

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Webový repozitář vědeckých prací



2016

Vedoucí práce: Mgr. Petr Osička,
Ph.D.

Ivoš Apolenář

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Ivoš Apolenář
Název práce: Webový repozitář vědeckých prací
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2016
Studijní obor: Aplikovaná informatika, prezenční forma
Vedoucí práce: Mgr. Petr Osička, Ph.D.
Počet stran: 44
Přílohy: 1 CD/DVD
Jazyk práce: český

Bibliographic info

Author: Ivoš Apolenář
Title: Web repository of scientific papers
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2016
Study field: Applied Computer Science, full-time form
Supervisor: Mgr. Petr Osička, Ph.D.
Page count: 44
Supplements: 1 CD/DVD
Thesis language: czech

Anotace

Práce pojednává o naprogramované aplikaci sloužící jako úložiště vědeckých publikací. Představuje její technické aspekty a popisuje řešení vybraných problémů při vývoji. Hlavní funkcí aplikace je automatické vyplnění metadat publikace při ukládání a export seznamu publikací do souboru ve formátu Bibtex.

Synopsis

Thesis deal about programmed application serving as repository of scientific papers. Presents its technical aspects and describing solution of selected problems at develop. Main function of application is automatic filling metadata of scientific paper at inserting and list of scientific papers export to Bibtex format.

Klíčová slova: repozitář vědeckých publikací, webová aplikace, JavaServer Faces

Keywords: repository of scientific papers; web application; JavaServer Faces

Děkuji Mgr. Petru Osičkovi, Ph.D. za rady a připomínky při vedení této práce. Dále chci poděkovat rodičům a všem, kteří mě po celou dobu studia podporovali.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Úvod	8
1.1	Vlastnosti implementovaného repozitáře	8
2	Aplikace s podobným účelem	10
2.1	Mendeley	10
2.2	JabRef	11
2.3	RefMe	11
3	Technická dokumentace	13
3.1	Použité technologie	13
3.1.1	Java	13
3.1.2	Java Server Faces	13
3.1.3	Apache Maven	15
3.1.4	Apache Tomcat	16
3.1.5	PostgreSQL	16
3.1.6	BibTex	16
3.1.7	Ostatní	16
3.2	Struktura projektu	17
3.2.1	Složka java	17
3.2.1.1	Annotation	17
3.2.1.2	Bean	18
3.2.1.3	Database	18
3.2.1.4	Develop	18
3.2.1.5	EnumType	18
3.2.1.6	Exception	18
3.2.1.7	Jsf	19
3.2.1.8	MetaDataExtraction	19
3.2.1.9	Servlet	19
3.2.1.10	Util	19
3.2.2	Složka webapp	19
3.3	Struktura databáze	20
3.4	Zabezpečení	21
3.5	Popis řešení	22
3.5.1	Vyplňování metadat	22
3.5.2	Export do formátu bibtex	24
3.5.3	Podobnost řetězců	24
3.5.3.1	Levenshteinova vzdálenost	25
3.5.3.2	Implementace	25

4	Uživatelská dokumentace	27
4.1	Přihlášení	27
4.2	Obnova hesla	27
4.3	Správa účtu	27
4.4	Tabulka zobrazující data	27
4.4.1	Tabulka pro publikace	28
4.4.1.1	Tlačítka s akcemi pro tabulku publikací	28
4.5	Repozitář	30
4.6	Publikace	30
4.6.1	Přidání	30
4.6.2	Přehled	30
4.6.3	Úprava	31
4.7	Seznamy publikací	32
4.7.1	Správa	32
4.7.2	Přehled	32
4.7.3	Úprava	32
4.8	Zobrazení autora nebo kolekce	33
4.9	Vyhledávání	33
4.10	Odhlášení	34
4.11	Administrace	34
4.11.1	Správa publikací	34
4.11.2	Přehled publikace	34
4.11.3	Správa autorů a kolekcí	34
4.11.4	Zobrazení autorů a kolekcí	34
4.11.5	Úprava autorů a kolekcí	35
4.11.6	Uživatelé	35
4.11.7	Vyhledávání	35
5	Rozšíření aplikace	36
	Závěr	37
	Conclusions	38
	A Nasazení a testování aplikace	39
	B Obsah přiloženého CD/DVD	42
	Literatura	43

Seznam obrázků

1	Webová verze aplikace Mendelej	10
2	Aplikace JabRef na platformě Windows	11
3	Webová verze aplikace RefMe	12
4	Grafické znázornění JSF architektury	14
5	Grafické znázornění životního cyklu JSF [7]	15
6	Základní struktura projektu, vygenerovaná Mavenem	17
7	Schéma databáze	21
8	Záhlaví aplikace po přihlášení uživatele	27
9	Tabulková tlačítka	28
10	Náhled publikace po kliknutí na tlačítko s ikonou lupy v tabulce	29
11	Tabulka publikací s vybranými publikacemi	29
12	Dialog pro výběr metadat při přidávání článku z časopisu	31
13	Dialogové okno s bibtexem aktuálně zobrazovaného seznamu	32
14	Ukázka vyhledávání publikací	33
15	Dialogové okno pro slučování autorů	35

Seznam vět

1	Definice (Mapa metadat)	22
---	-----------------------------------	----

Seznam zdrojových kódů

1	Ukázka vygenerovaného bibtextu pro seznam publikací	25
2	Výpočet procentuální podobnosti dvou řetězců	26

1 Úvod

Vědečtí pracovníci se neustále snaží svůj obor rozvíjet a posouvat kupředu. Vypracovávají vědecké publikace obsahující nové poznatky a zjištěné skutečnosti. V každé takové publikaci je potřeba uvést citace odkazující se na jiné publikace, ze kterých autor vycházel. Usnadnit a zefektivnit práci s těmito publikacemi mohou repozitáře vědeckých prací.

Uživatel repozitáře si může jednoduše udržovat všechny potřebné publikace na jednom místě. Díky tomu může publikace vyhledávat, třídit je podle vybraných kritérií apod.

Přidávat nové publikace do repozitáře může být v závislosti na množství ukládaných metadat¹ časově náročná a nezajímavá činnost. Proto ji chceme co nejvíce zjednodušit a urychlit. K tomu lze využít již existující repozitáře poskytující webové služby. Skrze tyto služby získat metadata vkládané publikace a ty nabídnout uživateli.

1.1 Vlastnosti implementovaného repozitáře

Repozitář implementovaný v rámci této práce je webová aplikace využívající open source technologií s podporou nasazení na unixových systémech². Aplikace rozlišuje 2 typy uživatelských účtů. Oba typy mají k dispozici následující funkcionalitu:

- Vkládání nových publikací s umožněním automatického vyplnění metadat.
- Vytváření a správu seznamů publikací.
- Vyhledávání v metadatech uložených publikací a komfortní práce s výsledky vyhledávání.
- Export seznamů publikací do formátu Bibtex³.

Funkcionalita administrátorského účtu je rozšířena o následující funkce:

- Registrace nebo odstranění běžného členského účtu.
- Správa publikací, autorů a kolekcí⁴. Např.: odstranění publikace z repozitáře nebo slučování autorů či kolekcí.

Aplikace podporuje 3 typy publikací s rozdílnými metadaty:

- **Článek v časopise:** titul, autoři, rok vydání, časopis, svazek, číslo, strany, DOI⁵, URL.

¹Metadata – data poskytující informace o jiných datech. Např.: titul, autoři knihy.

²Druh operačních systémů pro počítače.

³Formát pro sazbu použité literatury v systému LateX, popsán v kapitole 3.1.6.

⁴Termín používán pro souhrnné označení časopisů, sborníků a sérií podle nichž lze sdružovat publikace jednotlivých typů.

⁵Digital Object Identifier – identifikátor publikací v digitální podobě.

- **Článek na konferenci:** titul, autoři, rok vydání, sborník, strany, DOI, URL.
- **Kniha:** titul, autoři, rok vydání, série, vydavatel, svazek, ISBN⁶, DOI, URL.

Aplikace je uzavřená. Bez platných přihlašovacích údajů nelze do aplikace vstoupit a používat ji. Přihlášení členové mají k dispozici všechny publikace vložené do systému. Ty si mohou přidávat do soukromého repozitáře a pracovat s nimi. Pokud uživatel z nějakého důvodu publikaci už nepotřebuje, může ji ze soukromého repozitáře odebrat. Tímto krokem ovšem zapříčiní také odebrání publikace ze všech svých seznamů.

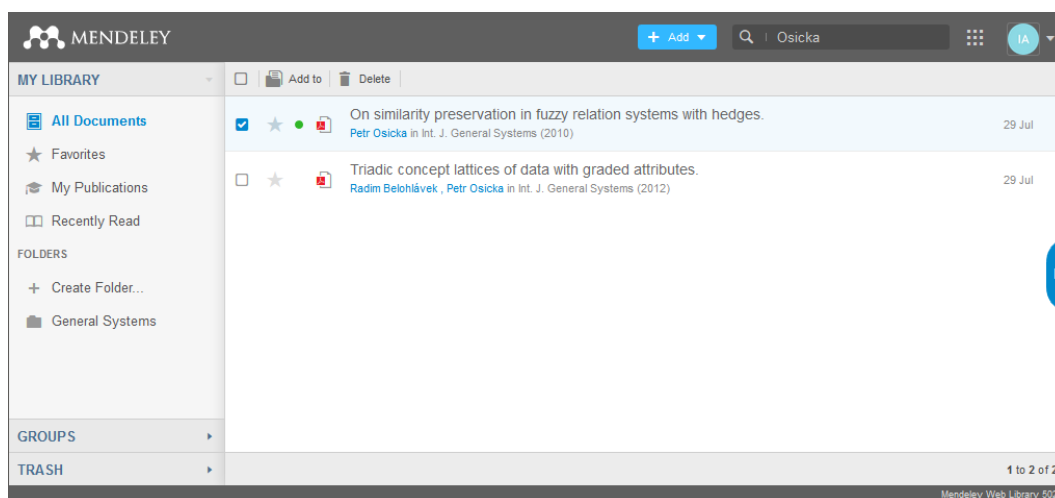
⁶International Standard Book Number – číselný kód jednoznačně identifikující knihu.

2 Aplikace s podobným účelem

2.1 Mendeley

Známý a oblíbený program pro správu a sdílení vědeckých publikací a jejich citací [1]. Jde o webovou aplikaci, avšak nabízí i aplikace pro desktopové (Windows, Mac OS, Linux) i mobilní (iOS, Android) platformy. V roce 2013 byl Mendeley koupen vydavatelstvím Elsevier, podle spekulací, za 65 milionů amerických dolarů. Nabízí placenou verzi i verzi zdarma. Mezi hlavní funkcionality patří:

- Automatické získání metadat z vloženého pdf souboru.
- Vyhledávání, filtrování a řazení publikací dle metadat.
- Export souboru ve formátu bibtex.
- Prohlížeč pdf souboru s rozšířenou funkcionalitou (poznámky, zvýrazňování určitých částí textu).
- Kategorizace prací do složek a označování publikací tagy.
- Statistiky publikací a autorů.
- Sdílení prací.



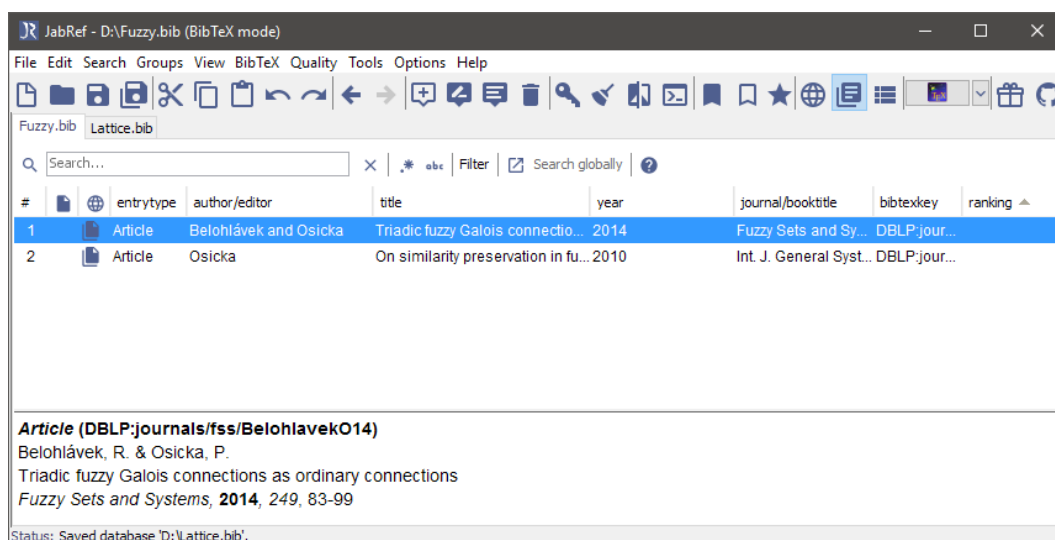
Obrázek 1: Webová verze aplikace Mendeley

Na obrázku 1 je zobrazena webová verze aplikace Mendeley – správa publikací. Vlevo je menu, kde lze ovlivnit zobrazení publikací podle kategorií. V menu je možné také vytvářet nové vlastní složky do kterých lze publikace třídit. Po kliknutí na publikaci se zobrazí detail se všemi informacemi, které lze upravovat. Tuto nabídku lze také zobrazit kliknutím na modré tlačítko při pravém okraji. V horním panelu je nabídka pro přidání nové publikace (souborem, manuálně), vyhledávání v metadatech publikace a správa účtu.

2.2 JabRef

Systém pro správu citací, který využívá bibtex jako základní formát [2], je k dispozici zdarma. Vývoj probíhá pro platformy Windows, Mac OS i Linux. Umožňuje vyhledávání ve webových externích databázích. Podporuje import i export v různých formátech. Dovoluje přidat vlastní bibtex položku k libovolnému typu publikace. Zajímavou funkcí je slučování publikací.

Na obrázku 2 je zobrazena spuštěná aplikace JabRef na platformě Windows. Jsou v ní otevřeny dvě bibtex databáze, mezi kterými je možné přepínat kartami pod panelem nástrojů. Panel nástrojů mimo běžné funkce obsahuje například funkce pro slučování publikací, přiřazení značky nebo ohodnocení publikace. Kliknutím na publikaci se v dolní části okna zobrazí její detail. Při dvojitým kliknutím je k dispozici úprava metadat publikace přepsáním políček formuláře, nebo přímo úpravou metadat v bibtex formátu.



Obrázek 2: Aplikace JabRef na platformě Windows

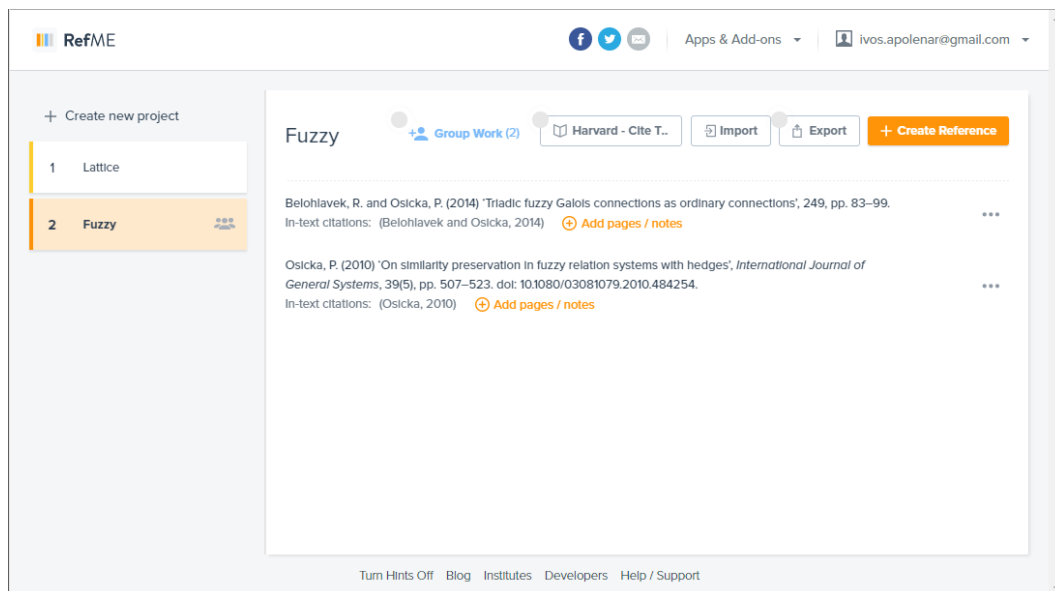
2.3 RefMe

V základní funkcionalitě je RefMe bezplatný systém pro správu citací [3]. Po uhrazení poplatku je možné používat bonusové funkce. Je k dispozici jako webová aplikace s klienty na iOS nebo OS Android. Je kompatibilní s jinými aplikacemi například s Mendeley. Umožňuje sdílení referencí v pracovních skupinách. Podporuje import ve formátu RIS⁷ a export do více formátů. Uživatel si může vybrat z více než 7500 citačních stylů.

Na obrázku 3 je zobrazena webová verze aplikace RefMe. V levém panelu lze přidávat a přepínat mezi projekty. Každý projekt má své vlastní publikace. Ty lze samozřejmě upravovat, přesouvat mezi projekty nebo exportovat. V panelu

⁷Citační formát pro výměnu citačních dat.

nad zobrazenými publikacemi je možné prostřednictvím mailu sdílet projekt s dalšími uživateli. Dále může uživatel nastavit citační styl, přičemž pro každý projekt lze přiřadit citační styl rozdílný.



Obrázek 3: Webová verze aplikace RefMe

3 Technická dokumentace

V této kapitole budou popsány technické rysy implementované aplikace. Seznámí čtenáře s použitými technologiemi, strukturou projektu a databáze. Na závěr budou popsány postupy řešení vybraných problémů při implementaci aplikace.

3.1 Použité technologie

Technologie musely být vybrány v souladu s požadavkem provozu aplikace na operačním systému unixového typu.

3.1.1 Java

Java [4] je objektově orientovaný programovací jazyk. Je vyvíjen firmou Sun Microsystems jako open source technologie. Dlouhodobě patří mezi nejpoužívanější jazyky vůbec. Používá silné typování se statickou kontrolou a podporuje automatickou správu paměti. Java se dělí na 3 edice:

- SE (Standard Edition) - umožňuje vývoj aplikací pro stolní počítače.
- EE (Enterprise Edition) - rozšíření Javy SE, je určena pro vývoj webových aplikací.
- ME (Micro Edition) - vývoj pro malá omezená zařízení jako jsou tiskárny, routery apod.

Díky JVM⁸ je platformově nezávislá a lze spustit na libovolné architektuře, pro kterou je k dispozici JVM. Java dodává aplikaci její dynamičnost. Java byla vybrána, protože jsem si v ní chtěl vyzkoušet naprogramovat webovou aplikaci.

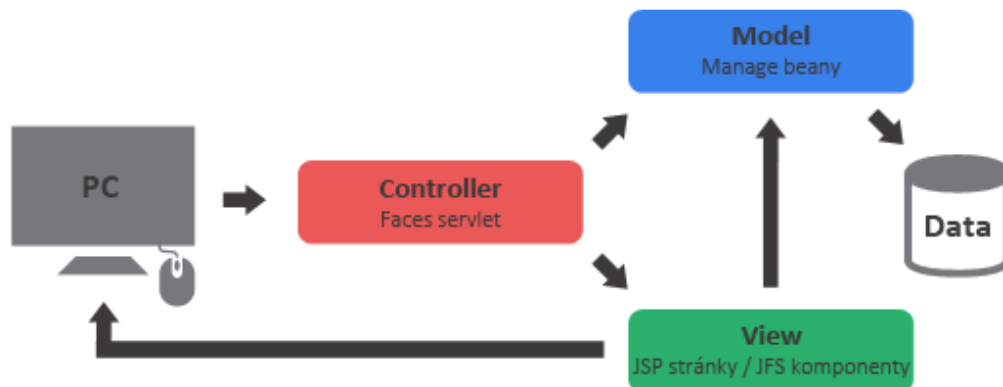
3.1.2 Java Server Faces

Java server Faces (JSF) je komponentově orientovaný framework⁹ pro vývoj webových stránek. Patří do specifikace Javy EE. JSF je založen na MVC (Model View Controller) architektuře [5], dělí aplikační logiku od prezentační, viz obrázek 4.

- Model – přístup k datům je realizován pomocí managed beanů, což jsou javovské třídy, poskytující přístup k zapouzdřeným datům pomocí tzv. getterů a setterů.
- View – neboli pohled, zajišťuje prezentaci dat uživateli. Používá JSF komponenty zapisované speciálními XML .

⁸Java Virtual Machine – software pro interpretaci Javy, poskytující abstrakci nad konkrétní počítačovou platformou.

⁹Sada hotových knihoven, řešící stereotypní úlohy při vývoji software.



Obrázek 4: Grafické znázornění JSF architektury

- Controller – faces servlet¹⁰ zpracovávající požadavky uživatele.

JSF projde od přijetí požadavku od uživatele a vrácením odpovědi 6 fázemi. Souhrnu těchto fází říkáme životní cyklus [6], viz obrázek 5. Nyní budou jednotlivé fáze stručně a zjednodušeně popsány.

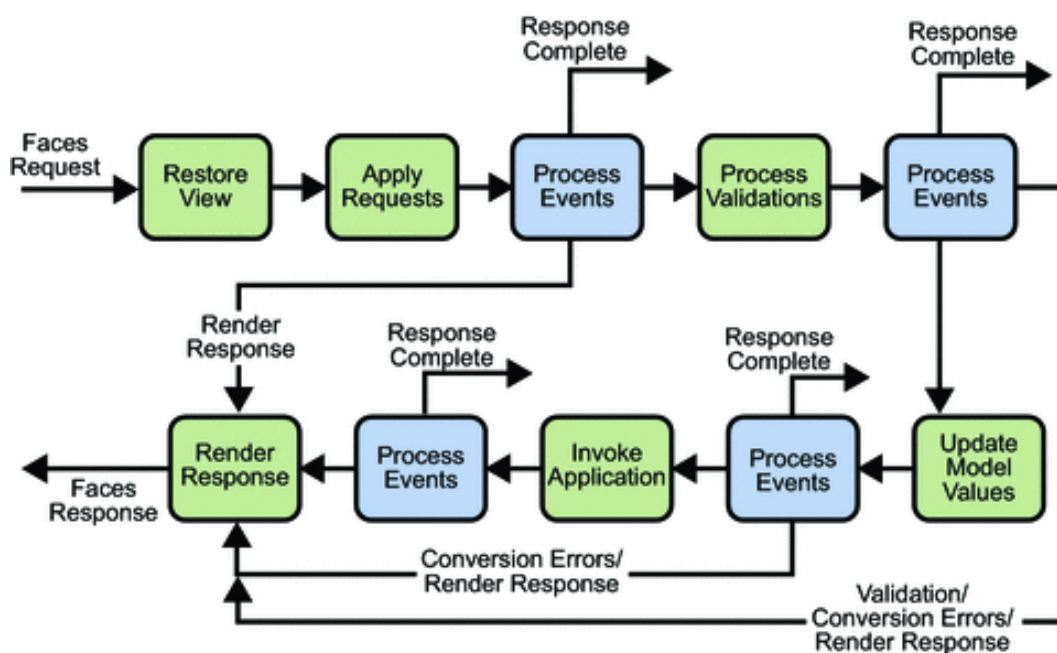
- **Restore view** – započne po přijetí požadavku faces servletem, který úvodní fází spustí. Je vytvořen a inicializován `FacesContext`, kde jsou uloženy informace potřebné pro zpracování požadavku. Je vytvořen prázdný pohled. Pokud se nejedná o postback¹¹ požadavek, přechází se rovnou k poslední fázi.
- **Apply request values** – parametry požadavku aktualizují hodnoty příslušných komponent (zatím si je drží lokálně samy komponenty). Parametry jsou v textové podobě, a proto dochází ke konverzi na vhodný datový typ. Pokud konverze selže, je vytvořena chybová zpráva. Ta je přidána do instance `FacesContextu`. Nakonec jsou vyvolané události zaslány svým listenerům¹².
- **Process validations** – jsou spuštěny validátory, které jsou přiřazeny ke komponentám. Validátory porovnávají hodnoty uložené lokálně v komponentách s validačními pravidly. Pokud validace selže, je vytvořena chybová zpráva a přidána do instance `FacesContextu`. Poté se přechází rovnou do poslední fáze a chyby jsou zobrazeny ve znovu vygenerovaném pohledu. Pokud validace proběhla bez chyb, jsou opět události předány svým listenerům.

¹⁰Třída jazyku Java zpracovávající HTTP požadavky.

¹¹Opakovaný požadavek na stejnou stránku, většinou v souvislosti s odesláním formuláře.

¹²Rozhraní zodpovědné za reagování na události.

- **Update model value** – nastává pokud předchozí fáze proběhla bez chyb. Nyní jsou již nastaveny příslušné modely (managed bean) a nakonec jsou události opět předány příslušným listenerům.
- **Invoke application** – v této fázi jsou zpracovány události jako například odeslání formuláře tlačítkem. Následně dochází k přesměrování na jiný pohled, případně aktualizování současného pohledu.
- **Render response** – zobrazení požadovaného pohledu a nebo zobrazení chybových zpráv.



Obrázek 5: Grafické znázornění životního cyklu JSF [7]

3.1.3 Apache Maven

Nástroj pro automatizaci a usnadnění sestavování aplikace [8]. Pomáhá vývojáři udržovat správnou strukturu projektu. Má sadu předpřipravených tzv. ačtyřpů, ze kterých si může vývojář vybrat a Maven poté vygeneruje základní a ověřenou strukturu projektu.

V Mavenu je možné si u projektu definovat závislosti na externích knihovnách. Maven tyto knihovny poté automaticky stáhne z veřejných repositářů a připojí k vašemu projektu. Vývojáři stačí, aby nadefinoval závislost pouze na knihovnu, kterou chce přímo používat a Maven se už o ostatní postará. Konfigurace maven projektu se provádí v konfiguračním souboru `pom.xml`.

Maven byl vybrán, jelikož je projekt závislý na velkém množství knihoven.

3.1.4 Apache Tomcat

Aplikační server pro Javu EE [9], který nesplňuje celou specifikaci a proto poskytuje pouze částečnou implementaci. V základu nepodporuje JSF. To lze vyřešit přidáním příslušných knihoven.

Apache Tomcat byl vybrán, protože je to open source projekt.

3.1.5 PostgreSQL

Relační databázový systém [10] vyvíjený primárně pro unixové systémy, nicméně existuje i pro Windows. Podporuje důležité prvky moderních databází jako například cizí klíče, transakční zpracování dat nebo také výčtový datový typ. K dotazování (jak napovídá název) používá dialekt dotazovacího jazyka SQL (Structured Query Language).

Výběr databázového systému nebyl nijak svazující, jelikož Java dokáže s použitím daného JDBC¹³ ovladače pracovat s mnoha databázemi. Pro správu databáze byl využit **pgAdmin** [11].

PostgreSQL byl vybrán díky tomu, že s ním mám zkušenosti a také proto, že je vydáván jako open source projekt.

3.1.6 BibTex

Nástroj pro správu a automatické generování použité literatury [12]. Typicky je užíván systémem LaTeX. Odděluje data od způsobu jejich prezentace. V aplikaci jsou používány tyto typy záznamů:

- **article** - článek z časopisu,
- **book** - kniha,
- **inproceedings** - článek z publikace.

Aplikace používá bibtex jako formát pro soubory exportovaných seznamů publikací.

3.1.7 Ostatní

- **Prime faces** - knihovna obsahující více než 100 komponent pro JSF. Podporuje Ajax a nabízí několik komunitně vytvořených vzhledů [13].
- **Spring JdbcTemplate** - součást frameworku Spring, poskytuje aparát pro zjednodušenou práci s databází [14].
- **Jsoup** - HTML a XML parser¹⁴ [15].
- **JSON-java** - knihovna pro zpracování JSON souborů [16].

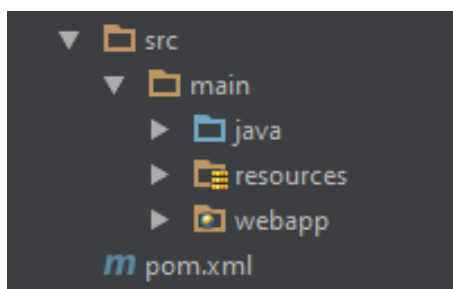
¹³Java Database Connectivity – rozhraní, které definuje jednotný přístup k databázím.

¹⁴Nástroj pro syntaktickou analýzu jazyka.

- **Apache Commons Lang** - knihovna pro práci s řetězci [17].
- **PrettyFaces** - knihovna pro mapování URL adres [18].
- **Apache PDFBox** - knihovna umožňující pohodlnou práci s pdf soubory [19].
- **Less** - preprocesor umožňující větší abstrakci při psaní kaskádových stylů [20].

3.2 Struktura projektu

Aplikace byla vyvíjena jako maven projekt, typu webová aplikace. V konfiguračním souboru `pom.xml` jsou zapsány informace pro sestavování aplikace a hlavně závislosti na externím knihovnách, které byly v projektu použity. Složka `src` obsahuje zdrojové kódy aplikace. Na obrázku 6 lze spatřit základní strukturu projektu. Složka `resources` obsahuje pouze jeden soubor – `base.less`, ve kterém je nadefinován vzhled aplikace za pomoci modifikovaného zápisu kaskádových stylů. Ty jsou poté do standardních kaskádových stylů překládány preprocesorem Less. Složky `java` a `webapp` jsou rozsáhlejší a budou popsány v následující kapitolách.



Obrázek 6: Základní struktura projektu, vygenerovaná Mavenem

3.2.1 Složka java

Obsahuje veškeré zdrojové kódy v jazyce Java. Třídy jsou rozděleny do balíčků, které zajišťují jejich logické uspořádání. Kořenovým balíčkem je `cz.upol.scientificRepository`. Následující kapitoly popíší balíčky, které kořenový balíček obsahuje.

3.2.1.1 Annotation

Obsahuje java anotace, které byly použity jako značky pro udržení pořádku v kódu. Používají je metody s přímým přístupem k databázi.

3.2.1.2 Bean

Složka obsahuje Java Bean třídy spolupracující s pohledy (získ dat z formulářů, obsluha událostí). Obsahuje další 2 balíčky.

- **Controller** - třídy s příponou „controller“. Zodpovídají zejména za zpracování událostí z pohledů. Připravují data k renderování výsledné HTML stránky. Obsahuje další balíčky a třídy provádějící operace zejména s entitami. Většinou jde o přidání, úpravu, zobrazení a správu entit z balíčku `bean.entity`. Mimo balíček `insertion`, který má kontrolery vkládané do jiných kontrolerů a balíček `search` s třídami realizujícími vyhledávání.
- **Entity** - třídy reprezentující entity vyskytující se v aplikaci, jako např. uživatel, publikace. Zároveň obsahuje rozhraní, která jsou implementována, vybranými entitami.

3.2.1.3 Database

Třídy provádějící operace asociované s databází. Každá třída pracující přímo s databází se nachází v tomto balíčku. Obsahuje další 3 balíčky:

- **Dao** - třídy s přímým přístupem k databázi. Většina tříd představuje operace nad jednou tabulkou v databázi. Přístup k databázi je kombinován s nativním JDBC API a Spring `JdbcTemplate`, které poskytuje vyšší programátorský komfort. Nativní JDBC bylo používáno především pro databázové transakce, kvůli přímočařejšímu použití.
- **RowContainer** - použito pro snadnější manipulaci s výsledky databázového dotazu, které nelze přímo mapovat na entity.
- **RowMapper** - obsahuje třídy implementující rozhraní `RowMapper` pro mapování výsledků výběru z databáze na entity či `rowContainery`. Používáno ve spolupráci se Spring JDBC.

3.2.1.4 Develop

Obsahuje třídy pro usnadnění testování při vývoji aplikace.

3.2.1.5 EnumType

Obsahuje výčtové typy pro definici a asociaci dat se společným účelem. Například `PublicationType` pro rozlišení podporovaných publikací.

3.2.1.6 Exception

Třídy, které dědí z třídy `Exception`. Slouží pro rozlišení a snadnější identifikaci chyb hlavně při práci s databází.

3.2.1.7 Jsf

Obsahuje třídy, které blíže souvisejí s frameworkem JSF. `AuthorizationFilter` slouží pro zajištění kontroly oprávnění přístupu na určité URL adresy aplikace. To znamená, že obyčejný uživatel nemá přístup na URL adresy administrativní sekce. Dále obsahuje tzv. validátory, implementující validaci specifických vstupních dat nebo statickou třídu `FacesAccessor` pro snadnější a rychlejší přístup k metodám instance `FacesContextu`.

3.2.1.8 MetaDataExtraction

Seskupuje třídy, které získávají metadata pro automatické vyplnění. Prvním způsobem je získání dat z webových repositářů. Druhý způsob extrahuje data z vkládaného pdf souboru. Dále obsahuje třídu pro analýzu získaných metadat.

3.2.1.9 Servlet

Balíček obsahující jedinou třídu, která je potomkem třídy `HttpServlet`. Stará se o zobrazení vybraného pdf souboru v prohlížeči.

3.2.1.10 Util

Obsahuje další používané třídy, které se nehodí do jiného balíčku a nemá pro ně smysl vytvářet balíček vlastní. Je zde například třída `BibTexBuilder`, která se stará o vytvoření obsahu bibtex souboru.

3.2.2 Složka webapp

Tato složka obsahuje soubory s kódy uživatelského rozhraní, tedy pohledů, v jazyce XHTML a konfigurační soubory v jazyce XML. Obsahuje také několik složek, které nyní budou popsány.

- **Admin** - pohledy přístupné pouze uživateli s právy administrátora.
- **Error** - pohledy pro zobrazení chybových zpráv uživateli.
- **Resources** - obsahuje kaskádové styly a obrázky v podsložkách. V podsložce `repository` najdeme uživatelsky definované, znovu použitelné komponenty uživatelského rozhraní.
- **Templates** - šablony pro omezení redundantního kódu pohledů a vkládané pohledy.
- **WEB-INF** - konfigurační soubory pro `FacesServlet` a pro mapování URL adres.

3.3 Struktura databáze

Jelikož je aplikace repozitář, bylo nutné navrhnout databázi pro uložení potřebných dat. Databáze obsahuje následující tabulky:

- **author** – autoři publikací,
- **author_publication** – vztah autorů k publikacím, tzn. který autor napsal jakou publikaci,
- **book** – kniha (typ publikace),
- **collection** – kolekce sdružující publikace. Časopis pro článek v časopise, sborník konference pro článek z konference a série pro knihu,
- **conference_article** – článek z konference (typ publikace),
- **document** – pdf soubor publikace,
- **journal_article** – článek z časopisu (typ publikace),
- **member** – uživatelé systému,
- **member_publication** – publikace, které mají uživatelé v soukromém repozitáři,
- **publication** – společná data všech typů publikací. Ve spojení vždy s jednou tabulkou (book, conference_article, journal_article) tvoří data jednoho typu publikace,
- **publication_list** - seznamy publikací,
- **publication_list_publication** - publikace patřící do konkrétního seznamu publikací,
- **security_token** - tokeny¹⁵ vygenerované při žádosti o obnovu hesla.

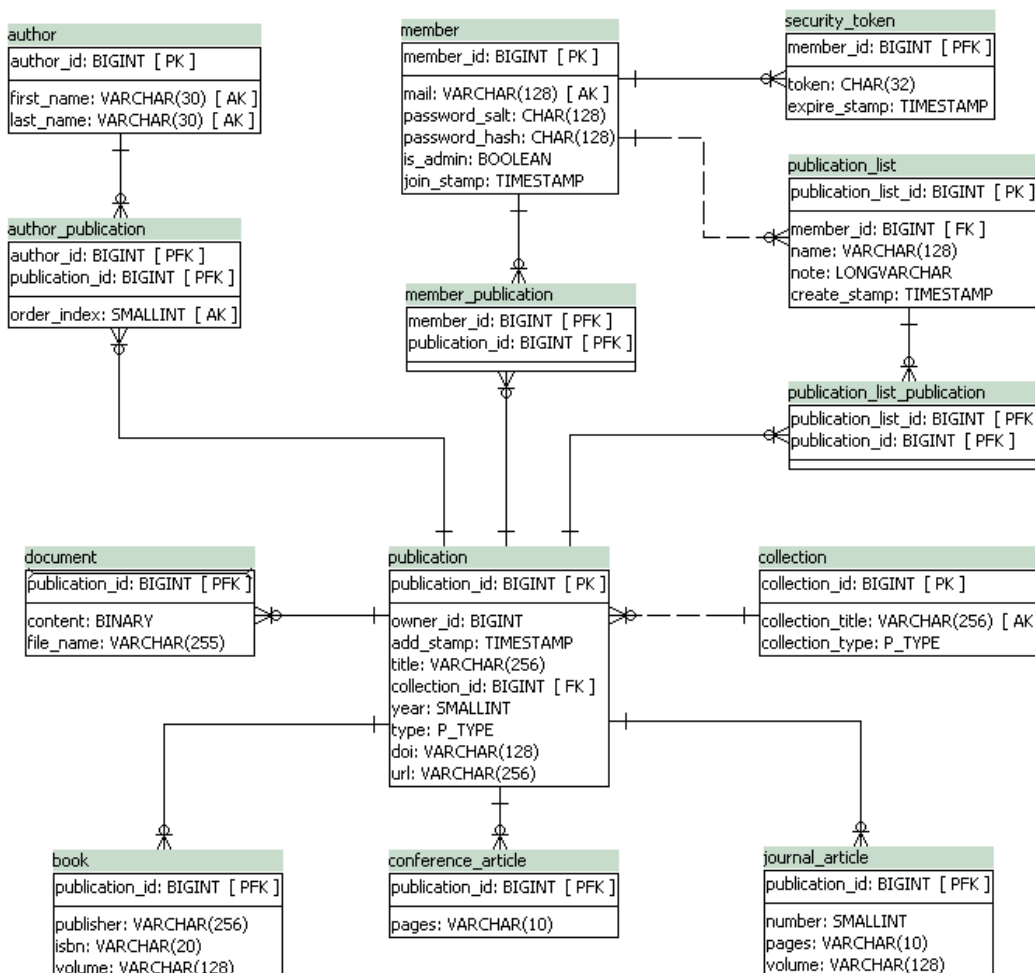
Integritu databáze zajišťují cizí klíče. Při vykonání dotazu „UPDATE“ respektive „DELETE“ je automaticky provedena akce „CASCADE“, která zapříčiní úpravu respektive smazání odkazujících se záznamů. Na obrázku 7 je zobrazeno schéma databáze, která byla navržena pro aplikaci. Ve schématu jsou použity následující zkratky:

- PK – primární klíč,
- FK – cizí klíč,
- AK – unikátní klíč,

¹⁵generované řetězce s časově omezenou dobou platnosti

- PFK – primární a zároveň cizí klíč.

Schéma také používá datový typ PTYPE, což je uživatelsky definovaný datový typ pro rozlišení typu publikace. Byl vytvořen kvůli udržení konzistence databáze a snadnému mapování na korespondující výčtový typ jazyku Java.



Obrázek 7: Schéma databáze

3.4 Zabezpečení

Heslo je ukládáno do databáze jako haš vytvořený algoritmem SHA-512. Před hašováním je heslo zřetězeno s vygenerovanou solí.¹⁶ Sol je generována pro každé heslo nová a je uložena v databázi společně s hašem hesla. Všechny databázové operace jsou řešeny pomocí parametrizovaných dotazů. JSF poskytuje ochranu

¹⁶Vygenerovaný řetězec, který se připojuje za heslo pro zvýšení jeho bezpečného uložení v databázi.

proti XSS (Cross-site scripting) útoku používáním komponenty `outputText` v pohledech. Komponenta provádí automatické escapování hodnot. Vestavěnou ochranu proti CSRF (Cross-site request forgery) poskytuje JSF, použitím autorizačního tokenu.

3.5 Popis řešení

V této kapitole budou popsány způsoby řešení vybraných problémů při vývoji aplikace. Zavedeme následující definici.

Definice 1 (Mapa metadat)

Mapou metadat nazveme objekt jazyku Java, který má následující strukturu: `Map<MetaDataItem, String>` kde objekt `MetaDataItem` je výčtový typ označující typ metadat nazývaný klíč. `String` je řetězec reprezentující samotná metadata, nazývaný hodnota.

3.5.1 Vyplňování metadat

Automatické vyplňování metadat je realizováno pomocí dvou veřejně přístupných webových repozitářů.

1. **Dblp** – referenční repozitář vědeckých publikací z oblasti informatiky. V květnu roku 2016 indexoval více než 3,3 miliony publikací napsaných více než 1,7 miliony autorů [21].
2. **Crossref** – nezisková organizace mimo jiné podporující vědecký výzkum. Poskytuje služby pro jednoduché citování publikací [22].

Na tyto repozitáře se aplikace připojí pomocí protokolu HTTP s požadavkem na vyhledání publikace. Poté jsou získána metadata v určitém formátu. Ta jsou zpracována a v případě, že nabízí relevantní výsledky, jsou nabídnuta uživateli k použití. Aplikace samozřejmě nemůže metadata úspěšně vyhledat, pokud nemá o publikaci nějakou vstupní informaci. Tuto roli plní titul publikace. K jeho získání jsou k dispozici dva způsoby. První způsob je získat titul přímo z nahraného pdf souboru pomocí knihovny Apache PDFBox, popsané v kapitole 3.1.7. Druhým způsobem je přímý zápis uživatele do vstupního formuláře. Následuje podrobný popis získání metadat z obou zmíněných repozitářů.

Dblp – Algoritmus je založen na dvou HTTP požadavcích na tyto URL adresy:

1. `http://dblp.uni-trier.de/search?q=<dotaz>`,
2. `http://dblp.uni-trier.de/rec/xml/<dblp identifikátor>`.

V první adrese je `<dotaz>` nahrazen titulem publikace. Následně je na adresu zaslán požadavek. Výsledkem je získaný HTML kód, jenž je zpracován pomocí

knihovny Jsoup, popsané v kapitole 3.1.7. Projde se prvních 5 výsledků, z nichž se titul každého porovnává s hledaným. Porovnání je provedeno prostřednictvím metody `getSimilarity` (viz kapitola 3.5.3.2), sloužící pro výpočet procentuální míry podobnosti dvou řetězců. Je vybrán titul s největší mírou podobnosti, a pokud ji má dostatečně vysokou, je podle něj vybrán dblp identifikátor publikace. V opačné situaci algoritmus končí a nebyla získána vhodná metadata.

Pokud úspěšně došlo k získání potřebného identifikátoru, pokračuje se požadavkem na druhou URL adresu. V ní je `<dblp identifikátor>` nahrazen získaným dblp identifikátorem. Provede se požadavek a je získána publikace zakódovaná v XML formátu. Zpracování XML je opět provedeno knihovnou Jsoup, které probíhá následovně:

1. Je přečten nezpracovaný tag.
2. Jeho jméno se porovná s názvy sbíraných metadat, které představují klíč.
3. Pokud je nalezena shoda, je hodnota tagu uložena do mapy metadat pod klíčem, který se shodoval se jménem tagu.

Tyto kroky se opakují do chvíle, až jsou zpracovány všechny tagy. Nyní máme získána všechna dostupná a pro aplikaci relevantní metadata, uložená v mapě metadat.

CrossRef – Algoritmus používá HTTP požadavek na jedinou URL adresu:

`http://api.crossref.org/works?query=<dotaz>&sort=relevance&rows=5`

Na adresu se pošle požadavek kde `<dotaz>` je titul hledané publikace. Je získán kód ve formátu JSON a následně zpracován knihovnou JSON-java popsanou v kapitole 3.1.7. Zpracování probíhá ve dvou fázích. V prvním fázi jsou získány tituly všech 5 výsledků a opět za pomoci metody `getSimilarity` porovnány s požadovaným titulem. Je vybrán titul s největší mírou podobnosti, a pokud ji má dostatečně vysokou provede se fáze druhá, jinak algoritmus končí.

Druhá fáze spočívá v iteracích nad množinou klíčů mapy metadat. Kdy každý klíč má jasně danou cestu k požadované hodnotě, která je vložena do mapy metadat. Po ukončení iterací jsou dostupná metadata uložena v mapě metadat.

Po ukončení obou předchozích algoritmů mohou nastat tyto případy.

1. Příslušná metadata nebyla získána z ani jednoho systému. Uživateli je to oznámeno. Ten se může pokusit akci opakovat, nebo vyplnit metadata manuálně.
2. Metadata byla získána z obou zdrojů. Mapy metadat jsou analyzovány – mohou nastat dva případy:
 - a) Shodují se a nezáleží na tom, která budou vybrána.

b) Neshodují se, uživatel je vyzván k výběru.

3. Metadata byla získána pouze z jednoho zdroje.

Poslední dva body zapříčiní, že jsou metadata úspěšně vyplněna z mapy metadat.

3.5.2 Export do formátu bibtex

Aplikace generuje bibtex (popsán v kapitole 3.1.6) záznam¹⁷ pro objekty implementující rozhraní `BibTexEntity` z balíčku `cz.upol.scientificRepository.bean.entity`. Rozhraní má následující metody:

- `String bibTexType()` - vrátí typ bibtex záznamu.
- `Map<MetaDataItem, String> getMetaDataMap()` - vrátí objekt popsáný v definici 1 s metadaty ke konstrukci bibtex záznamu.

Vytvoření jednoho záznamu a poté i celého bibtex kódu (zřetěžením jednotlivých záznamů) je docíleno pomocí třídy `BibTexBuilder` z balíčku `cz.upol.scientificRepository.util`.

Algoritmus pracuje v iteracích nad seznamem obsahující `BibTexEntity`. Pro každou entitu provede následující.

1. Získá typ entity a zapíše ve tvaru „@typ{“.
2. Z mapy metadat vezme hodnotu klíče identifikující záznam a zapíše s příponou „,“.
3. Následně pro ostatní metadata (která nejsou prázdná) z mapy provede zápis ve tvaru „klíč = {hodnota}“, který pro lepší čitelnost odsadí a zalomí řádek.
4. Odstraní poslední čárku a zapíše „}“.

Z vygenerovaného kódu vytvoří dočasný soubor s koncovkou `.bib`, který je jako instance třídy `FileInputStream` předán speciálnímu objektu frameworku `PrimeFaces`. Ten zapříčiní, že (po stisknutí příslušného tlačítka v pohledu) stáhneme vytvořený bibtex soubor. Zdrojový kód 1 zobrazuje ukázkou vygenerovaného bibtexu pro seznam publikací s dvěma položkami.

3.5.3 Podobnost řetězců

Schopnost vypočítat podobnost dvou řetězců je vzhledem k povaze implementované aplikace užitečná. Díky ní lze uživateli poskytnout vyšší komfort.

Například při získávání metadat. Uživatel si nemusí pamatovat přesný titul publikace a přesto může získat metadata, která skutečně zamýšlel. Také nemusí být obtěžován zobrazováním nevhodných výsledků či výběrem metadat, když se liší například jen v interpunkčním znaménku.

¹⁷Bibtex kód pro jednu publikaci.


```

1 @article{Osicka2010On,
2   AUTHOR = {Petr Osicka},
3   TITLE = {On similarity preservation in
4     fuzzy relation systems with hedges.},
5   VOLUME = {39},
6   YEAR = {2010},
7   PAGES = {507--523},
8   JOURNAL = {Int. J. General Systems},
9   NUMBER = {5}
10  }
11
12 @inproceedings{Konecny2010General,
13   AUTHOR = {Jan Konecny, Petr Osicka},
14   TITLE = {General Approach to Triadic Concept Analysis.},
15   YEAR = {2010},
16   BOOKTITLE = {CLA},
17   PAGES = {116--126}
18  }

```

Zdrojový kód 1: Ukázka vygenerovaného bibtextu pro seznam publikací

3.5.3.1 Levenshteinova vzdálenost

Algoritmus pro výpočet vzdálenosti dvou řetězců, pojmenovaný po svém autorovi Vladimíru Levenshteinovi [23]. Levenshteinova vzdálenost je definována jako minimální počet úprav nutných k transformaci jednoho řetězce na druhý. Úpravami se rozumí přidání, odebrání nebo náhrada písmene.

Uvažujme například tyto dva řetězce „Konvexní“ a „Konkávni“. Jelikož jsou stejně dlouhé, stačí nalézt počet nutných náhrad. První 3 písmena jsou shodná. Následující tři písmena už jsou rozdílná a proto se provedou náhrady:

1. v → k,
2. e → á,
3. x → v.

Poslední dva znaky jsou již opět shodné a proto je výsledná hodnota Levenshteinovy vzdálenosti mezi řetězci „Konvexní“ a „Konkávni“ rovna 3.

3.5.3.2 Implementace

Výsledkem aplikace levenshteinovy vzdálenosti na dva řetězce je celé číslo, které navíc může růst v závislosti na délce vstupních řetězců. Pomocí tohoto čísla by se dala jen těžko objektivně a obecně určit podobnost. Žádoucí proto je, získat relativní hodnotu vyjadřující podobnost řetězců v procentech.

Zdrojový kód 2 zobrazuje implementaci metody `getSimilarity` pro zjištění procentuální podobnosti dvou řetězců. Na řádce 6 je do proměnné `maxLength`

```

1 double getSimilarity(String str1, String str2) {
2     int str1Len = str1.length();
3     int str2Len = str2.length();
4
5     // uloženi maximalni delky retezce ze vstupu
6     int maxLength = (str1Len > str2Len) ? str1Len : str2Len;
7
8     // vypocet levenshteinovy vzdálenosti
9     int levDist = StringUtils.getLevenshteinDistance(str1, str2);
10
11    return 1 - ((double) levDist / maxLength);
12 }

```

Zdrojový kód 2: Výpočet procentuální podobnosti dvou řetězců

uložena maximální délka z řetězců ze vstupu. Na řádku 9 dochází k výpočtu levenshteinovy vzdálenosti ze vstupních řetězců, který obstará metoda z knihovny Apache Commons Lang (viz kapitola 3.1.7). Řádek 11 již vrací procentuální podobnost řetězců, jako podíl proměnné `levDist` a `maxLength` odečtený od hodnoty 1.

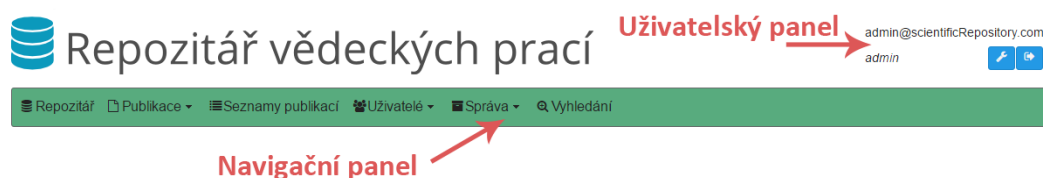
Je nutné si uvědomit význam odečtení podílu od hodnoty 1. Proměnná `levDist` obsahuje počet nutných úprav, aby byly řetězce ekvivalentní, a proto by samotný podíl vyjadřoval procentuální míru nepodobnosti. Cílem je však získat procentuální míru podobnosti, a proto je výsledkem rozdíl hodnoty 1 a vypočteného podílu.

4 Uživatelská dokumentace

Tato kapitola slouží jako manuál pro uživatele. Nejprve bude ukázána práce s aplikací z pohledu běžného uživatele. Poté bude ukázána i z pohledu administrátora, který může pracovat jako běžný uživatel, ale má samozřejmě k dispozici administrační funkce aplikace.

4.1 Přihlášení

Přihlášení k aplikaci probíhá standardním způsobem přes stránku s formulářem. Je nutné vyplnit login (používá se registrovaný mail) a heslo. Po úspěšném přihlášení je uživatel přesměrován na stránku se soukromým repozitářem.



Obrázek 8: Záhloví aplikace po přihlášení uživatele

Na obrázku 8 lze vidět záhlaví aplikace po přihlášení. Uživatelský panel zobrazuje e-mail a oprávnění uživatele. Obsahuje dvě tlačítka. Levé (ikona montážního klíče) pro přesměrování na stránku se správou účtu a pravé pro odhlášení (ikona s šipkou). Navigační panel obsahuje menu aplikace pro navigaci mezi stránkami.

4.2 Obnova hesla

V případě zapomenutého hesla, lze heslo obnovit. Je třeba přejít na stránku s formulářem pro obnovu. Toho je docíleno kliknutím na položku v menu na přihlašovací stránce. Je nutné vyplnit registrovaný mail. Pokud je zadán neregistrovaný mail, není o tom z bezpečnostních důvodů uživatel informován. V opačném případě jsou na mail poslány instrukce pro obnovu.

V mailu se nachází odkaz na formulář k obnově hesla. Možnost obnovy hesla je po zaslání požadavku časově omezena na 30 minut.

4.3 Správa účtu

Na stránce pro správu účtu může uživatel změnit e-mail a heslo. Aktualizovaný email samozřejmě nesmí být již registrován.

4.4 Tabulka zobrazující data

Data se na většině stránek aplikace zobrazují v tabulkách. Každý sloupec je pojmenován podle toho co zobrazuje. V záhlaví tabulky, lze ve většině tabulek

data řadit podle hodnoty sloupce. Stačí na něj kliknout v záhlaví, lze řadit vzestupně i sestupně. Pokud se ve sloupci v záhlaví zobrazuje pod popisem okénko formuláře, lze také data podle sloupce filtrovat.

Většina tabulek také obsahuje tlačítka související s položkou na řádku, kde se nacházejí. U levého okraje jsou to zaškrtačovací tlačítka pro výběr jedné či více položek tabulky. U pravého zase tlačítka s ikonami pro provedení akce s položkou tabulky (nemusí obsahovat všechny), které si lze prohlédnout na obrázku 9.

- **Lupa** – zobrazení detailu publikace v dialogovém okně (pouze u publikací).
- **Písmeno „i“** – odkaz na stránku zobrazující přehled položky.
- **Tužka** – odkaz na formulář pro úpravu položky.
- **Ikona pdf dokumentu** – slouží pro zobrazení pdf souboru publikace v nové kartě prohlížeče (pouze u publikací). Pokud je tlačítko zobrazeno světlejším odstínem než ostatní tlačítka není pdf dokument k publikaci nahrán.



Obrázek 9: Tabulková tlačítka

Pokud tabulka obsahuje více než 15 záznamů, zobrazí se stránkování tabulky.

4.4.1 Tabulka pro publikace

Tabulka s publikacemi se vyskytuje v aplikaci často, proto bude konkrétněji popsána.

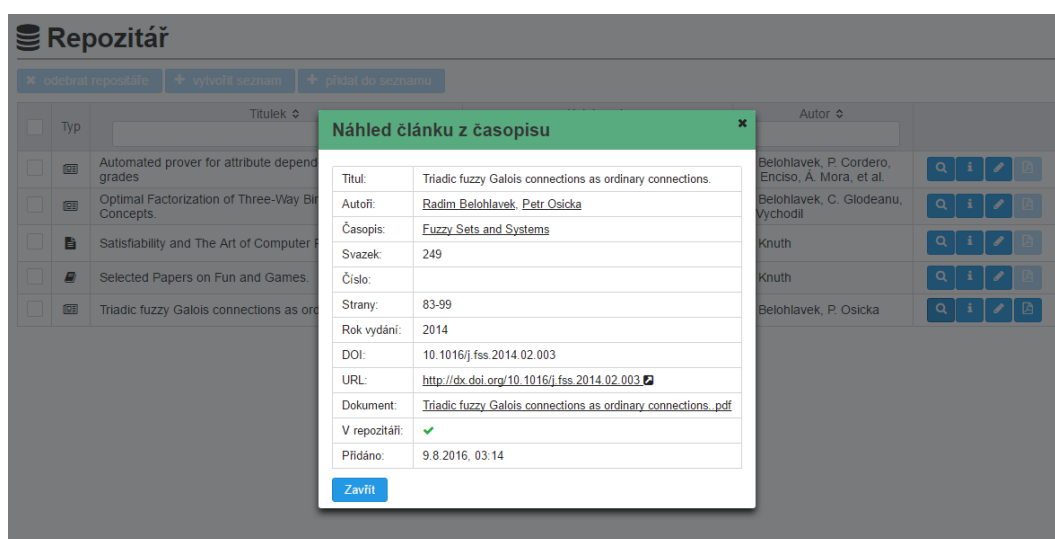
Tabulka přímo zobrazuje typ (ikona při najetí myši zobrazí popis), titul, kolekci (kniha – série, článek z konference – sborník, článek z časopisu – časopis) a autory. V případě, že je titul či kolekce příliš dlouhá, zobrazí se pouze určitý podřetězec, končící třemi tečkami. Pokud je autorů víc než se vejde do buňky, je zobrazeno pouze prvních několik autorů, počtem podle místa v buňce, zakončených zkratkou „et al.“. Tabulky, kde je to potřebné, zobrazují také ikony informující o tom, zdali má publikaci uživatel přidanou v soukromém repozitáři. Ikony lze spatřit na obrázku 11. Zelená ikona značí, že je publikace v soukromém repozitáři přidána. Červená značí opak.

Na obrázku 10 je zobrazen náhled publikace po kliknutí na tlačítko s ikonou lupy v tabulce publikací.

4.4.1.1 Tlačítka s akcemi pro tabulku publikací

Nad tabulkou zobrazující publikace se mohou nacházet tlačítka pro provádění operací s vybranými publikacemi.

- **Přidat do repozitáře** – přidá publikace do soukromého repozitáře.



Obrázek 10: Náhled publikace po kliknutí na tlačítko s ikonou lupy v tabulce

- **Obebrat z repozitáře** – odebere publikace ze soukromého repozitáře.
- **Vytvořit seznam** – zobrazí dialogové okno s formulářem pro přidání nového seznamu publikací. Pokud nebyly publikace v uživatelově soukromém repozitáři, jsou do něj přidány.
- **Přidat do seznamu** – zobrazí dialog s tabulkou již vytvořených seznamů, kde lze jeden vybrat a do něj publikace přidat. Pokud nebyly publikace v uživatelově soukromém repozitáři, jsou do něj automaticky přidány.

Jestliže není vybrána alespoň jedna publikace, jsou tlačítka neaktivní.

Na obrázku 11 je zobrazena tabulka s publikacemi. Modře jsou podbarveny vybrané řádky. Nahoře nad tabulkou jsou vidět tlačítka pro akce s vybranými publikacemi.

+ Přidat do repozitáře		* Odebrat z repozitáře		+ Vytvořit seznam		+ Přidat do seznamu	
<input type="checkbox"/>	Typ	Titulek	Kolekce	Author			
<input type="checkbox"/>	📄	Automated prover for attribute dependencies in data with grades	International Journal of Approximate Reasoning	R. Belohlavek, P. Cordero, M. Enciso, Á. Mora, et al.	✓	🔍	📄
<input checked="" type="checkbox"/>	📄	On similarity preservation in fuzzy relation systems with hedges.	International Journal of General Systems	P. Osicka	✓	🔍	📄
<input type="checkbox"/>	📄	Satisfiability and The Art of Computer Programming.	Theory and Applications of Satisfiability Testing – SAT 2012	D. Knuth	✓	🔍	📄
<input checked="" type="checkbox"/>	📄	Selected Papers on Fun and Games.	CSLI lecture notes series	D. Knuth	✗	🔍	📄
<input checked="" type="checkbox"/>	📄	Triadic concept lattices in the framework of aggregation structures.	Information Sciences	J. Konecny, P. Osicka	✓	🔍	📄
<input type="checkbox"/>	📄	Triadic fuzzy Galois connections as ordinary connections.	Fuzzy Sets and Systems	R. Belohlavek, P. Osicka	✓	🔍	📄

Obrázek 11: Tabulka publikací s vybranými publikacemi

4.5 Repozitář

Umožňuje správu publikací, které má uživatel přidány ve svém repozitáři. Publikace jsou přehledně zobrazeny v tabulce. Nad tabulkou se nachází tlačítka s akcemi: odebrat z repozitáře, vytvořit seznam, přidat do seznamu. Funkce tlačítek byly popsány v kapitole [4.4.1.1](#).

4.6 Publikace

4.6.1 Přidání

Každý typ publikace má k přidání rozdílný formulář s políčky pro metadata, která jej charakterizují. Metadata lze samozřejmě vyplnit ručně. Pohodlnější je však použít tlačítka na automatické vyplnění metadat. Stačí nahrát dokument nebo vepsat titul publikace. Po kliknutí na tlačítka „Vyplnit metadata“ se aplikace pokusí vyhledat data v repozitářích popsaných v kapitole [3.5.1](#). Mohou nastat následující možnosti:

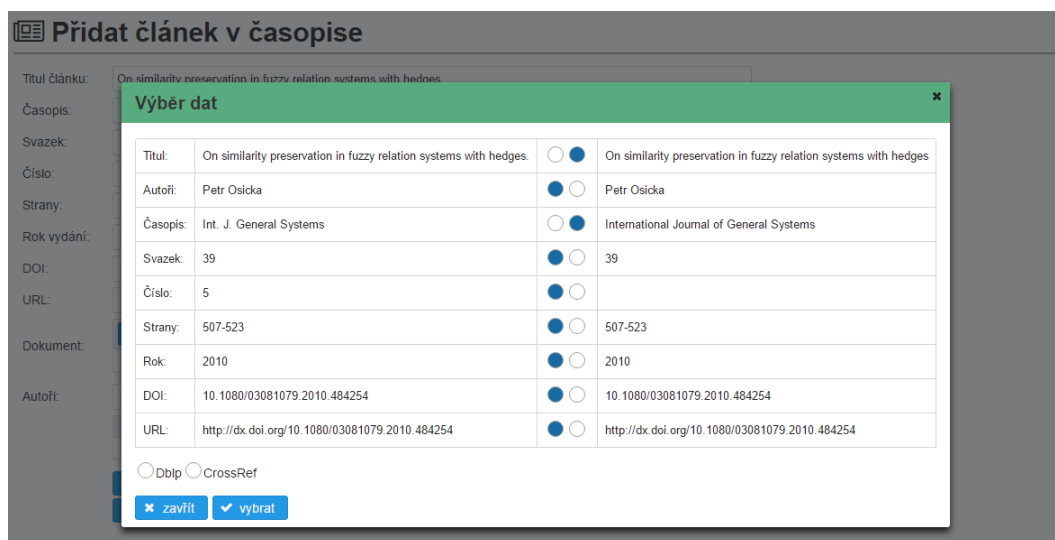
- Nepodařilo se připojit k jednomu nebo oběma repozitářům.
- Nepodařilo se získat žádná relevantní metadata.
- Metadata byla získána z jednoho repozitáře.
- Metadata se podařilo získat z obou repozitářů. Mohou nastat dva případy.
 - Metadata jsou shodná → není co vybírat.
 - Metadata jsou rozdílná a uživateli se zobrazí dialogové okno s možností výběru metadat. Metadata může uživatel vybrat podle repozitáře nebo je libovolně nakombinovat. Dialog je vidět na obrázku [12](#), přičemž pro ostatní typy publikací se liší pouze ve sbíraných metadatach.

Po kliknutí na tlačítka „Přidat“ aplikace porovná přidávaných titul publikace s tituly v databázi a pokud nalezne dostatečnou shodu, je uživateli zobrazen dialog s publikací, která má podobný titul. Na přehled této publikace se může nechat přeměřovat a místo přidání nové publikace zvolit publikaci již vloženou v repozitáři.

4.6.2 Přehled

Přehled publikace je pro všechny typy stejný. Jediné rozdíly jsou samozřejmě v zobrazovaných metadatach. Ta jsou přehledně zobrazena v tabulce. Nad tabulkou se nachází 3 tlačítka:

- **Upravit** – přeměřuje na stránku s úpravou publikace.



Obrázek 12: Dialog pro výběr metadat při přidávání článku z časopisu

- **Přidat do repozitáře** – přidá publikaci do repozitáře. Zobrazeno pouze pokud uživatel nemá přidanou publikaci v repozitáři.
- **Odebrat z repozitáře** – odebere publikaci z repozitáře. Zobrazeno pouze pokud uživatel má publikaci přidanou v repozitáři.
- **Přidat do seznamu** – zobrazí dialogové okno s nabídkou seznamů, do kterých lze publikace přidat (nezobrazuje ty v nichž je již publikace obsažena).
- **Zobrazit bibtex** – zobrazí bibtex záznam publikace v dialogovém okně.

Tlačítka jsou zobrazena v jednu chvíli skutečně pouze tři, jelikož tlačítka pro přidání a odebrání publikace z repozitáře se podle aktuálního stavu střídají.

Pod zobrazenými metadaty je vysouvací panel, který při otevření načte seznamy, v nichž je publikace obsažena. Má také k dispozici tlačítko pro odebrání publikace z vybraných seznamů.

4.6.3 Úprava

Úprava publikace logicky obsahuje formulářová políčka, stejná jako v případě přidání (v rámci stejného typu). Ta jsou po zobrazení vyplněna dostupnými metadaty která lze upravit. Publikaci může upravovat běžný uživatel, pouze pokud ji má ve svém repozitáři přidanou výlučně on.

4.7 Seznamy publikací

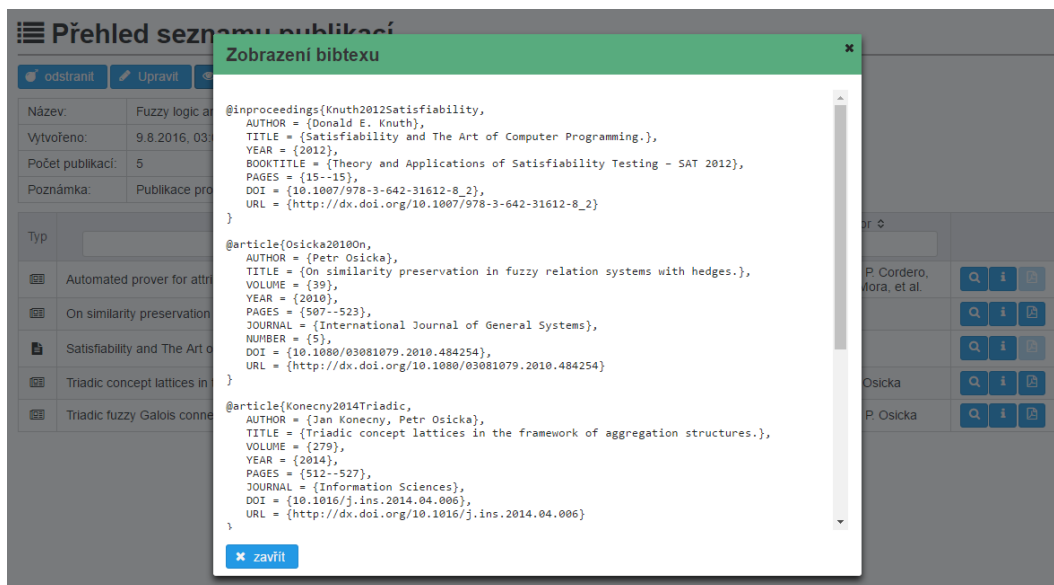
4.7.1 Správa

Zobrazuje uživatelské seznamy publikací v tabulce. Nad tabulkou je pouze jedno tlačítko pro odstranění vybraných seznamů.

4.7.2 Přehled

Zobrazuje informace o seznamu a publikace, které do něj patří. Nahoře jsou k dispozici tlačítka s akcemi.

- **Odstranit** – odstraní seznam.
- **Upravit** – přesměruje na stránku pro úpravu seznamu.
- **Zobrazit bibtex** – zobrazí dialogové okno a v něm bibtex s publikacemi z aktuálně zobrazovaného seznamu (viz obrázek 13).
- **Stáhnout bibtex** – vytvoří bibtex soubor s publikacemi z aktuálně zobrazeného seznamu a umožní jej uživateli stáhnout.



Obrázek 13: Dialogové okno s bibtexem aktuálně zobrazovaného seznamu

4.7.3 Úprava

Umožňuje úpravu seznamu přes formulář. Zobrazuje publikace patřící do seznamu s dvěma tlačítky s akcemi.

- **Odebrat** – odebere vybrané publikace ze seznamu.

- **Přidat** – zobrazí dialogové okno s výběrem publikací z uživatelského repozitáře, které lze přidat (nejsou v seznamu již obsaženy).

Na závěr je nutné potvrdit tlačítkem „upravit“, jinak změny nebudou uloženy.

4.8 Zobrazení autora nebo kolekce

Zde jsou zobrazeny informace o autorovi/kolekci. Je k dispozici také tabulka (viz kapitola 4.4.1) se všemi autorovými publikacemi, respektive se všemi publikacemi patřící do kolekce. Pro tabulku jsou k dispozici všechna tlačítka s akcemi (viz kapitola 4.4.1.1).

4.9 Vyhledávání

Vyhledávat je možné v následujících položkách.

- **Publikace** – pomocí zaškrtnutých tlačítek lze navolit typy publikací, které se budou prohledávat. Zaškrtnuté tlačítko s popiskem „Autoři společně“, rozhoduje zda-li musí být zadání autoři společní nebo se mají vyhledat všechny publikace s alespoň jedním vyplněným autorem.
- **Autoři** – vyhledávání autorů.
- **Kolekce** – vyhledávání kolekcí.

🔍 Vyhledávání

Publikace
Autoři
Kolekce

Parametry

Články z konferencí
 Články z časopisů
 Knihy
 🔄 reset
🔍 hledat
Autoři společně

titulek...	Osicka, Belohlavek
2016	DOI...
URL...	Journal of Approximate Reasoning
sborník...	série...
18	svazek...
isbn...	vydavatel...

+ Přidat do repozitáře
* Odebrat z repozitáře
+ Vytvořit seznam
+ Přidat do seznamu
🗑️ Odstranit ze systému (admin)

<input type="checkbox"/>	Typ	Titulek ↕	Kolekce ↕	Author ↕				
<input type="checkbox"/>		Automated prover for attribute dependencies in data with grades	International Journal of Approximate Reasoning	R. Belohlavek, P. Cordero, M. Enciso, A. Mora, et al.	✓			

Obrázek 14: Ukázka vyhledávání publikací

Při vyhledávání publikací se výsledky zobrazí v tabulce (kapitola 4.4.1), přičemž akční tlačítka jsou zobrazena všechna.

4.10 Odhlášení

Pokud je uživatel 20 minut neaktivní, je automaticky ohlášen. Pro manuální odhlášení stačí kliknout na ikonku vpravo v uživatelském panelu.

4.11 Administrace

Tato kapitola pojednává o rozšířených možnostech práce s aplikací z účtu administrátora.

4.11.1 Správa publikací

Správa publikací zobrazuje všechny publikace uložené v databázi aplikace. Publikace jsou zobrazeny v tabulce (viz kapitola 4.4.1), která má navíc sloupec s počtem udávajícím, v kolika soukromých repozitářích je publikace vložena. Administrátor je může mazat z databáze. Tímto krokem o publikaci přijdou všichni uživatelé, kteří ji měli ve svých soukromých repozitářích či seznamech.

Může se stát, že nějaký uživatel přidá publikaci a pak o ni ztratí zájem a ze svého repozitáře ji odebere. Publikaci pak nemusí nikdo používat. Administrátor může nepoužívané publikace odstranit z databáze aplikace pomocí tlačítka „odstranit publikace bez použití“.

4.11.2 Přehled publikace

Je stejné jako pro běžného uživatele (viz kapitola 4.7.2) s tím rozdílem, že má k dispozici administrační menu „administrace“ u pravého okraje stránky. Administrační menu má tyto nabídky.

- **Upravit** – přesměruje na stránku pro publikace.
- **Odstranit** – odstraní publikaci z databáze.

4.11.3 Správa autorů a kolekcí

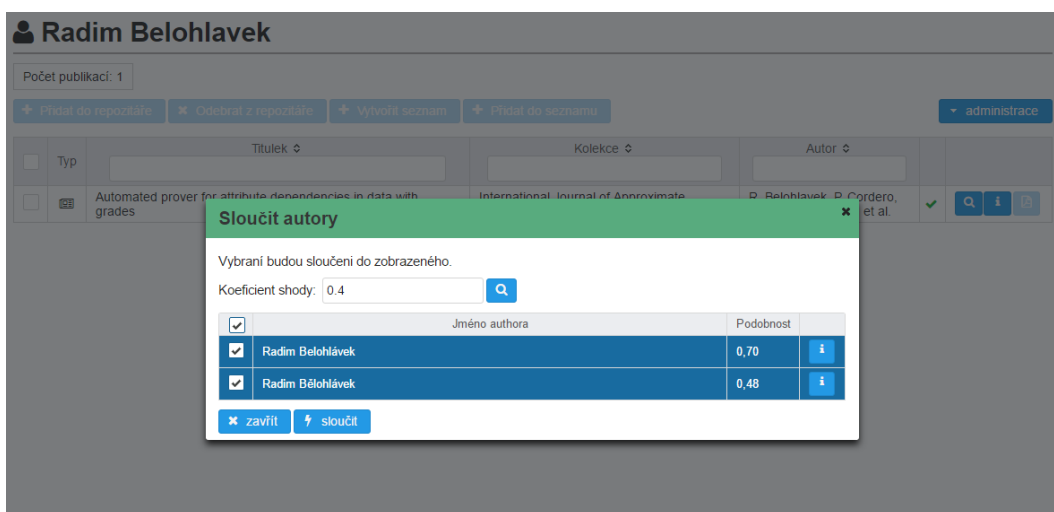
Autoři respektive kolekce jsou zobrazeny v tabulce. Tabulka autorů obsahuje sloupce jméno autora a počet publikací které naspal. Tabulka kolekcí má sloupce vyjadřující: název kolekce, typ kolekce a počet publikací, které kolekce sdružuje. Obě tabulky mají tlačítka pro zobrazení a úpravu.

Správa autorů respektive kolekcí má tlačítko, které odstraní autory bez publikace respektive odstraní prázdné kolekce.

4.11.4 Zobrazení autorů a kolekcí

Je stejné jako pro běžného uživatele (viz kapitola 4.8) s tím rozdílem, že má k dispozici administrační menu „administrace“ u pravého okraje stránky. Administrační menu má tyto nabídky.

- **Sloučit** – zobrazí dialogové okno pro autory/kolekce, kde lze podle zadaného koeficientu podobnosti vyhledat podobné položky a vybrané sloučit. Jako nástupce je zvolen autor/kolekce ze které byla akce vyvolána. Po sloučení samozřejmě nástupce přebírá všechny publikace od sloučených autorů/kolekcí. Na obrázku 15 je zobrazena ukázka slučování autorů.
- **Upravit** – přesměruje na stránku pro úpravu autora/kolekce.
- **Odstranit** – odstraní autora/kolekci. Odstranění je povoleno pouze v případě, že autor/kolekce nemá žádnou publikaci.



Obrázek 15: Dialogové okno pro slučování autorů

4.11.5 Úprava autorů a kolekcí

Úprava autorů/kolekcí spočívá v korekci předvyplněného formulářového políčka. Jméno/název nesmí být již obsazen.

4.11.6 Uživatelé

Administrátor může odstranit vybrané uživatele na stránce se správou uživatelů. Dále může nové uživatele do aplikace registrovat. Na vyplněný mail je zaslána zpráva o registraci a heslo pro první přihlášení.

4.11.7 Vyhledávání

Je stejné jako pro běžného uživatele (viz kapitola 4.9) s tím rozdílem, že má k dispozici administrační tlačítko „Odstranit ze systému (admin)“ u pravého okraje stránky. Kliknutím na tlačítko se po potvrzení vymažou vybrané publikace z databáze.

5 Rozšíření aplikace

Jako každý software tak i v případě mé aplikace se nabízí mnoho potencionálních vylepšení, které by bylo možné a užitečné implementovat.

- Rozšířit možnosti vyhledávání metadat na další volně přístupné repozitáře a nabídnout uživateli více kvalitních metadat.
- Při vkládání nové publikace hledat duplicitní publikaci porovnáním všech metadat, ne pouze titulu.
- Při automatickém vyplnění nabídnout uživateli korekci autorů a kolekcí porovnáním s databází.
- Přidat možnosti importu bibtexu (RISU) – automatické přidání publikací a možnosti vytvořit rovnou seznam publikací.
- Umožnit uživateli export do více formátů (RIS, PDF).
- Možnost slučovat publikaci kombinací metadat.
- Detekce duplicit.
- Zasílání zpráv mezi uživateli (např. automaticky, když administrátor smaže publikaci, kterou má uživatel v repozitáři apod.).

Závěr

V rámci bakalářské práce byla implementována webová aplikace sloužící jako re-
pozitář vědeckých publikací. Aplikace splňuje zadané požadavky. Hlavní funkcí
implementované aplikace je automatické vyplňování metadat publikací při je-
jich vkládání. To je docíleno stažením metadat z veřejně přístupných webových
repozitářů.

Tento dokument pojednává o podobných aplikacích, technických aspektech
aplikace a popisuje řešení vybraných problémů při implementaci aplikace. Sou-
částí dokumentu je také uživatelská dokumentace a příloha s instrukcemi pro
instalaci a spuštění aplikace.

Conclusions

Web application serving as repository of scientific papers has been developed as part of the bachelor's thesis. The application meets requirements. Main function of developed application is automatic filling metadata at inserting publication. This is achieved downloading from public access web repositories.

This document deals about similar applications, technical aspects and describing selected solution of problems at developing application. User documentation and appendix with instruction for install and run application is document's part too.

A Nasazení a testování aplikace

Požadavky

Pro nasazení aplikace je nutné mít nainstalovány následující technologie.

1. Java JRE nebo JDK pro Javu 8 –
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. PostgreSQL 9.4 (nebo verze vyšší, ale netestováno) –
<https://www.postgresql.org/download>
3. Apache tomcat 8 (nebo verze vyšší, ale netestováno) –
<http://tomcat.apache.org/download-90.cgi>

Konfigurace

PostgreSQL

Po instalaci lze konfigurovat pomocí programu pgAdmin, který je součástí instalace. Případně lze použít příkazovou řádku, záleží na preferencích a zkušenostech uživatele. Je nutné provést tyto kroky:

1. Vytvořit uživatele „scientificRepository“
2. Vytvořit databázi „scientificRepository“
3. Obnovit databázi pomocí souboru z cd:
 - a) pgAdmin –
bin/postgreSQL/scientificRepository_backup.backup
 - b) příkazová řádka –
bin/postgreSQL/scientificRepository_backup.sql
a použít příkaz
„psql -U scientificRepository scientificRepository < cesta k souboru“

Apache Tomcat

Po instalaci je nejdříve nutné nastavit systémovou proměnnou JRE_HOME pro Java JRE respektive JAVA_HOME pro Java JDK na příslušnou cestu k instalované technologii. Poté přidat PostgreSQL JDBC driver do složky lib (cd: lib/postgresql-9.4.1208.jre7).

Nyní je nutné vytvořit účet administrátora v souboru conf/tomcat-users.xml. Mezi tagy <tomcat-users> a </tomcat-users> vložíme:

Kde si samozřejmě můžeme zvolit vlastní uživatelské jméno i heslo.

```

1 <role rolename="manager-gui"/>
2 <user username="admin" password="admin" roles="manager-gui"/>

1 <Resource name="jdbc/scientificRepository"
2     factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"
3     auth="Container"
4     type="javax.sql.DataSource"
5     maxActive="200"
6     maxIdle="200"
7     maxWaitMillis="10000"
8     username="scientificRepository" password="scientificRepository"
9     driverClassName="org.postgresql.Driver"
10    url="jdbc:postgresql://localhost:5432/scientificRepository" />

```

Nakonec je nutné nastavit přístup k databázi v souboru `conf/context.xml`. Mezi tagy `<Context>` a `</Context>` vložíme: Kde atributy `username` a `password` jsou uživatelské jméno a heslo k databázi. Atribut `url` je tzv. connection string a pokud nesouhlasí je nutné jej upravit:

- **localhost** – IP adresa databáze.
- **5432** – Port na kterém databáze naslouchá.
- **scientificRepository** – název databáze, pokud jste ji pojmenovali jinak.

Konfigurační soubory nalezneme na cd ve složce `conf`.

Nasazení

Musí být samozřejmě spuštěný Tomcat i PostgreSQL (popř. restartovat po na-konfigurování). Otevřeme internetový, prohlížeč zadáme adresu a port Tomcat serveru. Zobrazí se úvodní stránka serveru. Nahoře vpravo vybereme z menu položku `uvManager App` a přihlásíme se údaji, které jsme nakonfigurovali v souboru `conf/tomcat-users.xml`.

Najdeme panel s názvem „Deploy“ a v něm „WAR file to deploy“. Vybereme soubor z `cd: bin/ScientificRepository.war` a klikneme na tlačítko „Deploy“. Počkáme až se soubor nahraje a je stránka aktualizována. Nyní v tabulce „Applications“ v prvním sloupci klikneme na odkaz „ScientificRepository“ a pokud vše proběhlo v pořádku, zobrazí se přihlašovací stránka aplikace.

V databázi je k dispozici pro první přihlášení nativní účet s loginem: „`admin@scientificRepository.com`“ a heslem: „`admin`“. Po přihlášení si změňte email i heslo.

Podporované prohlížeče

Aplikace byla testována zejména pro Google Chrom (v. 51.0). Další bezproblémové používání umožňuje Mozilla Firefox(v. 47.0) a Opera (v. 39.0). Ostatní nejsou testovány nebo v nich aplikace nefunguje korektně.

Testování

Aplikace je pro účel testování nasazena na studentský server katedry informatiky. Adresa webu je <http://student.inf.upol.cz:8081/ScientificRepository>. K dispozici jsou účty (účet – login / heslo):

- Administrátor – admin@scientificRepository.com / admin
- Člen - member@scientificRepository.com / member

V době odevzdání práce se k Tomcatu, kde je aplikace nasazena, dalo bohužel připojit pouze ze serveru student.inf.upol.cz. Pokud bude problém i nadále přetrvávat může to komplikovat testování aplikace.

B Obsah příloženého CD/DVD

bin/

Obsahuje soubor pro nasazení aplikace na Tomcat server a soubory pro vytvoření databáze aplikace.

conf/

Konfigurační soubory serveru ApacheTomcat.

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

lib/

Ovladač k databázi PostgreSQL.

src/

Kompletní zdrojové kódy aplikace.

readme.txt

Instrukce pro spuštění webové aplikace.

Literatura

- [1] WIKIPEDIA, The Free Encyclopedia: Mendeley [online] [cit. 2016-08-10].
Dostupné z: <<https://en.wikipedia.org/wiki/Mendeley>>
- [2] JABREF.ORG: JabRef [online]. [cit. 2016-08-10]. Dostupné z:
<<https://www.jabref.org>>
- [3] REFME LTD: Referencing Made Easy [online]. [cit. 2016-08-10]. Dostupné z:
<<https://www.refme.com>>
- [4] ORACLE: What is Java technology and why do I need it? [online]. [cit. 2016-08-10]. Dostupné z:
<http://www.java.com/en/download/faq/whatis_java.xml>
- [5] TUTORIALSPPOINT: JSF - Architecture [online]. [cit. 2016-08-10]. Dostupné z:
<http://www.tutorialspoint.com/jsf/jsf_architecture.htm>
- [6] ORACLE: The Java EE 6 Tutorial [online]. [cit. 2016-08-10]. Dostupné z:
<<http://docs.oracle.com/javaee/6/tutorial/doc/bnaqq.html>>
- [7] ORACLE: The Java EE 5 Tutorial [online]. [cit. 2016-08-10]. Dostupné z:
<<http://docs.oracle.com/javaee/5/tutorial/doc/bnaqq.html>>
- [8] THE APACHE SOFTWARE FOUNDATION: Apache Maven Project [online]. [cit. 2016-08-10]. Dostupné z: <<https://maven.apache.org>>
- [9] THE APACHE SOFTWARE FOUNDATION: Apache Tomcat [online]. [cit. 2016-08-10]. Dostupné z: <<http://tomcat.apache.org>>
- [10] THE POSTGRESQL GLOBAL DEVELOPMENT GROUP: PostgreSQL [online]. [cit. 2016-08-10]. Dostupné z: <<https://www.postgresql.org>>
- [11] THE POSTGRESQL GLOBAL DEVELOPMENT GROUP: PgAdmin, PostgreSQL Tools [online]. [cit. 2016-08-10]. Dostupné z:
<<https://www.pgadmin.org>>
- [12] ALEXANDER FEDER: Your BibTeX resource [online]. [cit. 2016-08-10].
Dostupné z: <<http://www.bibtex.org/>>
- [13] PRIMETEK: PrimeFaces, Ultimate UI Framework for Java EE [online]. [cit. 2016-08-10]. Dostupné z: <<http://primefaces.org>>
- [14] PIVOTAL SOFTWARE: Accessing Relational Data using JDBC with Spring [online]. [cit. 2016-08-10]. Dostupné z:
<<http://spring.io/guides/gs/relational-data-access>>
- [15] JONATHAN HEDLEY: Jsoup, Java HTML Parser [online]. [cit. 2016-08-10].
Dostupné z: <<https://jsoup.org>>

- [16] SEAN LEARY: JSON-java, GitHub.com [online]. [cit. 2016-08-10]. Dostupné z: <<https://github.com/stleary/JSON-java>>
- [17] THE APACHE SOFTWARE FOUNDATION: Apache Commons Lang [online]. [cit. 2016-08-10]. Dostupné z: <<https://commons.apache.org/proper/commons-lang>>
- [18] OCPSOFT: PrettyFaces [online]. [cit. 2016-08-10]. Dostupné z: <<http://www.ocpssoft.org/prettyfaces>>
- [19] THE APACHE SOFTWARE FOUNDATION: Apache PDFBox® - A Java PDF Library [online]. [cit. 2016-08-10]. Dostupné z: <<https://pdfbox.apache.org>>
- [20] LESS: Getting Started [online]. [cit. 2016-08-10]. Dostupné z: <<http://lesscss.org/>>
- [21] DBLP COMPUTER SCIENCE BIBLIOGRAPHY: What is dblp? [online]. [cit. 2016-08-10]. Dostupné z: <<http://dblp.uni-trier.de/faq/What+is+dblp.html>>
- [22] CROSSREF.ORG: History/mission [online]. [cit. 2016-08-10]. Dostupné z: <<http://www.crossref.org/01company/02history.html>>
- [23] WIKIPEDIA, The Free Encyclopedia: Levenshtein distance [online]. [cit. 2016-08-10]. Dostupné z: <https://en.wikipedia.org/wiki/Levenshtein_distance>