Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Systém pro správu publikací

Plzeň 2017 Ondřej Pittl

Místo této strany bude zadání práce.

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 27. dubna 2017

Ondřej Pittl

Poděkování

Rád bych tímto poděkoval svému vedoucímu bakalářské práce Ing. Michalovi Nyklovi, Ph.D. nejen za vedení mé bakalářské práce, ale také za užitečné připomínky k mé práci a za čas strávený konzultacemi.

Abstract

This bachelor thesis aims to create a PHP library that manages the results of publication activities of the selected scientists. The information that the library will be processing will be obtained by web mining the selected bibliographic services. Based on the complex searches of available bibliographic services and a comparison of their positives and negatives, some of them have been selected. The data obtained will be stored in a database and presented to users in the form of a list of authors and a list of their publications. Additionally, the publications will be formatted in accordance with the citation standard ČSN ISO 690:2011 and the BiBTeX format. The contents of both the lists will be filtered and sorted, so that they can be exported in CSV and TXT formats.

Abstrakt

Tato bakalářská práce si klade za cíl vytvoření PHP knihovny spravující výsledky publikační činnosti zvolených vědců. Data, s nimiž bude knihovna pracovat, budou získávána dolováním webových stránek zvolených bibliografických služeb. Služby byly zvoleny na základě komplexní rešerše a porovnání jejich kladů a záporů. Získaná data budou uchovávána v databázi a strukturovaně prezentována uživateli formou seznamu autorů a seznamu jejich publikací. Publikace budou navíc formátovány v souladu s citační normou ČSN ISO 690:2011 a formátem BiBTeX. Obsah obou seznamů bude filtrován a řazen a u obou bude umožněn export ve formátech CSV a TXT.

Obsah

1 Úvod

S postupem času úroveň celosvětového vzdělání společně se zájmem o vědu dlouhodobě stoupá. Do řad výzkumníků přibývají noví členi, vědecká sféra se rozšiřuje a se zvyšujícím se počtem vědecky činných osob se zvyšuje i produkce vědeckých publikací. Během svých výzkumů výzkumníci potřebují spolupracovat, ať už aktivně formou spoluautorství, nebo pasivně poskytnutím své související publikace k danému výzkumu. K této kooperaci jsou využívány webové bibliografické služby obsahující informace o autorech a jejich publikacích. Bohužel, tato data nebývají napříč službami konzistentní.

Cílem této práce je navrhnout a implementovat systém, který bude uchovávat a prezentovat informace o autorech, spadajících pod vybraná pracoviště (např. Katedra informatiky a výpočetní techniky), a jejich publikační činnosti. Zdrojem těchto informací budou bibliografické služby, ve kterých si autoři informace udržují sami nebo jsou udržovány automaticky. Nashromážděná data budou společně s dostupnými statistikami prezentována uživateli.

V následující druhé kapitole je čtenář uveden do oboru bibliografie a je mu vysvětlena její náplň, druhy a související bibliografické činnosti. Dále je zde seznámen s výběrem základních statistických metrik, kterými budeme měřit kvalitu autorů. Následuje kapitola třetí, popisující vykonanou rešerši existujících bibliografických služeb. Rešerše sestává z komplexního přehledu popisujícího vlastnosti a výhody jednotlivých služeb a možností zahrnutí v tomto systému. Produktem je opodstatněný výběr vyhovujících služeb, založený na jejich kritickém zhodnocení.

V kapitole čtvrté je popsána architektura klient-server a programové vybavení, jež bude pro implementaci systému využito. Kapitola pátá se zaobírá analytickým rozborem úlohy, jejím cílem, včetně způsobů splnění. Odůvodňuje užití dolování webu a osvětluje způsob extrakce požadovaných informací z těchto dat a jejich následného zpracování.

Kapitolou šestou je popsán průběh implementace systému na základě důkladného rozboru řešeného problému z předchozí kapitoly. Řeší vyhodnocení uživatelského vstupu aplikace, získání dat a jejich zpracování. Osvětluje i způsob prezentace shromážděných dat.

Sedmá kapitola je zaměřena na ověření funkcionality systému a jeho bezpečného užívání. V kapitol e poslední je obsažen popis výsledného produktu, jeho kvality a kritické hodnocení. Jsou zde také navržena i možná další vylepšení pro případný budoucí vývoj systému.

2 Bibliografie

Bibliografie je vědní obor, který se zabývá studiem publikací, a to vědeckých, spisovatelských i například novinářských. Zaobírá se metodami práce s nimi a nikterak rozborem jejich obsahů [1]. Bibliografie pohlíží na publikace jako na množiny děl a zkoumá metody jejich evidence a zpracování. Soubor těchto metod společně se zmíněnými množinami děl tvoří systematicky strukturovaný systém evidence publikační činnosti.

Publikace nejsou omezeny na fyzická díla, neboť v dnešní době tvoří velkou část publikace ve formě digitální a právě s těmito publikacemi bude bude systém pro správu evidence pracovat.

2.1 Druhy bibliografie

V závislosti na míře použité abstrakce se bibliografická práce liší v množství detailů analyzovaných v jednotlivých publikacích, čímž se dělí do dvou kategorií – analytická bibliografie a enumerativní bibliografie. Obě tyto kategorie jsou podrobněji popsány v [1], a proto zde uvádím pouze jejich stručný popis.

2.1.1 Analytická bibliografie

Analytická bibliografie bývá označována rovněž jako kritická bibliografie a zabývá se studiem publikační produkce. Dle sledovaného faktoru lze analytickou bibliografii dále dělit na:

(a) popisnou – posuzuje fyzickou stránku publikace (rozměry, formát, vazbu apod.),

(b) historickou – hledí na historické souvislosti a okolnosti, za kterých publikace vznikla,

(c) textovou – je analogií textové kritiky zkoumající literární původ publikace.

2.1.2 Enumerativní bibliografie

Enumerativní bibliografie vytváří kolekce publikací majících společnou vlastnost, kterou může být například užitý jazyk, časové období vzniku díla, místo

vydání, tématika apod. Z publikací jsou zpracovávány především jméno autora, název díla, místo vydání, jméno nakladatelství, rok vydání, ISBN, rozsah dokumentu včetně počtu stran, číslo vydání, použitá literatura aj. Tato práce se bude zabývat pouze enumerativní bibliografií, a proto budeme dále v textu enumerativní bibliografii označovat pouze jako „bibliografie“.

2.2 Bibliografická citace

Bibliografická citace je uspořádaná množina údajů (pokud možno) jednoznačně identifikující publikaci. Takovými údaji jsou například název publikace, jméno autora, rok a místo vydání aj. Uváděné údaje a jejich uspořádání se liší podle typu publikace a použité citační normy. Použití citace ve vlastní práci slouží k podložení správnosti prezentovaných informací a k identifikaci zdroje informace, jejíž nejsme autory.

Citačních norem existuje mnoho (např.: ČSN ISO 690, APA, CBE/CSE, ACS, NLM, IEEE aj.)[1]. Požadována je pro tuto práci norma ČSN ISO 690:2011 a formát BibTeX, jenž jsou požadovány zadáním práce.

Rovněž typů publikací existuje velké množství, avšak následující rešerše bibliografických služeb (viz kapitola 3) odhalí, že nám pro tuto práci postačí pouze jejich omezená množina. Použity budou: časopisecký článek, konferenční příspěvek (článek ve sborníku), kapitola knihy, kvalifikační práce a elektronický zdroj.

2.2.1 Citační norma ČSN ISO 690:2011

Citační norma ČSN ISO 690:2011 popsaná v [2] byla zavedena v březnu 2011 a stala se jednou z nejužívanějších citačních norem na českém území[2]. Dále následují ukázky formátů citační normy pro zvolené typy publikací čerpané z [2]. Údaje označené zeleně jsou nepovinné.

Citace časopiseckého článku

Tvůrce. Název článku. *Název časopisu*. Vedlejší názvy. Místo: nakladatel, rok, číslování, strany. ISSN.

[1] Více například na webu: `http://www.lib.vt.edu/find/citation/` nebo `http://knihovna.vsb.cz/kurzy/citace/08.html`

[2] Příklady užití citační normy ISO 690: `http://www.iso690.zcu.cz`

Citace konferenčního příspěvku
Tvůrce. Název příspěvku. In: *Název sborníku.* Vedlejší názvy. Místo: nakladatel, rok, strany. ISBN (nebo ISSN).

Citace kapitoly knihy
Tvůrce. Název příspěvku. In: Tvůrce publikace. *Název publikace.* Vedlejší názvy. Vydání. Další tvůrce. Místo: nakladatel, rok, strany. ISBN.

Citace kvalifikační práce
Tvůrce. *Název.* Vedlejší názvy. Místo vytvoření, rok vytvoření. Rozsah. Druh práce. Název školy. Vedoucí práce nebo školitel.

Citace elektronického zdroje
Tvůrce. Název příspěvku. Další informace o příspěvku. *Název časopisu.* Vedlejší názvy [typ nosiče]. Místo: nakladatel, číslování, strany [cit. datum citování]. ISSN. Dostupné z: DOI nebo adresa.

2.2.2 BiBTeX

BibTeX je nástroj generující seznam referencí v prostředí LaTeX a stejnojmenně bývá označován i jeho výstupní formát. Do odděleného souboru formátu `bib` se zaznamenávají ve specifickém tvaru informace o citovaných publikacích. Na základě tohoto souboru bývá generován seznam referencí. Všechny typy publikací jsou zapisovány obdobně, avšak mění se definice typu publikace (`@article` v případě zápisu časopiseckého článku v následujícím příkladu) a její atributy.

Příklad zápisu časopiseckého článku

```
@article{unikátní_identifikátor,
  author  = {jméno_autora},
  title   = {název_publikace},
  journal = {název_časopisu},
  year    = rok_vydání,
  number  = číslování,
  pages   = {rozsah_stran_časopisu},
  month   = měsíc_vydání,
  note    = {poznámka},
  volume  = číslo_svazku
}
```

Definice typu konferenčního příspěvku je tvořena řetězcem `@conference`, kapitoly knihy `@inbook`, kvalifikační práce `@masterthesis` či `phdthesis` (podle typu kvalifikační práce) a elektronického zdroje `@misc`. V příloze B uvádím příklady zápisu těchto typů publikací.

2.3 Scientometrie

Jak objasňuje [3]: „Scientometrie je vědecká nauka, která studuje evoluci vědy využitím kvantitativních indikátorů.“ Metriky sloužící pro získávání kvantitativních indikátorů se nazývají metodami citační analýzy. Mezi nejpoužívanější (a mnohdy bibliografickými službami využívané) metody citační analýzy patří počítání citací, h-index, g-index, varianty i-indexu aj. H-index, g-index a varianty i-indexu jsou metody ohodnocující autory publikací, které budou v práci využity. Níže uvádím jejich základní definice.

2.3.1 H-index

H-index nebo též Hirshův index autora [4] udává počet publikací h daného autora, které mají alespoň h citací, přičemž všechny jeho ostatní publikace mají méně než h citací. Definici popisují také [3, 5].

Definice: Seřadíme-li všechny autorovy vědecké publikace sestupně dle počtu citací, tak i označuje pořadí publikace a c_i počet citací publikace s indexem i, pak lze výpočet Hirschova indexu h zapsat vzorcem (1).

$$h = max(i), \text{ kde } c_i \geq i \tag{1}$$

2.3.2 G-index

Metrika g-index je obdobou h-indexu a autor Egghe jej popisuje v [6]. Určení velikosti g-indexu také začíná seřazením publikací sestupně dle počtu citací. G-index je pak takové nejvyšší pořadové číslo publikace, jehož druhá mocnina je menší nebo rovna součtu citací, které obdržely publikace s pořadovým číslem menším nebo rovným g-indexu. Jak dále [3] uvádí, g-index je vždy větší či roven h-indexu.

Definice: Seřadíme-li všechny vědecké publikace sestupně dle počtu citací, tak i označuje index publikace a c_i počet citací dané publikace, pak lze výpočet g-indexu g definovat vzorcem (2).

$$g = max(i), \text{ kde } i^2 \leq \sum_{j=1}^{i} c_j \tag{2}$$

2.3.3 I-index a jeho varianty

Roku 2011 byla společností Google Inc. zavedena metrika I10-index [7] a byla zahrnuta do statistik poskytovaných službou Google Scholar. Velikost i10-indexu určuje počet publikací autora, které mají alespoň 10 citací. Modifikacemi této jednoduché metriky vznikly například i5-index či i20-index popsané například v [8].

Definice: Máme-li seřazeny všechny vědecké publikace autora sestupně dle počtu citací, tak i označuje pořadí vědecké publikace a c_i označuje počet citací dané vědecké publikace s indexem i, pak lze iN-index iN definovat vztahem (3).

$$iN = max(i), \text{ kde } c_i \geq N \tag{3}$$

3 Rešerše bibliografických služeb

Tato kapitola popisuje provedenou rešerši bibliografických databází a služeb, přičemž kriticky hodnotí vlastnosti rešeršovaných služeb, významnost a případně množství zastoupených publikací, je-li známo. Výsledkem provedené rešerše je množina vybraných služeb, které svými vlastnostmi a možnostmi zahrnutí do tohoto systému pro správu publikací vyhovují nejvíce. Tento výběr bude v práci využit jako zdroj bibliografických dat.

3.1 Bibliografická služba

Webové bibliografické služby ve svých databázích shromažďují informace o publikační činnosti autorů. Nad obsaženými záznamy jsou obvykle vykonávány metody, které např. počítají statistiky autorů na základě jejich publikační činnosti. Tyto statistiky autory (resp. publikace) hodnotí a vyjadřují například míru jejich vědeckého přínosu. Obsah databází, včetně vypočtených statistik, je následně strukturovaně prezentován uživateli.

Služeb pro správu publikací existuje napříč internetem mnoho. Každá poskytuje jiný typ informací a je určena k jinému účelu. Některé jsou pro využití v této práci přívětivější, jiné naopak svojí špatnou přístupností k informacím činí jejich využití pro tuto práci nereálným (například registrace uživatelů, placený přístup apod.).

3.2 Rešerše dostupných služeb a jejich výběr

Původní ideou této práce bylo vyhledat bibliografické služby poskytující veřejná API[3] a zahrnout je do navrhovaném systému. Funkce veřejného API jsou určeny k používání a bývají dokumentovány, díky čemuž bych byl schopen plně využít dat poskytovaných služeb a vytěžit z nich maximum. Existence veřejného API ale není jediným kritériem pro výběr vhodných služeb a absence API nemusí znamenat jejich zavržení.

[3]API je zkratkou *Application Programming Interface* a jedná se o rozhraní, tedy kolekci, veřejně přístupných funkcí a metod, které mohou být programátorem využívány.

Mezi hlavní sledované faktory bibliografických služeb, které ovlivňují jejich využitelnost, patří:

a) Veřejné API

   Využíváním dostupného API můžeme maximalizovat objem informací, který je možné ze služby získat. Budou-li funkce navíc dokumentovány, proces získání informací bude významně urychlen.

b) Přístupnost a registrace

   Některé služby jsou limitovány povinnou registrací, zpoplatněním přístupu nebo striktní uzavřeností komunity. Všechny tyto restrikce významně omezují možnosti získání dat z dané služby.

c) Indexace

   Procesem indexace označujeme způsob získávání a mapování informací, které budou pro tento systém nezbytné. Zdrojem těchto informací může být člověk nebo automatizovaný systém. V případě, že informace obsažené v databázích služeb udržuje člověk, označujeme tento způsob indexace jako manuální. Druhou možností je automatická indexace, spočívající v autonomním prohledávání dat na internetu a aktualizování informací v databázi dané služby. Posledním způsobem je poloautomatická indexace, při níž jsou automaticky získaná data kontrolována člověkem. V případě člověka lze navíc rozlišovat mezi školeným pracovníkem, jenž se v oboru vyzná, a běžným uživatelem z vědecké i nevědecké sféry.

   Hlavní výhodou autonomně indexujícího systému bývá především větší počet nashromážděných dat. Tento způsob indexování má však i stinné stránky, především chybovost (chybné rozpoznání jmen či afiliací, duplicity apod.). Služby využívající manuální indexování publikací mívají zpravidla nižší chybovost, a tedy vyšší důvěryhodnost (za předpokladu pravdivosti vyplněných informací).

d) Typ služby

   Rozlišujeme mezi službami primárně zaměřenými na poskytnutí informací běžnému uživateli formou vyhledávače[4] a službami orientovanými na podporu komunikace a spolupráce členů komunity dané služby formou sociální sítě. Nás budou zajímat především ty služby, které budou vhodně poskytovat informace, ať už bude služba jakéhokoliv typu z výše uvedených.

[4]Vyhledávač je služba, která vyhledává zadaný textový řetězec (tzv. dotaz) ve svých databázích a prezentuje uživateli informace, které s řetězcem souvisejí.

e) Objem uchovávaných informací

V mnoha případech sama služba neuvádí, kolik bibliografických záznamů databáze obsahuje, a nezbývá, než důvěřovat službám nebo výzkumníkům, které službu využívali.

Tabulka 3.1 vyobrazuje základní srovnání vlastností služeb, které ovlivňují jejich využitelnost a přínos nejen pro tuto práci, ale i pro její následné využití v praxi. Poskytuje ucelený náhled na množinu všech rešeršovaných služeb z odlišných úhlů pohledu pro zdůraznění jejich rozdílů.

Informace obsažené ve zmiňované tabulce a následujícím textu byly získány především z oficiálních stránek služeb a publikací citovaných níže, ale také jsem vycházel ze svých vlastních poznatků nabytých rešerší jednotlivých služeb. Rešerši byly podrobeny služby BibSonomy, CiteSeerX, DBLP, GoogleScholar, Mendeley, ResearchGate a Web of Science.

3.2.1 BibSonomy

BibSonomy[5] je kombinací sociální sítě a vyhledávače bibliografických dat, který prohledává záznamy z bibliografické databáze této služby. Hlavní funkce této služby spočívá ve sdílení publikací mezi uživateli, včetně případných záložek označených hashtagy[6]. Klíčová slova vyplněná uživatelem do vstupního vyhledávače, tzv. dotaz, jsou vyhledána službou v databázi na základě shody s názvem publikace, hashtagem, jménem autora apod.

Kolik publikací je indexováno BibSonomy na svých stránkách neuvádí, avšak podle [9] počet indexovaných publikací k roku 2016 převyšuje 3.45 milionů a počet aktivních uživatelů převyšuje 8.900.

Služba neumožňuje seskupení informací ve formě profilové stránky obsahující informace o autorovi, instituci apod. Další nevýhodou webových stránek této služby je nedostatečné prolinkování stránek, které ztěžuje jak manuální, tak automatické procházení.

Vzhledem ke zmíněným nedostatkům shledávám BibSonomy za nevyhovující pro svoji práci.

3.2.2 CiteSeerX

Jak uvádí CiteSeer^x[7] na webové stránce se svou historií[8], služba známá pod původním názvem *CiteSeer* vznikla v roce 1997 a stala se první digitální kni-

[5]Web: `https://www.bibsonomy.org`

[6]Tzv.: hashtag obecně slouží ke kategorizaci a seskupení logicky souvisejících položek.

[7]Web: `http://csxstatic.ist.psu.edu`

[8]Web: `http://csxstatic.ist.psu.edu/about/history`

hovnou a vyhledávačem používajícím autonomní indexaci. Výhodou tohoto přístupu je maximální množství získaných dat při minimálním úsilí. Má to však i stinné stránky, a to především chybovost, duplicity aj. (viz část 3.2).

Indexovány jsou především články z časopisů a sborníků, technické zprávy a knihy. Podle informací uvedených v [3] služba k roku 2011 shromažďuje přes 32 milionů záznamů a dle [10] se by se mohlo jednat o vhodný zdroj dat pro citační analýzu.

3.2.3 DBLP

DBLP (*DataBases and Logic Programming*)[9] je manuálně udržovaná bibliografická databáze publikací zaměřených na obory počítačových věd. Díky manuální správě obsahu lze předpokládat nižší chybovost (např.: eliminace duplicit, chybějících či částečných informací atd.). K vyhledávání a datům DBLP je poskytován volný a bezplatný přístup.

Služba uvádí několik vizualizací statistik obsažených záznamů[10]. Například prezentuje množství informací nashromážděných v jednotlivých letech, viz obrázek A.1 v příloze A. K roku 2016 počet záznamů převyšoval 3.7 milionů. Vzhledem k ročnímu nárůstu indexovaných záznamů (například za rok 2016 přibylo přes 260 tisíc publikací), jak informuje obrázek A.2 v příloze A, má služba velký potenciál i v budoucích letech. K únoru 2017 služba obsahuje především konferenční příspěvky a časopisecké články a v menším zastoupení i např. kvalifikační práce, knižní kapitoly aj.

Na základě těchto informací jsem se rozhodl službu zahrnout do této práce.

3.2.4 Google Scholar

Veřejný vyhledávač bibliografických dat Google Scholar[11] je od společnosti Google[12] a patří k bezkonkurenčním velikánům současných automatizovaně udržovaných bibliografických služeb. Kromě samotného obsahu publikací jsou indexována také spjatá metadata. Mezi jeho výhody patří volná dostupnost dat, přičemž bezpodmínečná není ani registrace.

Služba je strukturována do profilů osob obsahujících publikace autorů, afiliace a základní automaticky počítané statistické údaje.

[9] Web: `http://dblp.uni-trier.de`
[10] Web: `http://dblp.uni-trier.de/statistics`
[11] Web: `https://scholar.google.com`
[12] Google Inc., web: `www.google.com`

Jak služba uvádí na svých webových stránkách[13], indexovány jsou publikace nejrůznějších typů (např. kvalifikační práce, příspěvky ve sbornících, články, knihy, abstrakty aj.) ze všech vědních oborů. U každé publikace služba zpřístupňuje minimálně název publikace, jména autorů, abstrakt a odkaz na zdroj informace.

3.2.5 Mendeley

Mendeley[14] je služba autory označovaná jako správce referencí a akademická sociální síť. Jako své hlavní kvality uvádí úschovu vědeckých publikací v zabezpečeném cloudovém úložišti[15] na serverech služby Mendeley a jejich přístupnost odkudkoliv, čímž je uživatelům umožněna práce i na cestách – psaní poznámek, generování citací v nejběžněji podporovaných formátech apod. Další užitečnou funkcí může být sdílení publikací mezi autory a možnost týmové spolupráce. Mendeley nabízí i bezplatnou verzi s omezenou velikostí uživatelova úložiště. Služba poskytuje veřejné API [11].

Podle informací na oficiálních webových stránkách[16] službu k prosinci 2016 využívá přes 6 milionů uživatelů. U publikací je udržována škála automaticky počítaných statistik[17] (např. počet jejich citací, zobrazení, přečtení apod.). Služba se zdá být vhodným zdrojem informací pro tuto práci.

3.2.6 ResearchGate

Služba ResearchGate[18] (RG) byla podle informací na oficiálních stránkách[19] založena roku 2008 dvěma lékaři a vědcem z oblasti počítačových věd a zprovozněna byla v květnu téhož roku. Autoři zdárně propojili uživatele ve vědeckou komunitu sdílející výzkumy a poznatky, aby usnadnili a podnítili komunikaci a spolupráci. Jak dále uvádí[20], RG je jedna z největších manuálně udržovaných bibliografických služeb. Dle oficiálních webových stránek se do ResearchGate zapojilo přes 12 milionů uživatelů. Ani o publikace nashromážděné tímto zdrojem není nouze, služba se pyšní více než 100 miliony publikací a každoročním průměrným přírůstkem přes 2 miliony nových publikací.

[13] Web: `https://scholar.google.com/intl/en/scholar/about.html`
[14] Web: `https://www.mendeley.com`
[15] Tato úložiště umožňují úschovu dat na serverech a bývají multiplatformně přístupná.
[16] Web: `https://www.mendeley.com/research-network/community`
[17] Web: `https://www.mendeley.com/reference-management/stats`
[18] Web: `https://www.researchgate.net`
[19] Web: `https://www.researchgate.net/about`
[20] Web: `https://www.researchgate.net/press`

Data této sociální sítě jsou udržována primárně manuálně (vyjma statistik), neboť každý uživatel si spravuje sám informace, jimiž je prezentován.

Webové stránky prezentující data návštěvníkům jsou organizovány v profily autorů, institucí apod. obsahující mnoho užitečných informací. Zároveň jsou tyto profily vhodně strukturovány a prolinkovány. Díky uvedeným vlastnostem by využití služby ResearchGate v této práci mohlo být nejen prospěšné, ale také snadné.

3.2.7 Web of Science

Poslední službou popsanou v této rešerši je jedna z nejstarších a nejuznávanějších bibliografických databází, databáze Web of Science[21] (či jen WoS). Její vznik se datuje již k roku 1955. „Aktuálně WoS shromažďuje vědecké články z více než 12 000 vlivných časopisů a více než 15 000 konferenčních sborníků," jak uvádí [5]. Databáze WoS zastřešuje 7 online databází[22], jejichž data se skládají především z konferenčních sborníků, odborných časopisů a knih nejrůznějších vědeckých disciplín, jejichž počet převyšuje 250. Podle informací obsažených na svých stránkách[23], WoS obsahuje přes 59 milionů záznamů.

Bohužel, WoS neposkytuje možnost volně dostupného vyhledávání, protože se jedná o službu placenou, a proto ji nebudeme v práci využívat.

3.3 Průzkum vhodných služeb

Na základě provedené rešerše jsem se rozhodl blíže seznámit se službami ResearchGate, GoogleScholar, DBLP a Mendeley.

3.3.1 Průzkum ResearchGate

Osobní profil, tedy veřejná strukturovaná stránka každého člena komunity sociální sítě ResearchGate, obsahuje vedle informací, které si autor udržuje samostatně (např.: jméno, příjmení, tituly, kontaktní informace, vzdělání, znalosti a dovednosti, vědecké zaměření, zájmy, dosavadní vědecká činnost

[21] Web: `https://webofknowledge.com`

[22] Databáze spravované WoS: Science Citation Index (SCI), Social Sciences Citation Index (SSCI), Arts & Humanities Citation Index (A&HCI), Index Chemicus, Current Chemical Reactions, Conference Proceedings Citation Index: Science, Conference Proceedings Citation Index: Social Science and Humanities.

[23] Web: `http://wokinfo.com`

aj.), také automaticky počítané statistiky jako např. RG score, Impact Faktor aj. Dlouhé seznamy publikací jsou řešeny stránkováním[24].

Registrace do služby není povinná, ale existují soukromé profily, které nejsou veřejně přístupné. Tyto profily nebudou do práce zahrnuty. Vedle osobních profilů uživatelů existují i profily institucí seskupující příslušníky instituce. Na těchto profilech nalezneme (vedle seznamu příslušných autorů) souhrnné informace a statistiky vztahující se k celé množině autorů.

Služba neposkytuje API, a proto je pro nás přínosem, že je na profily institucí nahlíženo velice podobně jako na profily uživatelů. Přínosem je také stejná šablona, do které jsou profily vykreslovány, a tedy jednotná struktura stránek usnadňující dolování informací.

Kombinace kvantity dat s jejich snadnou přístupností činí tuto službu prioritním zdrojem dat pro tuto práci.

3.3.2 Průzkum Google Scholar

Navzdory minimálně 7 let trvajícím žádostem uživatelů o zveřejnění API, Google Scholar veřejné API neposkytuje. Služba neprozrazuje množství záznamů v databázi. Mezi nejčastěji indexované typy publikací se řadí články, technické zprávy, kvalifikační práce, knihy nebo výběr webových stránek s akademickým zaměřením. Ve většině případů se jedná o placené publikace, u kterých je zdarma přístupný pouze abstrakt.

I tato služba uspořádává informace v profily autorů a institucí se stránkováním. Profily jsou doplněny o automaticky počítané statistiky typu počet publikací, počet citací, h-index a další. Profily jsou obohaceny i o grafické znázornění statistik ve formě grafů, v nichž je kupříkladu vyjádřena závislost počtu obdržených citací na čase (viz obrázek A.3 v příloze A). Vedle profilů samotných autorů lze zobrazit množinu autorů příslušejících instituci. Na rozdíl od RG však neobsahuje toto zobrazení žádnou přidanou hodnotu ve formě souhrnných statistik.

Služba poskytuje rozšířené vyhledávání, jímž lze hledání blíže specifikovat, např. určit zda hledáme autora (resp. publikaci), jehož jméno se přesně shoduje s hledanou frází, či zda je fráze podmnožinou jména autora apod.

Služba Google Scholar se zdá být kvalitním zdrojem automaticky indexovaných informací a bude v této práci zahrnuta.

[24]Objemné stránky uživatele zatěžují nejen objem přenesených dat, ale i nepřehledností a nepraktičností. Takové stránky bývají zpravidla ošetřeny *lazy loadingem* (tedy postupným načítáním obsahu závisejícím na skrolování s využitím AJAXu) nebo právě stránkováním, které objemnou stránku rozdělí na několik podstránek.

3.3.3 Průzkum DBLP

Plně bezplatná bibliografická databáze DBLP je zcela přístupná a k jejímu používání tedy nebude zapotřebí žádná registrace či poplatek.

Webové stránky neobsahují stránkování. Nevýhodou služby je absence indexace citací a stránek institucí se seznamy autorů s institucí spojených, a proto budou muset být odkazy na jednotlivé autory předány výhradně uživatelem systému. DBLP (stejně jako služba GS) umožňuje specifikovat více kritérií hledání než pouze hledanou frázi.

Služba DBLP obsahuje velké množství dat prezentovaných na přehledně strukturovaných stránkách, a proto bude DBLP v této práci využívána.

3.3.4 Průzkum Mendeley

Poslední zkoumanou službou je služba Mendeley, která se od ostatních liší svým přístupem k práci s publikacemi. Uživatelé mají možnost ukládat si články, jež mají rozečtené či v plánu studovat, které mohou mezi sebou sdílet.

Podmínkou využívání služby Mendeley k vývoji vlastní aplikace službu využívající, je registrace aplikace u služby Mendeley. Aplikaci jsou přiřazeny identifikátory využívané při autentizaci se serverem. Jako jedna z mála služeb Mendeley poskytuje API s podrobnou dokumentací [11], a proto by registrace s autentizací, předcházející využívání služeb, neměla být problém.

Průzkum Mendeley zahrnoval nejen teoretickou studii dokumentace API a toho, co služba poskytuje, ale také praktickou část, v níž jsem API testoval. Během praktického průzkumu jsem narazil na několik komplikací, z nichž bych zdůraznil především neaktuální dokumentaci, která byla problematická zvláště v místech popisu autentizace aplikace na serveru Mendeley. V dokumentaci se nachází již nevyužívaný autentizační protokol *OAuth 1.0*[25] a o aktuálně používané verzi *OAuth 2.0*[26] není v dokumentaci ani zmínka. K procesu autentizace jsou vyžadovány přesné URL adresy, na něž se odesílají požadavky. Po mnoha různých kombinacích a modifikacích URL adres zmíněných v dokumentaci se podařilo testovací aplikaci spojit se serverem.

Nicméně, čím déle testuji, co mi je služba Mendeley schopna poskytnout užitečného či jedinečného, tím více se přikláním k jejímu vyřazení z výběru bibliografických služeb. Služba nabízí zcela jiný aspekt využití, protože není orientována na prezentaci osoby jakožto autora publikací, ale slouží ke shromažďování publikací k osobním účelům. Poskytovaná data tedy nevyhovují požadavkům práce a služba nebude v práci využita.

[25] Web: `https://oauth.net/1`
[26] Web: `https://oauth.net/2`

služba	**API**	**přístup, registrace**	**indexace**	**typ služby**	**statistiky**
BibSonomy	ano	bezplatný, nepovinná	manuálně	vyhledávač, sociální síť	3.4+ mil. záznamů 8.900+ uživatelů
CiteSeerX	ne	bezplatný, nepovinná	autonomně	vyhledávač	32.2+ mil. záznamů
DBLP	ano	bezplatný, nepovinná	manuálně	vyhledávač	3.7+ mil. záznamů
Google Scholar	ne	bezplatný, nepovinná	autonomně	vyhledávač	–
Mendeley	ano	bezplatný, nepovinná	–	sociální síť	6+ mil. uživatelů
ResearchGate	ne	bezplatný, nepovinná	manuálně	sociální síť	12+ mil. uživatelů 100+ mil. publikací
Web of Science	ano	placený, povinná	manuálně	vyhledávač	59+ mil. záznamů

Tabulka 3.1: Přehled majoritních vlastností rešeršovaných služeb.

3.4 Vybrané bibliografické služby

Vybraní zástupci bibliografických služeb, jenž jsem se rozhodl blíže prozkoumat, patří k největším službám svého druhu, pokrývají široké spektrum publikací a zahrnují oba typy indexování dat – manuální i automatické.

Bohužel, služby vyhovující přístupností a způsobem indexování dat neposkytují API a naopak. Proto se odkláním od původní myšlenky využívání veřejných API a dostávám se k získávání dat přímo z webových stránek technikami tzv. text nebo web mining (více popsáno v části 5.1).

Majoritními faktory, které ovlivnily výběr množiny služeb, jež budou v práci využívány, byly především dostupnost informací, způsob indexování dat a bezplatný přístup k datům, díky němuž užívání navrhované aplikace zůstane bezplatné.

Pro účely této bakalářské práce nejvíce vyhovují služby ResearchGate, GoogleScholar a DBLP, s nimiž budu dále pracovat. Myslím, že tato kombinace by se mohla vhodně doplňovat způsobem indexování a poskytovanými informacemi.

3.5 Poskytované informace

Rešerše bibliografických služeb mi poskytla přehled o službami prezentovaných informacích o autorech a publikacích. Žádná ze služeb neposkytla pouhý výpis seznamu autorů a jejich publikací. Každá ze služeb poskytuje dle svého uvážení přidanou hodnotu, kterou mohou být statistiky autorů (např. h-index, g-index atd., viz část 2.3) či publikací (např. počet citací, spoluautorů, stran publikace atd.), grafy, podrobné informace o autorech (resp. publikacích), včetně autorových dovedností či specializace atd.

Rešerše také odhalila, že existují publikace, u nichž některé služby neindexují řádně všechny údaje potřebné pro naše účely. Avšak vzhledem k volbě tří velkých bibliografických služeb lze předpokládat, že se z těchto služeb ve většině případů povedou agregovat potřebné informace o publikacích a jejich autorech pro potřeby systému (tj. pro výpočet statistik, výpis publikací v požadovaném formátu apod.).

V následující části je popsán výběr vhodných informací, které nás zajímají, a které jsou žádané či potřebné k výpisu ve formátech ČSN ISO 690:2011 a BibTeX.

3.5.1 Volba důležitých informací

Jedním z úkolů tohoto systému je filtrovat výpis autorů a jejich publikací na základě instituce autora. Proto první informací, která nás zajímá, je právě instituce autora.

K normalizovanému výpisu publikací v souladu s citační normou ČSN ISO 690:2011, viz část 2.2, bude zapotřebí shromáždit dostatek informací. Požadované údaje se dle typu publikace liší. Seznam atributů autorů a publikací napříč jejich typy je následující: jméno autora či autorů, název publikace, rok a místo vydání, ISBN[27] a ISSN[28] publikace, číslo vydání, druh kvalifikační práce, jméno nakladatele, jméno vedoucího práce či školitele, název školy, rozsah publikace, rok vytvoření, typ nosiče a url adresa publikace (je-li dostupná online). V případě, že je publikace součástí publikace jiné (např. kapitola knihy, časopisecký článek, či příspěvek ve sborníku), bude zapotřebí i název publikace v níž je obsažena (např. kniha, časopis či sborník). Vedle základních informací budou stahovány akademické tituly, počet citací publikace, URL adresa profilové fotografie autora a specializace autora.

Obvykle služby nabízejí i základní statistická ohodnocení publikací či autorů, ta však budou v práci počítána na základě dostupných informací a nebudou stahována.

[27]EN: *International Standard Book Number*
[28]EN: *International Standard Serial Number*

4 Softwarové prostředky a metody

Produktem této práce bude PHP knihovna, jejíž funkcionalita bude demonstrována prostřednictvím webových stránek, které společně s danou knihovnou budou tvořit webovou aplikaci. Webová aplikace se dělí obecně na dvě části:

(a) **backend** – serverová část (tj. program pracující na serveru) a administrace, kterou bude knihovna řízena,

(b) **frontend** – klientská část (tj. program fungující na klientovi, tedy v prohlížeči uživatele), která bude uživateli prezentovat výsledné informace.

Níže jsou popsány programovací jazyky, technologie a knihovny, které budou k vývoji systému požity.

4.1 Backend

Administrace bude ovládat získávání a zpracování informací. Serverová část také řeší přípravu výstupu vykreslovaného v klientovi. Tato část práce informuje o technologiích, které budou na straně serveru použity.

4.1.1 PHP

Objektově orientovaný a multiplatformní programovací jazyk PHP je využívaný nejčastěji k programování logiky dynamických internetových aplikací na straně serveru. Syntaxe jazyka je popsána např. v [12, 13].

Samozřejmou součástí vývoje softwaru je testování aplikace. Jak je uvedeno v [14], testování neznamená pouze kontrolu správnosti několika případů užití na samém konci vývoje, ale obnáší testování jednotlivých částí aplikace vznikajících už během vývoje. Díky tomu pak máme kontrolu nad každou testovanou částí kódu a ve finále i nad kódem celým. Jedním ze způsobů testování jsou jednotkové testy, tzv. unit testy, jež jsou velice podobné JUnit testům programovacího jazyka Java[29].

[29] Web: `https://java.com`

4.1.2 Databázový systém

Pro uložení dat bude využit nejčastěji užívaný databázový systém MySQL a relační model dat. Interakce s databází bude umožněna jazykem SQL.

V souladu s informacemi nabytými z [15], volím pro tuto práci druh databázového úložiště InnoDB. Toto robustní úložiště podporující transakce a silnou referenční integritu umožňuje uchovávat velké množství dat.

4.1.3 Knihovna PDO

Knihovna PDO zastřešuje rozdílnost implementací komunikace s různými druhy databázových systémů a poskytuje tak jednotné API pro jejich správu. Nespornou výhodou knihovny PDO je (za předpokladu správného použití) ochrana vůči SQL injection[30] technikou vázání parametrů na parametrizované dotazy (namísto přímého vložení vstupu uživatele do SQL dotazu), jak popisuje [16]. Pro svoje kvality se PDO stává nejvyužívanější PHP knihovnou pro komunikaci s databázemi, jejíž využití je doporučeno i v [17].

Pro snazší tvorbu dotazů bude v práci využita i knihovna DSQL[31], jejíž funkcí je stavba SQL dotazů a sami tvůrci ji označují jako *SQL query builder*.

4.1.4 XML

Jazyk XML (*eXtensible Markup Language*[32]) je značkovací jazyk, jehož primárním využitím je, vedle výměny dat mezi aplikacemi, také definice struktury. Konkurovat by mu mohl formát JSON[33] s obdobnými vlastnostmi a hojným rozšířením napříč aplikacemi všeho druhu. S formáty XML i JSON se v programovacích jazycích snadno operuje. Ani jeden z formátů nedominuje žádnou vlastností, které by mohlo být při vývoji systému v jazyce PHP využito. a proto bude v aplikaci použit formát XML.

4.1.5 Composer

Composer[34] je nástroj pro správu závislostí, jehož úkolem je zajistit snadno a automatizovaně přítomnost požadovaných knihoven v odpovídajících verzích, na nichž bude tento systém závislý.

[30] SQL injection je úmyslný útok na databázi s cílem získání či smazání dat.
[31] Web: `https://github.com/atk4/dsql`
[32] Web: `https://www.w3.org/XML`
[33] Web: `http://www.json.org`
[34] Web: `https://getcomposer.org`

4.2 Frontend

Frontend představuje prezentační vrstvu vykreslující obsah. Součástí frontendu bude i skript ve skriptovacím jazyce JavaScript (JS), který bude zajišťovat interakci vyvíjené aplikace. V případě JS aplikací je provádění logických operací na klientovi akceptovatelná, jinak bývá prováděna na serveru. Výpočet v prohlížeči by měl být maximálně jednoduchý a neměl by obtěžovat klienta výpočetní náročností a zdlouhavým čekáním.

Průvodcem nových technologií (zejména HTML5 a CSS3, ale i JavaScriptové knihovny jQuery) mi je již po několik let kniha [18], z níž jsem čerpal v následujícím textu.

4.2.1 HTML a šablonovací systém Twig

Značkovací jazyk HTML (*HyperText Markup Language*) definuje strukturu webových stránek. Stránky tohoto systému budou definovány na serveru využitím HTML5[35], které oproti předešlým verzím poskytuje nové formulářové a sémantické prvky a nové funkce, například geolokace, *Web Socket*, *Web Workers*, *Local Storage*, offline úschovu dat aplikací a mnoho dalšího popsaného v [18, 19].

K sestavování výsledné stránky je využit šablonovací systém Twig[36] [20]. Šablonovací systém bývá používán u projektů, jejichž stránky jsou z důvodu velkého počtu generovány dynamicky.

4.2.2 CSS

Jazyk CSS (*Cascading Style Sheets*)[37] definuje vzhled vykreslované stránky. Aktuální verze CSS3 rozšiřuje dosavadní množinu CSS selektorů[38], obohacuje CSS např. o *Media Queries*, definující zobrazení webových stránek dle zařízení a šířky okna webového prohlížeče, a další. Dostatek informací, včetně příkladů, je poskytnuto v [18, 19].

4.2.3 JavaScript

Skriptovací jazyk JavaScript[39], užívaný především k obsluze interakce uživatele s webovými stránkami, dokáže řešit i složitější animace nebo například

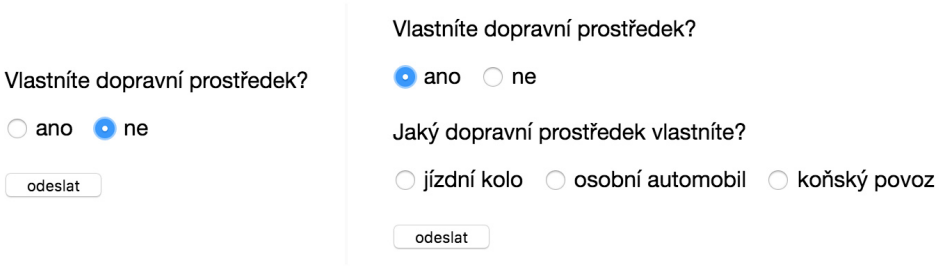
[35] Web: `https://www.w3.org/TR/html5`
[36] Web: `http://twig.sensiolabs.org`
[37] Web: `https://www.w3.org/Style/CSS`
[38] CSS selektor je řetězec, jímž se adresují prvky HTML struktury
[39] Web: `https://www.javascript.com`

posílat asynchronní dotazy na server (tzv. AJAX, *Asynchronous JavaScript and XML*). Odpověďmi těchto dotazů bude dynamicky měněn obsah stránek této webové aplikace.

JavaScriptový framework jQuery[40] usnadňuje práci zvláště s DOM (*Document Object Model*)[41]. Další výhodou jazyka je např. obsáhlá dokumentace s příklady a kvalitní podpora tvořená obrovskou komunitou a množstvím literatury [18, 19, 21].

4.2.4 AdvancedForm

JQuery plugin AdvancedForm byl vyvinut společností Etnetera a.s.[42] a je využíván pro vývoj dynamických formulářů. V těchto formulářích umožňuje definovat podmíněné zobrazení jakéhokoliv prvku HTML struktury formuláře na základě hodnoty jiného formulářového prvku. Na následujícím obrázku 4.1 je funkcionalita demonstrována příkladem, v němž je zobrazení volby dopravního prostředku závislé na volbě dopravního prostředku.



Obrázek 4.1: Ukázka funkcionality knihovny AdvancedForm – na základě výběru vlastnictví dopravního prostředku je (resp. není) zpřístupněna volba jeho typu.

4.2.5 Select2

Data prezentovaná uživateli budou filtrována podle kritérií. Minimálně výběr pracoviště či pracovišť autorů bude využit formulářový prvek `<select>` s atributem `multiple`.

Select2[43] je JavaScriptová knihovna, která transformuje běžný `<select>` na množinu HTML elementů (viz obrázek 4.2 znázorňující transformaci ele-

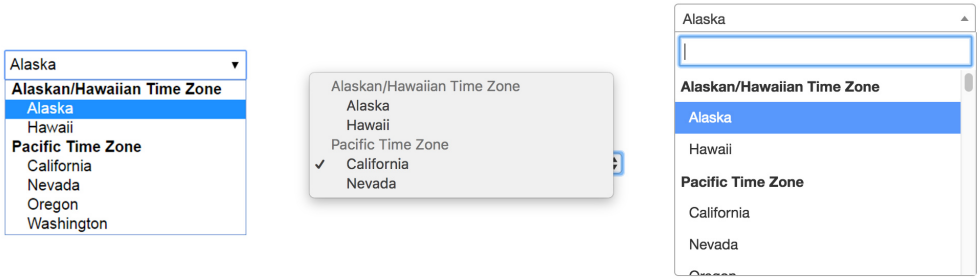
[40] Web: https://jquery.com

[41] Document Object Model je objektová reprezentace XML nebo HTML dokumentu ve formě stromové struktury.

[42] Web: `https://www.etnetera.cz`

[43] Web: `https://select2.github.io`

mentu). Tento nově vzniklý formulářový prvek tvoří nejen designově uspokojující provedení, ale především pohodlnější s novými možnostmi. Například je umožněno filtrování položek zadáním textového řetězce, pohodlný výběr (potažmo zrušení výběru) položek atd.



Obrázek 4.2: Ukázka transformace HTML elementu `<select>` knihovnou `Select2` v porovnání s běžným `<select>` pod dvěma rozdílnými operačními systémy (vlevo a uprostřed). Vzhled a funkcionalita nově vzniklého `<select>` (vpravo) je na platformě nezávislá.

5 Analýza a návrh řešení

Úkolem systému pro správu publikací bude získat potřebné informace o autorech a jejich publikační činnosti z vybraných bibliografických služeb. Prostřednictvím uživatelského vstupu bude systému předána informace, co a odkud bude získáváno. Uživatelský vstup bude sestávat především z URL adres profilových stránek autorů, o kterých chce uživatel získat informace. Systém projde kolekci příslušných stránek a získá z nich požadovaná data. Nad těmito daty budou spočteny základní statistiky, které budou společně s daty uchovány v databázi. Obsah databáze bude následně prezentován uživateli formou seznamu autorů vč. vypočtených statistik a seznamu publikací vypsaných v normalizovaném tvaru.

5.1 Získávání informací

Pro zpracování byly zvoleny služby ResearchGate, GoogleScholar a DBLP, viz kapitola 3. Ani jedna z nich neposkytuje API, a proto budou data dolována přímo z odpovídajících stránek.

Při budoucím využívání aplikace by vybrané služby nemusely být dostatečné, například z důvodu vyřazení některé ze služeb z provozu, zamezení přístupu (třeba formou zabezpečení typu Captcha[44] aj.) nebo pouhou potřebou zahrnutí nového systému. Proto při návrhu architektury aplikace bude kladen důraz na její obecnost a snadnou rozšiřitelnost.

5.1.1 Dolování webu

Dolování webu neboli *web mining* je autonomní proces, jehož cílem je získání dat z internetových stránek. Lze jej dělit na tři základní druhy:

(a) **dolování obsahu webu** (*web content mining*) zabývající se dolováním prezentovaných dat, tedy obsahu,

(b) **dolování struktury webu** (*web structure mining*) orientující se na zjišťování provázanosti stránek,

(c) **dolování způsobu užití webu** (*web usage mining*) získávající informace o způsobech použití (IP, autorizace apod.).

[44] Jedná se o sadu testů různého charakteru rozlišujících člověka od robota, tj. skriptu vydávajícího se za člověka. Sadou testů se zamezuje robotovi v přístupu na stránky, odesílání formuláře apod.

Pro získání informací bude v této práci využito dolování obsahu webu. Postup získávání dat touto cestou není flexibilní na změny struktury stránek, neboť Web mining je silně závislý na struktuře webu, a jakákoliv změna webu může do budoucna znamenat komplikace. Dolovat se budou řádově stovky stránek a to z mnoha důvodů, z nichž bych zmínil zejména využití vícera služeb a velké množství informací o autorech a publikacích, které chceme dolováním získat. S tím se pojí i stránkování objemného obsahu webových stránek služeb.

Mnohaprvkové seznamy položek, jimiž zde mohou být především autoři či jejich publikace, bývají strukturovány do vícera podstránek využitím stránkování. Dlouhý seznam je tedy rozdělen do částí na oddělených stránkách. Nejčastěji se setkáváme se stránkováním na základě parametrů URL adresy stránky. Adresa sestává obvykle z parametru označujícího číslo stránky a případně z počtu položek na stránce, viz následující příklady:

```
http://example.com/article/new-research-released?page=3

http://example.com/article/new-research-released?page=3&size=10
```

kde `page` označuje číslo stránky a `pagesize` definuje počet položek na stránce. Jiný možný způsob oddělení stránek je analogií způsobu předchozího s tím rozdílem, že číslo stránky není předáváno parametrem URL adresy, ale je přímo její součástí, jak je znázorněno níže:

```
http://example.com/article/new-research-released/1/

http://example.com/article/new-research-released/2/
```

Další možnou (a nejméně častou) realizací stránkování je formou zcela jedinečných URL adres, ve kterých se nemění jen hodnota parametru, viz následující příklad:

```
http://example.com/article/abc/

http://example.com/article/def/
```

5.1.2 Dolování bibliografických služeb

Bibliografická data budou získávána z webových stránek vybraných bibliografických služeb. Ty však nepočítají se zahlcováním požadavky. Na tomto principu funguje tzv. útok DOS (*Denial Of Service*), který úmyslným zahlcením serveru požadavky server přetíží a vyřadí z provozu. Proto bývají přístupy na servery evidovány. Běžnými praktikami v případech neobvyklé

aktivity bývá uživateli zobrazena *captcha* nebo je uživateli zcela odříznut přístupu ke službě. Při vývoji i používání tohoto systému bude potřeba služby nepřetěžovat, proto jednotlivé požadavky bude vhodné prokládat časovými prodlevami v řádu sekund.

Na základě výše uvedených informací lze předpokládat, že doba běhu procesu dolování nebude otázkou několika sekund, ale bude se pohybovat v řádu minut až hodin. Abychom nemuseli proces opakovat při každém použití systému, budou data uchovávána v databázi (viz části 4.1.2 a 6.5.1).

Služby ResearchGate a GoogleSchlar poskytují možnost přiřazení autora k instituci a obě umožňují seznam autorů, kteří jsou k instituci přiřazeni, vypsat na profilové stránce instituce. Seznam autorů obsahuje rovněž URL adresy směřující na profilové stránky autorů. Seznam autorů bude systém autonomně procházet a bude z něj vyjímat URL adresy autorů, které budou následně zahrnuty v procesu získávání informací. Možnost výběru této autonomního průchodu a kolekce URL adres bude ponechána uživateli prostřednictvím uživatelského vstupu.

5.1.3 Volba parseru

K zisku informací z jednotlivých služeb bude přistupováno jednotně. Obsah každé stránky bude stažen z internetu funkcemi jazyka PHP. Co však už jednotné není, je struktura internetových stránek jednotlivých služeb, tj. jejich podstránek, rozložení a označení jednotlivých prvků layoutu[45], zabezpečení přístupu k samotnému webu apod. Po získání struktury požadovaných internetových stránek následuje vyjmutí požadovaných informací. Modul, který se bude starat o extrakci informací, bude v této práci nazván parser[46].

Naskýtají se dvě možnosti přístupu k parsování struktur jednotlivých stránek. První možností je implementace samostatného parseru pro každou ze zvolených služeb. Pomineme-li, za předpokladu obdobného postupu, větší pracnost a časovou náročnost implementace duplicitních konstrukcí, zůstává zde problém, která podrývá myšlenku jednoduchosti zahrnutí nové služby. Pro každou novou službu je potřeba vytvořit nový parser. Druhou možností je vytvoření obecného parseru, který se bude řídit sadou pravidel definovaných konfiguračním souborem ve formátu XML.

Abychom co nejvíce usnadnili případné zahrnutí nové služby do systému, bude pro tuto práci implementován obecný parser pro všechny vybrané služby.

[45]Layoutem se označuje struktura či rozvržení (v tomto případě webové stránky).

[46]Parser je algoritmus, jenž na základě kolekce pravidel dělí sekvenci znaků na logické bloky textu.

5.1.4 Konfigurace parseru

Internetové stránky jednotlivých služeb se liší. Úkolem konfiguračního souboru parseru je dostatečně popsat strukturu jednotlivých stránek tak, aby byl parser schopen obsah stránek projít a vyjmout potřebné informace.

Konfigurační soubor bude muset obsahovat především informace o organizaci podstránek služby (kterými může být navíc i realizace stránkování mnohaprvkových seznamů) a stavbě příslušných URL adres, definici hledaných informací na jednotlivých stránkách a případně existenci stránky instituce se seznamem příslušných autorů. Tyto informace potřebné pro chod parseru budou definovány strukturou XML a hodnotami jejích značek (a případně atributů). Mimo obecných řetězců budou tyto značky obsahovat také XPath selektory a regulární výrazy.

XPath je zkratkou pro *XML Path Language* [22], kterým lze adresovat části HTML (či obecně XML) dokumentů. Vedle samotných prvků dokumentu umí adresovat také jejich obsahy (tj. hodnoty) i atributy. Adresující výraz vzniklý sekvencí klíčových slov jazyka je nazýván XPath selektor. XPath selektor může být použit např. pro selekci obrázku a přístup k jeho atributu definujícímu zdrojovou URL adresu souboru s obrázkem, viz následující příklad 5.1 a 5.2:

Zdrojový kód 5.1: **Příklad XPath:** HTML struktura.

```
<section class="personal">
    <aside>
      <img src="img/profile_pic.jpg">
    </aside>
    <div>
      <h1 class="name"> ... </h1>
      <p class="address"> ... </p>
    </div>
  <!-- ... -->
</section>
```

Zdrojový kód 5.2: **Příklad XPath:** selektor adresující URL obrázku.

```
//section[contains(@class, 'personal')]/aside/img/@src
```

Jazyk XPath bude využit při samotné extrakci žádaných hodnot dokumentu, protože v konfiguračním souboru parseru budou hodnoty jednotlivých atributů, které jsou ze stránek získávány, adresovány tímto jazykem.

Pro validaci (např. správnosti vstupu uživatele), extrakci (oříznutí vydolované hodnoty) a modifikaci (odebrání nežádoucích částí získaného řetězce apod.) budou v této práci použity regulární výrazy. Regulární výraz[47] je speciální řetězec představující určitý vzor či masku textového řetězce, který se vyskytuje se téměř ve všech programovacích jazycích.

Stránkování seznamů autorů a publikací je ve vybraných službách realizováno dvěma způsoby – jedinečnými adresami nebo předáním čísla stránky v parametru. Protože oba způsoby vyžadují jiné informace a oba způsoby bude potřeba v konfiguračním souboru rozlišit. Stránkování seznamu realizované předáním čísla stránky v parametru URL bude označováno jako *3-prvková paginace*, neboť pro její realizaci bude zapotřebí tří parametrů:

- **parametr čísla stránky** – parametr URL adresy uchovávající hodnotu aktuální stránky,
- **inkrement** – navýšení hodnoty čísla stránky pro přechod na stránku následující,
- **zastavovací podmínka** – definována XPath selektorem detekujícím konec stránkování.

Tento způsob využívá např. služba GoogleScholar, viz následující příklad:

```
https://scholar.google.cz/citations?user=BaJ-Ql8AAAAJ
            &hl=cs&cstart=60&pagesize=20
```

kde `cstart` je analogií čísla stránky (i v tomto případě značí nějakou vzdálenost od začátku seznamu, zde však představuje vzdálenost vyjádřenou počtem položek nikoliv počtem stran[48]) a `pagesize` počet položek na stránce.

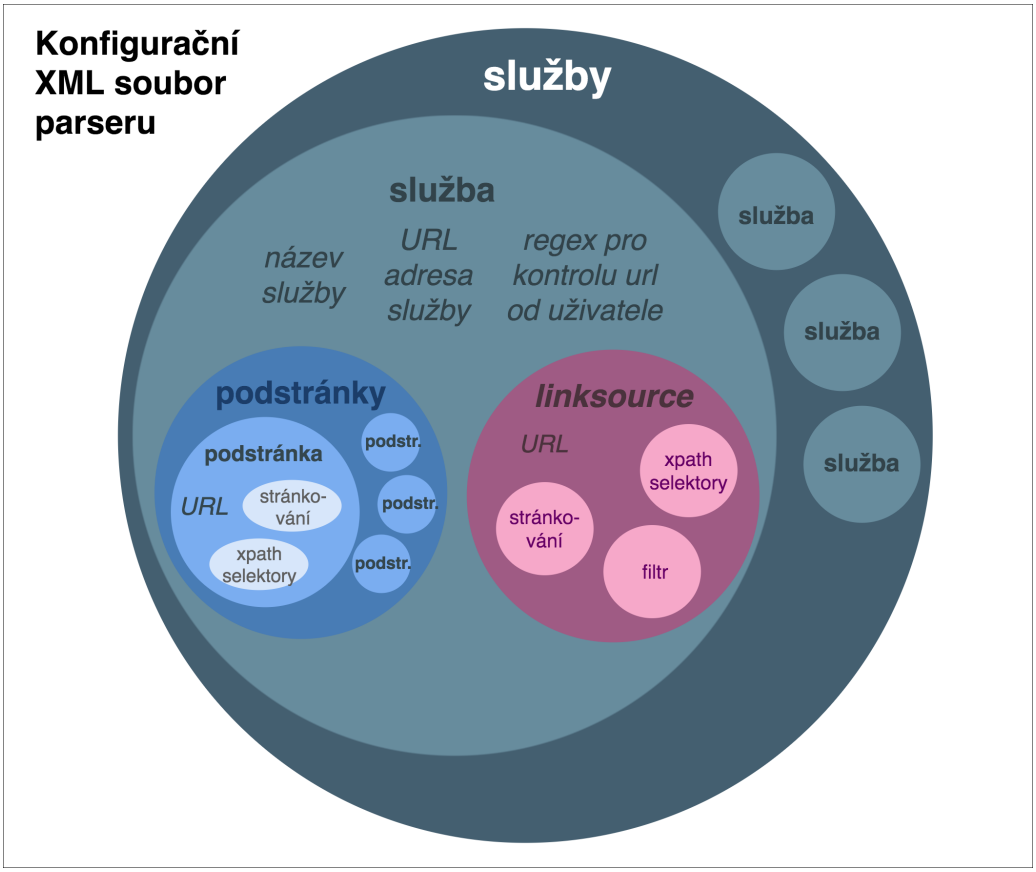
Druhý využívaný způsob realizace stránkování (formou jedinečných URL adres) je využíván např. službou ResearchGagte. Tento způsob stránkování bude nazýván *2-prvkovou paginací*, protože vyžaduje 2 parametry:

- **selektor následující stránky** – XPath selektor pro získání adresy následující stránky,
- **zastavovací podmínka** –XPath selektor pro detekci konce stránkování.

[47] Web: `http://www.regularnivyrazy.info`

[48] Např.: pro `example.com?page=3` značí page počet stran od začátku seznamu, avšak v případě `example.com?cstart=260` značí počet položek seznamu od jeho začátku.

Následující obrázek 5.1 znázorňuje návrh struktury konfiguračního souboru parseru.



Obrázek 5.1: Vizualizace návrhu struktury konfiguračního souboru parseru.

5.2 Zpracování získaných informací

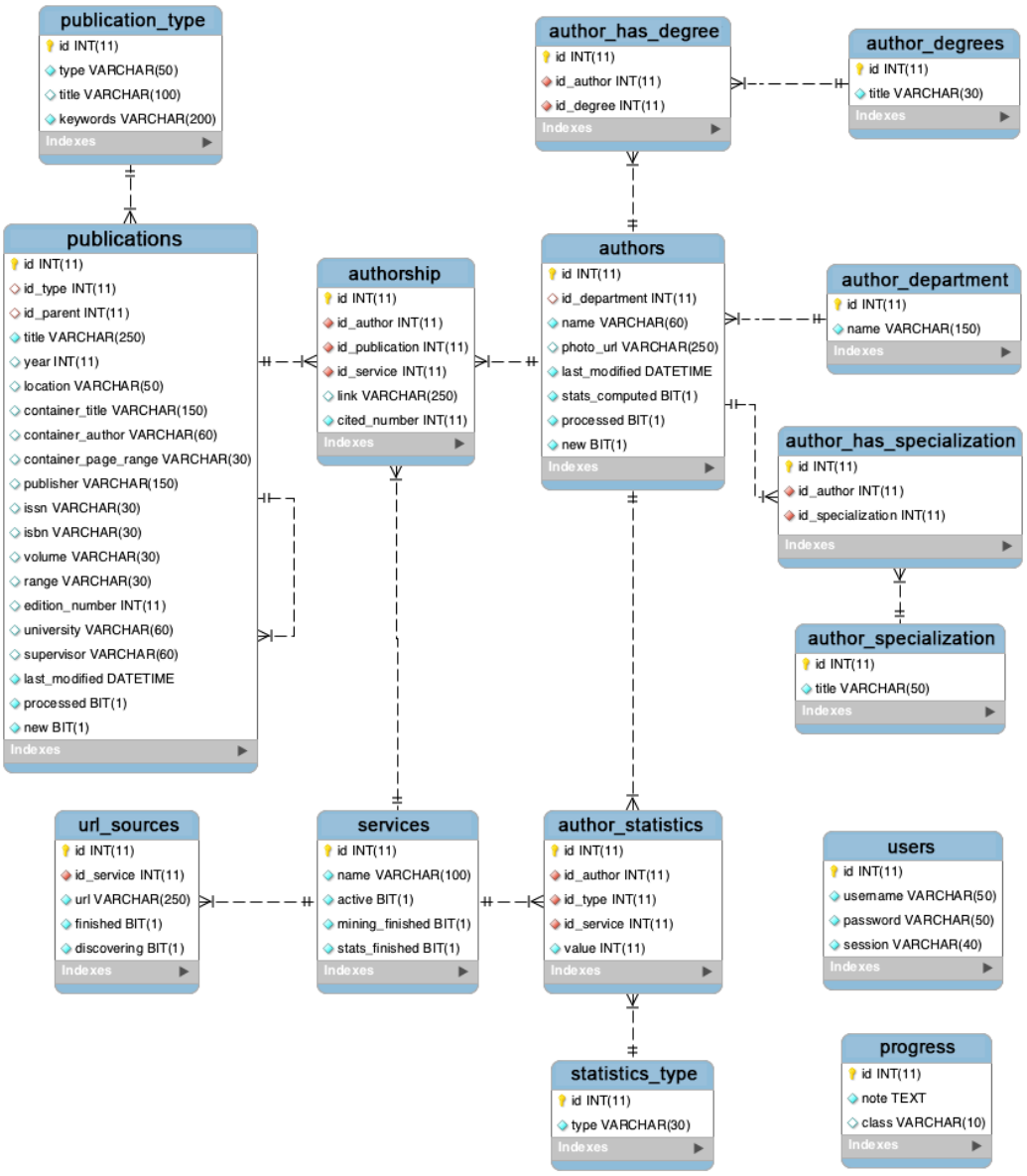
Na základě uživatelského vstupu tvořeného URL adresami a příznaky, jaké služby mají být prozkoumány, budou jednotlivé stránky stahovány a následnou aplikací XPath selektorů budou extrahována důležitá data.

Informace budou získávány ze tří bibliografických služeb. Většina akademiků se prezentuje ve víceru službeb. Může se stát, že informace nebudou napříč službami konzistentní a například názvy publikací nebudou identické. Obdobná komplikace může nastat i v rámci jedné služby v případě, kdy spoluautoři uvedou publikaci s odlišnými či neúplnými informacemi.

Má aplikace bude se zmíněnými odchylkami počítat a podobnost publikací kontrolovat. Zkoumána bude především podobnost názvů publikací

společně se shodou spoluautorů a roků vydání. Kontrola podobnosti názvů publikací bude spočívat ve srovnání četností jednotlivých slov a dle míry jejich shody bude rozhodnuto o totožnosti porovnávaných publikací. Totožné publikace budou sjednoceny.

Z informací získaných ze služeb budou počítány statistiky, kterými jsou h-index, g-index, i5-index a i10-index, viz část 2.3. Na základě informací, jež budou uchovávány v databázi (viz část 3.5.1), jsem se rozhodl pro následující návrh struktury databáze, viz ERA model[49] 5.2 databáze.

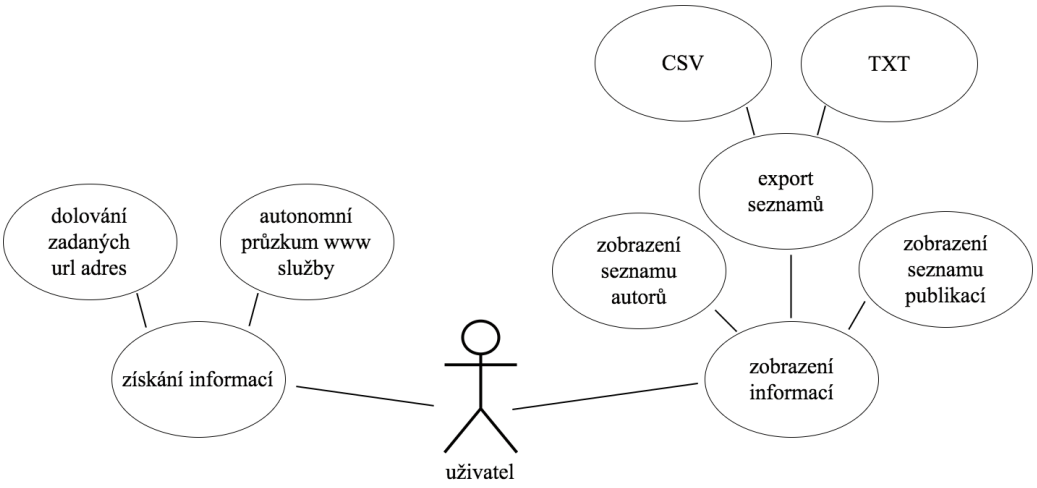


Obrázek 5.2: ERA model navržené databáze systému.

[49]ERA model je obvyklé schéma znázorňující strukturu relační databáze.

5.3 Návrh uživatelského rozhraní

Součástí systému pro správu publikací budou, vedle knihovny implementované v jazyce PHP, také webové stránky. Protože samotný chod vyvíjené knihovny bude situován na serveru a s knihovnou budeme moci interagovat prostřednictvím webových stránek, tak lze o tomto celku hovořit jako o webové aplikaci. Mimo ovládání budou webové stránky sloužit i k prezentaci získaných dat. Při návrhu webové aplikace je vhodné si nejprve uvědomit, co všechno bude uživatel po této aplikaci požadovat. Jednotlivé požadavky jsou nazývány případy užití a tyto případy lze vizualizovat diagramem případů užití (*use case diagram*) následovně, viz obrázek 5.3.



Obrázek 5.3: Diagram užití vytvářené webové aplikace.

Z diagramu jsou zřejmé dva hlavní případy užití této aplikace, tj. získání informací a zobrazení informací. Každý z těchto dvou případů užití bude oddělen na samostatné webové stránce. Stránka, na které bude uživatel moci ovládat získávání informací bude označena jako administrace. Na druhé stránce budou uživateli zobrazovány získané strukturované informace.

Při návrhu grafického uživatelského rozhraní (*GUI*) bude kladen důraz především na přehlednost a jednoduchý design, viz tzv. *wireframy*[50] na obrázcích G.1 až G.3 v příloze G.

[50]Wireframem je označován návrh layoutu aplikace s důrazem na její použitelnost.

6 Implementace systému pro správu publikací

Tato kapitola vysvětluje postup implementace systému pro správu publikací na základě provedeného analytického rozboru úlohy a jejích majoritních dílčích částí. Popisuje realizaci těchto částí od přijetí požadavků zadaných od uživatele, přes proces dolování dat z webových stránek bibliografických služeb, následné zpracovávání a uchovávání informací, až po sjednocování publikací, kalkulaci statistik a prezentaci výsledků renderováním šablon s daty.

6.1 Architektura aplikace

Aplikace není založena na žádném frameworku[51], přesto je postavena na třívrstvé architektuře a její adresářová struktura odděluje logicky související třídy. Serverová část této aplikace je navržena tak, aby byla schopna současně s chodem reagovat na asynchronní požadavky klienta, příkladem může být požadavek na seznam jmen autorů konkrétní služby.

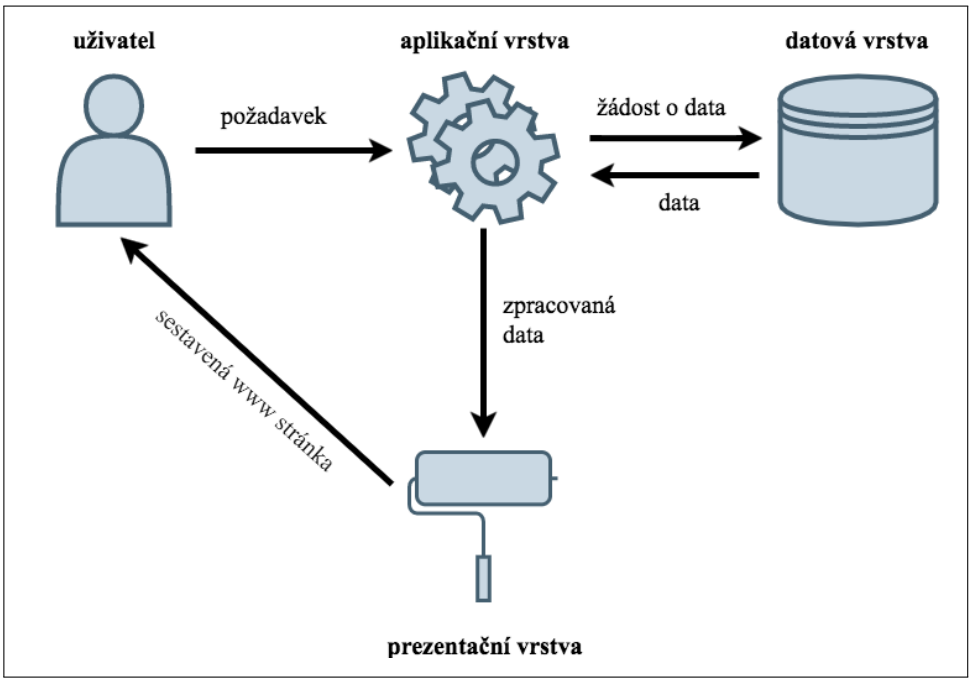
6.1.1 Třívrstvá architektura

Struktura tohoto systému bude založena na třívrstvé architektuře, neboť byla vyvíjena s ohledem na snadnou rozšiřitelnost, přehlednost, intuitivní ovládání a údržbu. Funkcionalita aplikace je dekomponována do tří spolupracujících komponent, tedy datové vrstvy, aplikační (či logické) vrstvy a prezentační vrstvy. Datová vrstva tohoto systému se skládá z databáze a tříd databázi obsluhujících. Ve vrstvě aplikační je zahrnuta všechna logika, výpočty a také samotné řízení chodu aplikace. Prezentační vrstva je určena k prezentování dat uživateli.

Výhodou třívrstvé architektury je vzájemná nezávislost jednotlivých komponent[52], lepší přehlednost a snazší rozšiřitelnost a údržba kódu. Následující obrázek 6.1 znázorňuje oddělení a kooperaci jednotlivých vrstev.

[51] Framework je podpůrný základ aplikace vedoucí k vhodné organizaci projektu při jeho vývoji. Součástí mohou být i knihovny a návrhové vzory.

[52] Změna implementace jedné komponenty má minimální vliv na komponenty ostatní. Například změna systému řízení báze dat a implementace obslužných tříd, tj. datové vrstvy, bude mít minimální vliv vrstvy aplikační a prezentační.



Obrázek 6.1: Vizualizace komponent třívrstvé architektury.

6.2 Konfigurace aplikace

Oddělení nastavení aplikace do samostatných souborů v adresáři `config/` usnadňuje některé změny funkcionality aplikace bez nutnosti zdlouhavého dohledávání odpovídajících částí ve zdrojovém kódu. V tomto adresáři se nalézá konfigurace aplikace s konstantami definovaným výčtem jednotlivých požadavků, přicházejících na server které server a které rozpoznává (`/config/config.inc.php`). Dále je zde umístěna konfigurace databázového spojení (`/config/dbconnection-config.inc.php`), konstanty s názvy databázových entit a jejich atributů (`/config/db-config.xml`), chybové hlášky (`/config/err-config.xml`), číselná reprezentace jednotlivých typů metrik (`/config/stats-config.xml`), konstanty s názvy jednotlivých šablon, jimiž se vykresluje obsah, a atributy s defaultními hodnotami formulářových prvků (`/config/view-config.xml`).

6.2.1 Obecná konfigurace aplikace

Konfigurován je například typ (`text/html`) a znaková sada (`UTF-8`) obsahu odesílaného serverem klientovi prostřednictvím hlavičky[53] `Content-Type` pro-

[53]HTTP hlavičky (*HTTP headers*) obsahují metadata související s obsahy zpráv, které jsou vyměňovány mezi klientem a serverem.

tokolu HTTP (*HyperText Transfer Protocol*)[54]. Lokalizace, úroveň reportovaných oznámení a chybových hlášení, *autoloading* tříd[55] apod.

Zdůrazním ještě konfiguraci časového limitu běhu skriptu, který byl nastaven na dobu neomezenou, neboť u této aplikace lze předpokládat dobu chodu v řádech hodin.

6.2.2 Konfigurace databázového spojení

Informace potřebné pro připojení k databázi jsou rovněž separovány v samostatném souboru. Následující informace jsou v souboru definovánax formou , přičemž zde uvádím pouze přehled hodnot, se kterými je pracováno:

název hostitele:	`127.0.0.1`
port:	`3306`
název databáze:	`publication_management_system`
uživatelské jméno:	`root`
heslo:	`root`

Název hostitele (tzv. *hostname*), číslo portu a název databáze společně tvoří tzv. DSN (*Data Source Name*)[56] ve tvaru:

```
mysql:host=127.0.0.1:3306;dbname=pms_database
```

6.2.3 Konfigurace parseru služeb

Součástí konfigurace aplikace je i konfigurace parseru pro dolování informací z bibliografických služeb `/config/services-config.xml`. V analytické části 5.1.4 jsme si nastínili obsah a strukturu konfiguračního XML souboru, viz obrázek 5.1.

V průběhu vývoje aplikace byly do tohoto souboru zahrnuty další informace, které se staly z důvodu změn ze strany služeb nezbytnými pro chod parseru. Například byla službou ResearchGate implementována ochrana zamezující přístup robotům[57] formou ReCaptcha[58]. Z tohoto důvodu byla do

[54]HTTP protokol definuje přenos hypertextových dokumentů formátu HTML a obvykle využívá port `80`. Existuje v několika verzích. Komunikace je iniciována dotazem klienta.

[55]*Autoload* je technika automatického importování (`include`) instancovaných tříd, díky němuž není třeba každý soubor s třídou *includovat*.

[56]DSN je datová struktura obsahující informace o databázi, k níž je přistupováno. Je vyžadována pro iniciování spojení s databází knihovnou `PDO`.

[57]Robotem je označován program vykonávající rutinní činnosti místo svého majitele.

[58]ReCapcha je ochrana typu capcha, která byla vyvinuta společností Google Inc. Web: `https://www.google.com/recaptcha/intro/invisible.html`

konfigurace zahrnuta data `cookies`[59], která byla překopírována z paměti webového prohlížeče po úspěšné autentizaci reálného uživatele služby. Díky tomuto řetězci lze dočasně obejít konstrukci ReCaptcha.

Podrobný popis konfigurace parseru bibliografických služeb se nalézá v příloze D a vzorový konfigurační soubor v příloze E.

6.3 Spuštění aplikace

Stránka s administrací (viz část 6.6.2) je vyhrazena řízení procesu získávání bibliografických dat. Přístup k administraci je podmíněn přihlášením uživatele. Administrace uživateli umožňuje zadání:

(a) kolekce URL adres profilů autorů, která je získána jejich manuálním vložením (tj. `custom_urls`, viz zdrojový kód 6.1),

(b) kolekce autorů tvořené výběrem z nabídky autorů objevených dopředným průchodem služby (tj. `discover_service_authors`, viz zdrojový kód 6.1 a níže),

(c) kolekce příznaků indikujících, jaké služby mají být autonomně procházeny (tj. `search_services`, viz zdrojový kód 6.1),

(d) volby ze dvou možností –uchování dat v databázi nebo jejich zobrazení (tj. `pms_process`, viz zdrojový kód 6.1),

(e) volby ze dvou možností – aktualizace existujících záznamů v databázi nebo jejich přeskakování (tj. `existing_update`, viz zdrojový kód 6.1).

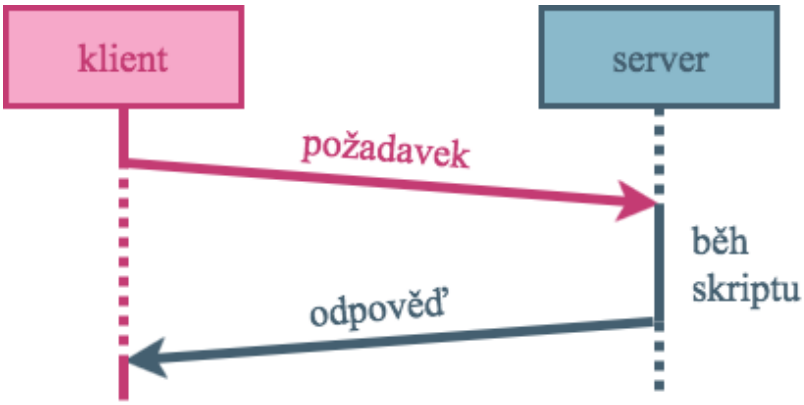
Kolekce **(a)** je získávána manuálním vepsáním URL adres autorů či jejich zkopírováním z internetového prohlížeče. U služeb ResearchGate a GoogleScholar je umožněna volba autonomního průchodu stránek dané instituce (viz část 5.1.2). Kolekce **(c)** je tvořena výběrem z těchto služeb. Stránky se seznamem autorů instituce poskytují také možnost dopředného průchodu webovými stránkami služby a získání jmen autorů (vč. URL adres) ještě před zahájením procesu získávání dat. Získaná jména autorů jsou uživateli poskytnuta s možností jejich výběru a zahrnutí do procesu dolování. Tato možnost **(b)** nabízí pohodlnější způsob výběru kolekce autorů, než kopírováním URL adres jednotlivých autorů. Dalším volitelným parametrem **(d)** je, zda mají být získaná data uchována v databázi či pouze zobrazena uživateli. V případě ukládání dat do databáze má uživatel navíc možnost výběru, zda

[59]Cookies jsou data zasílaná serverem klientovi, který data uchová v internetovém prohlížeči, a jsou serveru odesílána a serverem zpracovávána při dalších návštěvách webu. Mohou sloužit například k identifikaci uživatele apod.

se mají informace o již existujících autorech a jejich publikacích aktualizovat (v případě, že se zpracovávaný autor v databázi již vyskytuje) nebo se mají pouze zavádět do systému autoři noví **(e)**. Průběh procesu získávání dat a jejich následného zpracování je vizualizován, viz část 6.6.2.

6.3.1 Zpracování vstupních dat

Webové aplikace jsou založeny na protokolu HTTP, který pracuje na principu dotaz-odpověď. Spuštění skriptu této knihovny je iniciováno dotazem klienta, konkrétně odesláním formuláře s konfiguracemi, viz obrázek 6.2. Data z formuláře jsou serializována a zaslána na server metodou `HTTP POST`.



Obrázek 6.2: Inicializace komunikace serveru s klientem.

Získaná data jsou v prostředí backendu zpřístupněna proměnnou `$_POST`[60] a mohou vypadat například následovně, viz zdrojový kód 6.1.

[60] Automatická superglobální proměnná přístupná z jakékoliv části skriptu PHP.

Zdrojový kód 6.1: Příklad uživatelského vstupu aplikace.

```
$_POST = [
    [search_services] => [
        [0] => researchgate
    ],
    [discover_service_authors] => [
        [0] => http://example.com/user/abc
    ],
    [custom_urls] => [
        [0] => http://anotherexample.com/user/def
        [1] => http://example.com/user/ghi
    ],
    [pms_process] => mine_store,
    [existing_update] => update
]
```

U URL adres zadaných uživatelem jsou odstraněny případné duplicitní hodnoty. Předtím než budou adresy jakkoliv dále zpracovávány, je potřeba množinu adres nejprve validovat, neboť má uživatel možnost zadat obecný textový řetězec. Každé validní URL adrese je následně přiřazena služba na základě výskytu interního identifikátoru služby definovaného v konfiguračním souboru (`<service name>`.

Na základě rozpoznaných služeb z URL adres jsou instancovány objekty třídy `Service`, které s přiřazenými URL adresami manipulují, tj. specifičtěji je validují regulárním výrazem definovaným v konfiguračním souboru (`<checkRegex>`, viz příloha C), uchovávají příslušné URL adresy, řídí proces dolování atd. Každé službě je rovněž nastaven příznak, zda bude autonomně prozkoumána. Validované URL adresy jsou uchovány v databázi v tabulce `url_sources`.

6.4 Realizace získávání informací ze služeb

Má-li instance třídy `Service` příznak autonomního průzkumu, pak je průzkum služby, spočívající v průchodu stránkovaným seznamem autorů instituce, započat. Postupně jsou na základě pravidel definovaných konfiguračním souborem skládány URL adresy jednotlivých stran seznamu, jejichž struktura je stahována. Z těchto struktur jsou vyjímány URL adresy profilů autorů. Procházíme pouze seznamem těchto autorů, samotné stránky profilů dolovány nejsou. Získané URL adresy jsou následně uchovány v databázi pro budoucí Web mining. Postup stahování struktur a vyjmutí dat

bude vysvětlen v části 6.4.1.

Nastane-li situace, kdy zbudou v databázi nezpracované URL adresy z posledního použití (taková situace nastane, pakliže je chod přerušen), pak při prvním dalším spouštění aplikace je uživatel o vzniklé skutečnosti informován a je mu vypsán seznam URL adres, které nebyly zpracovány a budou zahrnuty do aktuálního zpracovávání. Tuto situaci modeluje obrázek G.6 v příloze G, v němž je vedena nedokončená činnost u všech 3 služeb.

6.4.1 Proces dolování a extrakce dat

V této části je předpokládána znalost podrobné struktury konfiguračního souboru, viz příloha C. Konfigurační soubor definuje pro každou službu podstránky `<subpages>`, jejichž URL adresy jsou na základě tohoto souboru skládány z bázové adresy služby (`<baseUrl>`) a atributu přípony URL dané podstránky (`urlSuffix`) následovně:

```
<baseUrl> + urlSuffix
```

Struktura stránky nacházející se na každé takto sestavené adrese je stahována. Nachází-li se na stránce stránkovaný seznam, pak URL adresy jednotlivých stran pro *3-prvkovou paginaci* jsou skládány takto:

```
<baseUrl> + urlSuffix + "?" nebo "&" + <pageParam> + "=" + [strana]
```

kde výběr mezi "?" a "&"závisí na tom, zda adresa již nějaký parametr obsahuje, a `[strana]` je číslo procházené stránky, jež byla získána součtem:

```
[hodnota předchozí stránky] + <pageParamIncrement>
```

Pro demonstraci uvedu příklad:

```
"http://example.com" + "/publications" + "?" + "page" + "=" + "1"
            http://example.com/publications?page=1
            http://example.com/publications?page=2
                                ...
```

V případě *2-prvkové paginace* je URL adresa na následující stránku získána vyjmutím jejím selektorem `<nextPageLink>` ze struktury stránky aktuální. Takto je stránkovaný seznam procházen, dokud zastavovací podmínka tvořená XPath selektorem `<stopCondition>` nedetekuje konec seznamu. Takový stav se značí tím, že vyjmutý obsah tímto selektorem bude prázdný.

K samotnému stažení obsahu stránky slouží třída `DataLoader`, která rovněž hlídá časové prodlevy mezi jednotlivými požadavky zasílanými na

server bibliografické služby. Časové prodlevy jsou voleny náhodně z intervalu $< 3; 9 > \cap \mathbb{Z}$ (tj. $\{3; 4; 5; \ldots; 9\}$) sekund, abychom co nejméně zatěžovali servery služeb a co nejvíce eliminovali možnost zamezení přístupu k jejich webovým stránkám. `DataLoader`u je předána URL adresa, ten zkontroluje časový odstup od posledního stahování (abychom nezahltili server služby požadavky a eliminovali podezřelou aktivitu, viz část 5.1.2) a případně počká. Obsah stránek je stahován funkcí `curl_exec()` knihovny `cURL`[61].

DOM reprezentace stránky usnadňuje manipulaci s HTML strukturou, a proto je stažená struktura převedena do její objektové reprezentace. Třída jazyka PHP `DomDocument` se o parsování struktury postará sama. Z takto reprezentované struktury postupnou aplikací příslušných XPath selektorů vyjímáme požadované informace, viz následující výňatek zdrojového kódu 6.2:

Zdrojový kód 6.2: Ukázka aplikace XPath selektoru na vytvořenou DOM reprezentaci dokumentu.

```
1 $xp = new DomXPath($dom);
2 $value = (string) $xp->query($xpath);
```

6.5 Proces zpracování získaných dat

Získanými informacemi bibliografických služeb jsou inicializovány instance `Author` (reprezentující autory) a `Publication` (reprezentující publikace) s náležitými atributy odpovídajícími sloupcům příslušných tabulek `authors` a `publications`, viz ERA model na obrázku 5.2.

6.5.1 Uchování dat

Než bude s instancemi autorů a publikací dále nakládáno, jsou předány databázi obsluhujícím třídám datové vrstvy (především `AuthorDatabaseManager` a `PublicationDatabaseManager`[62]). Informace jsou do databáze vkládány formou nových záznamů nebo jsou v souladu s příznakem o aktualizaci existujících záznamů (viz část 6.3.1) záznamy aktualizovány anebo přeskakovány. Záznam o autorství s odkazem na publikaci a počty citací publikace

[61] Web: `http://php.net/manual/en/book.curl.php`

[62] `AuthorDatabaseManager` obsluhuje tabulky vázající se k autorovi, tedy tabulky `authors`, `authorship`, `author_degrees`, `author_department`, `author_specialization`, `author_has_degree` a `author_has_specialization`, zatímco `PublicationDatabaseManager` obsluhuje pouze `publications` a `authorship`.

jsou uchovávány pro každou službu zvlášť. Vhodně zvoleným dotazem lze z databáze získat informaci, odkud byla která publikace získána.

Součástí databáze je tabulka `users` seskupující záznamy s přihlašovacími údaji potřebnými pro přístup ke stránce s administrací. Heslo a autentizační token jsou chráněny hashovací funkcí SHA1[63]. Autentizační token je klientovi zasílán serverem po úspěšném přihlášení a klientem je uchováván v paměti prohlížeče pro opětovné ověření identity.

Kromě zmíněných tříd datová vrstva obsahuje třídy `Db` (která využívá knihovnu `DSQL` pro stavbu SQL dotazů), `ProgressDatabaseManager` obsluhující log sestavovaný v průběhu procesu a `ServiceDatabaseManager` obsluhující tabulku `services`. Tabulka `author_statistics` obsahující statistické údaje o autorech je obsluhována třídou `StatisticsDatabaseManager` a tabulku `users` spravuje `UserDatabaseManager`.

6.5.2 Sjednocování identických záznamů

Jakmile jsou veškerá data ze všech služeb získána a uložena v databázi, začneme s nimi pracovat. Jak bylo již vysvětleno v části 5.2, v databázi se mohou vyskytovat duplicitní záznamy – ať už budou téměř identické či nikoliv. Téže publikaci je přiřazen jeden či více záznamů v `publications` a to například z důvodu nepatrně rozdílných názvů. Řešením tohoto jevu je sdružení souvisejících záznamů tak, že jeden záznam je vybrán jako majoritní či rodičovský a zbylé na něj budou uchovávat referenci. V praxi hovoříme o cizím klíči nad jedním sloupcem `id_parent` téže tabulky `publications` odkazujícím na rodičovský záznam.

Modul, který sdružování vykonává, je nazván `PairProcessor`. Funkce tohoto modulu je začíná dotazem do databáze a ziskem `id` autorů, majících alespoň jednu publikaci, které není přiřazeno žádné `id_parent` (mající tedy novou publikaci). Abychom neporovnávali každou publikaci s každou, čímž by výrazně stoupla časová náročnost, budeme porovnávat hledanou publikaci pouze v omezené množině. Má-li tato publikace v databázi duplicitní záznam, bude vlastněna stejnou množinou autorů. Autorství je nejdůležitější informace vedená o publikaci, na jejíž správnost se lze spolehnout, protože je autory hlídána. Proto sestavíme množinu publikací majících stejnou množinu autorů jako publikace nespárovaná. Touto množinou iterujeme, přičemž v každé iteraci kontrolujeme podobnost publikace nespárované s publikací procházenou. Podobnost je vyhodnocována na základě četností výskytů jednotlivých slov v názvech a na základě shody roků vydání publikací.

[63]Hashovací funkce převádí libovolný vstup na řetězec tzv . DSn ní délky v nečitelné podobě. Tento proces je nezvratný.

Výskyt každého slova názvu publikace jedné je ověřován v názvu publikace druhé. Podíl počtu pozitivních ověření (tedy počtu společných slov) s celkovým počtem slov vyjadřuje míru podobnosti p názvů v intervalu $p \in < 0; 1 >$. Druhým ověřovaným faktorem je již zmiňovaná shoda roků vydání. Jsou-li vedeny roky vydání u obou publikací a shodují-li se, jedná se o publikaci totožnou, pakliže podobnost názvů p činí alespoň 80 % (tj. $p \geq 0.8$). Nemají-li publikace roky vydání vedené či neshodují-li se, pak jsou publikace identické, jestliže je podobnost názvů alespoň 90% (tj. $p \geq 0.9$).

Každá publikace z množiny publikací, které byly vyhodnoceny jako identické, odkazuje na *rodičovskou publikaci*. Databázové záznamy této množiny publikací obsahují stejnou hodnotu `id_parent`, konkrétně hodnotu `id` rodičovské publikace. V následujícím příkladu, viz tabulka 6.1, jsou publikace s ID rovným 1, 2 a 4 (odkazující na 1 publikaci) spárovány a publikace s ID rovným 1 (odkazující sama na sebe) je publikací rodičovskou:

id	**id_parent**	**...**
1	1	...
2	1	...
3	3	...
4	1	...

Tabulka 6.1: Příklad spárovaných publikací.

6.5.3 Počítání statistických údajů

Nashromážděná data tímto systémem tvoří statistický soubor pro čtyři následující základní metriky autorů, které budou počítány pro každou službu zvlášť – h-index, g-index, i5-index a i10-index. Každému autorovi tedy bude spočteno 12 hodnot[64], které budou uchovány v tabulce `author_statistics`.

Výpočet všech metrik probíhá obdobně. Pro každou službu a každého autora zvlášť je postupně sestavena sestupná posloupnost počtů citací autorových publikací (přiřazených k dané službě). Nad takto sestrojenou posloupností hodnot jsou počítána zmíněná statistická ohodnocení. Postup jejich výpočtu je popsán v kapitole 2.3.

[64] 3 služby * 4 metriky = 12 spočtených metrik.

6.6 Prezentační vrstva

Prezentační vrstva slouží k prezentování dat uživateli. Kromě prezentace zajišťuje rovněž interakci s uživatelem, viz část 6.6.5. Grafické zpracování uživatelského rozhraní je obsaženo v přílohách G.

6.6.1 Struktura webu

Webové stránky systému se skládají celkem ze 3 stránek:

(a) *Domovská stránka*
- **url:** /
- **popis:** Stránka typu rozcestník.

(b) *Administrace*
- **url:** /admin
- **popis:** Administrace systému.

(c) *Prezentace informací*
- **url:** /explore
- **popis:** Prezentace bibliografických dat.

6.6.2 Administrace

Administrací je označena stránka umožňující ovládání procesu získávání bibliografických dat prostřednictvím jednoduchého formuláře. Přístup na tuto stránku je podmíněn ověřením totožnosti přihlášením.

Průběh procesu získávání a zpracování dat je vizualizován formou animací a tzv. *progress barů*[65]. Mimo těchto grafických prvků obsahuje stránka také sekci, do níž je průběžně vypisován log systému podrobně informující o průběhu. Vizualizaci obstarává knihovna PMSMiningPage, viz 6.6.5.

6.6.3 Prezentace informací

Na stránce s výsledky tohoto systému se nalézají seznam autorů se seznamem publikací. Položky obou z nich jsou členěny dle pracoviště autora a je umožněna rozšířená možnost jejich filtrace a řazení, viz G.8 či G.9 v příloze a následující obrázek 6.3.

[65]Progress bar je komponenta znázorňující průběh libovolné činnosti.

Obrázek 6.3: Možnosti filtrování a řazení seznamu autorů (nahoře) a seznamu publikací (dole).

Seznam autorů

Jednotlivé položky seznamu (tj. autoři) obsahují především jméno autora, akademické tituly, specializace, datum poslední změny záznamu a vypočtené statistiky. Seznam autorů lze filtrovat podle jejich jména a pracoviště a řadit je umožněno dle jména, pracoviště, počtu publikací a data poslední úpravy záznamu. Autoři jsou řazeni v rámci jednotlivých pracovišť (tj. seřazena je vždy podmnožina autorů vztahující se k danému pracovišti). Autoři tak i nadále zůstávají seskupeni podle pracovišť.

Položky seznamu autorů jsou zde vypisovány až teprve, když mají spočtené hodnoty metrik. Tyto hodnoty jsou počítány okamžitě po stahování dat.

Seznam publikací

Druhým seznamem je seznam publikací, jehož položky nabízejí kromě formátování výpisu publikací v normalizovaném tvaru v souladu s citační normou ČSN ISO 690:2011 a BibTeX, také možnost určení typu publikace, počet citací, odkaz na publikaci a informaci o datu poslední modifikace záznamu. Seznam publikací je možné filtrovat podle jejich názvů, jmen a pracovišť autorů a řadit je umožněno dle jmen autorů, názvu publikace, počtu citací publikace, typu publikace a data poslední modifikace záznamu. Rovněž i publikace jsou shlukovány a řazeny v rámci pracovišť svých autorů (tj. seřazena je vždy podmnožina publikací, jejichž autoři se vztahující k danému pracovišti). Publikace tak i nadále zůstávají seskupeny podle pracovišť svých autorů.

6.6.4 Export dat do CSV a TXT

Stránka s výsledky pro oba výše uvedené seznamy poskytuje možnost jejich exportování do formátů TXT a CSV, u kterého je ponechána volba oddělujícího znaku na uživateli. Vítána by mohla být kromě možnosti filtrování exportovaných záznamů také implementovaná volba zahrnutí (resp. nezahrnutí) statistik.

6.6.5 Interaktivní prostředí

Interakce uživatele s webovou aplikací je umožněna díky třem JavaScriptových knihovnám *PMSMiningForm*, *PMSResultForm* a *PMSDataExporter*, které jsem rovněž vyvíjel.

Knihovna *PMSMiningForm* obsluhuje formulář na stránce s administrací. Průběh získávání a zpracování bibliografických dat knihovnou vizualizuje tato knihovna v reálném čase formou jednoduchých grafů a statistik.

Další knihovnou je knihovna *PMSResultForm*, která zajišťuje inicializaci formulářových prvků typu `<select>`. V případě řazení je nabídka měněna dynamicky podle vybraného seznamu (seznamu publikací či autorů).

PMSExoprtData je poslední knihovnou. Zajišťuje serializaci položek seznamů, formátování a závěrem samotný export do souboru.

7 Testování systému

Tento systém pro správu publikací byl vyvíjen v prostředí operačního systému macOS Sierra na zařízení Apple MacBook (viz níže část 7.1.1) primárně pro webový prohlížeč Google Chrome (viz níže část 7.1.2).

Testování probíhalo již v průběhu vývoje aplikace. Funkčnost jednotlivých dílčí bloků kódu byla okamžitě s dokončením ověřována sérií testů. Každý takový otestovaný blok byl napojován na již existující celek a společně s ním byla správnost implementace celého celku ověřena. Nejobsáhlejší testování proběhlo ke konci vývoje.

Uživatelského testování (viz část 7.2.5) se vyjma mě účastnily 2 nezávislé osoby (23 let, grafik, pokročilá uživatelská úroveň a 24 let, student IT, programátor), kterým bylo zapůjčeno zařízení s nainstalovaným webovým serverem včetně veškerého potřebného programového vybavení. Cílem nebylo testovat instalaci aplikace, nýbrž její použití a spolehlivost.

7.1 Testovací podmínky

Hardwarové a softwarové prostředky byly vybrány různé, aby byla testy pokryta co nejširší škála podmínek, za nichž může být aplikace spuštěna.

7.1.1 Hardwarové vybavení pro testovací účely

Závěrečná testování byla vykonávána paralelně na dvou rozdílných zařízeních s rozdílnými operačními systémy:

HW.1) `Apple MacBook Pro, 13" retina`
`macOS Sierra 64-bit ver.10.12.4`
`Intel Core i5 o~2,7 GHz`
`8 GB RAM`

HW.2) `HP ProBook 4530s, 15.6"`
`MS Windows 7 Home Premium x64, SP1,`
`Intel Core i5-2410M, 2x2,3 GHz`
`8 GB RAM`

Kromě displejů zařízení byl také využit následující externí monitor:

HW.3) `ASUS MX239H, LED`
`23", 1920x1080 px`

7.1.2 Softwarové vybavení pro testovací účely

Na obou výše zmíněných zařízeních byla webová aplikace testována v následujících prohlížečích:

SW.1) Google Chrome[66], verze 57.0.2987.133, 64-bit,

SW.2) Mozila Firefox[67], verze 53.0, 64-bit,

SW.3) Safari[68], verze 10.1, 32-bit,

SW.4) Internet Explorer[69], verze 11.0.9600.18638.

7.2 Provedená testování

Celkové testování můžeme označit za dynamické manuální, tj. všechny dílčí testy byly provedeny člověkem (popřípadě byla člověkem zkontrolována systémem zpracovaná data) a výhradně za běhu aplikace. Příloha F obsahuje příklad provedených testů.

7.2.1 Test kompatibility

Prvním testovaným kritériem systému pro správu publikací byla kompatibilita, která odhalila, za jakých podmínek je aplikace plně schopna chodu. Během testování bylo objeveno několik závažných komplikací, které znemožňovaly chod aplikace.

Jedním z největších problémů byla nemožnost dolovat webové stránky pod operačním systémem Windows. Ukázalo se, že tzv. certifikáty autority (*CA Root Certificates*) nebyly aktuální a před používáním této aplikace bylo potřeba je aktualizovat. Tyto certifikáty jsou používány při ověřování SSL certifikátů zabezpečených webových stránek. Aktualizace těchto certifikátů se tak stala součástí instalace.

Následující testování kompatibility webových prohlížečů odhalilo komplikace s odlišnými webovými prohlížeči, než ve kterém byl systém vyvíjen. Aplikace byla vyvíjena primárně pro webový prohlížeč Google Chrome a výborná podpora se projevila také v prohlížeči Safari. V obou z nich jsou plně podporovány veškeré funkce systému. Obstojně dopadl také prohlížeč Mozilla Firefox, ve kterém není možný pouze export dat z důvodu slabé podpory HTML5 (konkrétně atributu `download`). Nejhůře dopadl Internet

[66] Web: `http://google.com/chrome`
[67] Web: `https://www.mozilla.org/cs/firefox/new/`
[68] Web: `https://www.apple.com/safari/`
[69] Web: `https://www.microsoft.com/cs-cz/download/internet-explorer.aspx`

Explorer, který nepovoluje aplikaci ani plně ovládat. Kompatibilita webových prohlížečů je srovnána následující tabulkou 7.1.

webový prohlížeč	**web mining**	**výpis, filtrování a řazení**	**export**	**ovládání**	**design**
Google Chrome	ANO	ANO	ANO	ANO	ANO
Safari	ANO	ANO	ANO	ANO	ANO
Mozilla Firefox	ANO	ANO	NE	ANO	ANO
Internet Explorer	ANO	ANO	NE	NE	NE

Tabulka 7.1: Test kompatibility webových prohlížečů.

Dalšími odlišnostmi, vztahujícími se ke grafickému prostředí, byly drobné výchylky v zarovnání textu či HTML prvků, nepatrné překrývání prvků layoutu apod. Tyto odlišnosti však nebránily použití aplikace. Přesto ale byly po testování opraveny.

Shrnu-li tuto část testování, systém je možné používat pod operačním systémem tak MS Windows i macOS Sierra. V souladu s informacemi z výše uvedené tabulky 7.1 je doporučeno tento systém používat výhradně prostřednictvím webového prohlížeče Google Chrome či Safari. Pro další testování bude využíváno právě těchto dvou prohlížečů.

Nutné je však zdůraznit, že těmito testy kompatibility byla podrobena *fontendová část* systému, zatímto samotná PHP knihovna není webovým prohlížečem nijak omezována.

7.2.2 Testování vůči předpokládaným výsledkům

Testování probíhalo na základě porovnání předpokládaných výsledků s reálnými výsledky získanými aplikací.

Byla vybrána množina zástupců (tj. autorů) z každé z využívaných služeb a tito autoři byli systémem zpracováváni. Získané výsledky byly zkontrolovány vůči očekávané předloze díky znalosti dat vyskytujících na profilových stránkách autorů.

Testování odhalilo odlišný výstup exportu, než bylo očekáváno. Export umožňuje volbu zahrnutí (či nezahrnutí) vypočtených statistik do exportovaného výstupu. Chyba, vyskytující se při každém vykonaném exportu,

spočívala ve volbě zahrnutí statistik do exportovaných informací a jejich následné absenci v exportovaném souboru a naopak.

Během testování se také projevila komplikace s nejednotností jmen autorů publikací. Vyskytly se případy, kdy byla zadána jména neúplně (části jmen zcela chyběly), kdy byla zadána zkratkou nebo nebyla jména jednotlivých spoluautorů správně oddělena. Z těchto důvodů se nacházejí v databázi redundantní záznamy reprezentující jednu reálnou osobu. Například pro prof. Ing. Václava Skalu, CSc. bylo za dobu testování uchováno 10 různých výskytů jeho jména, viz obrázek F.6 v příloze F.

Zbylé testování neobjevilo žádnou jinou chybu. Očekávaná data získaná systémem byla uchována v databázi, statistiky byly spočteny správně, identické publikace byly sjednoceny a vše bylo správně prezentováno uživateli.

7.2.3 Výkonnostní testování

Testování výkonu (*Performance Testing*) nebylo uskutečněno, neboť rychlost zpracování není majoritním faktorem tohoto systému, naopak je delší doba chodu aplikace očekávána. Mezi jednotlivými požadavky na servery využívaných služeb jsou voleny časové prodlevy v řádu sekund, které chod aplikace značně brzdí. Doba běhu aplikace je rovněž závislá na množství dolovaných dat, které se liší dle autora a dle toho, jaká uživatel bude chtít stahovat data. Testování výkonu tak pozbývá smyslu.

7.2.4 Test bezpečnosti

Webové stránky jsou použitím připravených SQL dotazů knihovny PDO ošetřeny vůči neoprávněné manipulaci útokem SQL Injection. Ten spočívá v narušení SQL dotazu prostřednictvím útočníkem vloženého úseku SQL kódu. Výsledkem může být přístup k informacím (například k přihlašovacím údajům uživatelů elektronického bankovnictví) nebo například smazání obsahu databáze.

Nejkritičtější část formuláře je část, kde je umožněno vkládat URL adresy. Hodnoty z těchto formulářových prvků jsou validovány a přípustné jsou pouze platné URL adresy služeb. U zbylých formulářových prvků je uživateli umožněna pouze volba ano/ne, přesto lze sekvenci SQL dotazu vložit do zdrojového kódu HTML jako například hodnotu vybraného formulářového prvku. Před tímto jednání chrání PDO knihovna formou vázání parametrů do dotazů SQL.

Ačkoliv jsou stránky vůči tomuto jednání chráněné, byla provedena série testů pokoušející se systém výše uvedeným způsobem narušit. Všechny do-

padly neúspěšně. Vybrané příklady jsou přiloženy v příloze F, viz obrázek F.5 v části Příloha F.5.

7.2.5 Uživatelské testování a test použitelnosti

Osoby účastnící se testování byly velmi stručně seznámeny s funkcí a přínosem aplikace a společně s webovými stránkami jednotlivých služeb jim byla předložena funkční aplikace. Odpovídajíce jejich studijním i profesním oborům, chvíli trvalo, než se v aplikaci zorientovali a naučili se aplikaci ovládat. Následně se však nevyskytl jediný problém, který by zamezil cílům (tj. požadavkům na zisk informací konkrétních autorů a jejich export), které si uživatelé vytyčili a také dosáhli.

Nevýhodu této aplikace, s níž bylo z důvodu Web miningu počítáno, uživatelé viděli v pomalém zpracování, avšak technická omezení po objasnění pochopili.

Zúčastněným osobám se zamlouvala především interaktivita a dynamičnost stránek společně s grafickým provedením. Ocenili rovněž možnost snadného výběru autorů (díky dopřednému průchodu službami, viz část 6.3).

8 Závěr

V rámci této práce byl vytvořen systém, který na základě konfigurace dokáže ze 3 bibliografických systémů (ResearchGate, GoogleScholar a DBLP) stahovat bibliografická data, která jsou následně uchovávána v databázi. Tato data je implementovaný systém schopen sám využít pro další zpracování, například pro výpočet statistik. Získaná a zpracovaná data umožňuje přehledně a responzivně (tj. čitelně na široké škále zařízení o různých rozlišeních displejů) prezentovat uživateli. Informace o publikacích jsou navíc formátovány v souladu s citační normou ČSN ISO 690:2011 a nástrojem BiBTeX. Mimo prezentování výsledků dokáže systém data exportovat ve formátech CSV a TXT.

Již od samotného návrhu aplikace jsem kladl důraz na snadnou rozšiřitelnost zahrnutím nové služby, a proto byla zvolena možnost jednoho obecného parseru (tedy modulu jednotného pro všechny služby, který extrahuje informace z dat daných služeb). Pro správu softwarových knihoven byl využit Composer, který správou závislostí projektu rozšiřitelnost aplikace rovněž usnadňuje.

Přestože již samo zadání bylo poměrně obsáhlé a zpracování jednotlivých bodů bylo časově náročné, zahrnul jsem do systému prvky, které nebyly součástí zadání. Rešerší bibliografických služeb jsem se rozhodl pro využívání pouze dvou služeb – ResearchGate a GoogleScholar. Přesto jsem se v průběhu vývoje uchýlil k zahrnutí i služby DBLP. Také jsem systém oživil interaktivitou, jednoduchými grafy a vizualizacemi, logováním v reálném čase a přehlednou prezentací získaných informací, včetně bohaté nabídky jejich filtrování a řazení. Administraci jsem zabezpečil přihlášením eliminujícím možné zneužití systému neoprávněnou osobou.

8.1 Kritické zhodnocení

Velká nevýhoda tohoto systému spočívá v možném zamezení budoucího využívání, např. z důvodu změny struktury webových stránek vybraných bibliografických služeb nebo z důvodu zamezení přístupu na jejich stránky této aplikaci. K takovému odříznutí (přestože jen dočasnému) již došlo během samotného vývoje, a to službami ResearchGate a GoogleScholar. Slabinu taktéž vidím v rychlosti aplikace ve fázi dolování z důvodu časových prodlev mezi požadavky, které proces stahování dat mnohanásobně prodlužují.

Důvodem zavržení používání tohoto systému by mohly být samy informace uváděné na webových stránkách využívaných služeb. Byly odhaleny případy, ve kterých jsou tyto informace uvedeny nepřesně či chybně (například uvedením zkratky jména či neoddělením výčtu jmen spoluautorů a jejich ponechání v celku jako jedno jméno). Právě tato nejednotnost informací (kdy se autory uváděné informace liší) do systému zanáší jistou entropii[70], jejíž důsledkem je například vystupování jedné reálné osoby pod více identitami, jak uvádím v části 7.2.2. Tento příklad demonstruje nejistotu uváděných informací ve službách, které jsou však tomuto systému pro správu publikací základem, na jehož bezchybnost bychom se potřebovali spolehnout.

8.2 Možná vylepšení

Systém umožňuje výpis seznamu publikací v souladu se dvěma formáty. Pro takový výpis je vyžadováno určité množství informací, které by však nemuselo být do budoucna z vybraných služeb získáno (například kdyby byla některá ze služeb vyřazena z provozu). Zahrnutí dalších bibliografických služeb by přineslo zisk objemnějšího množství dat, které by snáze pokrylo potřebné množství informací pro zmiňované výpisy.

Získaná data nemusí být na stránkách bibliografických služeb uvedena korektně (obzvláště u služeb s autonomní indexací) a možnost opravy údajů uživatelem aplikace by mohla být vítána. Stejně tak by využití přinesla i možnost výběru rodičovské (či majoritní) publikace v případě množiny sdružených publikací (viz část 5.2).

Dále tento systém počítá na základě získaných dat 4 metriky ohodnocující autory. Výběr těchto metrik by mohl být rozšířen o nové, společně metrikami ohodnocující publikace. Na základě těchto spočtených statistik by mohlo být umožněno řazení autorů i publikací.

Stránka administrace je zpřístupněna pouze přihlášeným uživatelům, jejichž účty však nelze nijak spravovat a ani vytvářet. Proto by mohl být implementován minimálně registrační formulář. Uvítána by mohla být i kompletní správa uživatelských účtů, kterým by navíc mohla být přidělována různá oprávnění.

[70]Entropie může být chápána jako míra informační ztráty či neurčitosti.

Literatura

[1] Hanke, T. *Tvorba datových zdrojů pro bibliometrická měření.* Diplomová práce, Západočeská univerzita v Plzni, Plzeň, 2014.

[2] Firstová, Z. *Nová citační norma ČSN ISO 690:2011 - Bibliografické citace* [online]. Univerzitní knihovna Západočeské univerzity v Plzni, 2011. [cit. 2017/03/08]. Citační norma ČSN ISO 690:2011. Dostupné z: `https://sites.google.com/site/novaiso690/home`.

[3] Danišík, J. *Integrace webových bibliografických služeb.* Diplomová práce, Západočeská univerzita v Plzni, Plzeň, 2011.

[4] Hirsch, J. E. An Index to Quantify An Individual's Scientific Output. *Proceedings of the National Academy of Sciences.* 2005, 102(46), s. 16569–16572. ISSN 1091-6490.

[5] Nykl, M. *Hodnocení významnosti variantami PageRanku.* Disertační práce, Západočeská univerzita v Plzni, Plzeň, 2016.

[6] Egghe, L. Theory and practise of the g-index. *Scientometrics.* oct 2006, 69, 1, s. 131–152. ISSN 0138-9130. doi: 10.1007/s11192-006-0144-7. Dostupné z: `https://doi.org/10.1007%2Fs11192-006-0144-7`.

[7] Connor, J. *Google Scholar Citations Open To All* [online]. Google Inc., 2011. [cit. 2017/03/14]. Google Scholar Blog: Google Scholar Citations Open To All. Dostupné z: `https://scholar.googleblog.com/2011/11/google-scholar-citations-open-to-all.html`.

[8] Noruzi, A. *Impact Factor, h-index, i10-index and i20-index of Webology* [online]. Webology, 2016. [cit. 2017/03/15]. Webology: Impact Factor, h-index, i10-index and i20-index of Webology. Dostupné z: `http://www.webology.org/2016/v13n1/editorial21.pdf`.

[9] Ortega, J. *Social network sites for scientists: a quantitative survey.* Chandos Publishing is an imprint of Elsevier, 2016. ISBN 978-0-08-100592-7.

[10] Fiala, D. Mining citation information from CiteSeer data. *Scientometrics.* 2011, 86(3), s. 553–562. ISSN 0138-9130.

[11] *Mendeley API Docs* [online]. Mendeley, 2015. [cit. 2015/12/11]. Mendeley: Documentation. Dostupné z: `https://api.mendeley.com/apidocs/docs`.

[12] *PHP: Documentation* [online]. The PHP Group, 2015. [cit. 2015/12/11]. PHP: Documentation. Dostupné z: `http://php.net/docs.php`.

[13] Procházka, D. *PHP 6: začínáme programovat.* Grada Publishing , a.s., 3. vydání, 2012. ISBN 978-80-247-3899-4.

[14] Schlossnagle, G. *Pokročilé programování v PHP 5.* Zoner Press, 1. vydání, 2004. ISBN 80-86815-14-5.

[15] Schneider, R. D. *MySQL: oficiální průvodce tvorbou, správou a laděním databází.* Grada Publishing a.s., 1. vydání, 2006. ISBN 80-247-1516-3.

[16] Huseby, S. H. *Zranitelný kód.* Computer Press, 1. vydání, 2006. ISBN 80-251-1180-6.

[17] Hopkins, C. *PHP okamžitě.* Computer Press, 2014. ISBN 978-80-251-4196-0.

[18] Hogan, B. P. *HTML5 a CSS3 : výukový kurz webového vývojáře.* Computer Press a.s., 1. vydání, 2011. ISBN 978-80-251-3576-1.

[19] *W3Schools Online Web Tutorials* [online]. W3Schools, 2015. [cit. 2015/12/11]. W3Schools: Online Web Tutorials. Dostupné z: `https://www.w3schools.com`.

[20] *Documentation – Twig – The flexible, fast and secure PHP template engine* [online]. Sensio, 2015. [cit. 2015/12/11]. Twig: Documentation. Dostupné z: `http://twig.sensiolabs.org/doc/2.x`.

[21] *jQuery API Documentation* [online]. The jQuery Foundation, 2015. [cit. 2015/12/11]. jQuery: Documentation. Dostupné z: `http://api.jquery.com`.

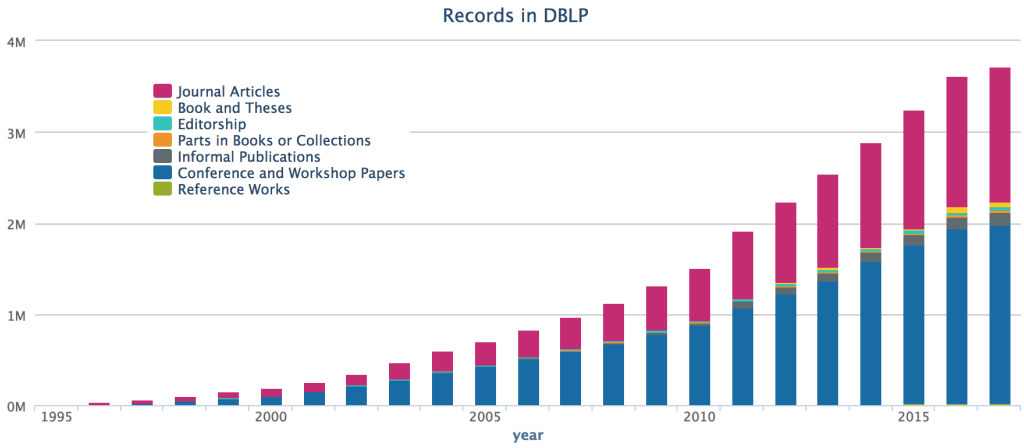
[22] *XML Path Language (XPath)* [online]. W3C, 1999. [cit. 2017/03/10]. XML Path Language. Dostupné z: `https://www.w3.org/TR/xpath`.

Přílohy

Obsáhlejší a méně relevantní materiály, které však rozšiřují přehled a informace obsažené v této práci, byly přesunuty do příloh, aby případně nenarušovaly její čitelnost.

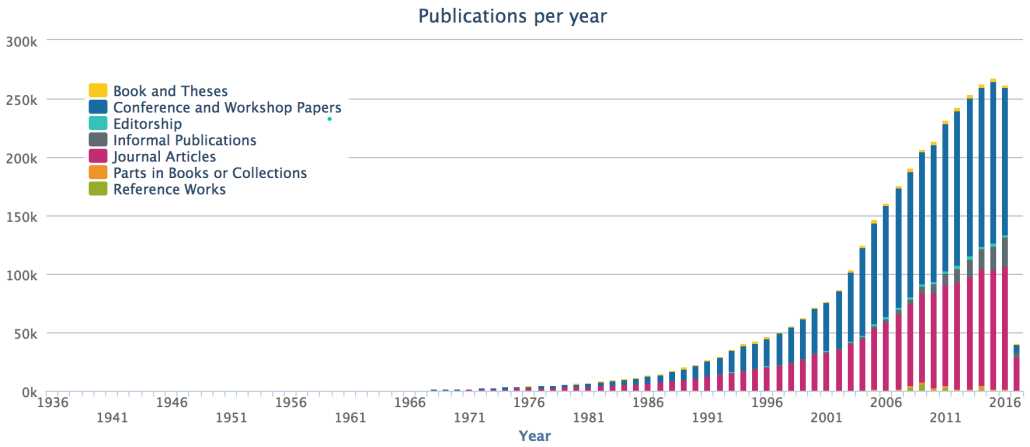
Příloha A Vizualizace statistik bibliografických služeb

V této části jsou obsaženy grafické vizualizace statistik poskytované samotnými službami DBLP a GoogleScholar. Níže uvedené vizualizace A.1 až A.3 byly nápomocny při výběru množiny bibliografických služeb využívaných implementovaným systémem pro správu publikací.



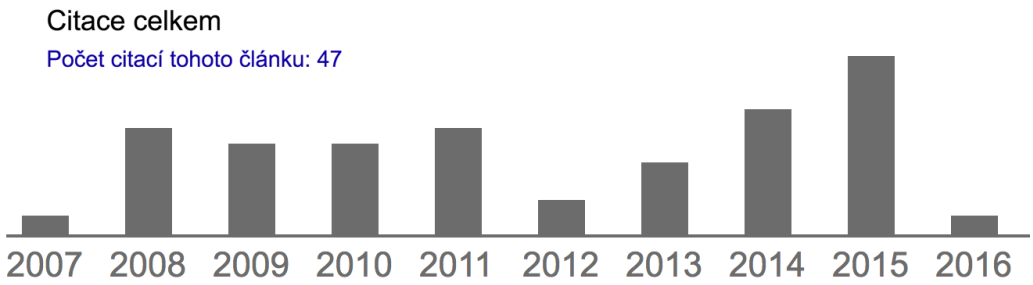
Obrázek A.1: Počet záznamů databáze DBLP k roku 2016.

(zdroj: `http://dblp.uni-trier.de/statistics/recordsindblp.html`)



Obrázek A.2: Počet indexovaných publikací databáze DBLP k roku 2016.

(zdroj: `http://dblp.uni-trier.de/statistics/publicationsperyear.html`)



Obrázek A.3: Příklad vizualizace závislosti citovanosti (počtu citací) publikace na čase službou Google Scholar.

(zdroj: `https://scholar.google.cz/citations?view_op=view_citation&hl=cs&user=BaJ-Ql8AAAAJ&citation_for_view=BaJ-Ql8AAAAJ:WJVC3Jt7v1AC`)

Příloha B Zápis publikací ve formátu BiBTeX

Níže se nalézají příklady zápisu publikací ve formátu určeném pro nástroj BibTeX, který slouží ke generování referencí (např. v prostředí LaTeX), viz část 2.2.2. Uvádím pouze typy publikací, které nás v této práci zajímají.

Zápis časopiseckého článku

```
@article { unikátní_identifikátor,
  author  = {jméno_autora},
  title   = {název_publikace},
  journal = {název_časopisu},
  year    = rok_vydání,
  number  = číslování,
  pages   = {rozsah_stran_článku},
  month   = měsíc_vydání,
  note    = {poznámka},
  volume  = číslo_svazku
}
```

Zápis konferenčního příspěvku

```
@conference { unikátní_identifikátor,
  author       = {jméno_autora},
  title        = {název_publikace},
  booktitle    = {název_knihy},
  year         = rok_vydání,
  editor       = {vydavatel},
  volume       = číslo_svazku,
  series       = označení_série,
  pages        = {rozsah_stránek_knihy},
  address      = {adresa_vydavatele},
  month        = měsíc_vydání,
  organization = {jméno_organizace},
  publisher    = {jméno_nakladatele},
  note         = {poznámka}
}
```

Zápis kapitoly knihy

```
  @inbook { unikátní_identifikátor,
    author       = {jméno_autora},
    title        = {název_publikace},,
    author       = {jméno_autora},
    title        = {název_publikace},
    chapter      = číslo_kapitoly,
    pages        = {rozsah_stránek_knihy},
    publisher    = {jméno_nakladatele},
    year         = rok_vydání,
    volume       = číslo_svazku,
    series       = označení_série,
    address      = {adresa_nakladatele},
    edition      = číslo_edice,
    month        = měsíc_vydání,
    note         = {poznámka}
  }
```

Zápis kvalifikační práce

Rozlišujeme dva typy kvalifikačních prací. První je diplomová práce (`@masterthesis`) a druhá je disertační práce (`@phdthesis`).

```
@masterthesis { unikátní_identifikátor
  author       = {jméno_autora},
  title        = {název_publikace},
  school       = {jméno_univerzity},
  year         = rok_vydání,
  address      = {adresa_nakladatele},
  month        = měsíc_vydání,
  note         = {poznámka}
}

@phdthesis { unikátní_identifikátor,
  author   = {jméno_autora},
  title    = {název_publikace},
  year     = rok_vydání,
  address  = {adresa_autora},
  school   = {název univerzity}
}
```

Zápis elektronického zdroje

```
@misc { unikátní_identifikátor,
  author       = {jméno_autora},
  title        = {název_publikace},
  howpublished = {způsob_publikování},
  month        = měsíc_vydání,
  note         = {poznámka}
}
```

Příloha C Konfigurace parseru bibliografických služeb

Analytická část 5.1.4 nastínila obsah tohoto konfiguračního XML souboru. V následujících blocích kódu je vysvětlena jeho struktura podrobněji. Červeně zvýrazněný text označuje povinný výskyt a naopak zelený výskyt volitelný.

Nejprve objasním dvě dílčí části `<xpaths>` a `<pagination>`, které se budou v tomto popisu vyskytovat na vícero místech.

Část `<xpaths>`

Ve značce `<xpaths>` je umístěna množina XPath selektorů separovaných v samostatných značkách. Přípustnými potomky této značky, obsahující XPath selektory informací o autorovi, jsou:

(a) `<name>`, `<department>`, `<degrees>`, `<specialization>`, `<photoUrl>`.

A značky se selektory adresujícími informace o publikaci jsou následující:

(b) `<title>`, `<link>`, `<type>`, `<coauthors>`, `<year>`, `<citationCount>`, `<location>`, `<minorTitle>`, `<containerTitle>`, `<containerAuthor>` `<containerPageRange>`, `<publisher>`, `<issn>`, `<isbn>`, `<volume>`, `<range>`, `<editionNumber>`, `<university>` a `<supervisor>`.

Tyto značky korespondují s pojmenováním atributů modelových tříd (třídy `Author` reprezentující autora a `Publication` reprezentující publikaci) a také pojmenováním atributů příslušných databázových entit (viz ERA model na obrázku 5.2).

Speciálními případy potomků značky `<xpaths>` jsou `<container>` obsahující selektor rozlišující jednotlivé položky seznamu a značka `<faculty>`, která je využita pro filtrování při autonomním průchodu službou, viz níže část `<linkSource>`. Obecně může značka `<xpaths>` obsahovat libovolné značky. Obsah jimi získaný nebude sice uchován v databázi, ale může se na jeho základě filtrovat, jako tomu je v případě značky `<faculty>`.

Veškeré prvky obsahující XPath selektory (tj. potomci značky `<xpaths>` a značky `<stopCondition>` a `<nextPageLink>`, které budou posané níže) povolují atributy `type`, `encoding` a `extractRegex`. V souladu s těmito atributy je vyjmutý obsah následně modifikován. `encoding="hexadecimal"` značí, že je získaný obsah vyjádřen šestnáctkovou soustavou a bude před dalším zpracováním převeden do kódování `utf-8`. `extractRegex` obsahuje regulární

výraz vyjímající žádanou informaci z přebytečného textu[71]. A v případě `type=url` je zkontrolována absolutnost URL adresy, kterou obsah představuje, a případně je prefixem tvořeným obsahem `<baseUrl>` náležitě upravena. Při případném dalším vývoji této aplikace lze atributy a jejich přípustné hodnoty rozšířit dle potřeb.

Vzorový příklad konfiguračního souboru služeb obsahuje v příloha E, viz zdrojový kód 1.

Část `<pagination>`

Stránkování, jak již víme z analytické části (viz část 5.1.4), rozlišujeme dvojího typu: *2-prvkovou paginaci* a *3-prvkovou paginaci*, oba typy jsou blíže popsány v části 5.1.1. Pro *2-prvkovou paginaci* jsou potřeba 2 parametry: `<nextPageLink>` a `<stopCondition>`. `<nextPageLink>` obsahuje XPath selektor získávající odkaz na další stranu seznamu a `<stopCondition>` obsahuje XPath selektor detekující poslední stránku seznamu.

```
<pagination>
   <nextPageLink></nextPageLink>
   <stopCondition></stopCondition>
</pagination>
```

Příklad:

```
<pagination>
   <nextPageLink>//a/@href</nextPageLink>
   <stopCondition>//ul/li</stopCondition>
</pagination>
```

Parametry `<pageParam>`, `<pageParamIncrement>` a `<stopCondition>` definují *3-prvkovou paginaci*. `<pageParam>` obsahuje parametr URL uchovávající hodnotu stránky, `<pageParamIncrement>` obsahuje navýšení hodnoty čísla stránky pro přechod na stránku následující a `<stopCondition>` je definován XPath selektorem detekujícím konec stránkování (viz část 5.1.4).

[71] Příklad: Selektor adresuje prvek obsahující mimo žádané hodnoty h (př.: "2017") také další text t (např. "roku ") a společně h a t tvoří hodnotu adresovaného prvku (př.: "roku 2017"). Regulární výraz definovaný v atributu `extractRegex` (př.: "\4") žádanou hodnotu h od textu t separuje (výsledkem je podmnožina h původně adresovaného obsahu).

```
<pagination>
   <pageParam></pageParam>
   <pageParamIncrement></pageParamIncrement>
   <stopCondition></stopCondition>
</pagination>
```

Příklad:

```
<pagination>
   <pageParam>page</pageParam>
   <pageParamIncrement>1</pageParamIncrement>
   <stopCondition>//ul/li</stopCondition>
</pagination>
```

Část `<services>`

Kořenová značka `<services>` seskupuje konfigurace jednotlivých služeb obsažených ve značkách `<service>`.

```
<services>
   <service>
      <!- konfigurace služby ->
   </service>
   <!- definice zbývajících služeb ->
</services>
```

Příklad:

```
<services>
   <service>
      <!- konfigurace první služby ->
   </service>
   <service>
      <!- konfigurace druhé služby ->
   </service>
</services>
```

Část `<service>`

Značka `<service>` obsahuje atribut `name` identifikující službu (přípustné hodnoty jsou `dblp`, `researchgate` nebo `scholar.google`). Přímými potomky značky `<service>` jsou značky `<baseUrl>` definující URL adresu

domovské stránky služby (v práci bude označována jako *bázová URL*) a `<checkRegex>`. Bázová URL adresa je uváděna včetně protokolu `http://` nebo zabezpečené varianty `https://`[72]. Značka `<checkRegex>` obsahuje regulární výraz, jímž jsou validovány URL adresy získané od uživatele (viz část 6.3.1). V `<cookies>` jsou data `cookies`, která byla překopírována z paměti webového prohlížeče po úspěšné autentizaci reálného uživatele ke službě, a díky kterým lze dočasně konstrukci Captcha obejít. Organizace podstránek je dána strukturou `<subpages>`.

```
<!- konfigurace služby ->
<service name="identifikace služby">
   <baseUrl></baseUrl>
   <checkRegex></checkRegex>
   <cookies></cookies>
   <subpages></subpages>
   <linkSource></linkSource>
</service>
```

Příklad:

```
<!- konfigurace služby exmpleA ->
<service name="exmpleA">
   <baseUrl>http://example.com</baseUrl>
   <checkRegex>http:\/\/example\.com\/users\/.+</checkRegex>
   <cookies>examplecookiestring</cookies>
   <subpages>
      <!- konfigurace podstránek ->
   </subpages>
   <linkSource>
      <!- konfigurace linkSource ->
   </linkSource>
</service>
```

Část `<subpages>`

Sekce `<subpages>` definuje organizaci podstránek služby, včetně tvarů příslušných URL adres. Seskupuje konfigurace podstránek, z nichž jsou data dolována. URL adresa každé z podstránek vybraných služeb je tvořena zá-

[72] Např. pro službu ResearchGate je ve tvaru `https://www.researchgate.net`

kladem (bázovou adresou) a příponou, definovanou atributem `urlSuffix` (jehož hodnota může být i prázdná, tj. `""`). Značka `<xpaths>` obsahuje kolekci XPath selektorů adresujících požadovaná data. Obsahuje-li podstránka mnohaprvkový seznam se stránkováním, pak se zde navíc vyskytuje značka `<pagination>`. Podstránky `<profile>` a `<publications>` jsou pro chod aplikace nezbytné a jsou tedy povinné.

```
<subpages>
   <profile urlSuffix="[přípona URL adresy]">
      <xpaths></xpaths>
   </profile>
   <publications urlSuffix="[přípona URL adresy]">
      <xpaths></xpaths>
      <pagination></pagination>
   </publications>
   <publication></publication>
   <citations></citations>
</subpages>
```

Příklad:

```
<subpages>
   <profile urlSuffix="/">
      <xpaths>
         <name>//h2/text()</name>
         <department>//p/text()</department>
      </xpaths>
   </profile>
   <publications urlSuffix="/publications">
      <xpaths>
         <container>//li[contains(@class, 'pub')]</container>
         <title>//span[contains(@class, 'tit')]/text()</title>
         <link type="url»//a/@href</link>
      </xpaths>
      <pagination>
         <pageParam>page</pageParam>
         <pageParamIncrement>1</pageParamIncrement>
         <stopCondition>//ul/li</stopCondition>
      </pagination>
   </publications>
</subpages>
```

Část `<linkSource>`

Posledním přímým potomkem `<service>` je značka `<linkSource>`, která definuje organizaci internetové stránky se seznamem autorů. `<linkSource>` je značka nepovinná, neboť např. služba DBLP neposkytuje seznam autorů vázajících se k instituci (viz část 5.1.2).

Značka `<searchUrl>` obsahuje URL adresu seznamu autorů přiřazených k instituci[73] a `<filter>` umožňuje výběr podmnožiny žádaných autorů ze seznamu. Filtrace probíhá na základě kritéria, které je definováno značkou `<criteria>` a jehož hodnotou je označení jednoho z XPath selektorů obsažených v `<xpaths>`. Poslední značkou je `<values>`, v níž se nalézá středníky oddělený výčet přípustných hodnot.

```
<linkSource>
   <searchUrl></searchUrl>
   <xpaths></xpaths>
   <filter>
      <criteria></criteria>
      <values></values>
   </filter>
   <pagination>
      <pageParam></pageParam>
      <pageParamIncrement></pageParamIncrement>
      <stopCondition></stopCondition>
   </pagination>
</linkSource>
```

Příklad:

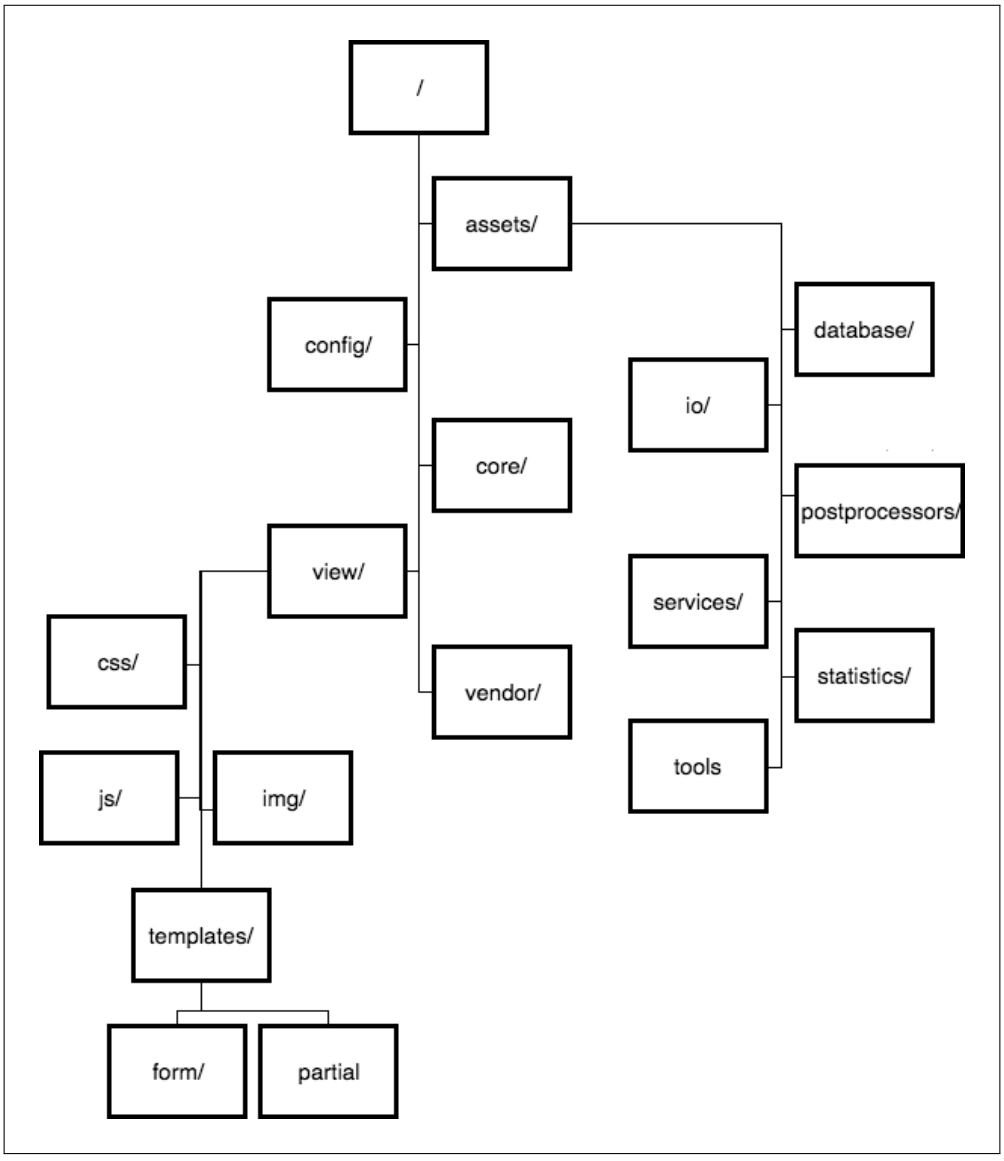
```
<linkSource>
   <searchUrl>http://example.com/users</searchUrl>
   <xpaths>
      <name>//h5/a/text()</faculty>
      <link type="url»//h5/a/@href</link>
      <faculty>//div[contains(@class, 'faculty')]</faculty>
   </xpaths>
```

[73] Například seznam příslušníků instituce *University od West Bohemia* vedený ve službě ResearchGate, web: `https://www.researchgate.net/institution/University_of_West_Bohemia/members`.

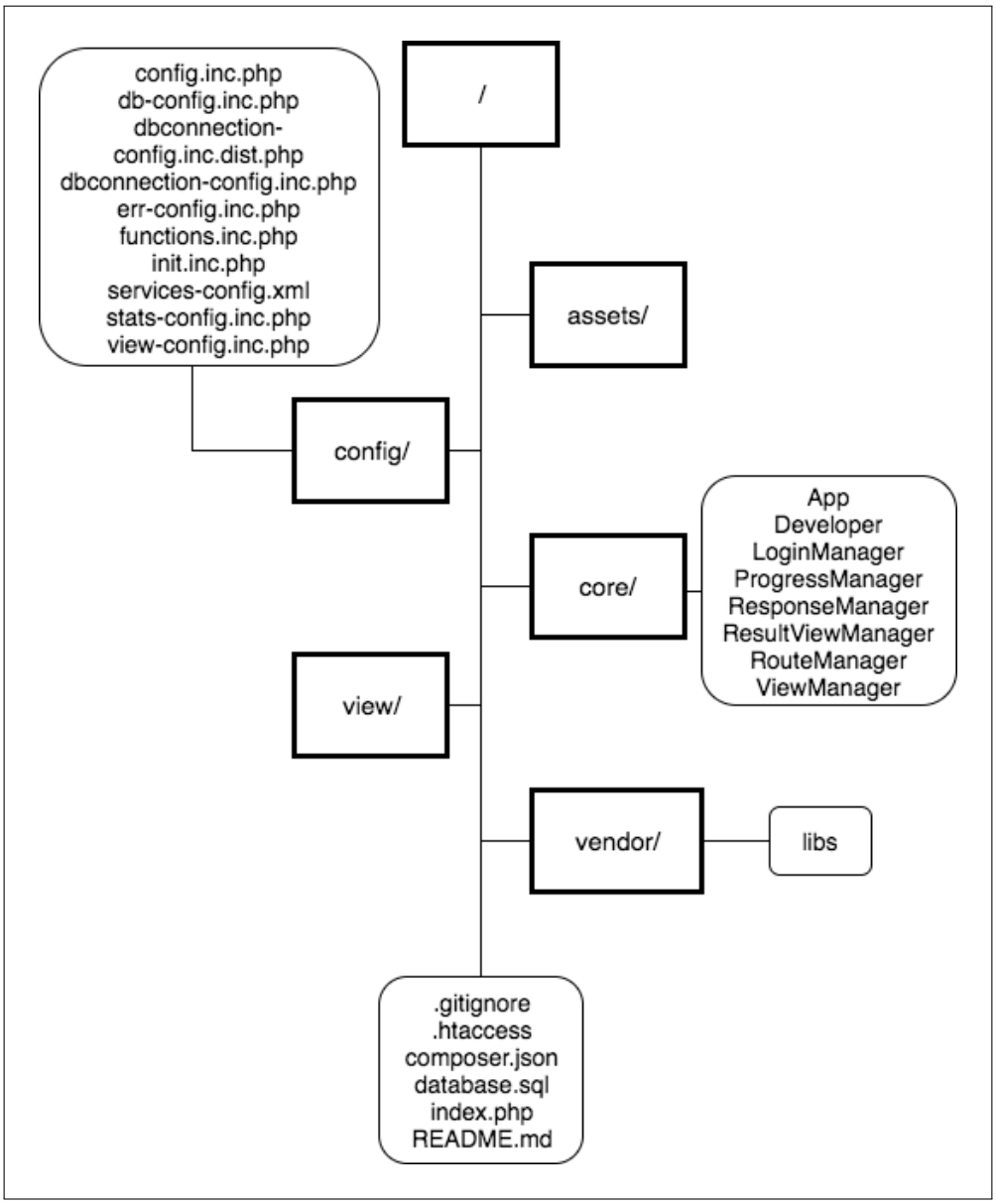
```
    <filter>
        <criteria>faculty</criteria>
        <values>KIV;Computer Science</values>
    </filter>
    <pagination>
        <pageParam>page</pageParam>
        <pageParamIncrement>1</pageParamIncrement>
        <stopCondition>//h2/span</stopCondition>
    </pagination>
</linkSource>
```

Příloha D Hierarchie adresářové struktury

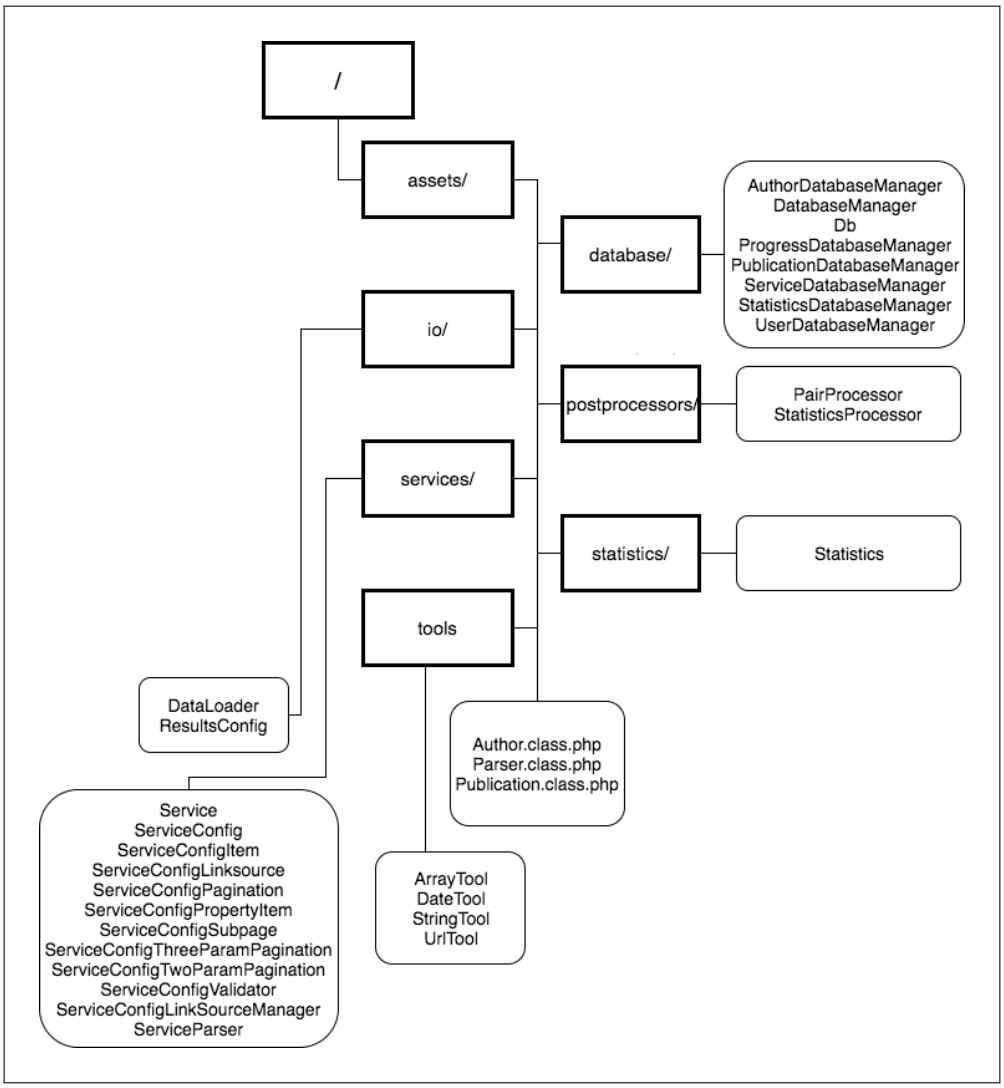
Projekt je strukturován dle logické souvislosti tříd do samostatných adresářů, viz obrázky D.1 až D.4.



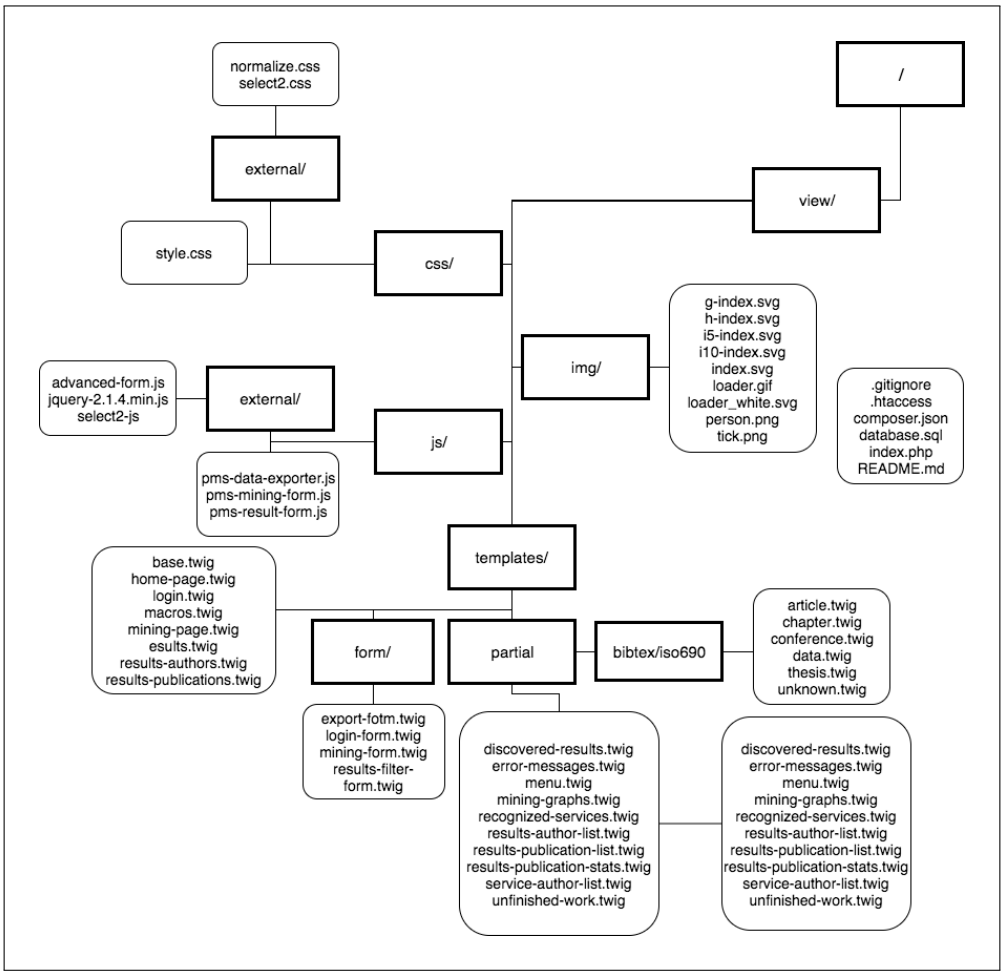
Obrázek D.1: Kostra adresářové struktury projektu.



Obrázek D.2: Adresářová struktura `root`, `/config/` a `/core/`.



Obrázek D.3: Adresářová struktura `/assets/`.



Obrázek D.4: Adresářová struktura `/view/`.

Příloha E Výběr zdrojových kódů

Zdrojový kód 1: Ukázka struktury konfiguračního souboru služeb s omezeným příkladem služby ResearchGate.

```
<services>
 <service name="researchgate">
  <baseUrl>https://www.researchgate.net</baseUrl>
  <checkRegex>https:\/\/(www\.)?researchgate\.net\/
     (profile|researcher)\/\S[^\/\n]+</checkRegex>
  <cookies><!-- cookies s autentiz. tokenem
     --></cookies>
  <subpages>
   <profile urlSuffix="/">
    <xpaths>
     <name>//a[contains(@class,
        'ga-profile-header-name')]/text()</name>
     <department>//div[contains(@class,
        'institution-dept')]/text()</department>
     <!--<degrees>-->
     <!--<specialization>-->
     <!--<photoUrl type="url">-->
    </xpaths>
   </profile>
   <publications urlSuffix="/publications">
    <xpaths>
     <container>//li[contains(@class,
        'li-publication')]</container>
     <title>//span[contains(@class,
        'publication-title')]/text()</title>
     <!--<link type="url">-->
     <!--<type extractRegex="[a-zA-Z ]*">-->
     <!--<coauthors>-->
     <!--<year extractRegex="\d{4}">-->
    </xpaths>
    <pagination>
     <pageParam>page</pageParam>
     <pageParamIncrement>1</pageParamIncrement>
     <stopCondition>//li[contains(@class,
        'li-publication')]</stopCondition>
    </pagination>
   </publications>
```

```
31      <citations
           urlSuffix="/citations?sorting=citationCount">
32       <xpaths>
33        <container>//li[contains(@class,
             'li-publication')]</container>
34        <title>//span[contains(@class,
             'publication-title')]/text()</title>
35        <citationCount>//div[contains(@class,
             'citation-count')]/text()</citationCount>
36       </xpaths>
37       <pagination>
38        <pageParam>page</pageParam>
39        <pageParamIncrement>1</pageParamIncrement>
40        <stopCondition>//li[contains(@class,
             'li-publication')]</stopCondition>
41       </pagination>
42      </citations>
43    </subpages>
44    <linkSource>
45     <searchUrl>https://www.researchgate.net/institution/
          University_of_West_Bohemia/members</searchUrl>
46     <xpaths>
47      <name>//li[starts-with(@class,'people-item')]/*
          /h5/a/text()</name>
48      <link type="url">//li[starts-with(@class,
          'people-item')]/*/h5/a/@href</link>
49      <faculty>//*/div[starts-with(@class,
          'truncate-single-line')]</faculty>
50     </xpaths>
51     <filter>
52      <criteria>faculty</criteria>
53      <values>KIV;Computer Science</values>
54     </filter>
55     <pagination>
56      <pageParam>page</pageParam>
57      <pageParamIncrement>1</pageParamIncrement>
58      <stopCondition>//li[starts-with(@class,
          'people-item')]</stopCondition>
59     </pagination>
60    </linkSource>
61   </service> <!-- dale: konfigurace GS a DBLP -->
62 </services>
```

Příloha F Výsledky testování

Testování přineslo velké množství dat, která se však v drtivé většině opakují, neboť např. použití různých internetových prohlížečů neovlivnilo výsledky aplikace. Různé prohlížeče však mohou ovlivnit ovladatelnost aplikace, například znepřístupněním ovládacích prvků administrace.

Testována byla aplikace na množině autorů, níže však uvádím pro demonstraci příklad z provedených testů. Testování je blíže popsáno v kapitole 7.

Příloha F.1 Kontrola stažených informací

Informace vybrané množiny zástupců z každé služby byly postupně stahovány. Na základě předpokládaného výsledku byla reálně získaná data vůči tomuto předpokladu porovnávána. Ani v jednom z provedených závěrečných testů se nevyskytla chyba a vždy bylo staženy informace, spočítány metriky a sjednoceny publikace, jaké být měly – tedy v souladu s předpokladem.

Kontrola vůči záznamům v databázi

Pro demonstraci byl vybrán profil autora Ing. Michala Nykla Ph.D. ve službě ResearchGate. Zisk informací obsažených v tomto profilu spočívá ve Web miningu následujících URL adres:

`https://www.researchgate.net`

... `/profile/Michal_Nykl`

... `/profile/Michal_Nykl/publications?page=0`

... `/profile/Michal_Nykl/publications?page=1`

... `/profile/Michal_Nykl/publications?page=2`

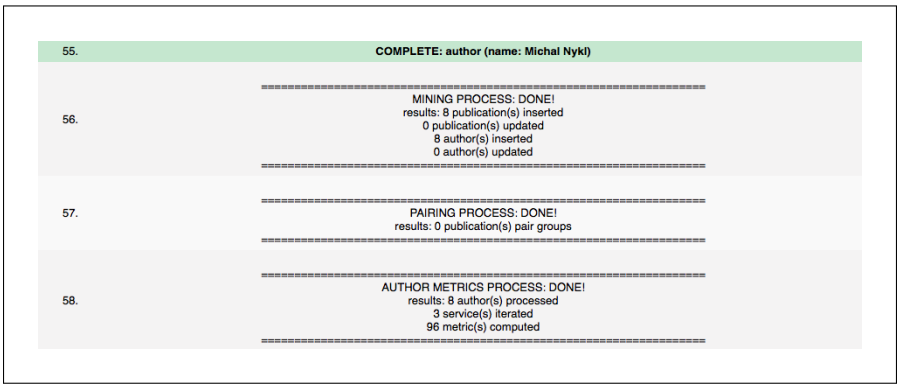
... `/profile/Michal_Nykl/citations?sorting=citationCount&page=1`

... `/profile/Michal_Nykl/citations?sorting=citationCount&page=2`

Byly sestrojeny platné URL adresy a stránky, které se na těchto adresách nacházejí, byly staženy správně, neboť všechny odpovědi na požadavky obsahovaly tzv. HTTP stavový kód o hodnotě 200. Z těchto URL adres byly získány a v databázi uchovány následující informace: **8 autorů**, **8 publikací** v letech **2011 –2013**. Jména autorů, názvy publikací, včetně jejich roků vydání byla stažena totožně s uváděnými informacemi na webu. Publikacím bylo správně přiřazeno autorství i typ publikace. Autoru bylo správně určeno pracoviště. URL adresa fotky je platná.

Kontrola vůči logu

Hladký průběh aplikace potvrzuje následující výňatek z výpisu logu (tj. podrobného záznamu o chodu aplikaci), viz obrázek F.1:

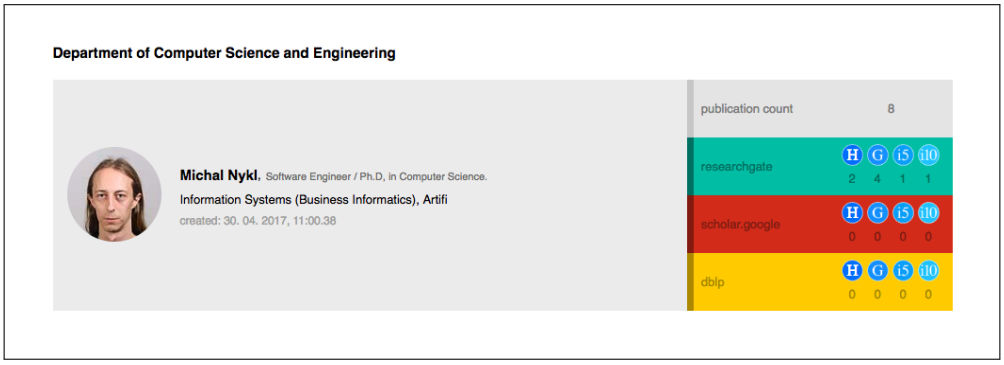


Obrázek F.1: Kontrola vůči výpisu logu.

Log také ukazuje, že nebyly nalezeny žádné identické publikace, a že bylo spočteno celkem 96 hodnot používaných metrik, tj. 8 autorů * 4 metriky * 3 služby.

Kontrola vůči výsledkům vypsaných v aplikaci

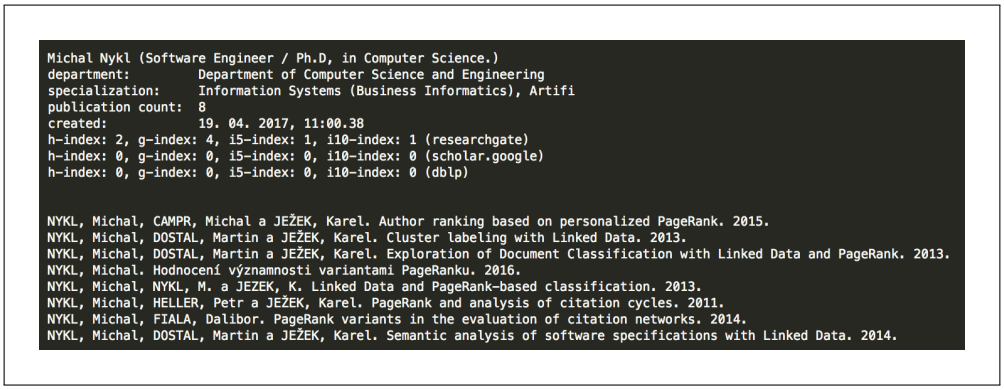
Bezchybný průběh dokládá i následující záznam ze seznamu autorů vypsaného aplikací, viz obrázek F.2.



Obrázek F.2: Kontrola výstupu aplikace vůči záznamu v seznamu autorů.

Příloha F.2 Kontrola exportovaných dat

Následující obrázek F.3 ukazuje výňatek exportovaných dat související s výše provedeným testováním.



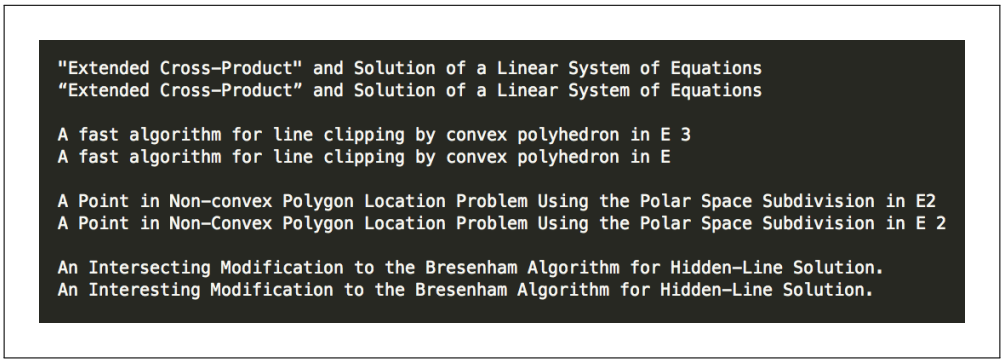
Obrázek F.3: Kontrola výsledku procesu vůči exportovaným datům.

Příloha F.3 Kontrola počítaných statistik

Na výše uvedeném příkladu lze zkontrolovat správnost spočtených statistik. Z autorových publikací byla sestavena sestupná posloupnost hodnot citací autorových publikací: `15, 4, 2, 2, 1, 1, 0, 0`. Na základě této posloupnosti byla manuálním přepočítáním ověřena správnost jednotlivých metrik: h-index o hodnotě 2, g-index 4, i5-index 1 a i10-index 1.

Příloha F.4 Kontrola sjednocování publikací

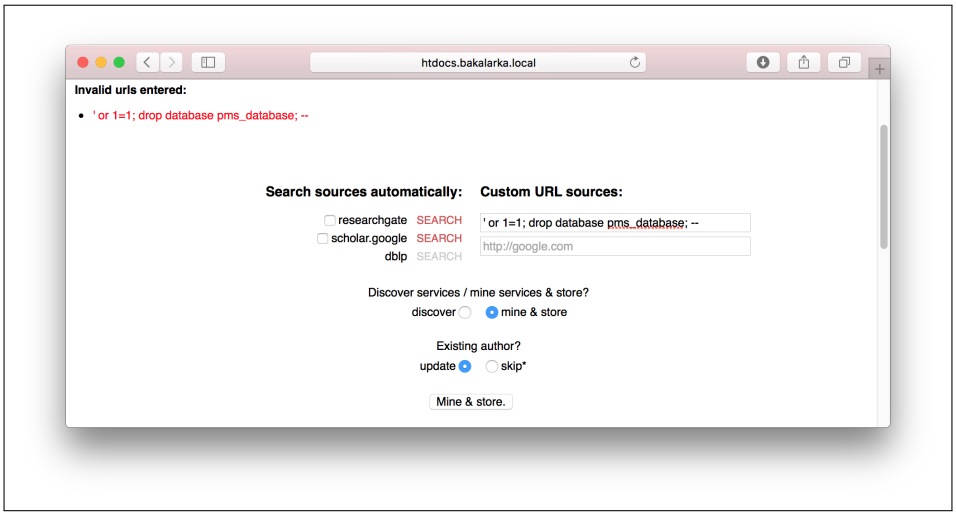
Během testování byla ověřována i správnost sjednocování publikací. Její správnost prokazuje například následující sjednocen dvojice publikací prof. Ing. Václava Skaly, CSc, viz obrázek F.4. Publikace byly získány ze služeb ResearchGate a GoogleScholar.



Obrázek F.4: Ukázka spárovaných publikací prof. Ing. Václava Skaly, CSc.

Příloha F.5 Kontrola bezpečnosti aplikace

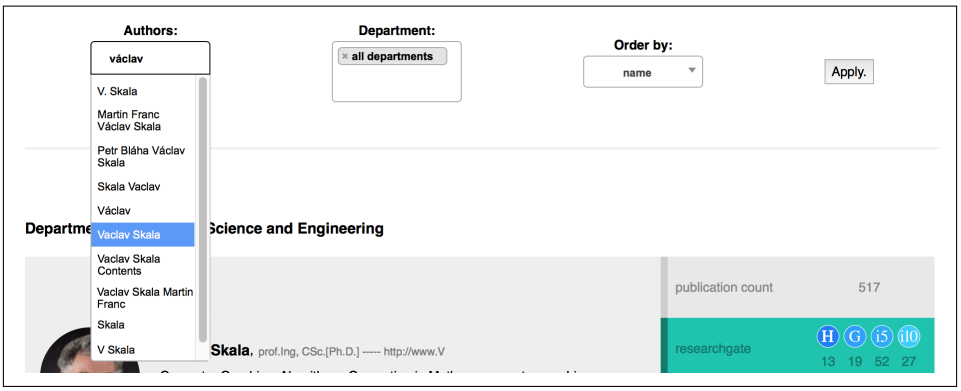
Tato část se věnuje tzv. SQL injection útokům. Série provedených testů, viz část 7.2.4, prokazuje odolnost vůči tomuto útoku. Následující obrázek F.5 ukazuje příklad pokusu o narušení chodu aplikace útokem SQL injection.



Obrázek F.5: Ukázka pokusu o narušení aplikace útokem SQL injection.

Příloha F.6 Kritické části aplikace

Testování odhalilo nejednotnost uváděných informací i v rámci jedné služby. Tato nejednotnost způsobuje redundantní záznamy v databázi, které ovlivňují použitelnost tohoto systému. Redundantní záznamy jsou ukázány v následujícím obrázku F.6, na němž je zaznamenáno 10 rozdílných autorů (vedených v databázi) reprezentující jednu reálnou osobu.



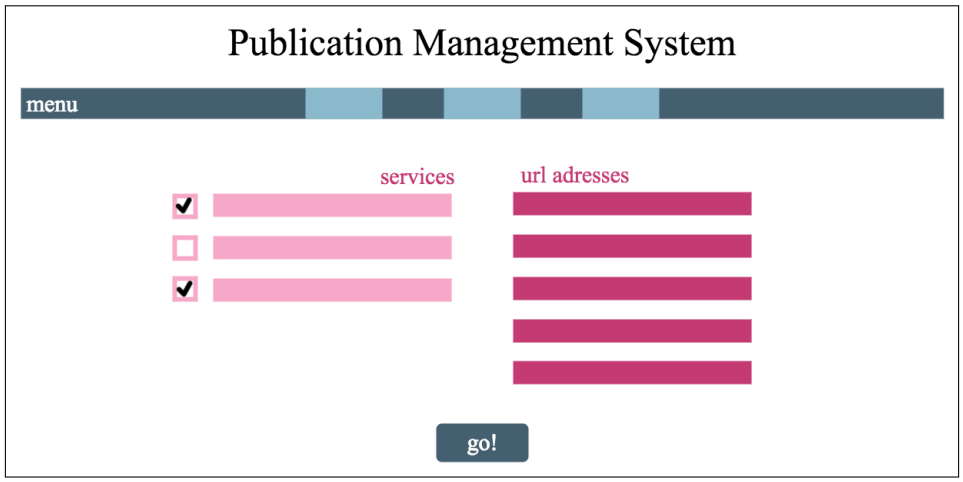
Obrázek F.6: Ukázka entropie zanesené do databáze nejednotnými informacemi uváděnými na stránkách služeb.

Příloha G Grafické uživatelské rozhraní

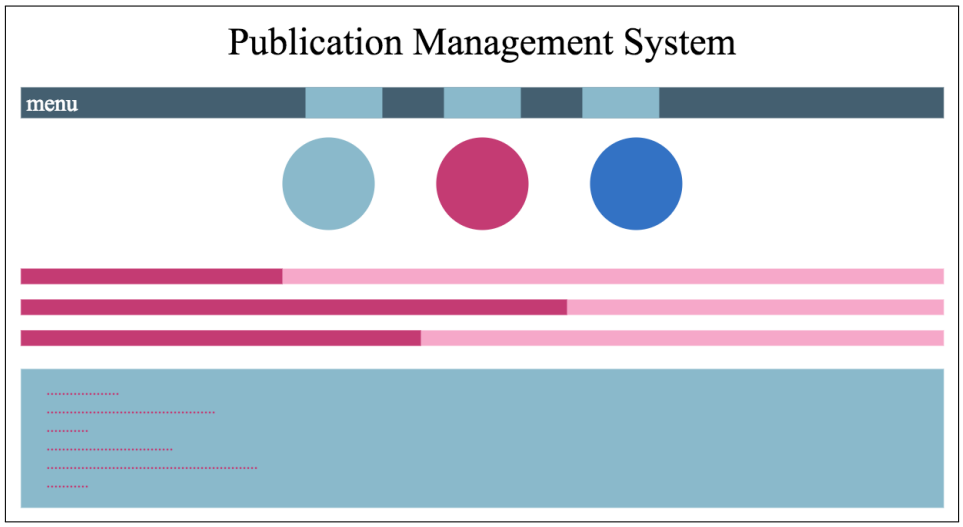
Realizaci GUI[74] předcházel jeho grafický návrh.

Příloha G.1 Wireframe webové aplikace

Při návrhu grafického zpracování byl kladen důraz na použitelnost a jednoduchý design aplikace, viz obrázky G.1 až G.3.

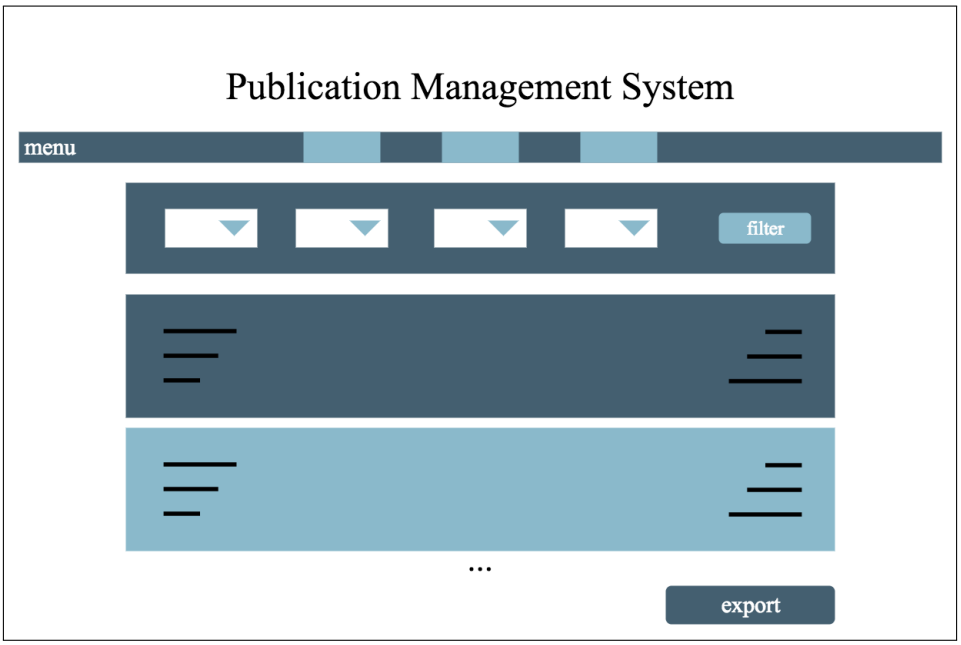


Obrázek G.1: Návrh GUI: administrace.



Obrázek G.2: Návrh GUI: administrace, vizualizace procesu zpracování dat.

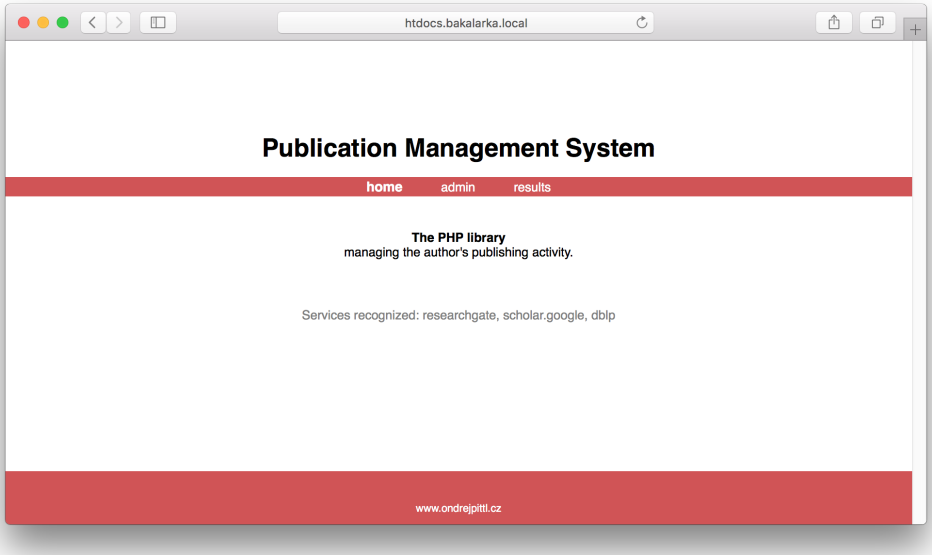
[74]Grafické uživatelské rozhraní bývá označováno GUI, tj. *Graphical User Interface*.



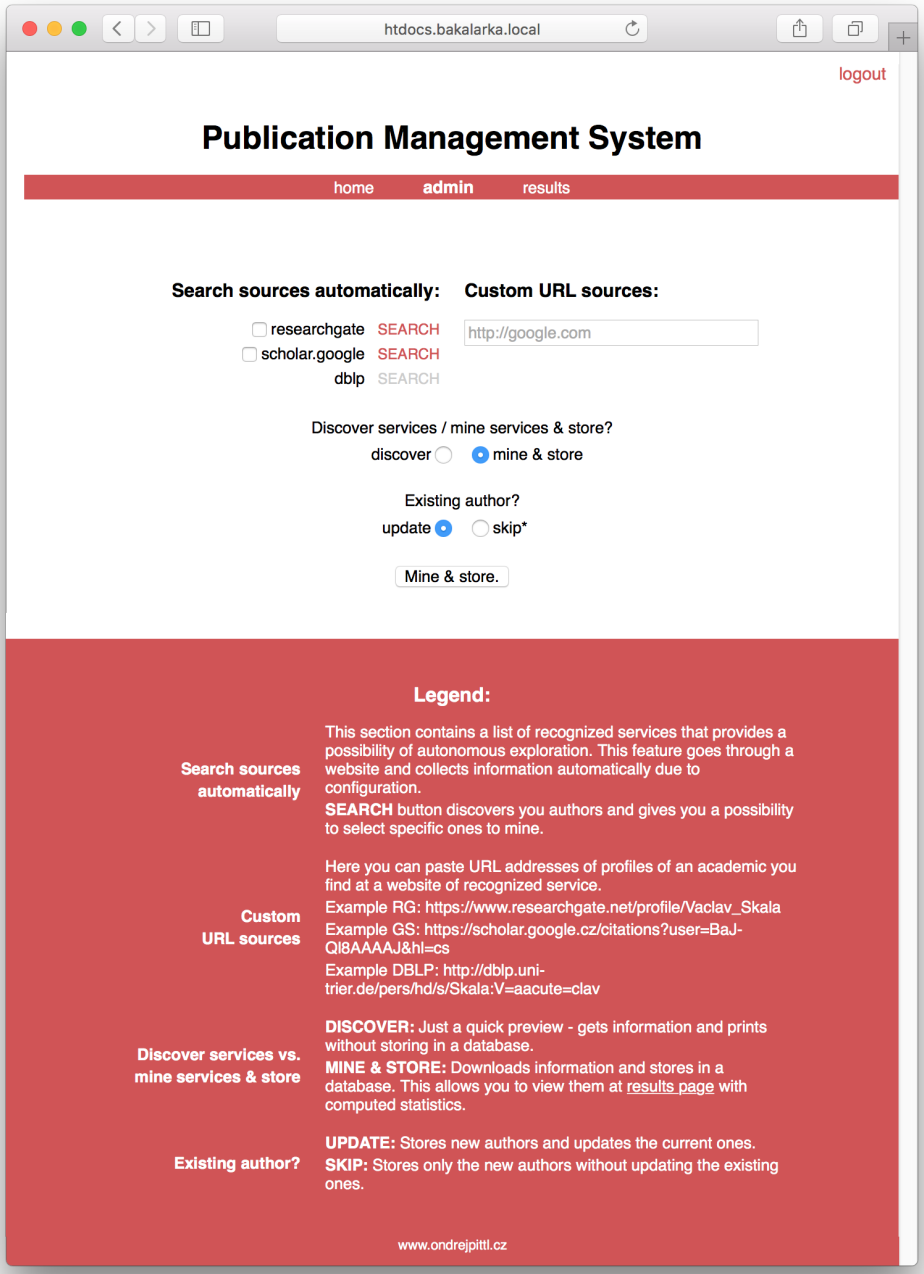
Obrázek G.3: Návrh GUI: stránka s výpisem seznamu publikací.

Příloha G.2 Realizace grafického uživatelského rozhraní

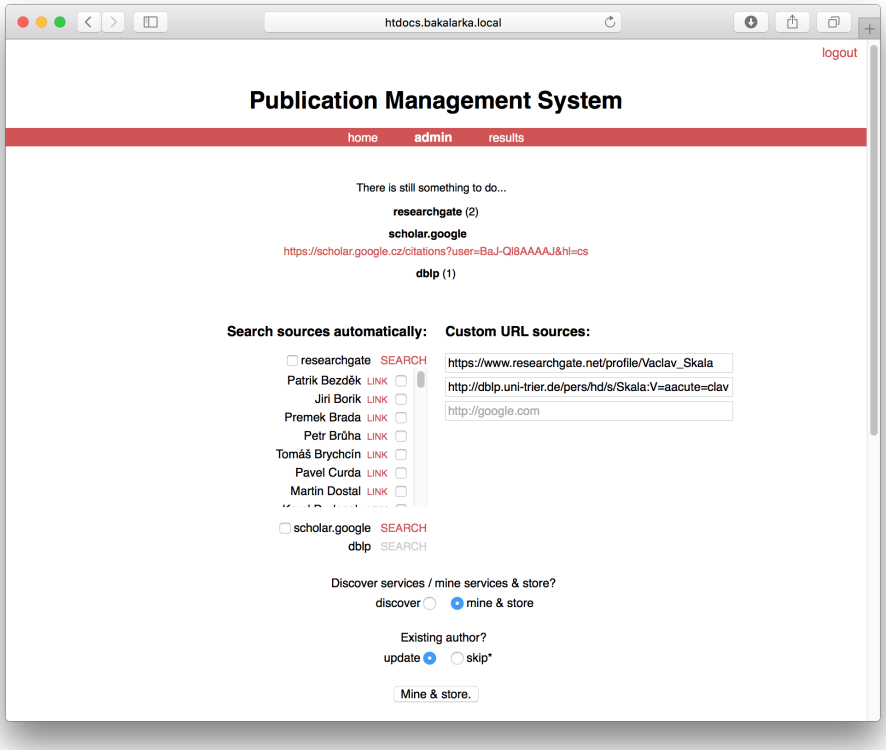
Realizace GUI vycházela z návrhu (viz příloha G.1). Byly však provedeny drobné změny z důvodu potřeb aplikace, viz obrázky G.4 až G.9.



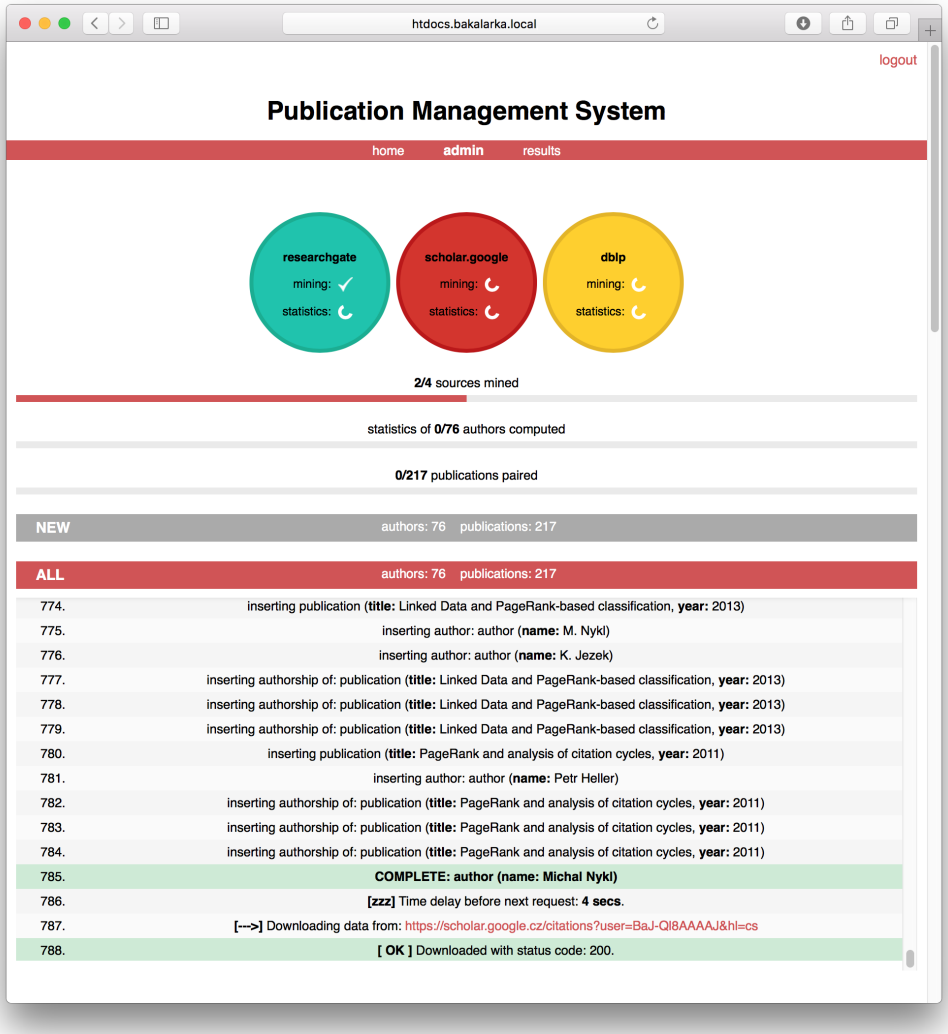
Obrázek G.4: GUI – hlavní stránka aplikace, rozcestník.



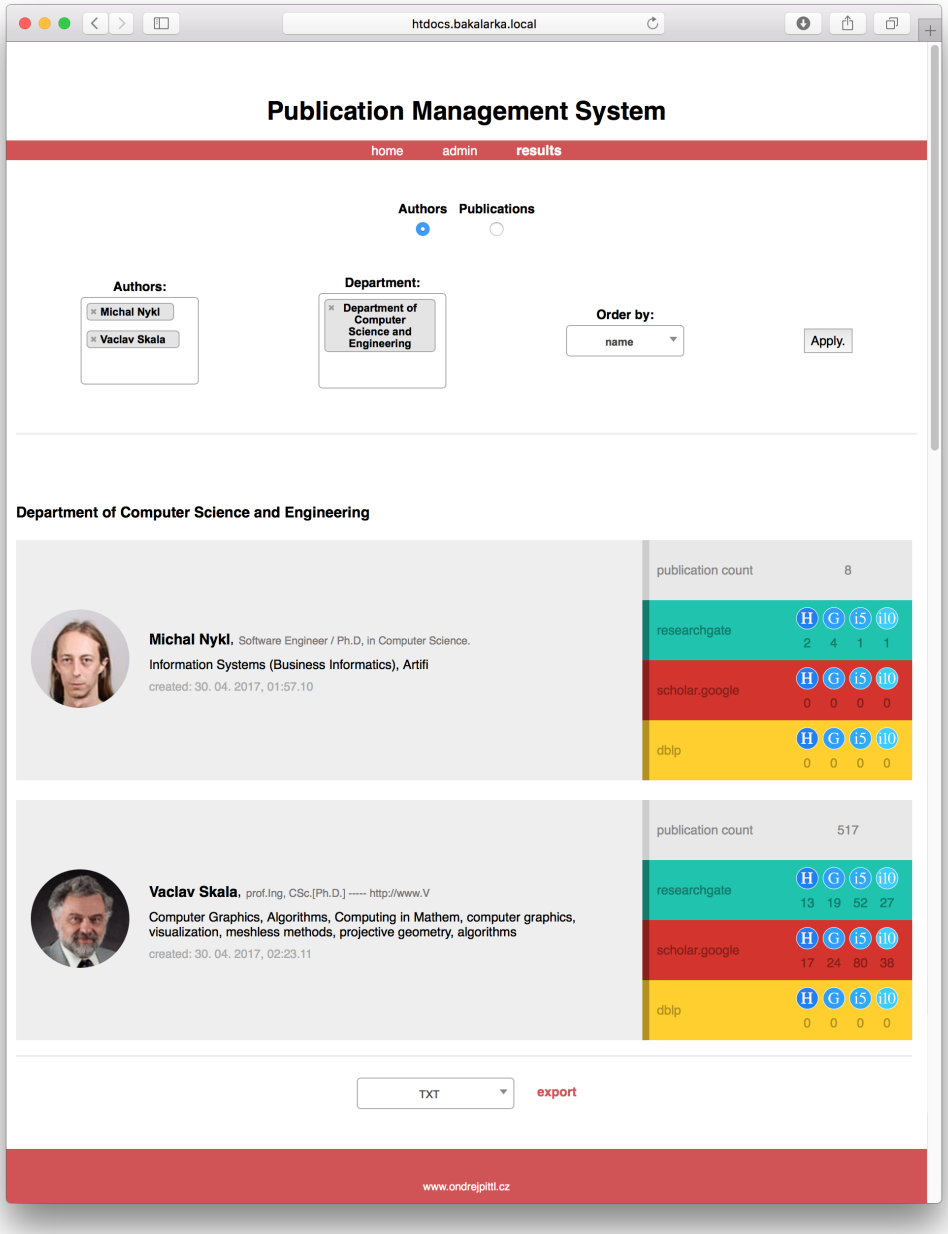
Obrázek G.5: GUI – prostředí administrace.



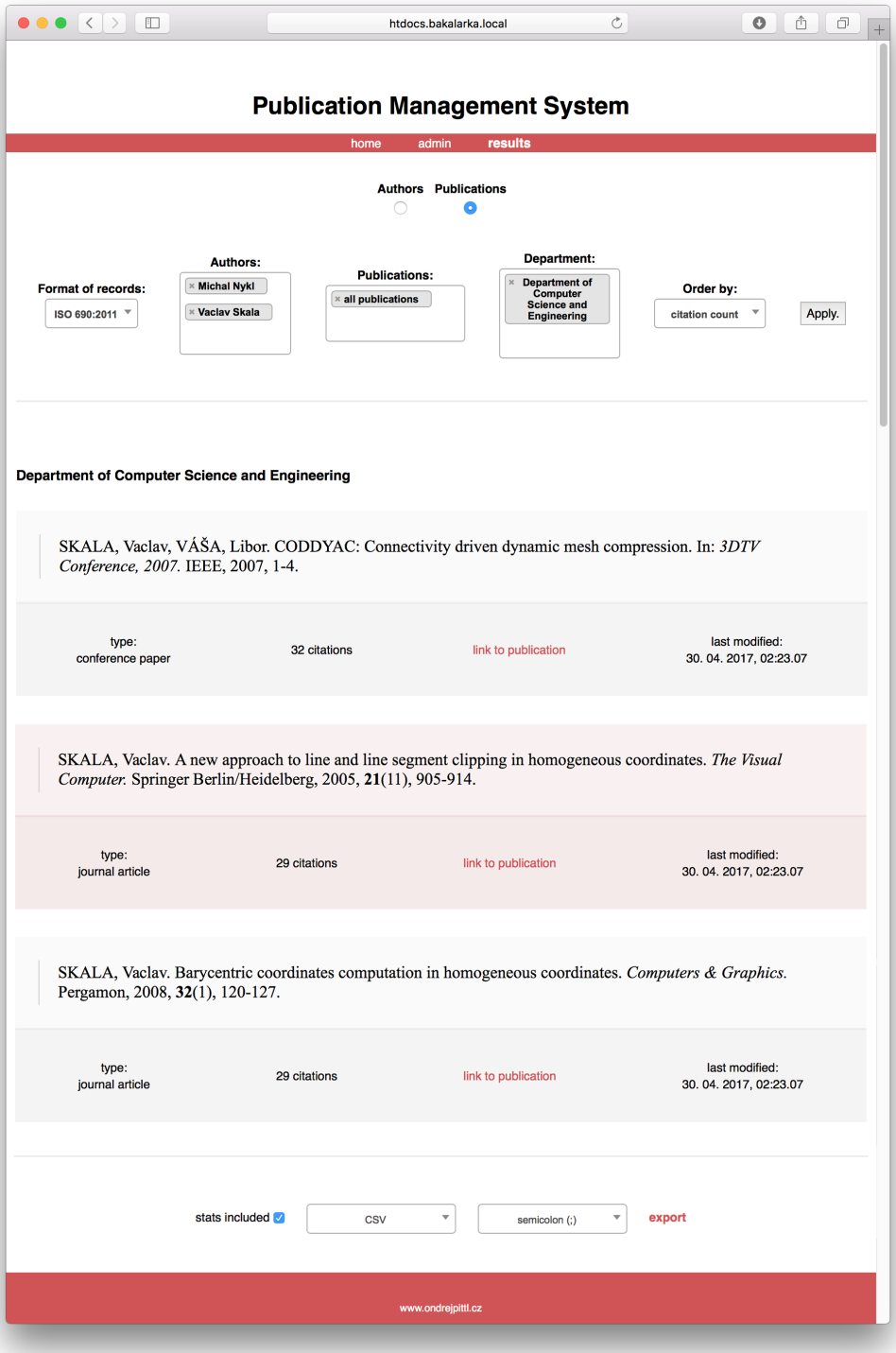
Obrázek G.6: GUI – prostředí administrace, ukázka dopředného průchodu a upozornění na nedokončenou činnost z minulého používání.



Obrázek G.7: GUI – prostředí administrace, vizualizace procesu knihovny.



Obrázek G.8: GUI – prezentace výsledků, seznam autorů vč. spočtených statistik.



Obrázek G.9: GUI – prezentace výsledků, seznam publikací formátovaných v souladu s citační normou ČSN ISO 690:2011.

Příloha H Uživatelský manuál

Systém pro správu publikací sestává z knihovny implementované v jazyce PHP, uchovávající data v MySQL databázi, a webových stránek využívajících JavaScript.

Příloha H.1 Minimální požadavky

Ke zprovoznění a používání této webové aplikace jsou zapotřebí minimálně **webový server** (např. *Apache HTTP Server*[75]) s nainstalovaným **PHP**[76] a **databázový server MySQL**[77]. K jejich instalaci je možné využít *cross-platformní* řešení, např. balík XAMPP[78], který jedinou instalací zprovozní Apache server, MySQL databázi, jazyk PHP a příp. další podpůrné nástroje. **Composer**[79] je nástroj, starající se o dostupnost podpůrných knihoven třetích stran a je rovněž umístěn na CD v `/support/`. Aplikace je zpřístupněna skrze **webový prohlížeč Google Chrome**[80].

Souhrn požadavků:

1. webový server,
2. databázový server MySQL,
3. PHP verze min. 5.6.30,
4. Composer,
5. Google Chrome s povoleným JavaScriptem.

Poznámka: V následujících krocích je postupováno nainstalovaným balíkem XAMPP (Apache 2.4.25, PHP 5.6.30).

Příloha H.2 Doporučené programové vybavení

Mimo programového vybavení popsaného v předchozí části, doporučuji využití nástroje phpMyAdmin[81] či nástroje Adminer přiloženého na CD adresáři v `/support/` poskytující pohodlnou správu databáze v prostředí webového rozhraní.

[75] Web: `https://httpd.apache.org`

[76] Popis instalace PHP viz: `http://php.net/manual/en/install.php`

[77] Stažení a instalace MySQL viz: `https://dev.mysql.com/doc/refman/5.7/en/installing.html`

[78] Stažení a instalace viz: `https://www.apachefriends.org/download.html`

[79] Stažení a instalace viz: `https://getcomposer.org/download`

[80] Stažení: `http://google.com/chrome`

[81] Popis instalace: `https://docs.phpmyadmin.net/cs/latest/setup.html`

Příloha H.3 Instalace aplikace

Instalace aplikace je závislá na operačním systému, webovém serveru a jeho nastavení. Obecně postup zahrnuje následující kroky:

1. **Založení databáze:** Spuštěním skriptu `/src/datbase.sql` v prostředí phpMyAdmin či Adminer nebo příkazem:

   ```
   mysql -u [username] -p < database.sql
   ```

   je vytvořena databáze pojmenovaná `pms_database`, v níž se nacházejí všechna data aplikace.

   Poznámka: Je nutné mít zapnutý databázový (příp. i webový) server.

2. **Umístění aplikace:** Adresář `/src/pms/` z přiloženého CD zkopírujeme do adresáře tzv. `DocumentRoot`, což je kořenový adresář webového serveru, kde jsou uchovávány zdrojové kódy jednotlivých webových stránek či aplikací. Umístění `DocumentRoot` je závislé na operačním systému, webovém serveru a jeho nastavení. Obvyklým pojmenováním `DocumentRoot` bývá `htdocs`, `httpdocs`, `www` či `public_html`.
3. **Správa závislostí** (např. knihoven): Z kořenového adresáře s projektem (kde se nachází `composer.json`) stáhneme využívané knihovny, na nichž je systém závislý, a to následujícím příkazem v příkazové řádce:

   ```
   composer install
   ```

4. **Konfigurace:** V souboru `/config/dbconnection-config.php` změňte dle nastavení MySQL serveru přístupové údaje pro připojení k databázi dle zdrojového kódu 2:

   Zdrojový kód 2: Nastavení přihlašovacích údajů k databázi.

   ```
   1  // Přihlašovací jméno a heslo k databázi.
   2  define('DB_USER_LOGIN',      'root');
   3  define('DB_USER_PASSWORD',   'root');
   ```

   Konfigurovatelná jsou i následující nastavení (viz zdrojový kód 3), v případě potřeby změňte.

   Zdrojový kód 3: Nastavení přístupových údajů k databázi.

   ```
   1  // Název hostitele a číslo portu databázového spojení.
   2  define('DB_HOST', '127.0.0.1');
   3  define('DB_PORT', '3306');
   ```

5. **Konfigurace webového serveru:** Konfiguraci webového serveru lze snadno provést překopírováním následujících záznamů (přiloženy na CD v `/support/`) na příslušná umístění.

   Řádek (viz zdrojový kód 4) vložte do souboru `hosts`. Umístění tohoto souboru se opět liší dle operačního systému, obvyklá umístění však bývají následující:

   - **MS Win.**[82]**:** `C:\Windows\System32\drivers\etc\hosts`
   - **Unix**[83]**:** `/etc/hosts`

Zdrojový kód 4: Nastavení souboru hosts.

```
127.0.0.1    pms.local
```

Následující blok kódu, viz zdrojový kód 5, se vkládá do konfiguračního souboru `httpd-vhosts.conf` samotného webového serveru. Obvyklým umístěním tohoto souboru bývá:

- **MS Win.:** `C:\XAMPP\apache\conf\extra\httpd-vhosts.conf`
- **Unix:** `/etc/apache2/extra/httpd-vhosts.conf`

V tomto bloku kódu je potřeba změnit cestu `DocumentRoot` k adresáři s projektem dle Vašeho operačního systému a konfigurace webového serveru, např. pro `Xampp` pod MS Windows bývá: `C:\xampp\htdocs`.

Zdrojový kód 5: Nastavení virtuálního webu (*virtual host*).

```
<VirtualHost pms.local>
  DocumentRoot [path-to-DocumentRoot]/pms
  ServerName pms.local
  RewriteEngine Off
  <Directory />
    Options Indexes FollowSymLinks MultiViews
      AllowOverride All
      Order allow,deny
      Allow from all
      Require all granted
  </Directory>
</VirtualHost>
```

Upozornění: Po nastavení virtuálního webu je nutné webový server restartovat.

[82]Umístění platné pro verze MS Windows: NT, 2000, XP, Vista, 7, 8, 10

[83]Mezi operační systémy založené na unixovém systému se řadí zejména Linux a macOS

6. **Aktualizace CA root certifikátů:** Pro ověření SSL certifikátů webových stránek (např. webové stránky služeb ResearchGate a GoogleScholar jsou chráněny zabezpečeným protokolem `https`) je zapotřebí mít aktuální balík certifikátů. Sada těchto certifikátů se nachází na přiloženém CD `/support/ca_cert.pem`. Umístěte tento soubor na libovolné místo (doporučuji například k instalaci `PHP`, v případě instalace pomocí XAMPP pod MS Windows `C:\xampp\PHP`). Ve složce s instalací PHP se nachází rovněž konfigurační soubor `php.ini`, ve kterém vyhledejte řádek, který znázorňuje zdrojový kód 6:

   Zdrojový kód 6: Původní řádek s CA certifikáty v php.ini.

```
1  ;curl.cainfo=
```

   Tento řádek odkomentujeme (odebereme ";", nachází-li se na začátku řádku) a vyplníme cestu k certifikátům CA `ca_cert.pem`, například následovně (viz zdrojový kód 7):

   Zdrojový kód 7: Příklad úpravy řádku s CA certifikáty v php.ini.

```
1  curl.cainfo=c:\xampp\php\ca_cert.pem
```

7. **Spuštění aplikace:** Nyní by měla být aplikace připravena k provozu, přístupná na adrese `pms.local/` ve webovém prohlížeči.

   Upozornění: Pro chod aplikace je potřeba, aby byl webový server i MySQL server v provozu.

Příloha H.4 Ovládání aplikace

Spuštěním webového serveru a MySQL serveru a následným přístupem na stránku `pms.local/` se dostáváte na domovskou stránku, která je rozcestníkem této webové aplikace. Odtud máte možnost přejít na stránku s Administrací nebo stránku s výsledky, viz příloha H.4.

Administrace

Vstup do administračního rozhraní je zabezpečen přihlášením. Vyplněním následujících přístupových údajů se přihlaste.

- **přihlašovací jméno:** `admin`
- **heslo:** `admin`

Stránka s administrací se skládá z formuláře, kterým se ovládá proces získávání informací, včetně následovného spočtení metrik h-index, g-index, i5-index a i10-index.

Levá část formuláře obsahuje seznam rozpoznaných služeb. Tyto služby lze výběrem autonomně projít a získat tak veškerá dostupná data nebo je zde umožněn dopředný průchod, jehož produktem je seznam autorů, které lze výběrem zahrnout do procesu získávání dat (viz část 6.3). Pravá část slouží k manuálnímu zadávání URL adres profilů autorů libovolné z rozpoznaných služeb, například následující:

```
https://www.researchgate.net/profile/Vaclav_Skala
https://scholar.google.cz/citations?user=BaJ-Ql8AAAAJ&hl=cs
http://dblp.uni-trier.de/pers/hd/s/Skala:V=aacute=clav
```

V dolní části se výběrem z voleb rozhodnete, zda chcete data uchovávat či jen zobrazit a zda se mají ukládat pouze noví autoři, jež v systému nejsou vedení, nebo se mají zároveň i aktualizovat existující autoři.

Tyto položky jsou vysvětleny také v legendě přímo na webových stránkách.

Poznámka: Ve třídě `Developer` v `/core/Developer.class.php` jsou definovány 2 atributy, které omezují počet procházených stran stránkovaných seznamů. Tato omezení snižují možnost zamezení přístupu ke službě. Upravte dle svých potřeb.

Prezentace informací

Na této stránce se nacházejí zpracované informace systémem pro správu publikací. Informace jsou strukturovány do seznamu autorů a seznamu publikací. Oba seznamy je možné filtrovat a řadit dle různých kritérií. Tato kritéria se pro každý ze seznamů liší, viz část 6.6.3.

V dolní části obou seznamů se nachází možnost jejich exportu ve formátech CSV a TXT (viz část 6.6.4). Seznam bude exportován v takové podobě, v jaké je vypsán na této stránce, tj. filtrovaná seřazená podmnožina s možností zahrnutí statistik či nikoliv. U souboru CSV je umožněna i volba oddělujícího znaku.