.

Aplikace byla testována a je plně funkční v prohlížečích IE9, Opera 11.64, Chrome 19 a Firefox 12.

Informační systém byl nasazen do testovacího režimu, ze kterého již vzešly připomínky na úpravy, a také se ukázalo, že systém je schopný fungovat v reálném prostředí. Pokud fáze testování skončí úspěšně, bude katalog propojen se stránkami obce a spuštěn v ostrém provozu.

Další rozvoj knihovny je možné směřovat k větší spolupráci s webovou službou ObálkyKnih.cz. Server plánuje v budoucnu rozšířit API o zobrazování hodnocení, komentářů, anotací a dalších informací ke knihám. Do budoucna je počítáno se vznikem nových požadavků od uživatelů, které budou možnosti knihovny dále rozšiřovat.

SUMMARY

The main aim of this work was to extend functionality of village Library in Horní Břečkov.

The librarian has now a tool for easy data import into the library transfer system. It can make the library transfer system used more often and it helps to refresh books. Support of the MARC 21 format in its XML variant give to the library long lasting support from the big libraries, because it is most used like exchange protocol among the Library Information Systems.

When the librarian inserts new book record, system now validates ISBN in his new ISBN-13 format and old ISBN-10 format as well.

The biggest innovation for the readers will be implementation of the web service ObálkyKnih.cz. It gives to the system ability to display the cover of the book and later, there is a possibility to add more information in the future. Other changes are bigger options to filter list of the books and the direct hyperlink to the writers Wikipedia page.

Application was tested and is fully functional in web browsers IE9, Opera 11.64, Chrome 19 a Firefox 12.

Library Information System is now in a testing mode, which already provided two notes and it also was proved, that the System is capable to work in a real environment. If the testing mode ends with success, System will be connected to the web site of the village presentation and will be launched for a real operation.

Next development can be focused to a better cooperation with the ObálkyKnih.cz web service. This server has ambitions to provide more API functions to display ratings, comments, annotations and some more informations about the books. In the future there is room for new requests from the users, which will result into the upgrades of the library.

SEZNAM POUŽITÉ LITERATURY

- [1] GUTMANS, Andi, Stig Sæther BAKKEN a Derick RETHANS. Mistrovství v PHP 5. Vyd. 1. Překlad Bogdan Kiszka. Brno: CP Books, 2005, 655 s. ISBN 80-251-0799-X.[2] W3C. Extensible Markup Language (XML) [online]. 2012, 2012.01.24 [cit. 2012-05-16]. Dostupné z: <http://www.w3.org/XML/>
- [2] KOSEK, Jiří. PHP a XML. 1. vyd. Praha: Grada, 2009, 367 s. ISBN 978-80-247-1116-4.
- [3] KOSEK, Jiří. XML pro každého: podrobný průvodce. 1. vyd. Praha: Grada, 2000, 163 s. ISBN 80-716-9860-1.
- [4] RIORDAN, Rebecca M. Vytváříme relační databázové aplikace. Vyd. 1. Praha: Computer Press, 2000, 280 s. ISBN 80-722-6360-9.
- [5] MARC. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 3. 5. 2012 [cit. 2012-05-05]. Dostupné z: <http://cs.wikipedia.org/wiki/MARC>
- [6] LIBRARY OF CONGRESS. Appendix B - Full Level Record Examples [online]. 2008 [cit. 2012-05-16]. Dostupné z: <http://www.loc.gov/marc/bibliographic/examples.html>
- [7] Marc 21: bibliografický formát. 1. čes. vyd. Praha: Národní knihovna České republiky, 2003. ISBN 80-705-0427-7.
- [8] Mezinárodní registrační systémy. Národní knihovna České republiky [online]. © 2006 [cit. 2012-05-16]. Dostupné z: http://www.nkp.cz/pages/page.php3?navez=ISBN,_ISMN,_ISSN&submenu3=17
- [9] Struktura čísla ISBN a jeho rozsah. Národní knihovna České republiky [online]. © 2006 [cit. 2012-05-16]. Dostupné z: http://www.nkp.cz/pages/page.php3?page=isbn_5.htm
- [10] SEO master [online]. 2010 [cit. 2012-05-14]. Dostupné z: <http://www.seomaster.cz>
- [11] Národní knihovna České republiky. MARC21 – minimální záznam pro tištěné monografie [online]. 27. 10. 09 [cit. 2012-05-17]. Dostupné z: http://www.nkp.cz/pages/page.php3?page=fond_marc_tist_monogr.htm

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AJAX	Asynchronous JavaScript and XML – asynchronní JavaScript a XML
DOM	Document Object Model – objektový model dokumentu
EAN	European Article Number – Evropské číslo zboží
FTP	File Transfer Protocol – protokol pro přenos souborů
HTML	HyperText Markup Language – značkovací jazyk pro hypertext
ISBN	International Standard Book Number – Mezinárodní standardní číslování knih
MARC	MACHINE-Readable Cataloging – strojově čitelná katalogizace
SAX	Simple API for XML – jednoduché API pro XML
SEO	Search Engine Optimization – optimalizace pro vyhledávače
SGML	Standard Generalized Markup Language – univerzální značkovací metajazyk
SQL	Structured Query Language – strukturovaný dotazovací jazyk
W3C	World Wide Web Consortium
XML	eXtensible Markup Language – rozšiřitelný značkovací jazyk
XLS	eXtensible Stylesheet Language – rozšiřitelný stylový jazyk
XLST	XLS Transformations – transformace XLS

SEZNAM OBRÁZKŮ

Obrázek 1 – Relační schéma tabulek MARC 21	23
Obrázek 2 - Relační schéma databáze knihovny	30
Obrázek 3 – Nové tlačítko Import souboru MARC 21	32
Obrázek 4 – Formulář pro výběr importovaného souboru	33
Obrázek 5 – Detail knihy v základním zobrazení.....	33
Obrázek 6 – Detail knihy v zobrazení MARC 21	34
Obrázek 7 – Volby pro import souboru MARC 21	36
Obrázek 8 – Validátor W3C	37
Obrázek 9 – Zobrazení názvu knihy v titulku.....	38
Obrázek 10 - Úspěšná validace stránky	38
Obrázek 11 – Nové tlačítko Zobrazit výpůjčky.....	39
Obrázek 12 – Detail knihy s obrázkem obalu.....	40
Obrázek 13 - Odkaz na Wikipedii se stránkou o spisovateli	41
Obrázek 14 – Validace formulářového pole atributem „required“	42
Obrázek 15 - Potvrzení před vrácením knihy	44

SEZNAM TABULEK

Tabulka 1 - Výpočet kontrolní číslice ISBN-13	26
Tabulka 2 - Tabulka MARC21	31

SEZNAM PŘÍLOH

P I PHP funkce – Kontrola ISBN

PŘÍLOHA P I: PHP FUNKCE – KONTROLA ISBN

```
1.  /** Kontrola ISBN
2.   * Validují se oba formáty - ISBN-10 i ISBN-13
3.   * @param $isbn ISBN může být ve tvaru s pomlčkami nebo bez
   pomlček
4.   * @return boolean Vrátí TRUE pokud je ISBN platné, v opačném
   případě FALSE
5.   */
6.  function KontrolaISBN($isbn)
7.  {
8.   // Vynulování kontrolního součtu
9.   $kontrolniSoucet = 0;
10.
11.  // Odstranění pomlček a převedení na velká písmena
12.  $isbn = strtoupper(str_replace("-", "", $isbn));
13.
14.  // Pokud se na prvních 3 pozicích vyskytne číslo 978 nebo
   979, jedná se o ISBN-13
15.  $verze = substr($isbn, 0, 3);
16.
17.  // ISBN-13
18.  if ($verze == 978 | $verze == 979)
19.  {
20.   // Pokud ISBN-13 nemá předepsaný tvar, je automaticky
   zamítnuto
21.   if (!preg_match("/^97[89][0-9]{10}$/", $isbn))
22.     return FALSE;
23.
24.   for ($i = 1; $i <= 12; $i++)
25.   {
26.    // Výpočet kontrolního součtu - každou sudou číslici z
   ISBN vynásobíme třemi
27.    // ostatní se přičtou
28.    if ($i % 2 == 0)
29.      $kontrolniSoucet += intval(substr($isbn, $i-1, 1))*3;
30.    else
31.      $kontrolniSoucet += intval(substr($isbn, $i-1, 1));
32.   }
```

```

33.     // Výpočet kontrolního čísla
34.     $kontrolniCislo = 10 - ($kontrolniSoucet % 10);
35.
36.     // Pokud kontrolní číslo souhlasí s vypočetnou číslicí,
    vrátí se TRUE
37.     if ($kontrolniCislo == substr($isbn, 12, 1))
38.         return TRUE;
39.     else
40.         return FALSE;
41. }
42.
43. // ISBN-10
44. else
45. {
46.     // Pokud ISBN-10 nemá předepsaný tvar, je automaticky
    zamítnuto
47.     if (!preg_match("/^[0-9]{9}[0-9X]$/", $isbn))
48.         return FALSE;
49.
50.     // Výpočet kontrolního součtu - každé číslo vynásobíme
    váhovým číslem
51.     // které se zvyšuje od 1 do 9
52.     for ($i = 1; $i <= 9; $i++)
53.     {
54.         $kontrolniSoucet += intval(substr($isbn, $i-1, 1)) * $i;
55.     }
56.
57.     // Pokud vyjde kontrolní číslo 10, změníme ho na písmeno X
58.     $kontrolniCislo = $kontrolniSoucet % 11 == 10 ? "X" :
    $kontrolniSoucet % 11;
59.
60.     // Pokud kontrolní znak souhlasí s vypočetným znakem,
    vrátí se TRUE
61.     if ($kontrolniCislo == substr($isbn, 9, 1))
62.         return TRUE;
63.     else
64.         return FALSE;
65. }
66. }

```