

Sem vložte zadání Vaší práce.

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

Návrh a implementace katedrální knihovny

Adéla Blažková

Vedoucí práce: Ing. Dana Vynikarová, Ph.D.

11. května 2015

Poděkování

Ráda bych poděkovala své vedoucí Ing. Daně Vynikarové, Ph.D. za poskytnuté rady při psaní bakalářské práce. Poděkování patří také Bc. Dominiku Bláhovi za konzultace ohledně komunikace se systémem Aleph a zajištění přístupu do tohoto systému.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 11. května 2015

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2015 Adéla Blažková. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Blažková, Adéla. *Návrh a implementace katedrální knihovny*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací webové aplikace pro správu knihovní agendy Katedry informačního inženýrství Provozně ekonomické fakulty České zemědělské univerzity v Praze. Pro realizaci byl vybrán webový framework Ruby on Rails, frontendový framework Twitter Bootstrap a databáze MySQL. Nasazení aplikace do produkčního prostředí s webovým serverem Apache je řešeno modulem Phusion Passenger. Výsledkem práce je funkční aplikace, která umožňuje správu čtenářů, výpůjček, rezervací a stažení knihovního fondu z knihovního systému Aleph. Do budoucna je možné v aplikaci vytvořit podporu i pro ostatní katedry České zemědělské univerzity.

Klíčová slova Webová aplikace, knihovní systém, webový katalog, OPAC, systém Aleph, MARC, Ruby on Rails, Ruby

Abstract

This bachelor thesis deals with design and implementation of web application for library agenda management of Department of Information Engineering at Faculty of Economics and Management, Czech University of Life Sciences. Web framework Ruby on Rails, front-end framework Twitter Bootstrap and MySQL databasis were chosen to design the application. Implementation of the application into the production environment with a web server Apache is solved by using Phusion Passenger modul. Outcome of the work is a fully functional application enabeling management of registrated readers, book loans, bookings and library fund download from the library system Aleph. The application can be further modified to support also the rest of the Czech University of Life Sciences departments.

Keywords Web application, library system, online catalog, OPAC, system Aleph, MARC, Ruby on Rails, Ruby

Obsah

Úvod	1
1 Cíl práce	3
2 Knihovní terminologie	5
3 Analýza	9
3.1 Současný stav	9
3.2 Uživatelské role	9
3.3 Funkční požadavky	9
3.4 Nefunkční požadavky	11
4 Analýza existujících řešení	13
4.1 ALEPH 500	13
4.2 Verbis	14
4.3 Clavius	14
4.4 TabbyBook	15
4.5 Koha a Evergreen	15
4.6 OPAC VuFind	16
4.7 Závěr	16
5 Návrh	19
5.1 Vybrané technologie	19
5.2 Doménový model	21
5.3 Databázový model	23
6 Realizace	25
6.1 Autentizace a autorizace	25
6.2 Získání knihovního fondu katedry	26
6.3 Automatické spouštění úloh	30

6.4	Jednoduchý vyhledávací formulář	30
6.5	Adresářová struktura aplikace	30
6.6	Fungování aplikace	32
6.7	Nasazení aplikace	34
7	Testování	35
7.1	Testovací databáze	35
7.2	Testování jednotek	35
7.3	Akceptační testování	36
7.4	Knihovní fond	39
	Závěr	41
	Literatura	43
A	Seznam použitých zkratk	47
B	Ukázka aplikace	49
C	Obsah příloženého CD	53

Seznam obrázků

3.1	Diagram případů užití	12
5.1	Doménový model	22
5.2	Databázový model	24
6.1	Aplikační a datová vrstva produktu Aleph	27
7.1	Testování registrace	38
B.1	Informace o titulu	50
B.2	Katalog knih	51
B.3	Výpůjčky a rezervace čtenáře	52

Seznam tabulek

2.1	MARC 21	8
6.1	Konverze	29

Úvod

Práce je zaměřena na správu knihovny Katedry informačního inženýrství Provozně ekonomické fakulty České zemědělské univerzity v Praze (dále jen KII PEF ČZU). Agenda knihovny je momentálně spravována pomocí sešitu a tužky. V sešitě je veden knihovní fond a jsou do něj zapisovány jednotlivé výpůjčky. Knihovna je určena pouze pro pedagogy KII PEF ČZU.

Tento systém vedení knihovní agendy není efektivní. Práci knihovníka nejvíce ztěžuje fakt, že nelze vyfiltrovávat výpůjčky od jednotlivých čtenářů. Takový proces je potřeba, pokud například některý z pedagogů ukončí pracovní poměr na katedře a je nutné zkontrolovat, zda jsou vrácené všechny jeho vypůjčené knihy.

Procesy spojené s knihovní agendou se dají velmi dobře automatizovat, a tak jsou dnes v naprosté většině knihoven používány knihovní systémy a webové katalogy.

Knihovní systémy jsou zaměřeny na správu knihovny. Do té spadá organizace výpůjček, rezervací, čtenářů a knihovního fondu. Systém může také zajišťovat procesy spojené s obnovou knihovního fondu a nákupem nových výtisků.

Webové katalogy vystavují knihovní fond na internet. Katalog slouží především pro čtenáře, kteří tak mohou vyhledávat záznamy v knihovním fondu. Také se mohou do katalogu přihlásit a sledovat stav svých výpůjček, knihovních poplatků nebo si rezervovat knihy.

Knihovní fond KII PEF, stejně jako ostatních kateder ČZU, je součástí ústřední knihovny ČZU a je uložen v knihovním systému Aleph. Tento systém však není dostupný pro správce knihovny KII PEF, kterému by mohl usnadnit práci s knihovní agendou.

Obsahem práce je analýza požadavků správce knihovny KII PEF. Následuje rešerže existujících knihovních systémů a webových katalogů. Na základě analýzy a rešerže je navrženo vlastní řešení včetně výběru technologií. V rámci implementace jsou v práci popsány stěžejní body realizace, podpůrné knihovny

ÚVOD

a struktura aplikace. Na závěr je aplikace ověřena jednotkovým a akceptačním testováním.

Cíl práce

Cílem mé práce je navrhnout a implementovat webovou aplikaci pro správu knihovny KII PEF ČZU.

Náplní teoretické části je rešerše již existujících knihovnických systémů a zhodnocení, do jaké míry splňují požadavky správce KII PEF.

V praktické části je na základě analýzy navrženo vlastní řešení aplikace knihovny KII PEF a aplikace je implementována v open-source prostředí. Nad výslednou aplikací jsou provedeny akceptační testy.

Knihovní terminologie

V této kapitole jsou vysvětleny termíny, které jsou běžně užívané v knihovnictví a budou se vyskytovat v textu následujících kapitol.

2.0.1 Katalogizace

Katalogizace je zpracování dokumentu, který je zařazen do knihovního fondu, a umožňuje jeho vyhledávání v systému. [1] Tento proces může probíhat více způsoby. Pokud se jedná o automatické systémy, mohou být bibliografické údaje o knize staženy ze vzdálených databází, načteny čtečkou čárových kódů a nebo jsou do systému zadány ručně.

2.0.2 Akvizice

Akvizice je proces získávání nových přírůstků do knihovního fondu. Knihovní systémy často podporují i tuto činnost tím, že umožňují evidenci dodavatelů a faktur, sledování předplatného pro jednotlivé tituly a další činnosti, které jsou spojené s objednáváním nových knih. [2]

2.0.3 OPAC

OPAC (Online Public Access Catalog) je knihovní katalog dostupný přes internet, který slouží pro vyhledávání záznamů v knihovním fondu. OPAC poskytuje bibliografické údaje o knize, informace o dostupnosti jednotlivých titulů a nebo o fyzickém umístění knihy v knihovně.

Čtenářům OPAC umožňuje přihlásit se do systému, kde mají poté k dispozici přehled o svých výpůjčkách, rezervacích, knihovních pokutách a historii výpůjček. Čtenář může své výpůjčky prodlužovat nebo si v systému tituly rezervovat. Čtenář také může požádat o meziknihovní výpůjčku z jiné knihovny nebo podávat návrhy na nákup nových knih do knihovního fondu.

OPAC poskytuje obvykle dvě úrovně vyhledávání v katalogu:

- Jednoduchý vyhledávací formulář, který umožňuje uživateli vyhledávat podle názvu titulu, autora, ISBN knihy nebo klíčových slov a slovních spojení.
- Rozšířený vyhledávací formulář, ve kterém lze navíc specifikovat vyhledávaný typ dokumentu, jazyk, oborovou oblast (matematika, fyzika nebo počítačová věda a jiné), časové rozmezí vydání knihy a další možnosti.

Dnes se často používá pojem OPAC nové generace, nebo-li OPAC 2.0, a jedná se o vylepšenou verzi původního katalogu OPAC. Pro vyhledávání záznamů v katalogu nové generace slouží pouze jedno pole, jako je tomu u internetových vyhledávačů, a jsou prohledány všechny oblasti katalogu. Komunita čtenářů má možnost hodnotit a komentovat jednotlivé tituly, doplňovat informace o knihách do katalogu nebo si zasílat zprávy mezi sebou. Také se čtenářům zobrazují tzv. tag clouds. Ty nabízí čtenáři podle intenzity vyhledávání zvýrazněná slova. [3]

Součástí katalogu OPAC také bývá integrovaný modul Z39.50 klient-server, který je představen v následující kapitole.

2.0.4 Z39.50

Z39.50 je standardizovaný aplikační protokol pro komunikaci mezi klientem a serverem, který umožňuje vyhledávat a stahovat data z databází. Z39.50 je však velice rozsáhlým protokolem, proto každý Z39.50 server podporuje jen podmnožinu jeho vlastností. Není tedy zajištěna vzájemná kompatibilita mezi jednotlivými systémy založenými na protokolu Z39.50 a nelze vytvořit univerzálního Z39.50 klienta, který by uměl komunikovat s libovolným Z39.50 serverem. [4]

Pokud katalogy nabízejí možnost vystavit knihovní fond pomocí Z39.50 serveru, mohou ostatní knihovny stahovat tyto data pomocí Z39.50 klienta, který je kompatibilní s daným serverem.

2.0.5 Standard MARC

MARC, nebo-li MACHine-Readable Cataloging, je široce používaný standard pro reprezentaci a výměnu především bibliografických dat, které jsou uloženy v textové databázi. Tento standard je mezinárodně využíván a podporován téměř ve všech knihovnách, aby mezi sebou mohly knihovny snadno sdílet data. Existuje několik formátů podporujících standard MARC a jedním z nejvíce používaných je formát MARC 21. [5]

2.0.5.1 Struktura formátu MARC 21

Formát MARC 21 se podle [6] skládá z následujících částí.

Návěští

Návěští je první pole v záznamu, které se skládá z 24 znaků. Každý znak nese určitou informaci, která slouží pro zpracování záznamu.

Adresář

Adresář obsahuje seznam polí, která jsou v záznamu použita, a informaci o jejich délce a počáteční pozici v záznamu.

Pole

Pole slouží pro ukládání bibliografických informací. Pro identifikaci typu pole slouží tag, který se skládá ze tří číslic. Pole se dělí na datová a kontrolní.

Kontrolní pole

Kontrolní pole obsahují základní údaje potřebné pro zpracování záznamu. Pro kontrolní pole jsou vyhrazeny tagy od 001 do 009. Například tag 001 obsahuje jedinečné identifikační číslo záznamu a tag 008 obsahuje nejdůležitější informace o knize ve zkrácené formě, které mohou být použity k identifikaci záznamu.

Datová pole

Datová pole jsou označena tagem XX1 až XX9 vyjma kontrolních tagů a slouží pro bibliografické údaje, jako například titul, autor, ISBN a další. Pole se skládá z několika podpolí, do kterých se údaje ukládají.

Základní rozdělení polí

- 0XX Kontrolní informace, identifikační čísla, kontrolní pole, klasifikační znaky
- 1XX Hlavní záhlaví
- 2XX Názvy a údaje o vydání, nakladatelské údaje
- 3XX Údaje fyzického popisu
- 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
- 9XX Pole pro národní použití [6]

2. KNIHOVNÍ TERMINOLOGIE

Tabulka 2.1: Bibliografické údaje ve formátu MARC 21 [7]

FMT	BK
LDR	00000nam-a22——a-4500
001	czu000037498
003	CZ-PrCZU
005	20100223091344.0
007	tu
008	991115s1999—-xr-a—e-p—-000-0-cze—
015	a cnb000770328
020	a 80-213-0521-5 (Česká zemědělská univerzita : brož.)
035	a (OCoLC)84981509
040	a ABA001 b cze d ABD026
072	7 a 51 x Matematika 2 Konspekt 9 13
080	a 51 2 MRF
080	a (075.8) 2 MRF
1001	a Bican, Ladislav, d 1942- 7 jk01012005 4 aut
24510	a Matematika III pro PEF / c Ladislav Bican, Václav Slavík
250	a Vyd. 1.
260	a Praha : b Česká zemědělská univerzita : b NAROMA, c 1999
300	a 124 s. : b il. ; c 30 cm
500	a Tabulky
65007	a matematika 7 ph117231 2 czenas
65009	a Mathematics 2 eczenas
655 7	a učebnice vysokých škol 7 fd133772 2 czenas
655 9	a Textbooks 2 eczenas
7001	a Slavík, Václav, d 1949- 7 mzk2004236868 4 aut
KPW	a 16877
DPD	a Doporučená
DPP	a Matematika III pro PEF - INFO b Katedra matematiky c 31110 7 3079
910	a ABD026
SID	a Z39 b NKC-Z-SN
SYS	000037498

Analýza

3.1 Současný stav

Knihovní fond KII PEF ČZU obsahuje momentálně necelých 400 titulů, přičemž od řady z nich vlastní více než jeden výtisk. Pro správu knihovního fondu je založen sešit, do kterého jsou zapsány důležité bibliografické údaje. Ve stejném sešitu jsou také vedeny výpůjčky čtenářů.

3.2 Uživatelské role

Pro aplikaci knihovny KII PEF byly analyzovány tři uživatelské role:

- Nepřihlášený uživatel
- Čtenář - tato role je určena pouze pro pedagogy KII PEF ČZU.
- Knihovník - tato role má funkci správce aplikace.

3.3 Funkční požadavky

Kapitola obsahuje seznam funkčních požadavků na aplikaci, které jsou rozděleny podle uživatelských rolí.

3.3.1 Funkční požadavky společné pro všechny uživatelské role

1. Každý uživatel si bude moci prohlížet knihovní fond katedry a vyhledávat v něm záznamy pomocí jednoduchého vyhledávacího formuláře. Vyhledávat bude možné podle titulu, autora a ISBN (International Standard Book Number) knihy.

3.3.2 Funkční požadavky z pohledu čtenáře

1. Registrace
Každý čtenář, který si bude chtít vypůjčit knihu z knihovního fondu katedry, se bude muset zaregistrovat do aplikace. Při registraci bude třeba vyplnit jméno, příjmení, email a heslo. Než bude čtenář vpuštěn do aplikace, musí jeho registraci nejprve schválit knihovník. Tento požadavek schválení uživatele je nutný, jelikož aplikace je určena pro pedagogy KII PEF ČZU.
2. Přihlášení
Čtenář se do aplikace přihlašuje pomocí uživatelského emailu a hesla. Počet pokusů o přihlášení nebude v aplikaci omezen. Pokud čtenář zapomene své heslo, může zažádat o obnovu hesla, kde zadá svoji emailovou adresu. Na tu přijde obnovovací odkaz z aplikace knihovny, kam čtenář zadá nové heslo.
3. Správa čtenářského účtu
Čtenář má možnost editovat své uživatelské údaje, změnit heslo nebo svůj účet zrušit. V uživatelském nastavení si může čtenář zvolit, zda si přeje zasílat automatické upomínky z aplikace. Dále bude moci čtenář sledovat stav svých výpůjček, které lze prodlužovat, a rezervací.
4. Rezervace knihy
Přihlášený čtenář si bude moci v aplikaci rezervovat jednotlivé knihy.
5. Automatické upomínky
Pokud má čtenář aktivované automatické upomínky, bude 7 dní předem upozorněn na vrácení vypůjčených knih do knihovny pomocí e-mailové zprávy. Upomínka bude obsahovat seznam knih pro vrácení.

3.3.3 Funkční požadavky z pohledu knihovníka

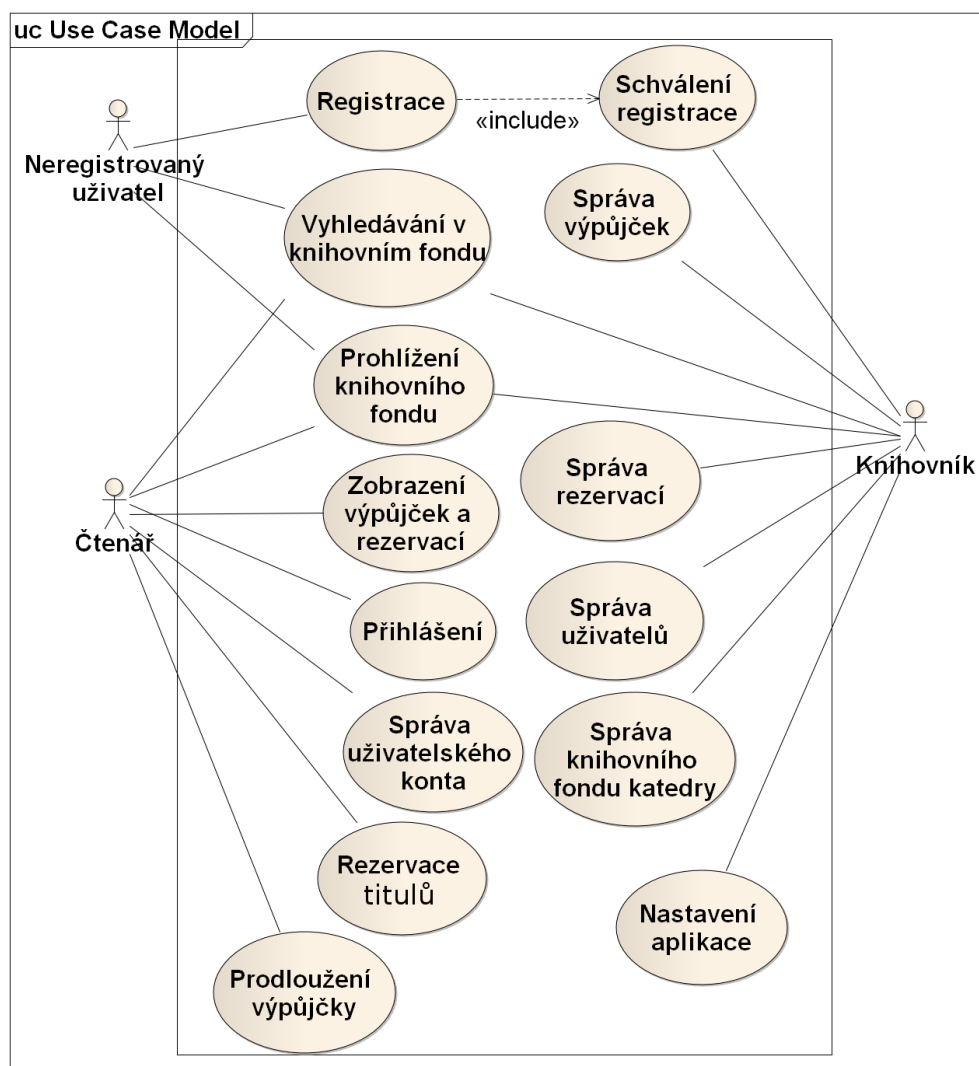
1. Přihlášení
Knihovník se bude do aplikace přihlašovat pomocí své emailové adresy a hesla. V aplikaci bude mít možnost editovat své uživatelské údaje včetně hesla. Knihovník nebude mít možnost registrovat se přímo v aplikaci.
2. Přehled o výpůjčkách a rezervacích
Knihovník si bude moci vypsát aktuální výpůjčky nebo rezervace čtenářů, které bude možné řadit dle zvolených parametrů (datum, jméno čtenáře, název knihy, autor). Na výpůjčky, které nebudou včas vráceny, bude knihovník upozorněn. Knihovník si bude moci vypsát také všechny rezervace, aktuální výpůjčky a jejich historii u konkrétního čtenáře.

3. Správa výpůjček a rezervací
Knihovník bude moci konkrétní výpůjčky buď prodloužit a nebo označit jako vrácené. Výpůjčku konkrétního čtenáře lze předat na jiného čtenáře, tedy původnímu čtenáři se výpůjčka automaticky vrátí a dalšímu čtenáři se výpůjčka vytvoří. U rezervací bude moci knihovník knihu buď vypůjčit nebo rezervaci čtenáři zrušit.
4. Správa knihovního fondu
Aplikace umožní knihovníkovi aktualizovat knihovní fond katedry pomocí interakce se systémem Aleph na ČZU. K jednotlivým záznamům v knihovním fondu bude moci knihovník přidávat evidenční čísla KII PEF a tyto změny budou lokálně uloženy (nebudou odeslány do systému Aleph).
5. Nastavení aplikace
Knihovník bude mít možnost aplikaci do určité míry přizpůsobit. Bude možné nastavit v ní počet dní pro vypůjčení a prodloužení, počet měsíců pro uložení historie výpůjček nebo počet knih zobrazených na stránce. V nastavení může knihovník také zrušit nebo povolit automatické zasílání e-mailů, které upozorňují na nové registrace čtenářů.
6. Správa uživatelů
Knihovník potvrzuje nebo zamítá nově zaregistrované čtenáře, má možnost editovat jejich uživatelské údaje nebo zrušit čtenářské konto, přičemž čtenář je o změnách knihovníkem informován emailem.

3.4 Nefunkční požadavky

1. Knihovna KII PEF bude dostupná jako webová aplikace.
2. Aplikace bude podporovat český jazyk.
3. Aplikace bude podporovat responzivní design.
4. Knihovní fond bude napojen na databázi knihovního fondu ČZU.
5. Aplikace bude nasazena na webový server Apache.

3. ANALÝZA



Obrázek 3.1: Diagram případů užití mapuje jednotlivé funkční požadavky

Analýza existujících řešení

Kapitola obsahuje rešerši popisující knihovní systémy a webové katalogy. Je zaměřená především na integrované systémy specializující se na univerzitní knihovny, webové katalogy nové generace a do rešerše jsou zahrnuty i dva úspěšné open source knihovní systémy.

V následujících kapitolách jsou používány pojmy knihovní agenda nebo základní knihovnické služby. Ty zahrnují katalogizaci, výpůjční protokoly, akvizice, Z39.50 klient-server, modul vyhledávání a správu čtenářů či zaměstnanců knihovny.

4.1 ALEPH 500

Produkt Aleph je při analýze stěžejní částí, jelikož knihovní fond KII PEF ČZU je uložen právě v tomto systému. Aleph je integrovaný knihovní systém vyvinutý izraelskou firmou EX Libris. Na trhu je již přes 20 let a u univerzitních knihoven v České republice tento systém dominuje. Jedná se o robustní systém zaměřený na správu velkých knihoven, které se skládají z dílčích jednotek.

Aleph 500 se skládá z několika modulů a každá knihovna si navolí jednotlivé komponenty právě tak, aby jí výsledný systém co nejvíce vyhovoval. Každý modul je také možné upravit na míru.

Systém podporuje veškerou knihovní agendu a je dodáván se základním webovým katalogem, který vystavuje knihovní fond na internet. Dodávaný OPAC nedosahuje takových možností jako konkurenční webové katalogy druhé generace, které čtenářům dovolují podílet se na obsahu katalogu, mají jednotné vyhledávání v katalogu a jsou napojené na webové portály Google Books nebo Wikipedia. Uživatelské a grafické rozhraní je dnes již zastaralé oproti konkurenci, jelikož nebylo během dvacetiletého provozu přizpůsobeno soudobým standardům. [8]

Aleph 500 je otevřený systém a umožňuje interakci s jinými aplikacemi. Podle [9] pro komunikaci se systémem Aleph existují dvě možnosti:

- X-Services - třetí strana odešle http požadavek, X-Server v systému Aleph požadavek zpracuje a odešle odpověď ve formátu XML (Extensible Markup Language). Za toto rozhraní je třeba platit licenci.
- RESTful API - funguje obdobně jako předchozí rozhraní. RESTful API a X-Server nabízí podobné možnosti požadavků, avšak ne úplně stejné. Výhodou RESTful API je, že je zdarma.

4.2 Verbis

Verbis je integrovaný knihovní systém vydaný v roce 2010. Systém Verbis nabízí řešení pro menší i větší knihovny, kde dovoluje definovat až třívrstvou hierarchii poboček. Tento produkt je vhodný pro školy využívající systém Bakaláři pro správu studentů a zaměstnanců, jelikož je s tímto systémem kompatibilní. Verbis nabízí základní balíčky pro různě velké knihovny, ke kterým je možné přikoupit doplňující moduly. Systém podporuje všechny knihovnické služby a různá vylepšení pro usnadnění práce knihovníka. Verbis nabízí uživatelsky přívětivé grafické rozhraní, ve kterém se lze rychle zorientovat. [10]

4.2.1 Portaro

Portaro je webový katalog druhé generace, který lze dokoupit k systému Verbis. Portaro zvládá všechny nedostatky, které byly vytknuty u webového katalogu systému Aleph. Portaro dále umožňuje knihovně upravit vzhled katalogu tak, aby byl katalog dobře zakomponovaný do webových stránek knihovny. [11]

K systému Portaro je možné doinstalovat následující tři moduly:

- OPAC edit - tento nástroj umožňuje spravovat výpůjčky a editovat záznamy přes internet.
- OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting) provider-harvester - komponenta pro stahování a poskytování dat z knihovního fondu. Modul harvester plní funkci sběrače dat a provider data poskytuje.
- Web-services - aplikace, která umožňuje poskytovat data třetím stranám.

4.3 Clavius

Integrovaný knihovní systém Clavius používají především městské a obecní knihovny, ale nabízí i speciální licenci pro základní nebo střední školy. Tato

licence se od standardního systému ve funkčnosti příliš neliší, pouze podporuje navíc evidenci tříd či zohlednění školního roku v systému. Výhoda školní licence spočívá především ve velké slevě na zakoupení produktu.

Systém se opět skládá ze samotných modulů, které je možné různě kombinovat. Moduly systému zahrnují veškerou knihovnickou agendu jako produkty zmíněné výše. Pro školy je vytvořen základní balíček s moduly, které poskytují katalogizaci, výpůjční protokol, webový katalog a sdílenou katalogizaci. Webový katalog systému Clavius nabízí čtenářům přehled o výpůjčkách, možnost rezervovat tituly a vyhledávání v katalogu. Škola může dokoupit i ostatní moduly systému Clavius, ovšem za plnou cenu. Nevýhodou produktu Clavius a webového katalogu může být, že fungují pouze na operačním systému Windows. [12]

4.3.1 Clavius REKS a Carmen

Systém Clavius REKS a katalog Carmen řeší závilost na konkrétní platformě u předchozích produktů Clavius a jsou již multiplatformní. Carmen je navíc OPAC nové generace a nabízí čtenářům mnohem více možností. Licence pro školy se ovšem na tyto produkty nevztahuje a lze je zakoupit pouze za plnou cenu. [12]

4.4 TabbyBook

Knihovní systém TabbyBook od české společnosti Computer Aided Technologies funguje jako webová aplikace a je to odlehčená, jednoduchá varianta oproti předchozím systémům.

TabbyBook je vhodný pro menší knihovny, zejména pro základní nebo střední školy, které nepotřebují složitější knihovní systém. Produkt nabízí téměř všechny základní knihovnické služby včetně jednoduchého webového katalogu. Mezi doplňující služby patří napojení na informační systém školy, přenesení evidence knih do systému a podpora pro čtečky čárových kódů.

Oproti výše zmíněným produktům nenabízí systém automatické stahování bibliografických dat a modul akvizice. Produkt TabbyBook je nabízen zdarma, zákazník platí až za technickou podporu a doplňující služby. [13]

4.5 Koha a Evergreen

Koha a Evergreen jsou dva úspěšné open source knihovní systémy. Koha je prvním integrovaným knihovním systémem vydaným pod open-source licencí GPL. Evergreen je vydaný pod stejnou licencí a byl primárně navržen pro velké knihovní konsorcium, systém tedy nabízí možnost hierarchie knihoven v rámci systému.

Oba dva produkty jsou naprosto plnohodnotné knihovní systémy, které jsou konkurenceschopné komerčním systémům zmíněných výše. Jejich velikou výhodou je to, že jsou nabízeny zdarma. Existuje kolem nich velká komunita uživatelů, kteří systémy posouvají stále dál a tvoří podporu pro ostatní uživatele. Oba produkty se prozatím používají především v zahraničí, v České republice používá momentálně pouze jedna knihovna systém Evergreen. [14]

4.6 OPAC VuFind

VuFind je webový katalog navržený a vyvinutý ostatními knihovnami z Falvey Memorial Library na Villanovské univerzitě v Pensylvánii v roce 2007 a je poskytován zdarma pod GPL open source licencí. VuFind je zcela modulární, lze tedy nainstalovat přesně ty komponenty, které knihovna potřebuje.

Jedním z cílů tohoto systému bylo přiblížit vyhledávání v katalogu vyhledávání dat na internetu, tedy pouze jedno okno pro vyhledávání a prohlížení všech knihovních zdrojů (katalogizační záznamy, časopisy, digitální zdroje a další). Výsledky vyhledávání lze zpřesňovat pomocí fazet, které slouží k definici různých vlastností a atributů. VuFind je OPAC druhé generace, čtenáři tedy mohou doplňovat tagy k jednotlivým knihám, komentovat, přidávat citace nebo si přečíst biografii autora. A samozřejmě nabízí VuFind čtenářům přehled vypůjček, zasílání zpráv mezi uživateli a rezervaci titulů.

Katalog VuFind podporuje napojení na několik knihovních systémů, mezi výše zmíněnými například na systém Aleph a Clavius. Jelikož je VuFind pod open source licencí, existuje kolem něho velká komunita uživatelů, kteří systém dále rozšiřují, vylepšují a tvoří podporu pro ostatní uživatele.

Aktuálně podporuje VuFind 11 jazyků, mezi které patří i čeština. Katalog VuFind u nás využívá například Národní technická knihovna v Praze nebo Moravská zemská knihovna v Brně. [15]

4.7 Závěr

Knihovních systémů existuje v dnešní době mnoho s různými specializacemi. Existují jak systémy řešící problematiku rozsáhlých knihovních zařízení, tak systémy vhodné pro obecní a městské knihovny, nebo systémy zaměřené na školní a univerzitní knihovny.

Podíl na trhu tvoří i open source produkty, které již nabízejí plnohodnotné služby jako jiné placené a drahé systémy a díky své komunitě se stále vyvíjí a zlepšují.

Zmiňované knihovní systémy nejsou však vhodným řešením pro knihovnu KII PEF ČZU, jelikož knihovní fond katedry je uložen v systému Aleph a jedním z požadavků na aplikaci je interakce s tímto systémem. Tuto možnost z výše zmíněných produktů umožňuje sice webový katalog VuFind, ten ovšem neslouží jako knihovní systém, kde by se daly vypůjčovat jednotlivé

knihy. Analyzované systémy jsou navíc zbytečně robustní pro potřeby malé katedrální knihovny.

Jako řešení pro katedrální knihovnu je tedy zvolen vlastní návrh webové aplikace, která bude sloužit jako knihovní systém a zároveň webový katalog. Ten si bude umět stahovat data ze systému Aleph 500.

Návrh

5.1 Vybrané technologie

5.1.1 Ruby

Ruby je multiplatformní open source programovací jazyk. Autorem tohoto jazyka je japonský programátor Jukihiro Macumoto. Jazyk Ruby byl navržen tak, aby byl dostatečně jednoduchý pro začátečníky a zároveň dostatečně výkonný i pro zkušené programátory. Jedná se o čistě objektový jazyk. To znamená, že objektem je naprosto vše, i jednoduché datové typy jako například číslo. Při návrhu byly převzaty vybrané prvky z ostatních programovacích jazyků, konkrétně z jazyka Smalltalk, Perl a Lisp. [16]

Ruby je skriptovací jazyk, neexistuje v něm tedy typová kontrola. Proto je možné do proměnné, která obsahuje řetězec, přiřadit číslo a konverze mezi řetězcem a číslem je velice snadná. Také umí velmi dobře pracovat s textem, obvykle s použitím regulárních výrazů. Další výhodou je, že v něm lze snadno spouštět externí programy. To vše usnadňuje programátorům práci při vývoji. [17]

Jednoduchá syntaxe jazyka Ruby umožňuje psát kód rychleji a udržovat ho čitelný. Pro tvorbu různorodých aplikací v jazyce Ruby jsou k dispozici knihovny pro síťové protokoly, hry, grafické uživatelské rozhraní, XML nástroje a další. Silnou stránkou je také balíčkovací systém RubyGems, který umožňuje snadno importovat podpůrné knihovny a software do aplikace. [16]

5.1.2 Ruby on Rails

Ruby on Rails je open source webový framework založený na programovacím jazyku Ruby. Zakladatelem je David Heinemeier Hansson. Webový framework usnadňuje běžné úkoly pro vývoj webové aplikace jako generování HTML, práci s databází a routování.

Rails jsou jedním z prvních frameworků s plnou podporou REST návrhu aplikace. Rest, nebo-li Representational State Transfer, architektura umožňuje

přistupovat k datům pomocí standardních http metod get, post, put a delete.

Nemalou výhodou je aktivní komunita uživatelů, která tvoří podporu pro ostatní uživatele a rozšiřuje možnosti Ruby on Rails pomocí knihoven. [18]

Programátorské principy podle [19] dodržované v Rails:

- **DRY**

Dont Repeat Yourself – tato zásada říká, že bychom neměli v programu opakovaně psát stejný kód, ale snažit se vytvořit metodu se znovupoužitelným kódem.

- **Konvence nad konfigurací**

Aby nemusel programátor trávit čas s nastavováním konfiguračních souborů, řídí se Ruby on Rails danými pravidly. Pokud například vytvoříme model „User”, Rails automaticky ukládá data do tabulky „users”, aniž by vývojář musel cokoli nastavit.

5.1.2.1 MVC Architektura

Ruby on Rails podporují architekturu MVC (Model, View, Controller). MVC je velmi častý architektonický vzor u webových frameworků. Základní myšlenkou je oddělení logiky od výstupu. Skládá se z následujících komponent:

- Model

Část, která obsahuje logiku aplikace. Spadají sem jakékoliv výpočty, databázové dotazy nebo validace. Model přijímá vstup, zpracuje ho a odešle výsledek. Neví, odkud vstupní data přijal, ani jak budou následně zobrazena na výstupu.

- Pohled

Pohled se stará o zobrazení výstupu uživateli, obvykle se jedná o html šablonu. Obsahuje pouze minimální množství logiky, která je pro výstup nutná.

- Řadič

Tato komponenta zajišťuje propojení celé aplikace, tedy komunikaci mezi uživatelem, modelem a výstupem. Pokud uživatel zadá URL (Uniform Resource Locator) adresu do webového prohlížeče, odešle se na webový server požadavek. Ten se předá správnému řadiči, který zavolá příslušné akce pro jeho zpracování. Výsledek je předán pohledu a zobrazen uživateli. [20]

5.1.3 HTML

HTML, nebo-li Hyper Text Markup Language, je značkovací jazyk pro tvorbu webových stránek. HTML se skládá z jednotlivých tagů a každý tag má určený význam obsahu. [21]

5.1.4 CSS

CSS, nebo-li Cascading Style Sheets, je soubor, díky kterému je možné oddělit vzhled stránek od jejich obsahu. Tento soubor obsahuje popis, jakým způsobem se jednotlivé komponenty HTML zobrazí v prohlížeči. [22]

5.1.5 Twitter Bootstrap

Twitter Bootstrap je open source frontendový framework, který nabízí kolekci nástrojů pro snadnou a rychlou tvorbu uživatelského rozhraní. Kolekce obsahuje HTML a CSS šablony pro navigaci, tlačítka, formuláře a další komponenty. [23]

Tento nástroj usnadňuje také tvorbu responzivního designu. Responzivní design znamená, že vzhled stránky se optimalizuje podle typu používaného zařízení. Stránka se tedy zobrazí jinak na mobilním zařízení, tabletu nebo notebooku.

5.1.6 MySQL

MySQL je výkonný multiplatformní databázový systém se snadnou konfigurací a správou pod open source licencí GPL GNU. Výhodou MySQL oproti konkurenčním produktům je jeho rychlost, stabilita a velká komunita uživatelů.

MySQL podporuje drtivou většinu funkcí, které jsou u databází potřeba (transakce, uzamykání na úrovni řádků, cizí klíče, poddotazy a fulltextové hledání). Oproti konkurenci vyžaduje MySQL malý diskový prostor. [24]

5.2 Doménový model

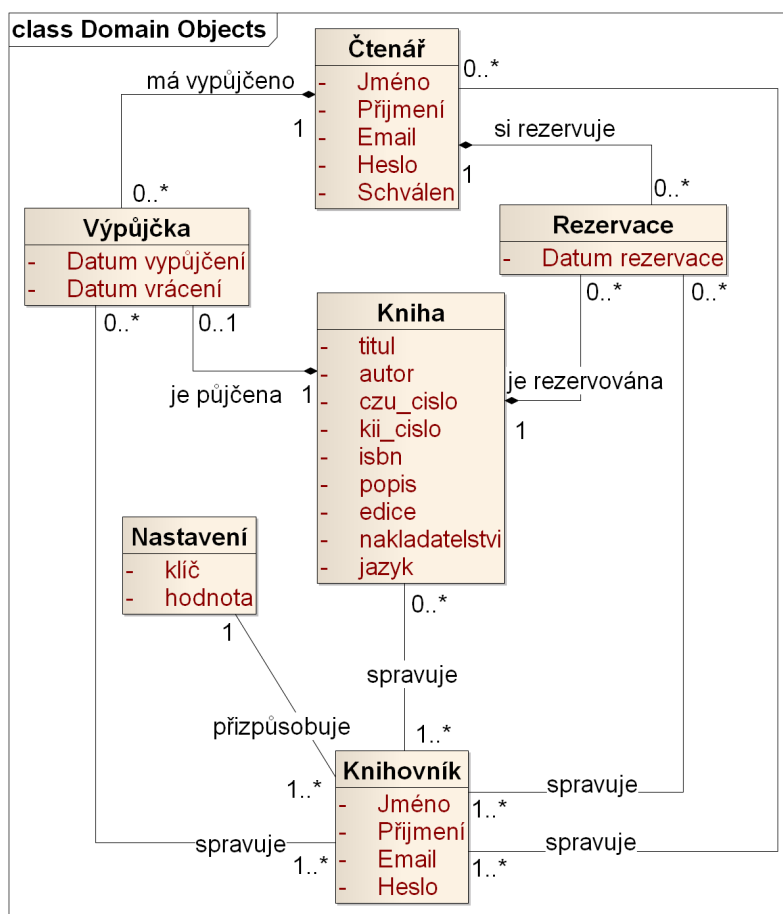
Doménový model 5.1 vychází ze zadání klienta a slouží pro identifikaci klíčových entit a vztahů mezi nimi. Jednotlivé třídy obsahují pouze důležité atributy. Mezi entitami jsou použity následující vztahy:

- Asociace - znázorňuje se plnou čarou a vyjadřuje, že entity mohou existovat nezávisle na sobě.
- Kompozice - znázorňuje se plnou čarou s vyplněným kosočtvercem na konci. Vyjadřuje vztah celek-část, přičemž část nemůže existovat bez celku.

V následujících podkapitolách jsou podrobněji vysvětleny vybrané entity.

5.2.1 Knihovník

V aplikaci je role knihovníka jako samostatná entita, jelikož jsou pro ni nastavena specifická práva.



Obrázek 5.1: Doménový model

5.2.2 Nastavení

Entita nastavení je zvolena jako jednoduché řešení přizpůsobení aplikace. Třída obsahuje atributy klíč a hodnota. Při prvním spuštění aplikace je třída naplněna předpřipravenými daty, jejichž hodnoty může knihovník následně upravovat v aplikaci.

5.2.3 Kniha

Knihovní fond stažený ze systému Aleph je uložen v textové databázi. Práce s textovou databází v aplikaci není efektivní, a to zejména při vyhledávání. Data z textové databáze jsou tedy převedena do relační databáze MySQL. Práce s relační databází má výhodu v rychlosti operací a snazší implementaci. V relační databázi jsou uloženy pouze důležité bibliografické údaje. Pokud by do budoucna bylo nutné vést další údaje, lze je do databáze snadno přidat.

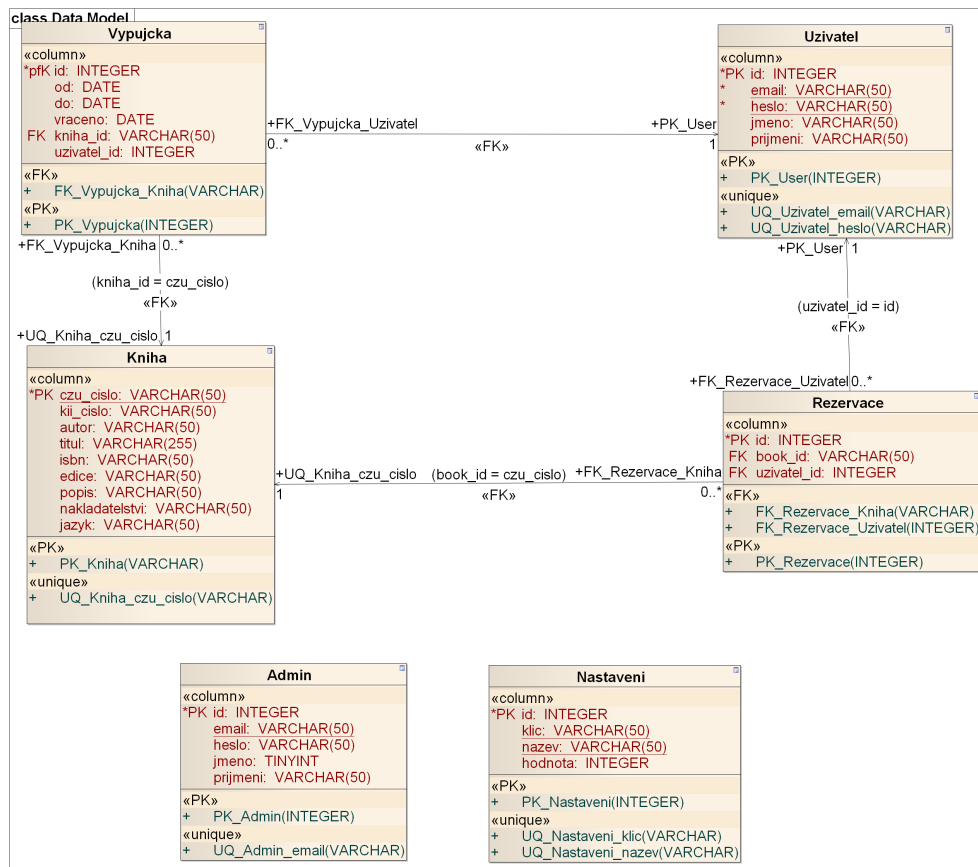
5.2.4 Výpůjčka

Tato třída je určena jak pro aktuální výpůjčky, tak pro jejich historii. Vrácená výpůjčka má vyplněné časové razítko u atributu „Datum vrácení“. Knihovník má možnost v aplikaci nastavit časový interval, po jehož uplynutí se historie výpůjček odstraní.

5.3 Databázový model

Databázový model 5.2 popisuje uložení dat v relační databázi. Databázový model je vytvořen podle jednotlivých entit doménového modelu. U každé entity jsou automaticky ukládána časová razítka vytvoření a poslední úpravy.

5. NÁVRH



Obrázek 5.2: Databázový model

Realizace

Tato kapitola obsahuje nejdůležitější části aplikace, způsob jejich řešení a jaké knihovny pro ně byly vybrány. V kapitole je také stručně shrnuto, jak aplikace funguje.

6.1 Autentizace a autorizace

Autentizace a autorizace uživatelů je v aplikaci řešena pomocí knihovny Devise. Devise podporuje rozlišování více typů uživatelů, v případě knihovny KII PEF ČZU tedy čtenáře a knihovníka.

Devise se skládá z několika modulů a v aplikaci jsou použity následující:

- **Autentizace pomocí databáze**
Šifruje a ukládá hesla do databáze pro ověření uživatelů při přihlášení.
- **Registrace**
Modul poskytuje registrační formulář, umožňuje editovat uživatelské údaje a zrušit účet.
- **Potvrzení**
Modul zajišťuje odeslání e-mailu s instrukcemi pro potvrzení účtu po zaregistrování nového uživatele.
- **Obnovení**
Umožňuje resetovat uživatelské heslo. Při vyžádání obnovy odešle aplikace na zadaný e-mail instrukce pro reset hesla.
- **Stopování**
Ukládá do databáze časové razítko, IP adresu a počet přihlášení konkrétního uživatele.
- **Zapamatování**
Umožňuje zapamatování přihlášení uživatele do aplikace.

- **Validace**

Umožňuje kontrolovat emailovou adresu a heslo. [25]

6.2 Získání knihovního fondu katedry

6.2.1 Interakce se systémem Aleph

Komunikace se systémem Aleph, ve kterém je uložen knihovní fond katedry, je řešena pomocí rozhraní X-Server. X-Server je součástí vícevrstvé architektury systému Aleph, která lze rozdělit na prezentační, aplikační a datovou vrstvu. 6.1

Datová vrstva zajišťuje přístup do databáze a poskytuje data vyšší vrstvě. Datová vrstva se skládá z komponenty RDBMS, nebo-li Relational DataBase Management System. Jedná se o databázový server, který spravuje data a jejich integritu. Vstupně-výstupní komponenta řídí komunikaci mezi aplikační a datovou vrstvou.

Aplikační vrstva poskytuje data prezentační vrstvě. Skládá se z komponenty, která zajišťuje zpracování dat, a rozhraní X-Server. X-Server se stará o přijetí požadavků a odeslání odpovědí. Odpovědi X-Serveru jsou zasílané ve formátu XML. [8]

Interakce se systémem Aleph probíhá pomocí HTTPS (Hypertext Transfer Protocol Secure) protokolu, jelikož požadavky obsahují citlivé údaje. Protokol HTTPS přenášená data šifruje a ověřuje identitu komunikujících stran.

Při přístupu do knihovního systému Aleph je v každém požadavku vyžadováno uživatelské jméno a heslo. To bylo aplikaci poskytnuto od informačního oddělení ČZU.

6.2.2 Požadavek find

V prvním kroku komunikace se systémem Aleph je zaslán požadavek find, ze kterého jsou získány informace potřebné pro požadavek present. Těmi jsou parametry `set_number` a `no_records`. `Set_number` určuje množinu záznamů knihovny dané katedry a `no_records` udává počet nalezených záznamů.

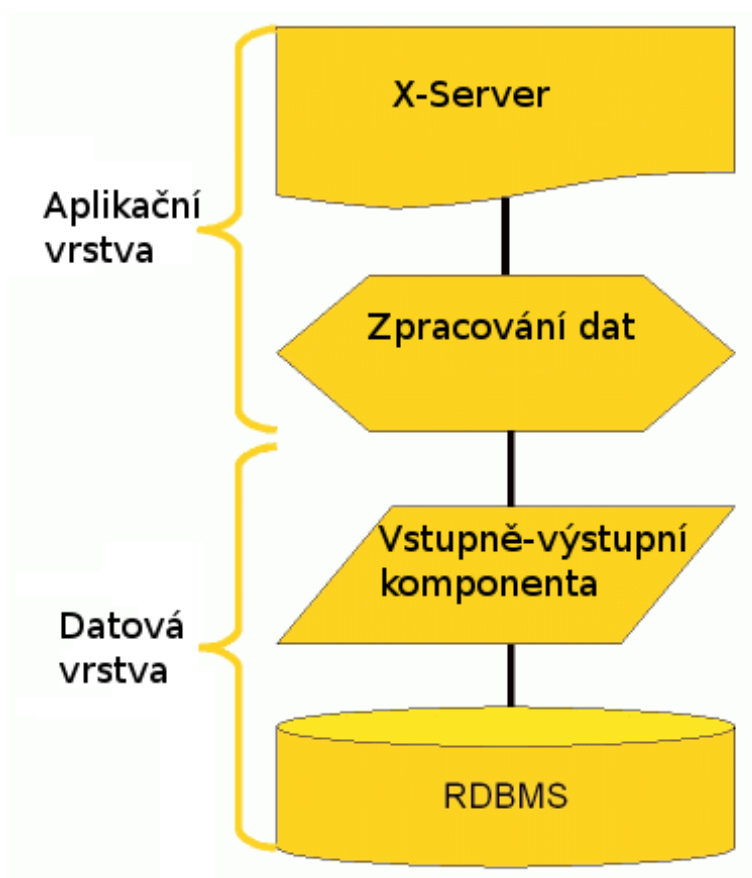
Vstupní parametry požadavku:

- **Base**

V parametru `base` se specifikuje knihovní katalog, do kterého se přistupuje.

- **Code**

`Code` slouží pro určení indexu, ve kterém se má hledat. Pro knihovny kateder je určen kód WSL (Sub-Library).



Obrázek 6.1: Aplikační a datová vrstva produktu Aleph [8]

- **Request**
Zde se vkládá hledaný výraz. Pro identifikaci konkrétní katedry se do parametru vloží její číselný kód.

6.2.3 Požadavek present

Služba present poskytuje bibliografické údaje knihovního fondu ve formátu OAI XML.

Vstupními parametry tohoto dotazu jsou:

- **Set_number**
V set_number se specifikuje množina záznamů, která má být prohledána. Tento parametr byl získán v předchozím požadavku find. Jednotlivé záznamy jsou v hledané množině očíslovány od 0 do počtu záznamů.
- **Set_entry**
V parametru se definuje rozmezí očíslovaných záznamů, které mají být

poslány (tzn. že lze poslat například všechny záznamy od čísla 80 do čísla 120). Úskalím tohoto parametru je, že dovoluje rozmezí maximálně pro sto záznamů.

Knihovní fond KII PEF aktuálně obsahuje téměř 400 záznamů. Požadavek `present` není tedy pro získání knihovního fondu úplně vhodný, jelikož neumožňuje zaslat více než sto záznamů v jedné odpovědi. X-Server ovšem nenabízí lepší možnost, proto musí aplikace odeslat požadavek `present` tolikrát, aby byl získán celý knihovní fond katedry. To je možné udělat díky získanému parametru `no_records`, ze kterého se v aplikaci vypočítá potřebný počet volání služby `present` a nastaví parametr `set_entry`. Takto se po částech stáhne celý fond katedry.

6.2.4 Sloučení dat

Po skončení volání požadavků `present` je knihovní fond uložen v několika souborech, které je nutné sloučit. Data jsou ve formátu XML, lze je tedy sloučit pomocí XSLT šablony. XSLT, nebo-li eXtensible Stylesheet Language Transformations, umožňuje převést vstupní XML soubor na jiný XML nebo HTML soubor.

Pro zpracování stažených souborů je použita XSLT šablona od Olivera Beckera z Humboldt University Berlin [26]. Ta umí sloučit dva XML soubory do jednoho. Na soubory je šablona aplikována externím nástrojem `saxon-xslt`, který je z aplikace spouštěn pomocí skriptu.

Šablona je určena pro sloučení dvou souborů. Proto je aplikována nejprve na první dva, a poté se k nim slučují popořadě zbylé soubory. Výsledný soubor tak obsahuje celý knihovní fond.

6.2.5 Konverze formátu

Záznamy získané ze systému Aleph jsou ve formátu OAI XML. Ten je dnes již nahrazen formátem MARC XML a ve většině knihoven přestal být podporován. Knihovna Ruby-Marc [27], která umožňuje číst, zapisovat a vyhledávat bibliografické údaje v textové databázi standardu MARC, soubory OAI XML neumí přečíst. Formát se však příliš neliší od MARC XML, se kterým lze pracovat, a proto je provedena konverze na tento formát.

Pro migraci OAI na XML MARC existuje XSLT šablona poskytnutá organizací Library of Congress. Tato šablona však nefungovala správně na soubory získané ze systému Aleph.

Pro převod na požadovaný formát byla tedy zvolena jiná cesta a to použití ruby knihovny Nokogiri, která je zaměřena na práci s XML soubory. Pomocí CSS nebo XPath selektorů lze vyhledávat jednotlivé uzly a editovat je. [28]

Při převodu je do původního souboru OAI vložen nový kořenový uzel „collection” s příslušnými atributy, který je vyžadován pro správné čtení souboru. Následně jsou všechny názvy elementů a atributů v OAI XML přepsány

Tabulka 6.1: Konverze OAI XML na MARC XML

Původní název uzlu/atributu	Nový název
fixfield s atributem „LDR”	leader
fixfield	controlfield
varfield	datafield
tag	id
ind1	i1
ind2	i2
code	label

podle formátu MARC XML. Jejich konverze jsou popsány v tabulce 6.1. Tímto je získán požadovaný formát MARC XML, ze kterého lze číst bibliografické údaje.

6.2.6 Požadavek item-data

Posledním krokem pro získání kompletních dat je zjistit, jaký počet výtisků od každé knihy vlastní knihovna KII PEF ČZU. Tato informace se dá ze systému Aleph získat pomocí požadavku item-data. Odpověď obsahuje informace o všech výtiscích konkrétního záznamu. Jednotky patřící KII PEF ČZU se dají vyhledat podle uzlů „collection”. Ty obsahují kód katedry, která je vlastní.

Vstupní parametry požadavku:

- **Base**
V parametru base se specifikuje knihovní katalog, do kterého se přistupuje.
- **Doc_number**
Parametr doc_number specifikuje konkrétní záznam z knihovního fondu a to pomocí unikátního identifikačního čísla záznamu v rámci ČZU.

Tento požadavek musí být odeslán pro každý záznam z knihovního fondu, proto je metoda pro získání počtu jednotek v knihovním fondu časově náročnější. Získané údaje jsou zapisovány do souboru s knihovním fondem katedry.

6.2.7 Převedení na relační databázi

Bibliografické údaje uložené v textové databázi je možné načítat do aplikace pomocí nástroje Ruby-Marc [27]. Ovšem v aplikaci je často potřeba načíst bibliografické údaje konkrétního záznamu a jeho vyhledávání v textové databázi nástrojem Ruby-Marc je pomalé.

Za účelem zrychlit práci s daty jsou vybrané bibliografické údaje ze souboru převedy do databáze MySQL. Při převodu jsou všechny záznamy v textové

databázi postupně čteny nástrojem Ruby-Marc a ukládány do relační databáze.

Pokud v databázi již záznam existuje, tak se aktualizují bibliografické údaje kromě atributu, který obsahuje evidenční číslo KII PEF. Tento údaj neexistuje v knihovním fondu vedeném na ČZU. Je veden pouze v rámci katedry a je možné ho v aplikaci k záznamům přidávat.

6.3 Automatické spouštění úloh

Automatické zasílání upomínek je v aplikaci řešeno knihovnou Whenever [29], která poskytuje jednoduchou syntaxi pro psaní CRON úloh. CRON je unixový systémový nástroj, který spouští předdefinované úkoly v naplanovaných časových intervalech.

V aplikaci jsou naplánované dvě automatické úlohy. Jedna spouští metodu pro mazání historie výpůjček v databázi. Časový interval pro mazání je nastaven knihovníkem přímo v aplikaci.

Druhá úloha kontroluje data pro vrácení výpůjček u všech čtenářů. Jestliže se blíží datum, kdy má být kniha vrácena do knihovny, odešle aplikace čtenáři email, pokud má povolené automatické upomínky. Email obsahuje seznam výpůjček pro vrácení do knihovny.

6.4 Jednoduchý vyhledávací formulář

Záznamy v knihovním fondu lze vyhledávat pomocí titulu, autora nebo ISBN knihy. Pro vyhledávání je v aplikaci využito regulárních výrazů. Regulární výraz představuje určitý vzor pro textové řetězce.

Z řetězce, který obsahuje dotaz, jsou nejdříve odstraněny nadbytečné mezery, čárky a jiné rušící znaky. Poté je každý záznam z knihovního fondu (dle dotazu buď titul, autor nebo ISBN) porovnáván, zda se v něm nachází hledaný řetězec. Tento jednoduchý způsob vyhledávání neumožňuje vyhledávat podle klíčových slov nebo řadit výsledky podle největší shody.

6.5 Adresářová struktura aplikace

Tato kapitola obsahuje výčet nejdůležitějších souborů a složek, ze kterých se aplikace skládá, a jejich stručný popis.

- **Gemfile** - soubor obsahuje všechny podpůrné knihovny z RubyGems používané v aplikaci.

- **app**
 - **assets** - pod složkami javascripts a stylesheets se zde nachází soubory application.js a application.css.sass. Tyto soubory obsahují seznam všech CSS nebo JS (JavaScript) modulů a souborů, které mají být naimportovány do aplikace. Vlastní nadefinované styly pro aplikaci se nachází v souboru customize.css pod složkou stylesheets.
 - **controllers** - pro každou entitu je zde vytvořen jeden soubor reprezentující řadič. Tyto soubory obsahují akce, které volají metody v modelech a odesílají výsledky na výstup uživateli.
 - **helpers** - soubory ve složce helpers obsahují pomocné metody, které lze sdílet navzájem mezi entitami.
 - **mailers** - zde se nachází soubory s akcemi, které řídí zasílání e-mailových zpráv z aplikace. V souboru admin_mailer.rb je možné nastavit email, na který si přeje knihovník zasílat upomínky.
 - **models** - zde se nachází jádro aplikace. Pro každou třídu je zde vytvořen jeden soubor, který obsahuje logiku aplikace a práci s databází.
 - **views** - pro každou entitu zde existuje jedna složka s HTML šablonami, které zobrazují uživateli výstup aplikace. Společná pro všechny entity je složka layouts. Ta definuje základní rozložení aplikace, tedy záhlaví, menu, obsah a zápatí. Do obsahu nadefinovaném ve složce layouts se pak generují šablony jednotlivých entit.
- **bin** - ve složce bin jsou uloženy skripty, které jsou připraveny pro instalaci a nasazení této webové aplikace.
- **config** - zde se nachází konfigurační soubory, které je možné v aplikaci nastavit.
 - **application.yml** - do application.yml se zadávají přihlašovací údaje pro email, ze kterého jsou z aplikace posílány zprávy uživatelům.
 - **database.yml** - tento soubor obsahuje nastavení databáze pro vývojové, testovací a produkční prostředí.
 - **routes.rb** - zde se nachází definice všech cest v aplikaci. Ty jsou převáděny v prohlížeči na URL adresy a naopak z prohlížeče jsou skrze ně zasílány požadavky do aplikace.
 - **schedule.rb** - soubor obsahuje rozvrh pro plánování automatických úloh CRON.
 - **locales** - tato složka obsahuje soubory s překlady chybových hlášek a oznámení z anglického do českého jazyka.

- **environments** - ve složce environments se nachází konfigurační nastavení pro vývojové, produkční a testovací prostředí. Je zde například nastaven emailový účet pro zasílání zpráv z aplikace.
- **data** - složka data obsahuje textovou databázi s knihovním fondem KII PEF v souboru fond.xml. Do této složky se také ukládají stažené soubory ze systému Aleph, které se po zpracování smažou.
- **db** - složka db obsahuje potřebné soubory pro inicializaci relační databáze. V souboru seeds.rb jsou zapsaná inicializační data pro nastavení aplikace a účet knihovníka. Dále složka obsahuje soubory s jednotlivými migracemi, které slouží pro vytvoření databázové struktury a jednotlivých tabulek.
- **features** - tato složka je vygenerována nástrojem pro akceptační testování Cucumber. Soubory s formátem feature obsahují jednotlivé scénáře testování. Ve složce support se nachází konfigurační soubory pro nastavení nástroje Cucumber. Složka step_definitions obsahuje konkrétní metody testování, které jsou aplikovány na kroky scénářů.
- **lib** - ve složce se nachází bezpečnostní certifikát rapid_czu.crt, který je importován při komunikaci se systémem Aleph do https protokolu.
- **public** - ve složce public jsou uloženy obrázky použité v aplikaci.
- **script** - v této složce se nachází skripty, které jsou použity pro slučování souborů při stažení knihovního fondu katedry.
- **test** - tato složka je určena pro jednotkové a integrační testování aplikace. Obsahuje složku fixtures, ve které jsou definovány testovací data.

6.6 Fungování aplikace

6.6.1 Účet knihovníka

Pro uživatelskou roli knihovníka není v aplikaci podporována registrace. Účet knihovníka je předpřipraven v souboru db/seeds.rb, kam se před prvním spuštěním aplikace vloží uživatelské údaje správce KII PEF. Těmi jsou email a heslo. Spuštěním tohoto souboru se účet uloží do databáze.

Jakmile má knihovník vytvořen svůj účet v databázi, může se přihlásit a uživatelské údaje lze již editovat přímo v aplikaci.

6.6.2 Přihlášení

Aplikace podporuje dvě uživatelské role, čtenáře a knihovníka. Proto jsou v aplikaci dva různé formuláře pro přihlášení, každý pro danou uživatelskou roli.

V menu aplikace se nachází odkaz pro přihlášení čtenáře a v přihlašovacím okně pro čtenáře lze formulář převést na okno pro knihovníka pomocí odkazu.

6.6.3 Správa čtenářů

Knihovník má v aplikaci možnost spravovat konta jednotlivých čtenářů, smí tedy editovat jejich uživatelské údaje nebo zrušit čtenářské konto.

Při registraci nového čtenáře aplikace knihovníka upozorní zasláním emailové zprávy. Registraci knihovník v aplikaci buď schválí, nebo zamítne.

6.6.4 Výpůjčky

Vypůjčovat knihy smí v aplikaci pouze knihovník. Konkrétní titul knihovník nalezne nejspíše pomocí vyhledávacího formuláře nebo v přehledu rezervací. Pokud je kniha dostupná pro vypůjčení, otevře knihovník výpůjční formulář, vybere konkrétního čtenáře a uloží výpůjčku.

Knihovník si v aplikaci může zobrazit seznam aktuálních i vrácených výpůjček. Výpůjčky lze mezi čtenáři převádět a to tak, že knihovník u výpůjčky konkrétního čtenáře zvolí možnost předat tuto výpůjčku na jiného čtenáře, kterého vybere ze seznamu. Výpůjčka se tak prvnímu čtenáři označí jako vrácená a připíše se zvolenému čtenáři.

Čtenář i knihovník může výpůjčky pomocí aplikace prodloužovat. Počet dní, o který lze prodloužit, a maximální dobu vypůjčení nastavuje knihovník v aplikaci.

6.6.5 Rezervace

Knihu lze rezervovat čtenářem i knihovníkem. Pokud knihovník bude chtít rezervovanou knihu vypůjčit, bude ve výpůjčním protokolu upozorněn, že jsou na knihu rezervace čtenářů.

6.6.6 Správa knihovního fondu

Textová databáze s knihovním fondem je v aplikaci uložena v souboru fond.xml ve složce data. Databázi může knihovník aktualizovat v nastavení aplikace volbou stažení dat ze systému Aleph.

Pro převod dat z textové databáze do relační existuje v nastavení samostatná volba. Převod dat je oddělený od stažení bibliografických údajů, jelikož tato akce může být potřeba, aniž by musel být aktualizován knihovní fond.

K jednotlivým záznamům lze v aplikaci přidat evidenční číslo vedené v rámci knihovny. Buď označením titulů v katalogu knih a otevřením hromadného formuláře, ve kterém se editují všechny vybrané knihy. Nebo lze evidenční číslo přidat při zobrazení bibliografických údajů knihy. Evidenční číslo KII PEF je uloženo pouze v relační databázi. Knihovník má také možnost knihu z databáze odebrat.

6.7 Nasazení aplikace

Jedním z nefunkčních požadavků na aplikaci je nasazení do produkčního prostředí s webovým serverem Apache. Apache není v Ruby on Rails standardně podporován, proto je třeba doinstalovat modul Phusion Passenger.

Tento modul lze integrovat do nastavení serveru Apache, na kterém je následně možné spustit aplikaci napsanou v jazyce Rails. Aplikační server Phusion Passenger je navržen tak, aby byl rychlý, stabilní, spolehlivý a bylo snadné ho použít. [30]

Testování

Funkčnost aplikace je ověřena testováním jednotek a akceptačním testováním pomocí předpřipravených dat.

7.1 Testovací databáze

Pro testování aplikace slouží databáze, která je naplněna ukázkovými daty. Tato data jsou zapsaná v tzv. fixtures ve formátu YAML (Ain't Markup Language), díky kterému jsou pro člověka snadno čitelná. Pro každou entitu je v aplikaci vytvořen jeden soubor fixtures.

Před spuštěním automatických testů je nutné testovací databázi vytvořit a naplnit pomocí předpřipraveného skriptu:

```
$ ./bin/prepare_test_database.sh
```

7.2 Testování jednotek

Testování jednotek ověřuje jednotlivé metody a akce tříd. U modelů se porovnává, zda metody správně zpracují vstupní data a vrací očekávané výsledky. U řadičů se testuje přesměrování na správné stránky a volání správných chybových hlášek nebo oznámení. Podporu pro psaní těchto testů nabízí přímo framework Ruby on Rails.

V příkazové řádce je možné spustit vybraný test příkazy:

```
$ rake test test/controllers/<test>
$ rake test test/models/<test>
```

Nebo lze spustit všechny napsané jednotkové testy pomocí:

```
$ rake test
```

7.3 Akceptační testování

Akceptační testování spočívá v přípravě jednotlivých scénářů, které jsou vyrobené podle specifikace zadavatele. Tyto scénáře jsou pak v aplikaci prováděny buď týmem testerů nebo automatickým nástrojem. [31]

Akceptační testování je v aplikaci zautomatizováno nástroji Cucumber [32] a Capybara [33]. Cucumber umožňuje psát testované scénáře prostým textem a jsou tak snadno čitelné pro každého. Na jednotlivé věty scénářů jsou pak volány akce, které jsou napsané pomocí knihovny Capybara, a vykonávají nadefinované úkony v aplikaci.

V příkazové řádce je možné spustit buď konkrétní množinu scénářů a nebo otestovat celou aplikaci.

7.3.1 Testované scénáře

V následujících podkapitolách jsou stručně napsány nejdůležitější body, které jsou ve scénářích testovány.

7.3.1.1 Správa knihovního fondu

- Úspěšné uložení evidenčních čísel v rámci katedry při editaci vybraných záznamů.
- Validace při otevření formuláře pro editaci, zda byl vybrán alespoň jeden záznam.
- Správné zobrazování bibliografických údajů, počtu výtisků od každé knihy a informace o dostupnosti záznamů (k vypůjčení, rezervována, vypůjčená).

7.3.1.2 Správa výpůjček

- Vytvoření výpůjčky konkrétnímu čtenáři.
- Předání výpůjčky jednoho čtenáře na jiného čtenáře v přehledu výpůjček.
- Validace formuláře pro výpůjčku.
- Prodloužení výpůjčky čtenáře.
- Potvrzení vrácení výpůjčky čtenáře.
- Správné zobrazování dostupných jednotek výtisků.

7.3.1.3 Správa rezervací

- Úspěšné rezervování výtisku.
- Správné zobrazení přehledu o rezervacích.

7.3.1.4 Vyhledávací formulář

- Nalezení správného výtisku při vyhledávání podle titulu knihy. Je testováno hledání podle celého názvu knihy a podle jeho podřetězců.
- Vyhledání knihy podle jejího autora. Jsou testovány kombinace se jménem i příjmením (nezáleží na pořadí) a pouze podle jména nebo příjmení.
- Nalezení knihy podle jejího ISBN.

7.3.1.5 Přihlášení uživatele

- Úspěšné přihlášení čtenáře do aplikace.
- Validace přihlašovacího formuláře.

7.3.1.6 Registrace uživatele

- Úspěšná registrace uživatele do aplikace.
- Odeslání e-mailu s pokyny pro potvrzení účtu čtenáři.
- Odeslání informativního e-mailu knihovníkovi o registraci nového čtenáře.
- Schválení účtu zaregistrovaného čtenáře.
- Zaslání informativního e-mailu čtenáři o schválení.
- Validace registračního formuláře - email, délka hesla a správné potvrzení hesla.

7.3.1.7 Správa čtenářů

- Editace čtenářského konta knihovníkem.
- Správné informace v přehledu výpůjček čtenáře.

7.3.1.8 Nastavení aplikace

- Vyplnění nastavení a jeho úspěšné uložení do databáze.

7. TESTOVÁNÍ

```
features/step_definitions/sinatra_steps.rb:62

5 scenarios (5 passed)
45 steps (45 passed)
0m2.369s
adela@adela-LIFEBOOK-UH552:~/skola/BAP/library$ cucumber features/sign_up.feature
Using the default profile...
Feature: Viewer signs up for the newsletter
  In order to sign up
  As a viewer
  I want to be able to sign up into application

  Scenario: View form page # features/sign_up.feature:6
    Given I am on "/" # features/step_definitions/sinatra_steps.rb:6
    And I click "Registrovat" # features/step_definitions/sinatra_steps.rb:84
    Then I should see "Registrovat" button # features/step_definitions/sinatra_steps.rb:70

  Scenario: Successful fill out sign up form #
  features/sign_up.feature:11
    Given I am on "/" #
  features/step_definitions/sinatra_steps.rb:6
    And I click "Registrovat" #
  features/step_definitions/sinatra_steps.rb:84
    When I fill in "user[name]" with "Jan" #
  features/step_definitions/sinatra_steps.rb:80
    When I fill in "user[surname]" with "Novák" #
  features/step_definitions/sinatra_steps.rb:80
    And I fill in "user[email]" with "jan@novak.cz" #
  features/step_definitions/sinatra_steps.rb:80
    And I fill in "user[password]" with "heslo123" #
  features/step_definitions/sinatra_steps.rb:80
    And I fill in "user[password_confirmation]" with "heslo123" #
  features/step_definitions/sinatra_steps.rb:80
    And I click "Registrovat" #
  features/step_definitions/sinatra_steps.rb:84
    Then I should see "Vaše registrace byla úspěšná, Váš účet ale zatím nebyl schválen knihovníkem." #
  features/step_definitions/sinatra_steps.rb:62
    And "jan@novak.cz" receives an email with "Pokyny pro potvrzení" as the subject #
  features/step_definitions/email_steps.rb:1
    And "spravce@knihovny.cz" receives an email with "Do aplikace se zaregistroval nový ctenar" as the subject #
  features/step_definitions/email_steps.rb:1
    When I have logged in as admin #
  features/step_definitions/sinatra_steps.rb:36
    And I click "Čtenáři" #
  features/step_definitions/sinatra_steps.rb:84
    And I click on link "jan@novak.cz" #
  features/step_definitions/sinatra_steps.rb:10
    Then "jan@novak.cz" receives an email with "Schválení registrace" as the subject #
  features/step_definitions/email_steps.rb:1

  Scenario: Failed fill out sign up form - wrong email # features/sign_up.feature:30
    Given I am on "/" # features/step_definitions/sinatra_steps.rb:6
    And I click "Registrovat" # features/step_definitions/sinatra_steps.rb:84
    When I fill in "user[name]" with "Jan" # features/step_definitions/sinatra_steps.rb:80
    When I fill in "user[surname]" with "Novák" # features/step_definitions/sinatra_steps.rb:80
    And I fill in "user[email]" with "spatny_email" # features/step_definitions/sinatra_steps.rb:80
    And I fill in "user[password]" with "heslo123" # features/step_definitions/sinatra_steps.rb:80
    And I fill in "user[password_confirmation]" with "heslo123" # features/step_definitions/sinatra_steps.rb:80
    And I click "Registrovat" # features/step_definitions/sinatra_steps.rb:84
    Then I should see "Email není validní." # features/step_definitions/sinatra_steps.rb:62

  Scenario: Failed fill out sign up form - too short password # features/sign_up.feature:43
    Given I am on "/" # features/step_definitions/sinatr
  a_steps.rb:6
    And I click "Registrovat" # features/step_definitions/sinatr
  a_steps.rb:84
    When I fill in "user[name]" with "Jan" # features/step_definitions/sinatr
  a_steps.rb:80
    When I fill in "user[surname]" with "Novák" # features/step_definitions/sinatr
  a_steps.rb:80
    And I fill in "user[email]" with "jan@novak.cz" # features/step_definitions/sinatr
  a_steps.rb:80
    And I fill in "user[password]" with "kratke" # features/step_definitions/sinatr
  a_steps.rb:80
    And I fill in "user[password_confirmation]" with "kratke" # features/step_definitions/sinatr
  a_steps.rb:80
    And I click "Registrovat" # features/step_definitions/sinatr
  a_steps.rb:84
    Then I should see "Heslo je příliš krátké(musí obsahovat alespoň 8 znaků)" # features/step_definitions/sinatr
  a_steps.rb:62

  Scenario: Failed fill out sign up form - wrong password confirmation # features/sign_up.feature:55
    Given I am on "/" # features/step_definitions/sinatr
  a_steps.rb:6
    And I click "Registrovat" # features/step_definitions/sinatr
  a_steps.rb:84
    When I fill in "user[name]" with "Jan" # features/step_definitions/sinatr
  a_steps.rb:80
    When I fill in "user[surname]" with "Novák" # features/step_definitions/sinatr
  a_steps.rb:80
    And I fill in "user[email]" with "jan@novak.cz" # features/step_definitions/sinatr
  a_steps.rb:80
    And I fill in "user[password]" with "heslo123" # features/step_definitions/sinatr
  a_steps.rb:80
    And I fill in "user[password_confirmation]" with "spatne_zopakovane_heslo" # features/step_definitions/sinatr
  a_steps.rb:80
    And I click "Registrovat" # features/step_definitions/sinatr
  a_steps.rb:84
    Then I should see "Potvrzení hesla se neshoduje." # features/step_definitions/sinatr
  a_steps.rb:62

5 scenarios (5 passed)
45 steps (45 passed)
0m2.065s
```

Obrázek 7.1: Ukázka akceptačního testování na registraci čtenáře

7.4 Knihovní fond

Testování stažení knihovního fondu ze systému Aleph bylo provedeno manuálně. Při komunikaci se systémem Aleph je nutné mít povolený přístup a není tedy možné tuto funkčnost testovat z jakékoli IP adresy.

Při testování byla v sekci nastavení nejdříve zvolena možnost Stáhnout knihovní fond. Tato operace trvala přibližně tři minuty. Na systém Aleph se posílá téměř čtyřista požadavků (především kvůli získání počtu jednotek každého výtisku), které je nutné zpracovat a uložit. Po skončení operace stahování se v aplikaci zobrazila zpráva, že data byla úspěšně připravena. Data se nachází v souboru fond.xml, do kterého byly údaje staženy, a pak úspěšně převedeny do formátu MARC XML.

Převedení dat z textové databáze do relační je v aplikaci otestováno jednotkovým testem v modelu entity Book. Pro toto testování je připravena testovací textová databáze uložená ve složce /test/data. V testu je ověřen počet vytvořených záznamů v relační databázi a jejich odpovídající názvy.

Jako úspěšný výsledek tohoto testování je považováno naplnění databáze bibliografickými údaji, které se shodují s knihovním fondem katedry v systému Aleph, a že jejich množství odpovídá počtu záznamů ve fondu katedry v systému Aleph.

Závěr

Cílem práce bylo navrhnout, implementovat a otestovat webovou aplikaci pro správu knihovny KII PEF ČZU.

Nejdříve byla provedena analýza funkčních a nefunkčních požadavků zadavatele a existujících knihovnických systémů a webových katalogů. Z analýzy vyplynulo, že zkoumané knihovnické systémy nejsou vhodným řešením pro knihovnu KII PEF. Proto bylo navrženo vlastní řešení aplikace.

Aplikace je navržena pro podporu pouze základních knihovnických služeb, které jsou potřebné pro správu knihovny KII PEF. Nejedná se tedy o robustní aplikaci jako u knihovnických systémů, které byly analyzovány v rešeršní části.

Výsledkem praktické části je funkční webová aplikace, která slouží jako knihovnický systém i jako webový katalog. Podporuje základní knihovnickou agendu jako správu čtenářů, výpůjček, rezervací a knihovnického fondu. Specifikem aplikace je, že umožňuje stáhnout knihovnický fond katedry ze systému Aleph a jeho editaci pro přidání evidenčního čísla KII PEF. Funkčnost aplikace byla ověřena akceptačním a jednotkovým testováním.

Do budoucna je možné aplikaci rozšířit tak, aby byla využitelná i pro ostatní katedry ČZU. Takové rozšíření by bylo možné realizovat přidáním volby katedry při stahování knihovnického fondu ze systému Aleph. Dále lze v aplikaci vylepšit vyhledávací formulář, například o rozšířené vyhledávání a hledání podle klíčových slov a slovních spojení.

Literatura

- [1] Krajská knihovna v Pardubicích: *Katalogizace / Zpracování fondu / Služby / Krajská knihovna v Pardubicích* [online]. [cit. 2015-04-20]. Dostupné z: <http://kkpce.cz/cs/sluzby/zpracovani-fondu/katalogizace/>
- [2] LANius s.r.o.: *Akvizice* [online]. [cit. 2015-04-20]. Dostupné z: <http://www.kpsys.cz/verbis/index.php/cs/ct-menu-item-25>
- [3] LANius s.r.o.: *Nový WWW katalog OPAC 2.0* [online]. [cit. 2015-04-20]. Dostupné z: <http://www.lanius.cz/clavius/opac20.htm>
- [4] KRČMAŘOVÁ, G.: *Z39.50 - protokol z minulého století* [online]. Ikaros, [cit. 2015-04-20]. Dostupné z: <http://ikaros.cz/node/10880>
- [5] Library of Congress: *Understanding MARC Bibliographic: Parts 1 to 6* [online]. [cit. 2015-04-20]. Dostupné z: <http://www.loc.gov/marc/umb/um01to06.html>
- [6] Masarykova univerzita: *Pravidla a formáty / Sémantické aspekty katalogizace* [online]. [cit. 2015-04-20]. Dostupné z: <http://is.muni.cz/do/rect/el/estud/ff/js14/katalogizace/web/pages/09-pravidla-a-formaty.html>
- [7] Česká zemědělská univerzita v Praze: *Katalog ČZU - Úplné zobrazení záznamu* [online]. [cit. 2015-04-20]. Dostupné z: <https://aleph.czu.cz>
- [8] Ex Libris Ltd.: *ALEPH 17.01 Staff User's Guide* [online]. [cit. 2015-04-20]. Dostupné z: http://www.lib.cas.cz/podpora/data/20_WWW_OPAC.pdf
- [9] Ex Libris Ltd.: *Apis* [online]. [cit. 2015-04-20]. Dostupné z: <https://developers.exlibrisgroup.com/aleph/apis>

- [10] KP-SYS spol. s r.o.: *Verbis* [online]. [cit. 2015-04-19]. Dostupné z: <http://kpsys.cz/verbis/index.php/cs/ct-menu-item-16/ct-menu-item-17>
- [11] KP-SYS spol. s r.o.: *Portaro* [online]. [cit. 2015-04-20]. Dostupné z: <http://kpsys.cz/verbis/index.php/cs/ct-menu-item-16/ct-menu-item-35>
- [12] LANius s.r.o.: *Základní informace o Claviu* [online]. [cit. 2015-04-20]. Dostupné z: <http://www.lanius.cz/clavius/info.htm>
- [13] Computer Aided Technologies, s. r. o.: *Knihovní systém TabbyBook - Vše, co Vaše knihovna potřebuje pro efektivní práci. Výpůjčky, katalogizace, e-knihy* [online]. [cit. 2015-04-20]. Dostupné z: <http://www.tabbybook.cz/podrobne.php>
- [14] ŽABIČKOVÁ, P.: *Nejrozšířenější open source knihovní systémy a jejich použitelnost* [online]. Duha, [cit. 2015-04-20]. Dostupné z: <http://duha.mzk.cz/clanky/nejrozsirenejsi-open-source-knihovni-systemy-jejich-pouzitelnostp>
- [15] OPÁLKOVÁ, M.: *OPACy nové generace II – VIRGObeta a VuFind* [online]. Ikaros, [cit. 2015-04-20]. Dostupné z: <http://ikaros.cz/node/13288>
- [16] About.com: *What is Ruby?* [online]. [cit. 2015-04-23]. Dostupné z: <http://ruby.about.com/od/beginningruby/a/WhatIsRuby.htm>
- [17] VÁCLAVOVIČ, J.: *Skriptovací programovací jazyky* [online]. reboot.cz, [cit. 2015-04-24]. Dostupné z: <http://reboot.cz/howto/programovani/skriptovaci-programovaci-jazyky/articles.html?id=153>
- [18] HARTL, M.: *What is Ruby on Rails?* [online]. [cit. 2015-04-21]. Dostupné z: <https://www.railstutorial.org/book/beginning#sec-introduction>
- [19] KEHOE, D.: *Ruby on Rails Tutorial (3rd Ed.)* [online]. RailsApps Project, [cit. 2015-04-21]. Dostupné z: <http://railsapps.github.io/what-is-ruby-rails.html>
- [20] ČÁPKA, D.: *MVC architektura* [online]. ITnetwork, [cit. 2015-04-21]. Dostupné z: <http://www.itnetwork.cz/mvc-architektura-navrhovy-vzor>
- [21] W3Schools.com: *HTML Introduction* [online]. [cit. 2015-04-27]. Dostupné z: http://www.w3schools.com/html/html_intro.asp

-
- [22] W3Schools.com: *CSS Introduction* [online]. [cit. 2015-04-27]. Dostupné z: http://www.w3schools.com/css/css_intro.asp
- [23] Tutorial Republic: *What is Bootstrap 3 - A Brief Introduction* [online]. [cit. 2015-04-21]. Dostupné z: <http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-introduction.php>
- [24] Welling, L.; Thomson, L.: *MySQL Průvodce základy databázového systému*. CP Books, a.s, 2005.
- [25] Plataformatec: *plataformatec/devise* [online]. [cit. 2015-04-27]. Dostupné z: <https://github.com/plataformatec/devise>
- [26] BECKER, O.: *XSLT Stylesheets* [online]. [cit. 2015-04-27]. Dostupné z: <http://www2.informatik.hu-berlin.de/~obecker/XSLT/>
- [27] CLARKE, K.; DUEBER, B.; GROPE, W.; aj.: *Ruby-marc* [online]. [cit. 2015-04-27]. Dostupné z: <https://github.com/ruby-marc/ruby-marc>
- [28] PATTERSON, A.; DALESSIO, M.; NUTTER, C.: *Nokogiri* [online]. [cit. 2015-04-27]. Dostupné z: <https://github.com/sparklemotion/nokogiri>
- [29] MAKHMALI, J.: *Whenever* [online]. [cit. 2015-04-27]. Dostupné z: <https://github.com/javan>
- [30] Phusion: *Phusion Passenger users guide, Apache version* [online]. [cit. 2015-05-1]. Dostupné z: <https://www.phusionpassenger.com/documentation/UsersguideApache.html>
- [31] HLAVA, T.: *Akceptační testování* [online]. Testování softwaru | Mantra, [cit. 2015-05-1]. Dostupné z: <http://testovanisoftware.cz/tag/akceptacni-testovani/>
- [32] HELLESØY, A.; BLÖTE, D.; HOLLAND, R.: *Cucumber* [online]. Cucumber Ltd, [cit. 2015-05-1]. Dostupné z: <https://cukes.info/>
- [33] NICKLAS, J.: *Capybara* [online]. [cit. 2015-05-1]. Dostupné z: <https://github.com/jnicklas/capybara>

Seznam použitých zkratk

API Application Programming Interface

CSS Cascading Style Sheets

ČZU Česká zemědělská univerzita

DRY Dont Repeat Yourself

GPL General Public License

HTML HyperText Markup Language

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

IP Internet Protocol

ISBN International Standard Book Number

JS JavaScript

KII Katedra informačního inženýrství

MARC MACHine-Readable Cataloging

MVC Model-view-controller

OAI Open Archives Initiative

OAI-PMH Open Archives Initiative Protocol for Metadata Harvesting

OPAC Online Public Access Catalog

PEF Provozně ekonomická fakulta

RDBMS Relational DataBase Management System

A. SEZNAM POUŽITÝCH ZKRATEK

REST Representational State Transfer

URL Uniform Resource Locator

XML Extensible Markup Language

XSLT eXtensible Stylesheet Language Transformations

YAML Ain't Markup Language

Ukázka aplikace

B. UKÁZKA APLIKACE

The screenshot displays a library catalog page for the Faculty of Informatics at the Czech University of Agriculture in Prague. The page features a header with the university's logo and name, a navigation menu, and a search bar. The main content area shows the title 'An introduction to formal languages and automata / Peter Linz.' and a list of metadata fields including availability, edition count, ISBN, and publication details. At the bottom, there are buttons for 'Rezervovat' (Reserve) and 'Vypůjčit' (Borrow).

Knihovna Katedry informačního inženýrství
Provozně ekonomická fakulta, České zemědělské univerzity v Praze

Členství Vypůjčené knihy Rezervace Katalog knih Nastavení aplikace Kontakt admin Dana Vynikarová

Hledat v katalogu... Hledat

An introduction to formal languages and automata / Peter Linz.
Linz, Peter.

Dostupnost	K vypůjčení
Počet jednotek	1/1
Evidenční číslo	<input type="text" value="Zadejte ev. číslo"/> Editovat
Číslo záznamu	000063205
ISBN	9781449615529 (casebound)
Vydání	5th ed.
Nakladatelské údaje	Sudbury, MA : Jones & Bartlett Learning, c2012.
Žánr/Forma	monografie
Popis	xiii, 437 s. : il. ; 25 cm. +

Rezervovat Vypůjčit

Copyright © 2015 Knihovna Katedry informačního inženýrství Provozně ekonomické fakulty České zemědělské univerzity v Praze | All rights reserved

Obrázek B.1: Informace o titulu

Databáze knih


1 2 3 4 5 ...

Počet titulů na stránku

Označit vše	Evidenční číslo	Číslo záz.:	Autor	Titul	ISBN	Vydání	Žánr	Smazat
<input type="checkbox"/>		czu000041718	Ziková, Jana	Zoner Callisto :tipy, efekty, kouzla /Jana Ziková	80-7226-793-0 (brož.) ;	Vyd. 1.	příručky	<input type="button" value="x"/>
<input type="checkbox"/>		czu000043801	Zelinski, Ernie J.	Nebojte se nepracovat /Ernie J. Zelinski ; [z anglického originálu ... přeložil Pavel Šroubek]	80-7178-782-5 (brož.) ;	Vyd. 1.	populárně-naučné publikace	<input type="button" value="x"/>
<input type="checkbox"/>		000054007	Zeigler, Bernard P.	Modeling & simulation-based data engineering :introducing pragmatics into ontologies for net-centric information exchange /Bernard Zeigler, Philip Hammonds.	9780123725158		monografie	<input type="button" value="x"/>
<input type="checkbox"/>		000057981	Witten, I. H.	Data mining :practical machine learning tools and techniques /Ian H. Witten, Elbe Frank.	978-0-12-088407-0	2nd ed.	příručka	<input type="button" value="x"/>
<input type="checkbox"/>		000056208	Wieggers, Karl Eugene,	Požadavky na software /Karl E. Wieggers ; [překlad Tomáš Znamenáček]	978-80-251-1877-1 (brož.)	Vyd. 1.	příručky	<input type="button" value="x"/>

1 2 3 4 5 ...

Obrázek B.2: Katalog knih



Knihovna Katedry informačního inženýrství
Provozně ekonomická fakulta České zemědělské univerzity v Praze

Hledat v katalogu...

Hledat

Čtenáři

Vypůjčené knihy

Rezervace

Katalog knih

Nastavení aplikace

Kontakt

admin Dana Vynikarová

Aktuální výpůjčky čtenáře Adéla Blažková

Třídění	Titul	Autor	Od	Do	Předat výpůjčku dalšímu čtenáři	Prodloužit	Potvrdit vrácení
	Meditace o základech vědy /Petr Vopěnka	Vopěnka, Petr,	2015-05-06	2015-07-05	➔	🕒	✔

Historie výpůjček čtenáře

Třídění	Titul	Autor	Datum vypůjčení	Datum vrácení
	Smalltalk, objects, and design /Chamond Liu.	Liu, Chamond.	2015-05-06	2015-05-06

Rezervace čtenáře

Třídění	Titul	Autor	Datum rezervace	Vypůjčit	Zrušit
	1000 nejkrásnějších obrazů historie /Wendy Beckettová ; odborná konzultantka Patricia Wrightová ; [z anglického originálu ...; přeložili Jan Šliva, Eva Klimentová a Milan Jára]	Beckett, Wendy,	2015-05-05	➔	✖

Copyright © 2015 Knihovna Katedry informačního inženýrství Provozně ekonomické fakulty České zemědělské univerzity v Praze | All rights reserved

Obrázek B.3: Výpůjčky a rezervace čtenáře

Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	src	
	impl.....	zdrojové kódy implementace
	thesis.....	zdrojová forma práce ve formátu \LaTeX
	text.....	text práce
	thesis.pdf.....	text práce ve formátu PDF