

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra počítačů

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Roman Kuchár**

Studijní program: Otevřená informatika
Obor: Softwarové systémy

Název tématu: **Vyhledávání zaměstnanců FEL ČVUT**

Pokyny pro vypracování:

Navrhněte a implementujte webovou aplikaci, která umožní vyhledávání lidí, výzkumných skupin a pracovišť na FEL ČVUT na základě jejich vědeckých výsledků.

Prozkoumejte existující datové zdroje (VVVS, KOS, Usermap) a relevantní aplikace na jiných katedrách (Profil zaměstnance na FIT ČVUT).

Pro serverovou část aplikace navrhněte REST api, pokud to bude nutné, rozšiřte stávající rozhraní pro získávání dat (VVVS api).

Uživatelské rozhraní pro vyhledávání by mělo být vzhledově kompatibilní se stávajícím vzhledem webových stránek FEL.

Navrhněte a implementujte automatické testovací skripty pro ověření funkčnosti REST rozhraní.

Seznam odborné literatury:

RESTful Web APIs, Leonard Richardson, Mike Amundsen, Sam Ruby, ISBN-13: 978-1449358068

NETTE framework, <https://netter.org/>

Learning XML, Second Edition, Erik T. Ray, ISBN-13: 978-0596004200

Vedoucí: Ing. Martin Klíma, Ph.D.

Platnost zadání: do konce zimního semestru 2016/2017


doc. Ing. Filip Železný, Ph.D.
vedoucí katedry



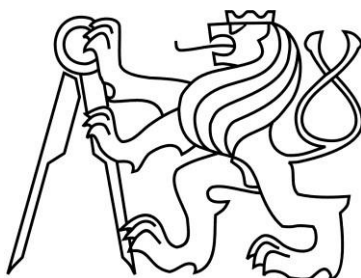

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 8. 10. 2015

České vysoké učení technické v Praze

Fakulta elektrotechniky

Katedra počítačů



Bakalárska práca

Vyhľadávanie zamestnancov FEL ČVUT

Roman Kuchár

Vedúci práce: Ing. Martin Klíma, Ph.D.

Študijný program: Otvorená informatika, Bakalársky

Odbor: Softwarové systémy

Január 2016

PodĎakovanie

Rád by som poĎakoval vĎetkým, ktorý mi pri tvorbe tejto práce podali informácie o internom fungovaní fakulty a ich dátových zdrojoch, na základe ktorých vznikala táto práca. Predovšetkým Ďakujem Ing. Martinovi Klímovi, Ph.D., vedúcemu bakalárskej práce za jeho vedenie, za pomoc, za kontakty a Ďalšie rady pri tvorbe tejto práce. V neposlednom rade by som chcel poĎakovať vĎetkým ľuďom, ktorý mi pomohli pri testovaní aplikácie.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval samostatne a že som uviedol všetky použité informačné zdroje v súlade s Metodickým pokynom o dodržiavaní etických princípov pri príprave vysokoškolských záverečných prác.

V Prahe dňa

.....

Roman Kuchár

Abstrakt

Cieľom tejto práce je navrhnuť a implementovať aplikáciu na vyhľadávanie ľudí a pracovísk Fakulty elektrotechnickej ČVUT na základe odbornosti. Odbornosť sa bude určovať podľa vedeckých článkov v časopisoch, zborníkoch konferencií a monografiách. Klientska časť aplikácie komunikuje pomocou REST API s webovou službou, ktorá spracováva a poskytuje všetky potrebné dáta. Design aplikácie bude poskytovať moderné užívateľské rozhranie kompatibilné s novo vytvorenými stránkami fakulty elektrotechnickej.

Klíčové slová

webová aplikácia, odbornosť, vyhľadávanie, zamestnanci, pracoviská

Abstract

The aim of this work is to design and implement an application to search people and departments of Faculty of Electrical Engineering on the basis of their expertise. Expertise will be determined according to their scientific articles in journals, conference proceedings and monographs. Client side of the application communicates using REST APIs from a web service that processes and provides all the necessary data. Design of the application will provide a modern user interface compatible with the newly created web sites of the Faculty of Electrical Engineering.

Keywords

web application, expertise, search, employees, departments

Obsah

1	Úvod	1
1.1	Motivácia	1
1.2	Cieľ	1
2	Analýza	2
2.1	Zainteresované osoby	2
2.1.1	Cieľová skupina	2
2.2	Scenáre použitia	2
2.2.1	Užívatelia	2
2.2.2	Prípady použitia	3
2.3	Požiadavky	4
2.3.1	Funkčné požiadavky	4
2.3.2	Nefunkčné požiadavky	5
2.4	Súčasný stav	6
2.4.1	Register informácií o výsledkoch	6
2.4.2	Publikačná činnosť	7
2.4.3	Vyhľadávanie publikácií a akcií	7
2.4.4	Aplikácia VVVS	8
2.4.5	Profil zamestnanca	9
2.4.6	Zhrnutie súčasného stavu	9
2.5	Dátové zdroje	10
2.5.1	VVVS (Veda, výskum, vonkajšie styky)	10
2.5.2	UDB	11
2.5.3	Usermap	12
3	Návrh	13
3.1	Architektúra	13
3.1.1	Vrstva dátového prístupu	14
3.1.2	Integračná vrstva	15
3.1.3	Biznis vrstva	16
3.1.4	Cache	18
3.1.5	Rest vrstva	18
3.1.6	Klientska aplikácia	21
3.2	Užívateľské rozhranie	22
3.2.1	Fáza 1	22
3.2.2	Fáza 2	23
4	Implementácia	25
4.1	Expertise search	25
4.1.1	Použité frameworky	25
4.1.2	Dátová vrstva (DTO)	25
4.1.3	Vrstva dátového prístupu (DAO)	26

4.1.4	Vrstva komponentov (Component)	26
4.1.5	Expertise service	27
4.1.6	Rest vrstva	27
4.1.7	Plánované úlohy	28
4.1.8	Nastavenie webovej aplikácie	28
4.1.9	Cache	28
4.2	Klientska aplikácia	29
4.2.1	Zobrazovacia logika	29
4.2.2	Aplikačná logika	30
5	Testovanie	31
5.1	Unit testing	31
5.1.1	JUnit (26)	31
5.1.2	Mockito (27)	31
5.1.3	Spring testing	31
5.2	FindBugs	31
5.3	Testovacie dátové sady	31
5.3.1	VVVS Mock API	32
5.3.2	UDB Mock API	32
5.4	Testovanie synchronizačných úloh	32
5.5	Testovanie klientskej aplikácie	32
5.6	Výsledok testovania	32
6	Záver	33
6.1	Splnenie cieľov	33
6.2	Možné vylepšenia	33
6.3	Budúcnosť aplikácie	33
7	Referencie	34
A.	Expertise search databázový model	36
B.	Obsah priloženého CD	37

Zoznam obrázkov

Obr. 2.1: Užívatelia.....	2
Obr. 2.2: Prípady použitia	3
Obr. 2.3: Funkčné požiadavky	4
Obr. 2.4: Nefunkčné požiadavky	5
Obr. 2.5: RIV - Vyhľadávací formulár (2)	6
Obr. 2.6: RIV – výsledky vyhľadávania (2).....	7
Obr. 2.7: Publikačná činnosť katedry počítačov (3)	7
Obr. 2.8: Vyhľadávanie publikácií a akcií – vyhľadávacie pole (5)	7
Obr. 2.9: Vyhľadávanie publikácií a akcií – výsledky vyhľadávania (5)	8
Obr. 2.10: Aplikácia VVVS – vyhľadávací formulár (6)	8
Obr. 2.11: Aplikácia VVVS – výsledok vyhľadávania (6).....	8
Obr. 2.12: Profil zamestnanca (7)	9
Obr. 3.1: Architektúra	13
Obr. 3.2: ER model internej databázy	14
Obr. 3.3: Zmeny okrajov a farby písma voči prototypu	23
Obr. 3.4: Zmena podfarbenia kontaktov a zobrazenie odkazu	24
Obr. 3.5: Stránkovanie	24
Obr. 3.6: Validáčna chyba	24
Obr. 3.7: Chyba nedostupnej aplikácie	24

1 Úvod

Aplikácia vyhľadávania zamestnancov FEL má byť prostriedkom pre jednoduché vyhľadávanie kontaktov najkompetentnejších ľudí a pracovísk na základe ich odbornosti. Pomocou jedného kliknutia je možné vyhľadať osoby majúce znalosť, resp. odbornosť vo vyhľadávanej problematike, ktoré následne môžeme s danou problematikou kontaktovať. Aplikácia má byť chýbajúcim kolieskom novo vzniknutých fakultných webových stránok, pomocou ktorého sa zlepší nielen interné fungovanie fakulty, či už medzi zamestnancami alebo aj medzi študentmi a zamestnancami, ale najmä prináša možnosť získať kontaktné údaje osobám mimo fakultu elektrotechnickú.

1.1 Motivácia

V nedávnej dobe prešli webové stránky fakulty elektrotechnickej veľkou zmenou. Vytvoril sa nový design a prerobila sa štruktúra obsahu. Aj napriek týmto novým zmenám sú webové stránky stále veľmi statické a ponúkajú možnosť prístupu k veľmi malému množstvu aplikácií, mimo nových stránok fakulty elektrotechnickej. Veľkým mínusom je chýbajúca možnosť vhodného vyhľadávania ľudí, ktoré momentálne existuje iba ako externá aplikácia UDB a najmä možnosť vyhľadávania odborníkov. Takéto vyhľadávanie je veľmi dôležité pre nadviazanie kontaktov, či už v rámci fakulty alebo aj mimo nej.

Predstavme si napríklad externú firmu, ktorá potrebuje vyriešiť konkrétny problém, no jej zamestnanci na to nemajú dostatočné znalosti. Potrebuje teda nájsť odborníka v danej problematike, ktorý by jej pomohol dopracovať sa k riešeniu problému. Vďaka aplikácii na vyhľadávanie podľa odbornosti jej bude stačiť vložiť kľúčové slovo problematiky do vyhľadávacieho poľa a ihneď sa dostane k niekoľkým kontaktom, či už priamo na osoby alebo na pracoviská fakulty elektrotechnickej. Na základe získaných kontaktov môže ihneď osloviť kompetentných ľudí, resp. pracoviská a dohodnúť sa na spolupráci.

Pre druhý príklad si skúsme predstaviť študenta fakulty. Má vlastný nápad na svoju záverečnú prácu a potrebuje si nájsť správneho vedúceho, ktorý by mu vedel s problematikou jeho nápadu najlepšie pomôcť. Priemerný študent poväčšine pozná iba zamestnancov fakulty, ktorý ho vyučovali nejaký predmet. Oproti reálnemu počtu zamestnancov je to malá množina ľudí. Vďaka tejto aplikácii by bol schopný vyhľadať a kontaktovať aj ľudí, ktorých nepozná, a ktorí by mu mohli ako vedúci práce vyhovovať oveľa viac práve vďaka ich odbornosti.

1.2 Cieľ

Cieľom projektu je vytvoriť intuitívnu a ľahko ovládateľnú aplikáciu zabezpečujúcu pohodlné vyhľadávanie osôb, pracovísk a ich kontaktov na základe odbornosti. Radiť tieto výsledky podľa počtu publikácií a ich váhy (počtu bodov publikácie) a zvýšiť tak spoluprácu fakulty s osobami mimo ČVUT, prípadne zjednodušiť študentom a zamestnancom hľadanie odborne znalých ľudí.

2 Analýza

Analýza popisuje budúcu funkcionálnosť aplikácie, dátové zdroje na základe ktorých bude aplikácia fungovať a aktuálne existujúce aplikácie.

2.1 Zainteresované osoby

Pre túto aplikáciu sú dôležité dve zainteresované osoby:

1. **Užívateľ aplikácie** je každá osoba, ktorá patrí do cieľovej skupiny (viď 2.1.1) užívateľov aplikácie.
2. **Správca dátových zdrojov** je osoba, ktorá je zodpovedná za dáta, na základe ktorých aplikácia pracuje.

2.1.1 Cieľová skupina

Aplikácia je zameraná predovšetkým na odbornú verejnosť, sekundárne na ľudí z ČVUT. Je nutné brať zvláštny ohľad na cudzojazyčných návštevníkov – anglická verzia webu.

2.2 Scenáre použitia

Scenáre použitia popisujú jednotlivé postupy pre vyhľadávanie podľa odbornosti. Pomocou nich je možné zapísať interakciu užívateľa a systému.

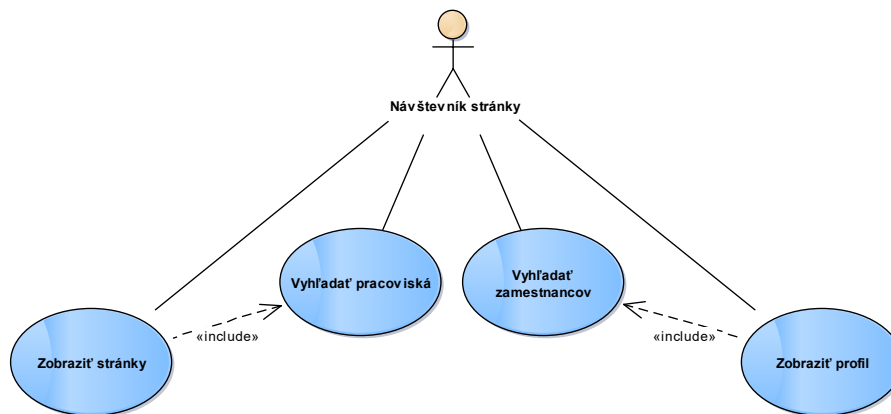
2.2.1 Užívatelia



Obr. 2.1: Užívatelia

Z dôvodu, že aplikácia poskytuje iba verejne prístupné dáta, nie je nutné riešiť autorizáciu užívateľa. Preto v aplikácii existuje iba jedna jediná rola používateľa, ktorou je každý jeden návštevník webových stránok. Tento užívateľ má prístup k všetkým funkcionalitám, ktoré webová aplikácia poskytuje.

2.2.2 Prípady použitia



Obr. 2.2: Prípady použitia

Vyhľadať zamestnancov

Aplikácia umožní užívateľovi vyhľadať zamestnancov podľa odbornosti vložením kľúčového slova do vyhľadávacieho poľa a zoradí výsledky podľa výšky vypočítanej odbornosti.

Scenár:

¹ Užívateľ zvolí vyhľadávanie zamestnancov a zadá vyhľadávanú odbornosť do vyhľadávacieho poľa. ² Systém vyhľadá zamestnancov podľa hľadanej odbornosti a zoradí ich podľa výšky odbornosti zamestnanca.

Vyhľadať pracoviská

Aplikácia umožní užívateľovi vyhľadať pracoviská podľa odbornosti vložením kľúčového slova do vyhľadávacieho poľa a zoradí výsledky podľa výšky vypočítanej odbornosti.

Scenár:

¹ Užívateľ zvolí vyhľadávanie pracovísk a zadá vyhľadávanú odbornosť do vyhľadávacieho poľa. ² Systém vyhľadá pracoviská, v ktorých pôsobia zamestnanci nájdení podľa hľadanej odbornosti a zoradí ich podľa súčtu výšky odbornosti zamestnancov.

Zobraziť profil

Aplikácia umožní zobraziť užívateľovi podrobné informácie o vyhľadanom zamestnancovi (číslo miestnosti kancelárie, telefónne číslo, emailová adresa).

Scenár:

¹ <include> Vyhľadať zamestnancov. ² Užívateľ klikne na meno zobrazeného zamestnanca z výsledkov vyhľadávania. ³ Systém načíta podrobné kontaktné informácie a zobrazí profilovú stránku zamestnanca.

Zobraziť stránky

Aplikácia umožní prejsť na oficiálne stránky vyhľadaného pracoviska.

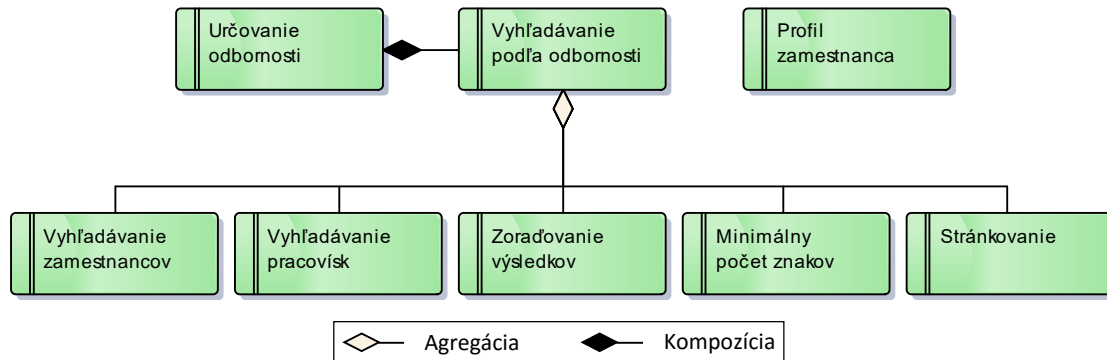
Scenár:

¹ <include> Vyhľadať pracoviská. ² Užívateľ klikne na názov zobrazeného pracoviska z výsledkov vyhľadávania. ³ Systém načíta podrobné informácie o zamestnancovi s UDB a zobrazí profilovú stránku zamestnanca.

2.3 Požiadavky

Požiadavky sú rozdelené na funkčné, ktoré popisujú funkcionality systému a jeho komponent a nefunkčné, ktoré špecifikujú nároky na systém a nasadenie aplikácie.

2.3.1 Funkčné požiadavky



Obr. 2.3: Funkčné požiadavky

Určovanie odbornosti

Na základe zadaných kľúčových slov sa určí odbornosť, ktorej výška sa odvíja od množstva a významnosti nájdených dát. Tieto dáta môžu pozostávať z vedecko výskumnej činnosti, pedagogickej činnosti, prípadne iných dostupných činnostiach.

Vyhľadávanie podľa odbornosti

Aplikácia bude vyhľadávať výsledky podľa odbornosti na základe zadaných kľúčových slov.

Vyhľadávanie zamestnancov

Výsledkom vyhľadávania budú nájdení zamestnanci, u ktorých bola určená odbornosť.

Vyhľadávanie pracovísk

Výsledkom vyhľadávania budú pracoviská a katedry, v ktorých pôsobia zamestnanci, u ktorých bola určená odbornosť. Výška odbornosti pracoviska sa určí sumou výšok odborností nájdených u zamestnancov pracoviska.

Zoradovanie výsledkov

Výsledky vyhľadávania budú zoradené zostupne podľa výšky určenej odbornosti.

Minimálny počet znakov

Možnosť vyhľadávania bude obmedzená na zadanie minimálne 2 znakov do vyhľadávacieho poľa.

2. Analýza

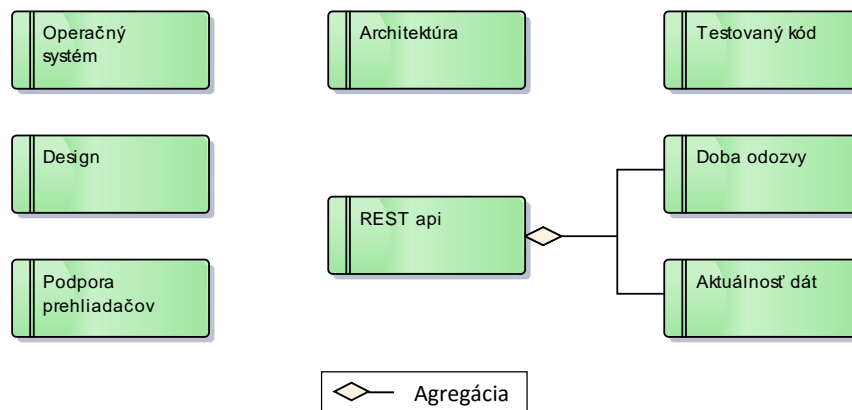
Stránkovanie

Výsledky vyhľadávania budú stránkované.

Profil zamestnanca

Profilová stránka zamestnanca bude poskytovať detailné údaje o zamestnancovi. Konkrétne: celé meno spolu s titulmi, pracovisko, emailovú adresu, telefónne číslo, číslo miestnosti, v ktorej má kanceláriu a náhľad všetkých jeho publikácií spojených z vyhľadávaným výrazom.

2.3.2 Nefunkčné požiadavky



Obr. 2.4: Nefunkčné požiadavky

Architektúra

Aplikácia bude rozdelená na serverovú časť a klientsku časť.

Operačný systém

Aplikáciu bude možné nasadiť na server, ktorý nie je závislý na operačnom systéme. Takýmto serverom môže byť napríklad Glassfish alebo JBOSS.

REST api

Serverová časť bude poskytovať vyhľadávané dáta pomocou aplikačného rozhrania REST api.

Doba odozvy

Doba odozvy aplikačného rozhrania pri priemernom používaní aplikácie nepresiahne hodnotu 0,5s.

Aktuálnosť dát

Zobrazované informácie budú synchronizované zo zdrojovými dátami tak, aby neboli staršie ako doba určená konštantou v aplikácii (napríklad jeden deň).

Testovaný kód

Kód aplikácie bude pokrytý testami pre prípadné odchytenie problémov počas možnej úpravy aplikácie.

2. Analýza

Design

Aplikácia bude poskytovať moderné užívateľské rozhranie kompatibilné s novými stránkami fakulty elektrotechnickej. Design bude podporovať zobrazenie aj na mobilných zariadeniach.

Podpora prehliadačov

Aplikácia bude podporovať všetky aktuálne verzie prehliadačov Mozilla Firefox, Google Chrome, Opera, Safari, Internet Explorer a Edge k dátumu 1.1.2015

2.4 Súčasný stav

V súčasnej dobe existuje niekoľko spôsobov na zobrazenie výsledkov vedy a výskumu zamestnancov. Napríklad na národnej úrovni Českej republiky existuje register informácií o výsledkoch (viď 2.4.1) formou ročného prehľadu (viď 2.4.2), vyhľadávania priamo v databáze VVVS, či už s verejným prístupom (viď 2.4.3), z autorizovaným prístupom (viď 2.4.4), alebo pomocou novovznikajúcej aplikácie Profil zamestnanca (viď 2.4.5).

2.4.1 Register informácií o výsledkoch

Register informácií o výsledkoch (RIV) (1) je jednou z častí (dátovou oblasťou) informačného systému výskumu, experimentálneho vývoja a inovácií (IS VaV). Aplikácia RIV poskytuje jednoduché a rozšírené vyhľadávanie vo verejne prístupných údajoch z RIV. Pojem jednoduchého vyhľadávania je mierne zavádzajúci, pretože toto vyhľadávanie sa skladá z väčšieho množstva polí (viď Obr. 2.5). Vyhľadané výsledky poskytujú náhľad k najdôležitejším informáciám (viď Obr. 2.6) a prípadne je možné prejsť na detail výsledku, kde sú poskytnuté všetky údaje, tvorcovia a ďalšie bližšie špecifikujúcejšie údaje o výsledku.

Podmienky pro vyhledávání: Jednoduché vyhledávání Rozšířené vyhledávání

Základní údaje o výsledku

Jazyk výsledku:

Dodavatel dat:

Identifikační kód:

Předkladatel výsledku

Název:

Tvůrce výsledku

Příjmení:

Výsledek

Název:

Rok uplatnění od do

Obor - skupina:

Klíčové slovo v anglickém jazyce:

ISBN nebo ISSN:

Vyhledat Vycištít

Obr. 2.5: RIV - Vyhľadávací formulár (2)

2. Analýza

Zpět na hledání Další >> Poslední >> Export

- 1. A 3D Approach to Facial Landmarks: Detection, Refinement, and Tracking**
Druh výsledku: D - Článek ve sborníku, Předkladatel výsledku: České vysoké učení technické v Praze / Fakulta elektrotechnická, Dodavatel výsledku: MŠMT, GA ČR, Název dokumentu: Proc. of 22nd International Conference on Pattern Recognition, Konsolidovaný rok uplatnění výsledku: 2014.
- 2. A 6DOF Motion Platform with Permanent Magnet Linear Motors**
Druh výsledku: J - Článek v odborném periodiku, Předkladatel výsledku: České vysoké učení technické v Praze / Fakulta elektrotechnická, Dodavatel výsledku: TA ČR, Název dokumentu: Transaction on Electrical Engineering, Konsolidovaný rok uplatnění výsledku: 2014.

Obr. 2.6: RIV – výsledky vyhľadávania (2)

2.4.2 Publikačná činnosť

Súčasťou webových stránok fakulty elektrotechnickej je verejne prístupný prehľad publikačnej činnosti, ktorý poskytuje ročné zobrazenie činnosti pracovníkov, resp. zamestnancov jednotlivých pracovísk. V prehľade nie je možné vyhľadávať a zobrazuje sa iba bibliografia bez ďalších podrobnejších informácií a bez akejkoľvek funkcionality.

K13136 - katedra počítačů

Publikační činnost 2015

Knihy, kapitoly v knihách a skripta, dizertace, výzkumné zprávy, nepublikované přednášky, stati ve sbornících, články v časopisech, patenty

Články v časopisech WoS

Anděl, M. - Kléma, J. - Krejčík, Z.: Network-Constrained Forest for Regularized Classification of Omics Data. *Methods*. 2015, vol. 83, p. 88-97. ISSN 1046-2023.

Bartoš, K. - Reháček, M.: IFS: Intelligent flow sampling for network security—an adaptive approach. *International Journal of Network Management*. 2015, vol. 25, no. 5, p. 263-282. ISSN 1055-7148.

Obr. 2.7: Publikačná činnosť katedry počítačov (3)

2.4.3 Vyhľadavanie publikácií a akcií

Táto aplikácia je webovým rozhraním pre vyhľadavanie verejne prístupných údajov o publikáciách a akciách (grantoch a výskumných zámeroch) v databáze VVVS (4). Poskytuje jednoduché vyhľadavanie pomocou jedného poľa, ktoré hľadá v priezviskách autorov publikácií a grantov v ich českých a anglických názvoch (viď Obr. 2.8). Pomocou rozšíreného vyhľadávania je možné radiť výsledky podľa zadaných kritérií. Rovnako ako v prehľade publikačnej činnosti sa zobrazuje iba bibliografia bez ďalších podrobnejších informácií (viď Obr. 2.9).

Jednoduché vyhledávání Rozšířené vyhledávání Přehledy po součástech Nápověda

Vyhledávání v záznamech VVVS

Jednoduché vyhledávání hledá v příjmeních autorů publikací a grantů a v jejich českých a anglických názvech.

Publikace Granty

Obr. 2.8: Vyhľadavanie publikácií a akcií – vyhľadávacie pole (5)

2. Analýza

Jednoduché vyhľadávání Rozšířené vyhľadávání Přehledy po součástech Náповěda

Výsledky hledání

Publikace

Malinský, R. - Jelínek, I.
Trend Analysis Framework
 In: Proceedings of the 14th International Conference WWW/INTERNET 2015. Maynooth, Greater Dublin: IADIS Press, 2015, vol. 14, p. 161-166. ISBN 978-989-8533-44-9.

Dytrych, T. - Hayes, A.C. - Launey, K.D. - Draayer, J.P. - Maris, P. - et al.
Electron-scattering form factors for 6Li in the ab initio symmetry-guided framework
 In: Physical Review C. 2015, vol. 91, no. 2, art. no. 024326, p. 024326-1-024326-8. ISSN 0556-2813.

Obr. 2.9: Vyhľadávání publikácií a akcií – výsledky vyhľadávania (5)

2.4.4 Aplikácia VVVS

Aplikácia VVVS je interný informačný systém ČVUT, ktorý eviduje výsledky vedy a výskumu (predovšetkým publikácie), akcie (granty, výskumné zámery a zmluvy) a ďalšie aktivity vedecko-výskumných pracovníkov vo vedeckej komunite. Aplikácia VVVS slúži k odosielaniam výsledkov do RIV (Register informácií a výsledkov), exportom pre statické analýzy aj k interným hodnoteniam vedeckovýskumných činností (4). Na prácu s touto aplikáciou je nutná autorizácia, ktorá otvorí prístup k neverejným dátam z databázy VVVS. Prístup k tejto aplikácii majú iba akademickí pracovníci, doktorandi zaradení na katedrách, prípadne ostatní zamestnanci, ktorí majú schválenú žiadosť vedúcim pracoviska. Aplikácia poskytuje možnosť vyhľadávania publikácií pomocou formulára, ktorý sa skladá z väčšieho množstva polí (viď Obr. 2.10). Výsledkom vyhľadávania je zoznam publikácií, ktorý zobrazuje názov a typ publikácie, autora a rok vydania (viď).

Agenda publikací

Kritéria pro výběr publikací

Typ	-- libovolný --	Fakulta autora	13000 - Fakulta elektrotechnická	moje
Katedra autora	13139	Jméno autora	Klíma Martin Ing. Ph.D.	ja
Období	2000 vloni letos - 2015 vloni letos	Stav	-- libovolný --	
Název v orig.		Název angl.		
Jazyk	angličtina	Ident.		Barva -- libovolná -- Druh hodnocení -- libovolný --
WoS		Scopus		doi
Vydavatel		Místo vydání		Stát vydání
ISBN		ISSN		Ext. číslo grantu
Vytvořil	ja	kdy	dnes	Opravil ja kdy dnes
RVV2011	je libovolné	RVV2012	je libovolné	RVV2013 je libovolné RVV2014 je libovolné
Ražení	nejprve podle typu	pak podle autorů	a nakonec podle názvu	

Zobrazit seznam Vyprázdnit položky Bibliografické citace česky Bibliografické citace anglicky BibTeX

Obr. 2.10: Aplikácia VVVS – vyhľadávací formulár (6)

Seznam publikací

vybraných podle kritérií zadaných na předchozí stránce

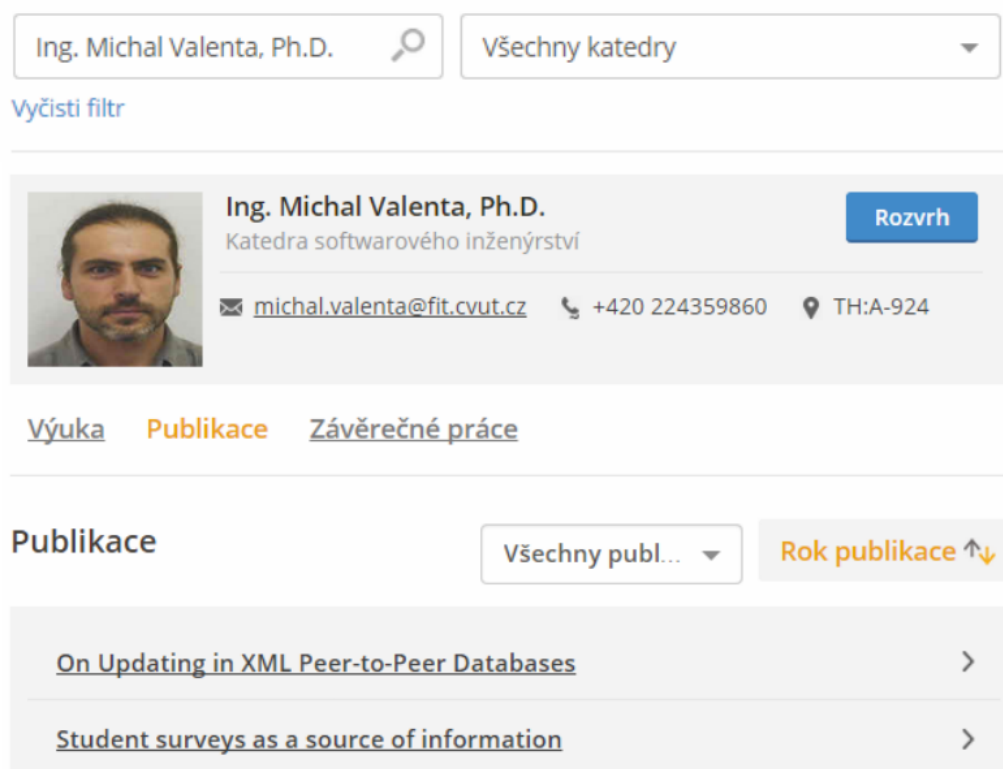
Seznam obsahuje 15 záznamů Vložit nový záznam

Č.	Typ	Autoři	Rok	Název (zkrácený)	Stav	Ident.	RIV	RVV2011	RVV2012	RVV2013	RVV2014	Druh	WoS	Scopus	doi
1	ASW	Macík.M., Bašek.J., Klíma.M.	2012	Forms602 to UIPProtocol Format Converter	Odesláno do RIV	197256	RIV								Detail Opravit
2	ASW	Macík.M., Bašek.J., Klíma.M.	2011	UIP Portal	Odesláno do RIV	197209	RIV								Detail Opravit

Obr. 2.11: Aplikácia VVVS – výsledok vyhľadávania (6)

2.4.5 Profil zamestnanca

Profil zamestnanca je novo vznikajúca aplikácia fakulty informačných technológií pod vedením Ing. Jakuba Jirůtku a Ing. Vladimíra Mezera, ktorá vychádza z požiadaviek dekana FIT ČVUT, Prof. Pavla Tvrdíka. Aplikácia má poskytovať zoznam akademických zamestnancov fakulty, ich kontaktné údaje a informácie o výkovej a publikačnej činnosti a má byť súčasťou fakultných webových stránok (viď Obr. 2.12). Jej cieľom má byť informovanie verejnosti a zvýšenie prestíže fakulty. V rámci ČVUT podobná aplikácia neexistuje, a preto je pri jej vývoji zohľadnená na možnosť nasadenia na iných fakultách.



The screenshot shows a user interface for a faculty member profile. At the top, there is a search bar containing 'Ing. Michal Valenta, Ph.D.' and a dropdown menu set to 'Všetchny katedry'. Below this is a 'Vyčisti filtr' link. The main profile card features a photo of Ing. Michal Valenta, Ph.D., his name, and his department 'Katedra softwarového inženýrství'. A blue 'Rozvrh' button is visible. Contact information includes an email 'michal.valenta@fit.cvut.cz', a phone number '+420 224359860', and a location 'TH:A-924'. Below the profile card are three tabs: 'Výuka', 'Publikace', and 'Závěrečné práce'. The 'Publikace' tab is active, showing a list of publications. The first publication is 'On Updating in XML Peer-to-Peer Databases' and the second is 'Student surveys as a source of information'. Each publication entry has a right-pointing arrow.

Obr. 2.12: Profil zamestnanca (7)

2.4.6 Zhrnutie súčasného stavu

V nasledujúcej tabuľke je vidieť, že neexistuje žiadna aplikácia, ktorá by spĺňala základné požiadavky na vyhľadávanie podľa odbornosti, ako sú pohodlné vyhľadávanie pomocou jedného poľa a následné zoradovanie výsledkov tak, aby mal užívateľ ihneď možnosť kontaktovať najkompetentnejšiu osobu alebo pracovisko bez žiadneho ďalšieho dodatočného hľadania.

2. Analýza

Aplikácia	Jednoduché vyhľadávanie publikácií	Radenie podľa váhy publikácie	Kontaktné údaje autorov alebo pracovísk	Verejný prístup
Register informácií o výsledkoch	✗	✗	✗	✓
Prehľad publikácií	✗	✓	✗	✓
Vyhľadávanie publikácií a akcií	✓	✗	✗	✓
Aplikácia VVVS	✗	✓	✗	✗
Profil zamestnanca	✗	✗	✓	✓

Tabuľka 2.1: Základné požiadavky voči súčasným aplikáciám

2.5 Dátové zdroje

Pre tento projekt je najdôležitejším dátovým zdrojom databáza VVVS (viď 2.5.1) uchovávajúca výsledky vedy a výskumu, na základe ktorých sa bude určovať odbornosť zamestnanca. Druhým dôležitým zdrojom je databáza UDB (viď 2.5.2), ktorá uchováva podrobné informácie o zamestnancoch a pracoviskách fakulty elektrotechnickej.

2.5.1 VVVS (Veda, výskum, vonkajšie styky)

Na získavanie dát z databázy VVVS je vytvorené aplikačné rozhranie (API) v podobe REST webových služieb (8). Toto API poskytuje iba určité, verejnosti prístupné dáta z databázy, ktoré vo väčšinovej miere postačujú na správne fungovanie tohto projektu. VVVS API je stále vo fáze vývoja, čo znamená, že v budúcnosti bude ponúkať viac údajov o publikáciách ako sú kľúčové slová a abstrakt publikácie, ale z dôvodu vydavateľských práv nemôže nikdy poskytovať obsah celej publikácie. Aplikácia teda bude určovať odbornosť iba na základe názvu, abstraktu a kľúčových slov publikácie.

2.5.1.1 Typy obsahu

VVVS API poskytuje niekoľko typov obsahu a z nich vypíšem tie, ktoré sú podstatné v mojej aplikácii.

Publication

Objekt reprezentujúci publikáciu. Aktuálne tento objekt neposkytuje informácie o abstrakte publikácie a o jej kľúčových slovách. Pre chod mojej aplikácie sa počíta taktiež s týmito dátami.

Atribúty:

biblio :string - biblio
id :int – id publikácie
language :string - jazyk
name :string – názov

releaseDate :string - dátum vydania
type :string - typ
url :string - url
year :int – rok

2. Analýza

Authorship

Objekt reprezentujúci autora publikácie, jeho typ účasti a podiel práce na danej publikácii. Aplikácia bude akceptovať iba typ účasti = 01 (autor)

Atribúty:

personalNumber :int - osobné číslo autora
type :string - typ účasti

Person

Objekt reprezentujúci a poskytujúci informácie o zamestnancovi ČVUT.

Atribúty:

firstName :string - Krstné meno
fullName :string - celé meno aj s titulmi
lastName :string - priezvisko
personalNumber :int - osobné číslo
username :string - užívateľské meno

Division

Objekt reprezentujúci katedru/pracovisko.

Atribúty:

code :int/string? - kód organizačnej jednotky. V aktuálnej verzii je označený ako id s dátovým typom integer. V budúcnosti má byť premenovaný na „code“, no jeho budúci dátový typ nie je jasne určený.
name :string - názov organizačnej jednotky

RIV

Objekt reprezentujúci hodnotenie (počet bodov) publikácie podľa RIV.

Atribúty:

divisionId :int - ID katedry
points :string - počet bodov
url :int - odkaz
year :int - rok

2.5.2 UDB

UDB je databáza v správe fakulty elektrotechnickej, ktorá udržiava podrobné informácie o svojich zamestnancoch a pracoviskách. K UDB bohužiaľ neexistuje žiadna dokumentácia, takže všetky informácie ohľadne poskytovaných dát a prístupu k nim mám iba z konzultácií s p. Ing. Janom Kočím a p. Ing. Pavelom Šedekom. Dáta s UDB sú prístupné pomocou protokolu LDAP. To je definovaný protokol pre ukladanie a prístup k dátam z adresárového serveru, kde sú jednotlivé položky uložené vo forme záznamov a usporiadané do stromovej štruktúry (9).

2.5.2.1 Typy obsahu

Pri jednej z konzultácií som mal možnosť nahliadnuť na poskytované dáta, no z dôvodu chýbajúcej dokumentácie bolo možné štruktúru UDB získať jedine priamym prístupom. Moja žiadosť o LDAP prístup k UDB bohužiaľ ostala nevypočutá, a preto nasledujúce typy obsahu sú zapísané abstraktne podľa webovej aplikácie UDB.

User

Objekt poskytujúci podrobné informácie o zamestnancovi FEL ČVUT.

Atribúty:

firstName :string - krstné meno
fullName :string - celé meno aj s titulmi
lastName :string - priezvisko
phone :string - telefónne číslo
email :string - emailová adresa
username :string - užívateľské meno
room :string - miestnosť kancelárie
department :string - kód pracoviska

2. Analýza

Department

Objekt poskytujúci podrobné informácie o pracovisku FEL ČVUT.

Atribúty:

code :string – kód pracoviska

name :string – Názov pracoviska

address :string – adresa

phone :string – telefón

homepage :string – domovská stránka

headmaster :string – užívateľské meno vedúceho pracoviska

2.5.3 Usermap

Usermap poskytuje základne informácie o osobách z celého ČVUT ako je meno, užívateľské meno, emailové adresy, telefóny atď. Fakulta informačných technológií vytvorila aplikačné rozhranie pre Usermap (10), ktoré prvotne bolo plánované použiť na získavanie podrobných údajov o zamestnancoch FEL a jeho pracoviskách. Po objavení UDB bolo použitie Usermap API zavrhnuté a jediným pozostatkom z neho je získavanie zverejnených fotografií zamestnancov, pomocou priameho odkazu na fotografie a zadaním užívateľského mena. Pokiaľ zamestnanec nemá vloženú fotografiu, vráti sa defaultne nastavený obrázok.

Príklad:

<https://usermap.cvut.cz/photos/kucharo2>

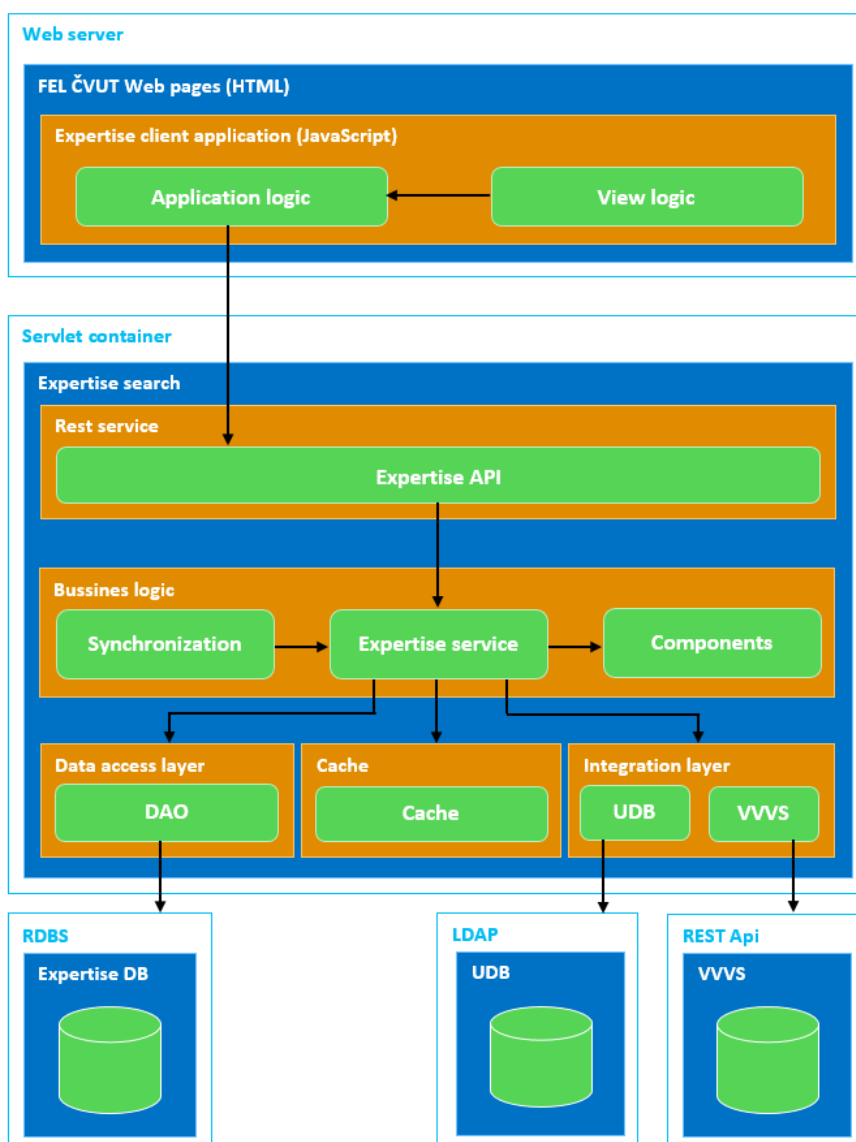
3 Návrh

Návrh aplikácie popisuje architektúru aplikácie a ako bude zabezpečená funkcionlita aplikácie.

3.1 Architektúra

Architektúra popisuje funkcionlitu a komunikáciu medzi jednotlivými komponentami aplikácie. Tie sú rozdelené do dvoch častí:

1. **Serverová časť** s pracovným názvom Expertise search, ktorá zabezpečuje biznis logiku (viď 3.1.3), komunikuje s internou databázou (viď 3.1.1), integruje spojenie s externými dátovými zdrojmi (viď 3.1.2) a ponúka aplikačné rozhranie (viď 3.1.5) s pracovným názvom Expertise API.
2. **Klientska aplikácia** (viď 3.1.6), ktorá reprezentuje dáta získane pomocou Expertise API.



Obr. 3.1: Architektúra

3.1.1 Vrstva dátového prístupu

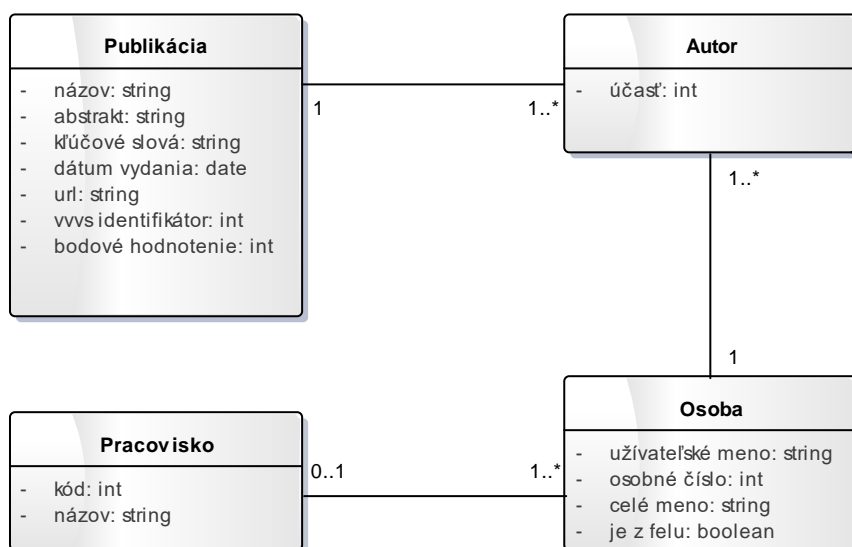
Vrstva dátového prístupu alebo odbornejšie nazvané DAO vrstva (viď Obr. 3.1 Data access layer) je vrstva, ktorá sa stará o spojenie z internou databázou a vykonávanie všetkých potrebných volaní na databázu. Interná databáza môže byť nasadená na inom serveri, ako je nasadená aplikácia Expertise search.

3.1.1.1 Interná databáza

Interná databáza, nazvaná Expertise DB (viď Obr. 3.1 Expertise DB), má za úlohu udržiavať dáta potrebné pre vyhľadávanie. Sú to primárne dáta z databázy VVVS, ktoré je nutné ukladať do internej databázy z niekoľkých dôvodov. Prvým hlavným dôvodom je aktuálne VVVS API a jeho komunikačný jazyk, vďaka ktorému by jedno vyhľadávanie v aplikácii znamenalo množstvo volaní VVVS API. Zároveň tým, že pre tento projekt vyhľadávame iba zamestnancov fakulty elektrotechnickej, je nutné vyhľadávať iba v kompetentných publikáciách a to s daným jazykom nie je možné dosiahnuť.

Druhým hlavným dôvodom je rýchlosť aplikácie. Tým, že sú v internej databáze uložené najnutnejšie dáta je zabezpečené, že je možné poskytnúť výsledok vyhľadávania bez nutnosti volania externých dátových zdrojov, čím sa ušetrí čas, ktorý by bol minúty na čakanie za odpoveďou od dátového zdroja. Interná databáza zároveň ponúka možnosť indexovania (11), ktorým sa zrýchlia vyhľadávacie procesy v databáze.

Interná databáza je vytvorená pomocou open source objektovo relačného systému PostgreSQL (12). Pozostáva zo štyroch navzájom prepojených tabuliek. Kompletný databázový model viď A. Expertise search databázový model.



Obr. 3.2: ER model internej databázy

Publikácia

Tabuľka udržiava informácie o publikáciách fakulty elektrotechnickej.

- **názov** – názov publikácie. Toto pole bude indexované.
- **abstrakt** – abstrakt publikácie. Toto pole bude indexované.

3. Návrh

- **klúčové slová** – klúčové slová publikácie oddelené pomocou bodkočiarky (;). Toto pole bude indexované.
- **dátum vydania** - dátum vydania publikácie bez časovej zložky.
- **url** – odkaz na celý obsah publikácie.
- **vvvs identifikátor** – identifikátor publikácie podľa databázy VVVS.
- **bodové hodnotenie** – bodové hodnotenia publikácie podľa RIV.

Osoba

Tabuľka udržiava informácie o osobách, ktoré sú autorom aspoň jednej publikácie.

- **užívateľské meno** – užívateľské meno osoby. Toto pole bude indexované.
- **osobné číslo** - osobné číslo osoby.
- **celé meno** – celé meno osoby spolu z jeho titulmi.
- **je z felu** – príznak označujúci osoby, ktoré sú zamestnancami FEL.

Autorstvo

Prepojovacia tabuľka spájajúca publikácie a osoby.

- **účasť** – výška účasti autora na tvorbe publikácie. Hodnota v percentách.

Pracovisko

Tabuľka udržiavajúca informácie o pracoviskách fakulty elektrotechnickej.

- **kód** – kódové označenie pracoviska. Toto pole bude indexované.
- **názov** – názov pracoviska.

3.1.2 Integrovaná vrstva

Integrovaná vrstva (viď Obr. 3.1 Integration layer) zaobstaráva integráciu s externými aplikáciami. Pre tento projekt sú to aplikácie na prístup k externým dátovým zdrojom VVVS (viď 3.1.2.1) a UDB (viď 3.1.2.2).

3.1.2.1 VVVS

Integrácia s externým dátovým zdrojom VVVS (viď Obr. 3.1 VVVS) je najdôležitejšou integráciou, pretože vďaka nej má aplikácia prístup k potrebným dátam na určovanie vyhľadávania. Táto integrácia sa neprevoláva v reálnom čase pri používaní aplikácie, ale prevoláva sa iba pri aktualizácii dát v internej databáze (viď 3.1.3.1). Zo všetkých ponúkaných dátových zdrojov VVVS api (8) aplikácia používa určitú množinu:

- GET /divisions/{divisionCode}/people
 - Vrátí zoznam pracovníkov na katedre.
- GET /divisions/{divisionCode }/publications
 - Vrátí zoznam všetkých publikácií pracovníkov katedry.
- GET /publications/{id}/rivs
 - Vrátí všetky RIV body danej publikácie.

3. Návrh

- GET /publications/{id}/authorships
 - Vrátí všetkých autorov danej publikácie.
- GET /people/{personalNumber}
 - Vrátí daného pracovníka.

3.1.2.2 UDB

Integrácia s externým dátovým zdrojom UDB (viď Obr. 3.1 UDB) poskytuje detailné informácie (najmä kontakty) o zamestnancoch a pracoviskách fakulty elektrotechnickej. Keďže k LDAP prístupu na UDB neexistuje žiadna dokumentácia a prístup k tomuto dátovému zdroju mi nebol poskytnutý, nedokážem presne popísať LDAP volanie. Aplikácia vyžaduje 3 rôzne volania na UDB, ktoré vďaka informáciám získaným pri konzultácii s p. Ing. Pavlom Šedekom, viem potvrdiť, že sú možné.

- Volanie, ktoré vráti všetky pracoviská FEL.
- Volanie, ktoré vráti práve jedno pracovisko podľa jeho kódu.
- Volanie, ktoré vráti zamestnanca FEL podľa jeho užívateľského mena.

3.1.3 Biznis vrstva

Biznis vrstva (viď Obr. 3.1 Bussines layer) je časť aplikácie, ktorá definuje jej pravidlá fungovania. Zabezpečuje synchronizáciu dát (viď 3.1.3.1), podporné výpočty a mapovanie (viď 3.1.3.2) a spojenie všetkých vrstiev dokopy (viď 3.1.3.3).

3.1.3.1 Synchronizácia

Najdôležitejšou úlohou aplikácie je synchronizácia dát v internej databáze (viď Obr. 3.1 Synchronization), ktorá sa vykonáva vďaka naplánovaným úlohám (tzv. scheduled jobs). Tieto úlohy sa spúšťajú dopredu definovanými časovými intervalmi. Vďaka týmto nastavovateľným intervalom je možné zabezpečiť, aby sa aktualizácia databázy vykonávala v predpokladanej dobe najnižšej aktivity užívateľov aplikácie, napríklad počas nočných hodín. Aplikácia ponúka tri úlohy, ktoré je možno plánovať.

1. Aktualizácia nových publikácií

Táto úloha je primárne zameraná na vytvorenie záznamov o novo pridaných publikáciách v databáze VVVS, u ktorých aspoň jeden z autorov je zamestnancom fakulty elektrotechnickej. Zároveň vytvára entity osôb, ktoré sú autorom resp. spoluautorom aspoň jednej publikácie a ukladá ich základné informácie. Táto úloha je najdôležitejšou úlohou pre tento projekt, pretože vďaka nej bude určovanie odbornosti vždy postavené na najnovších dátach. Predpokladaný interval tejto úlohy je raz denne.

2. Aktualizácia všetkých publikácií

Táto úloha kontroluje všetky publikácie vo VVVS pre fakultu elektrotechnickú. Úloha je tým pádom časovo náročná a keďže sa predpokladá, že záznamy vo VVVS sa spätne nebudú meniť, nebude sa ani táto úloha často spúšťať. Interval tejto úlohy predpokladám na rádovo niekoľko mesiacov až rok.

3. Aktualizácia zamestnancov FEL

Táto úloha slúži na aktualizáciu základných informácií o osobách uložených v internej databáze. Pomocou volania na získanie zamestnanca s UDB podľa jeho užívateľského mena skontroluje, či je osoba stále zamestnancom fakulty elektrotechnickej. Ak sa volaním nevrátia dáta o osobe, znamená to, že nie je zamestnancom FEL, čo sa internej DB prejaví príznakom "je z felu". Pokiaľ UDB vráti dáta o zamestnancovi, aktualizuje sa jeho meno, ktoré mohlo napr. nadobudnúť nový titul a nastaví sa príznak "je z felu" na hodnotu true (môže sa stať, že osoba bola v internej DB, ale nebola zamestnancom FEL).

Priebeh doťahovania nových publikácií

1. Pracoviská

Volanie:

Volanie LDAP s UDB, pomocou ktorého získam všetky pracoviská.

Poznámka:

Vďaka doťahovaniu dát s UDB mám zabezpečené, že sa mi vracajú iba pracoviská fakulty elektrotechnickej.

2. Osoby

Volanie:

VVVS API – GET /division/{code}/people

Poznámka:

Pomocou tohto volania získam zamestnancov katedry, ktorý sú autorom aspoň jednej publikácie. Pri týchto osobách mám istotu, že sú zamestnancami FEL, a preto sa do internej databázy uložia s príznakom "je z felu" nastaveným na hodnotu true.

3. Publikácie

Volanie:

VVVS API – GET /division/{code}/publications

VVVS API / GET /publications/{id}/rivs

Poznámka:

Prvým volaním získam všetky publikácie pre pracovisko (takto zabezpečím získavanie publikácií FEL). Pri aktualizácii nových publikácií sa toto volanie prevedie s filtrom na dátum vydania publikácie (dátum vydania je väčší ako najvyšší dátum vydania publikácie pre dané pracovisko).

Pomocou druhého volania získam bodové hodnotenie publikácie.

4. Autorstvo

Volanie:

VVVS API – GET /publications/{id}/authorship

VVVS API – GET /people/{personalNumber}

Podmienka:

Typ autora = 01 (autor)

3. Návrh

Poznámka:

Na základe osobného čísla sa pokúsi nájsť osobu v internej databáze. Pokiaľ sa osoba nenájde znamená to, že nie je zamestnancom FEL. V tomto prípade sa prevedie druhé volanie na získanie jeho základných údajov a osoba sa uloží s príznakom "je z felu" = false.

3.1.3.2 Komponenty

Komponenty (viď Obr. 3.1 Components) sa starajú o podpornú funkcionálnosť, ktorú je možné samostatne oddeliť a tým zabezpečiť možnosť samostatného testu. Komponenty sa delia na komponenty mapovania, výpočtu odbornosti, porovnávania a utilít.

- **Komponenty mapovania** - tieto komponenty slúžia na mapovanie jedného typu DTO na typ druhý.
- **Komponent počítania odbornosti** slúži na prepočet bodov publikácie medzi jednotlivých autorov, čím sa určí výška odbornosti zamestnanca.
- **Komponenty porovnávania** slúžia na zoraďovanie vyhľadávaných výsledkov.
- **Utility** slúžia na udržiavanie nastavení aplikácie a iné zjednodušenie činnosti aplikácie.

3.1.3.3 Expertise service

Expertise service (viď Obr. 3.1 Expertise service) má za úlohu zlúčiť implementáciu jednotlivých vrstiev a komponentov a zabezpečiť tak biznis logiku aplikácie vzhľadom na funkčné požiadavky. Zabezpečuje vyhľadávanie v internej databáze, mapovanie výsledkov podľa žiadaného formátu a ich stránkovanie.

3.1.4 Cache

Cache (viď Obr. 3.1 Cache) má za úlohu zvýšiť výkon aplikácie pomocou ukladania dát do medzipamäti. Pri používaní cache je nutné dávať veľký pozor na možné chyby, ktoré môžu vďaka cache nastať. Najväčším problémom cache je situácia, kedy sa dáta uložené v cache a reálne dáta nezhodujú, napríklad sa reálne dáta zmenia tesne po tom ako sa uložia do cache. Do cache sa budú ukladať výsledky vyhľadávania zamestnancov a pracovísk a dáta získané volaním integrácie z UDB. Keďže výsledky vyhľadávania sú založené na dátach z internej databázy, nikdy by nemal nastať problém kolízie dát v cache a v databáze, nakoľko DB sa aktualizuje pomocou plánovaných úloh, ktoré sa vykonávajú v dobe predpokladanej inaktivity užívateľov. U cache integrácie s UDB by tento problém mohol nastať, no z dôvodu, že detailné kontaktné údaje sa nemenia tak často, je tento problém zanedbateľný.

3.1.5 Rest vrstva

Rest vrstva (viď Obr. 3.1 Rest layer) vytvára aplikačné rozhranie Expertise API, ktoré poskytuje vyhľadávané dáta klientskej aplikácii. Nakoľko klientska aplikácia nijako neupravuje dáta, či už v internej DB alebo v externých dátových zdrojoch, aplikačné rozhranie ponúka iba volanie http metódou GET (13). Expertise API ponúka iba verejne prístupné dáta, a preto nevyžaduje žiadnu autorizáciu.

3. Návrh

3.1.5.1 REST zdroje

API poskytuje dáta vo formáte JSON (14), ktorý je najvhodnejší na prenos dát AJAX (15) volaním z klientskej aplikácie. URI k Expertise API sa skladá z URI k aplikácií nasadenej na nejakom serveri a prefixu „rest/v1“. Príklad URI pri nasadenej aplikácií na lokálnom serveri je:

localhost:8080/expertise_search/rest/v1

Parametre volaní

search – Hľadaný výraz, resp. hľadaná odbornosť

pageSize – Určuje maximálny počet výsledkov. Ak parameter nie je zadaný alebo je jeho hodnota nižšia ako 0, volanie vráti všetky výsledky.

page – Číslo strany. Parameter sa aktivuje iba v prípade, že je zadaný aj parameter pageSize a musí byť väčší ako 0. Ak parameter nie je zadaný a vo volaní je prítomný parameter pageSize volanie vráti prvú stranu výsledkov.

GET /persons

Vráti list zamestnancov so základnými údajmi, u ktorých bola nájdená odbornosť podľa zadaného hľadaného výrazu. Súčasťou odpovede je list publikácií, v ktorých bol nájdený hľadaný výraz.

Akceptované parametre:

search (povinný parameter), pageSize, page

Príklad obsahu:

```
{
  "peoples" : [
    {
      "username" : "example_username",
      "fullname" : "Ing. Engineer's name",
      "department" : {
        "name" : "Example department",
        "code" : "11111"
      },
      "publications" : [
        {
          "id" : 1234,
          "title" : " Publication title",
          "abstract" : "Abstract text",
          "releaseDate" : "yyyy-MM-dd",
          "keywords" : "joined by comma separator",
          "url" : "http:// publication.url.address"
        }
      ]
    }
  ]
}
```

GET /persons/{username}

Vráti list s jedným zamestnancom podľa jeho užívateľského mena, rozšírený o detailné kontaktné údaje. Súčasťou odpovede je list publikácií, v ktorých bol nájdený hľadaný výraz. Ak sa u zamestnanca nenájdu žiadne publikácie podľa hľadaného výrazu, volanie vráti prázdny list.

3. Návrh

Akceptované parametre:

search (povinný parameter)

Príklad obsahu:

```
{
  "peoples" : [
    {
      "username" : "example_username",
      "fullname" : "Ing. Engineer's name",
      "email" : "example_username@fel.cvut.cz",
      "phone" : "+420 123 456 789",
      "room" : "Praha 2, Karlovo náměstí 13, E-107",
      "department" : {
        "name" : "Example department",
        "code" : "11111"
      },
      "publications" : [
        {
          "id" : 1234,
          "title" : "Publication title",
          "abstract" : "Abstract text",
          "releaseDate" : "yyyy-MM-dd",
          "keywords" : "joined by comma separator",
          "url" : "http://publication.url.address"
        }
      ]
    }
  ]
}
```

GET /departments

Vráti list pracovišť, u ktorých zamestnancov bola nájdená odbornosť podľa zadaného hľadaného výrazu. Súčasťou odpovede je list publikácií a všetkých jej autorov, v ktorých bol nájdený hľadaný výraz.

Akceptované parametre:

search (povinný parameter), pageSize, page

Príklad obsahu:

```
{
  "departments" : [
    {
      "name" : "Example department",
      "code" : 11111,
      "address" : "Karlovo nám. 13 ,121 35 Praha 2",
      "phone" : "+420 123 465 789",
      "url" : " http://deparmtnet.url.address ",
      "head" : {
        "fullname" : "prof. Ing. Headmaster's name, CSc.",
        "username" : "headmaster_username"
      },
      "publications" : [
        {
          "id" : 1234,
```

3. Návrh

```
        "title" : " Publication title",
        "abstract" : "Abstract text",
        "releaseDate" : "yyyy-MM-dd",
        "keywords" : "joined by comma separator",
        "url" : "http://publication.url.address",
        "authors" : [
        {
            "fullname" : " Ing. Engineer's name",
            "username" : " example_username",
            "fromFel" : true
        }
    ]
}
}
```

GET /publications/{ids}

Vráti list publikácií podľa ich identifikátorov so všetkými údajmi o publikáciách, okrem ich počtu bodov, ktorý nie je zverejňovaný. V URI je možné zadať iba číselnú hodnotu identifikátora publikácie. Pre vyžiadanie viacerých publikácií naraz sú identifikátory oddelené čiarkou.

Príklad obsahu:

```
{
  "publications" : [
  {
    "id" : 1234,
    "title" : " Publication title",
    "abstract" : "Abstract text",
    "releaseDate" : "yyyy-MM-dd",
    "keywords" : "joined by comma separator",
    "url" : "http://publication.url.address",
    "authors" : [
    {
      "fullname" : " Ing. Engineer's name",
      "username" : " example_username",
      "fromFel" : true
    }
  ]
}
}
```

3.1.6 Klientska aplikácia

Klientska aplikácia (viď Obr. 3.1 Expertise client application) je navrhnutá ako single-page aplikácia (16). Všetok potrebný kód (HTML, JavaScript a CSS) je načítaný spolu s prvým načítaním stránky. Všetky ostatné prostriedky sú načítané dynamicky a vložené do stránky podľa potreby. Aj keď sa single-page aplikácia skladá z viacerých stránok, nikdy sa nepresmeruje na inú logickú stránku. Vďaka moderným webovým technológiám, ako napríklad HTML5 pushState() API (17), je možné poskytnúť užívateľom pocit navigácie na inú logickú

stránku, čo nám zabezpečí správnu funkčnosť aplikácie po obnovení stránky. Použitím frameworku AngularJS (18) zabezpečím možnosť rozdelenia klientskej aplikácie na aplikačnú a zobrazovaciu logiku.

3.2 Užívateľské rozhranie

Užívateľské rozhranie bolo vyvíjané v dvoch fázach. V prvej fáze vznikol prototyp užívateľského rozhrania (viď 3.2.1). V druhej fáze prebiehalo zjednotenie vytvorených prvkov s aktuálnymi webovými stránkami fakulty elektrotechnickej (viď 3.2.2).

3.2.1 Fáza 1

Fáza 1 zachytáva prvotný návrh prototypu, kde som sa zamerlal na vytvorenie moderného zobrazenia a zoskupenia vyhľadanych informácií.

Vyhľadávací formulár

Vyhľadávací formulár slúži na štart vyhľadávania na základe údajov zadaných vo vyhľadávacom poli a zvolením typu vyhľadávania, kde si užívateľ môže zvoliť vyhľadávanie zamestnancov, pracovísk alebo oboch. Kliknutím na ikonu lupy, alebo stlačením klávesy Enter vo vyhľadávacom poli sa spustí vyhľadávanie. Tento formulár je kľúčovou funkcionalitou celej aplikácie a z toho dôvodu sa nachádza na každej stránke aplikácie, aby užívateľ mohol zmeniť alebo začať nové vyhľadávanie kedykoľvek a kdekoľvek v aplikácii.

Výsledok vyhľadávania

Výsledky vyhľadávania sú rozdelené do dvoch stránok na základe zvoleného typu vyhľadávaného obsahu. V základnom nastavení (pokiaľ si užívateľ nezvolí inak) aplikácia vyhľadáva oba typy obsahu, kde prvým výsledkom sú zamestnanci a po nich nasledujú pracoviská. Na každej z týchto stránok sa zobrazí vyhľadávaný obsah súvisiaci s nájdenými publikáciami, u ktorého je možné dané publikácie zobraziť alebo skryť kliknutím na zobrazený počet nájdených publikácií. Užívateľ môže označiť publikácie, ktoré ho zaujali a následne prejsť na detail zvolených publikácií kliknutím na tlačidlo *Show detail of selected publications*, ktoré sa nachádza na začiatku a na konci vyhľadávaného obsahu.

Vyhľadání zamestnanci

U tohto obsahu sa užívateľovi zobrazí meno a fotografia zamestnanca spolu s jeho pracoviskom a preferovanou emailovou adresou. Užívateľ má možnosť po kliknutí na meno zamestnanca sa dostať na jeho profilovú stránku.

Vyhľadané pracoviská

U tohto obsahu sa užívateľovi zobrazí názov pracoviska, fakulta, pod ktorú patrí, vedúcu osobu pracoviska a kontakt na pracovisko, ktorý je umiestnený v samostatnom podfarbenom bloku. Po kliknutí na názov pracoviska bude užívateľ presmerovaný v novom okne na oficiálne stránky pracoviska. V tomto prípade sú pri vyhľadaných publikáciách zobrazené mena autorov, na ktorých profil je možné sa dostať po kliknutí na ich meno. Užívateľ taktiež môže prejsť na profilovú stránku vedúceho pracoviska pomocou kliknutia na jeho meno.

3. Návrh

Profil zamestnanca

Na stránke profilu zamestnanca sa zobrazia podrobnejšie informácie o zamestnancovi, a to konkrétne: fotografia zamestnanca, jeho meno, pracovisko a kontaktné údaje, ku ktorým patrí preferovaný email, telefónne číslo a číslo miestnosti, v ktorej má zamestnanec kanceláriu. Na profilovej stránke sa taktiež zobrazia všetky zamestnancove publikácie.

Detail publikácií

Na tejto stránke užívateľ môže vidieť všetkých autorov publikácie (po kliknutí na meno sa dostane na profil autora), jej názov, abstrakt, kľúčové slová, dátum vydania publikácie a odkaz na stránky, kde je daná publikácia publikovaná.

3.2.2 Fáza 2

V druhej fáze som sa zamerlal na zjednotenie celého užívateľského rozhrania spolu s grafickým návrhom webových stránok FEL a vytvorenie chýbajúcich prvkov prototypu, ako sú stránkovanie a error handling.

Okraje

Všetky používané okraje, či už okolo vyhľadávacieho poľa, alebo okolo vyhľadaných výsledkov boli zjednotené na rovnakú hrúbku a rovnaké zaoblenie hrán okrajov. Zaoblenie prvku je prevzaté z vyhľadávacieho poľa nachádzajúceho sa v pravom hornom rohu stránky FEL.

Farba písma

Na rozlíšenie dôležitosti textu sa v prototype používali rôzne farby písma. Táto rozdielnosť farieb bola odstránená, nakoľko design stránok FEL používa vždy iba čierne písmo a dôležitosť rozlišuje podľa veľkosti písma a podľa jeho hrúbky.

Vyhľadať podľa odbornosti

Search [Zobraziť vybrané články: 0](#) [Rozsireny filter](#)

Zamestnanci

 **prof. Ing. Janko Hrasko**
katedra trpaslíku a permoniku
omacka.pavel@fel.cvut.cz Publications: 2



Search by expertise

3D

Employees

 **Jan Čech**
Katedra počítačové grafiky a interakce
user0@fel.cvut.cz Publications: 1

Obr. 3.3: Zmeny okrajov a farby písma voči prototypu

3. Návrh

Odkazy

Farba a dekorácia textu odkazu pri prejení myšou je taktiež upravená podľa designu FEL, kde sa vždy používa modrá farba a podčiarknutý text.

Podfarbenie kontaktov pracovníka

Na podfarbenie kontaktov pracovníka bola použitá modrá farba používaná vo vrchnom panelu stránok FEL.

Pracoviska

Katedra počítačové grafiky a interakce Faculty of electrical engineering Head: prof. Ing. Jiří Žára, CSc.	Contact: Karlovo nám. 13 ,121 35 Praha 2 13139@fel.cvut.cz	Publications: 1
---	---	-----------------



Departments

Katedra počítačové grafiky a interakce Faculty of electrical engineering Head: prof. Ing. Jiří Žára, CSc.	Contact: Karlovo nám. 13 ,121 35 Praha 2 13139@fel.cvut.cz	Publications: 1
--	---	-----------------

Obr. 3.4: Zmena podfarbenia kontaktov a zobrazenie odkazu

Stránkovanie

Stránkovanie je navrhnuté ako číselné bloky, kde číslo aktuálnej stránky je podfarbené šedou farbou, ktorá sa používa v tabuľkách na stránkach FEL. Pri prejení myšou nad inou stránkou sa číslo danej stránky podfarbí slabšou šedou farbou, ktorá sa používa na rovnaký efekt prejdenia myšou u položiek bočného menu stránky FEL.

| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |

Obr. 3.5: Stránkovanie

Error handling

Jednou z funkčných požiadaviek je možnosť vyhľadávania pri zadaní minimálne dvoch znakov do vyhľadávacieho poľa. Pri snahe užívateľa vyhľadávať iba s jedným zadným znakom, aplikácia vypíše červenou farbou validačnú správu (viď Obr. 3.6) upozorňujúcu užívateľa o nutnosti zadania dvoch znakov, ktorá je zobrazená pod vyhľadávacím poľom.

Ak by nastala chyba komunikácie z Expertise API, napríklad z dôvodu nedostupného serveru, aplikácia upozorní užívateľa pomocou zobrazenia informácie o nedostupnosti aplikácie (viď Obr. 3.7).

Enter at least 2 characters for search

Obr. 3.6: Validačná chyba

The application is currently unavailable. Please try it later again.

Obr. 3.7: Chyba nedostupnej aplikácie

4 Implementácia

Táto kapitola popisuje implementáciu navrhutej aplikácie a jednotlivé frameworky, ktoré sú použité.

4.1 Expertise search

Pre túto časť aplikácie som zvolil programovací jazyk Java 7 kvôli jeho podpore vytvárania webových služieb a existujúceho množstva frameworkov. Pri implementácii som sa zamerlal najmä na štruktúru kódu a jeho rozdelenie do vrstiev, aby bolo možné čo najväčšiu časť kódu pokryť testami. Táto časť aplikácie je vytvorená ako maven projekt (19), ktorý sa stará o správu, riadenie a automatizáciu kompilácie aplikácie.

4.1.1 Použité frameworky

Pri implementácii serverovej časti aplikácie som použil niekoľko frameworkov na správu Java objektov (viď 4.1.1.1), vytvorenie REST služieb (viď 4.1.1.2) a spojenie z databázou (viď 4.1.1.3).

4.1.1.1 Spring Framework 4 (20)

Veľmi populárny open source framework, ktorý tvorí robustný základ aplikácií postavených na Java EE platforme. Hlavným bodom Spring Frameworku je jeho IoC (inversion of control) kontajner, ktorý spravuje javovské objekty pomocou referencie. Tieto objekty sa taktiež nazývajú beans, ktoré je možné konfigurovať pomocou XML súboru. Spring sa stará o vytváranie, inicializáciu a životný cyklus týchto objektov.

4.1.1.2 Jersey 2 (21)

Jersey framework je open source framework pre vývoj REST webových služieb. Ponúka vlastné API rozširujúce sadu nástrojov JAX-RS s ďalšími funkciami a nástrojmi na zjednodušenie REST služby. Tento framework je používaný v Rest vrstve (viď 4.1.6).

4.1.1.3 Hibernate 4 (22)

Hibernate poskytuje spôsob, pomocou ktorého je možné udržať dáta perzistentné. Pomocou anotácií JPA (Java persistence API) je možné mapovať objekty na entity v relačnej databáze. Objekty uložené v databáze je možné dotiahnuť pomocou jazyka HQL (Hibernate Query Language), ktorý je odvodený od jazyka SQL a je mu teda veľmi podobný. Tento framework je používaný vo vrstve dátového prístupu (viď 4.1.3).

4.1.2 Dátová vrstva (DTO)

Dátová vrstva sa stará o prenos dát medzi jednotlivými logickými vrstvami. Obsahuje teda tzv. DTO objekty (Data transfer object). Súčasťou tejto vrstvy je aj objekt udržiavajúci kritéria pre stránkovanie. Všetky DTO objekty sú implementované v balíčku „*cz.cvut.fel.ku-charo2.dto*“ a sú rozdelené na základe ich využívania:

- **Entity DTO** sú objekty, ktoré sa mapujú na internú databázu. Všetky tieto objekty sú nazvané podľa tabuľky v DB s postfixom „Entity“.

4. Implementácia

- **Rest DTO** sú objekty, pomocou ktorých sa vytvárajú odpovede pri volaní REST služby. Koreňové objekty sú nazvané prostredníctvom názvu volanej metódy a postfixu „RestResponse“. Súčasťou týchto DTO objektov je aj objekt, na ktorý sa pri volaní REST služby mapujú zadané url parametre.
- **Integration DTO** sú objekty, ktoré prenášajú dáta získané z integračných služieb.
 - **VVVS DTO** – objekty integrácie s VVVS. Ich názov sa skladá z názvu volanej entity a prefixu „Vvvs“.
 - **UDB DTO** – objekty integrácie s UDB. Ich názov sa skladá z názvu volanej entity a prefixu „Udb“

4.1.3 Vrstva dátového prístupu (DAO)

Pre každú tabuľku internej databázy a jej reprezentujúcu DTO entitu je vytvorený jeden DAO objekt. Pre zamedzenie duplikovania kódu je vytvorená abstraktná generická trieda, ktorá implementuje základnú funkcionálnosť každého DAO. DAO objekty sú implementované v balíčku „*cz.cvut.fel.kucharo2.backend.dao*“.

4.1.4 Vrstva komponentov (Component)

Implementované komponenty sa nachádzajú v balíčku „*cz.cvut.fel.kucharo2.backend.component*“.

- **Komponenty mapovania**
 - **FromEntityMapper** slúži na mapovanie 1:1 z Entity DTO na Rest DTO.
 - **ToRestResponseMapper** slúži na kompletné mapovanie databázových výsledkov (Entity DTO) na výsledný formát poskytovaný aplikačným rozhraním (koreňové Rest DTO).
 - **ToEntityMapper** slúži na 1:1 mapovanie z integračných DTO na Entity DTO.
- **Komponent počítania odbornosti**
 - **PointsCalculationComponent** slúži na prepočet bodov publikácie medzi jednotlivých autorov.
- **Komponenty porovnávania**
 - **DepartmentPointsDescComparator** slúži na zostupné zoradovanie pracovísk podľa výšky odbornosti.
 - **PersonPointsDescComparator** slúži na zostupné zoradovanie zamestnancov podľa výšky odbornosti.
 - **PublicationPointsDescComparator** slúži na zostupné zoradovanie publikácií podľa získaného počtu bodov.
- **Utility**
 - **PaginationUtil** slúži na aplikáciu stránkovania. Vrátí časť listu na základe stránkovacích kritérií.
 - **PropertiesUtil** udržuje nastavenia aplikácie.

4. Implementácia

- **UdbPropertiesUtil** udržuje nastavenia integrácie s UDB.
- **VvvsApiProperitesUtil** udržuje nastavenia integrácie s VVVS.

4.1.5 Expertise service

Servisná vrstva je implementovaná v balíčku „*cz.cvut.fel.kucharo2.backend.service*“. Táto vrstva obsahuje dve servisné triedy:

- **ExpertiseService** poskytuje rozhranie, ktoré využíva rest vrstva (viď 4.1.6). V tejto službe je implementovaná všetka biznis logika spojená s funkcionalitou vyhľadávania podľa odbornosti.
- **SynchronizationService** poskytuje rozhranie, ktoré využívajú plánované úlohy (viď 4.1.7). V tejto službe je implementovaná biznis logika na synchronizáciu internej databázy.

4.1.6 Rest vrstva

Rest vrstva implementuje aplikačné rozhranie a nastavuje mu funkčné filtre, pomocou ktorých je možné upravovať správanie aplikácie. Tieto filtre sa dajú rozdeliť do troch typov:

- **JSON filtre** umožňujú nastavenie mapovania Rest DTO na formát JSON.
 - **JacksonContextResolver** – nastavuje formátovanie JSON objektu (používanie odrážok a nových riadkov) a zabraňuje zápisu null hodnôt.
 - **JacksonFeature** – zapína používanie Jackson JSON poskytovateľa, vďaka čomu je aj možné použiť vyššie spomenutý JacksonContextResolver
- **Error filtre** umožňujú odchyťovanie chýb vzniknutých v aplikácii a ich spracovanie do zrozumiteľnej JSON odpovede, ktorú následne môže klientska aplikácia spracovať. Tento JSON objekt bude poskytovať http kód chyby, dôvod jej vzniku (exception message) a prípadne vývojársku informáciu, prečo chyba nastala.
 - **NotFoundExceptionHandler** – odchyťáva chyby vzniknuté nesprávnym volaním, napr. chybou v URI
 - **ServiceExceptionHandler** – odchyťáva chyby vzniknuté priamo v aplikácií, ktoré mohli vybublať z ktorejkoľvek vrstvy.
- **Ostatné filtre**
 - **LoggingResponseFilter** – nastavuje logovanie všetkých volaní na Expertise API a jeho následných odpovedí (JSON objektu).
 - **ResponseCorsFilter** – zabezpečuje nastavenie Access-Control-Allow hlavičky http odpovede, aby bolo možné použiť cross-domain volanie na Expertise API pomocou AJAXU.

4.1.7 Plánované úlohy

Spring Framework (20) poskytuje implementáciu spúšťania asynchrónnych úloh pomocou TaskScheduler rozhrania. Časové intervaly spúšťania úloh sa nastavujú pomocou cron výrazu (23), ktorý je možné upravovať v nastaveniach aplikácie.

4.1.8 Nastavenie webovej aplikácie

Implementácia poskytuje možnosť nastavenia aplikácie pomocou súboru *expertise-search.properties*. V tomto súbore je možné nastaviť časové intervaly spúšťania plánovaných úloh, zapnutie alebo vypnutie rovnocenného prepočítavania bodov publikácie a URI adresy pre integráciu s VVVS a UDB.

- **points.calculation.participation.divide** – zapína(true) alebo vypína(false) funkcionality prepočítavania bodov publikácie medzi autorov podľa ich účasti.
- **scheduled.allPublicationUpdate.cron** – cron hodnota nastavujúca interval spúšťania aktualizácie všetkých publikácií.
- **scheduled.latestPublicationUpdate.cron** – cron hodnota nastavujúca interval spúšťania aktualizácie najnovších, resp. posledných publikácií.
- **scheduled.updateEmployeesFromFel.cron** – cron hodnota nastavujúca interval spúšťania aktualizácie zamestnancov FEL.
- **vvvsApi.base.uri** – základná URI pre spojenie s VVVS API
- **vvvsApi.division.peoples** – VVVS URI na získanie ľudí podľa katedry
- **vvvsApi.division.publications** – VVVS URI na získanie publikácií podľa katedry
- **vvvsApi.publication.rivs** – VVVS URI na získanie RIV bodov pre publikáciu
- **vvvsApi.publication.authorship** – VVVS URI na získanie autorov publikácie
- **vvvsApi.person.byPersonaNumber** – VVVS URI na získanie osoby podľa jej osobného čísla
- **publication.releaseDate.format** – formát dátumu, ktorý používa VVVS API

Nasledujúce nastavenia sa môžu v produkčnej implementácii líšiť. Je to z dôvodu nastavení aplikácie v stave použitom pri testovaní, kde bolo nutné sa pripojiť k mockovým dátam UDB (viď 5.3.2).

- **udbApi.base.uri** – základná URI pre spojenie s UDB
- **udbApi.departments** – UDB URI na získanie všetkých pracovísk
- **udbApi.department.byCode** – UDB URI na získanie pracovísk podľa ich kódu
- **udbApi.employee.byUsername** – UDB URI na získanie zamestnanca podľa jeho užívateľského mena

4.1.9 Cache

Cache je implementovaná pomocou najpoužívanejšej open source distribuovanej Ehcache (24). Použitie ehcache umožňuje nastavovanie jednotlivých cache pomocou XML súboru *ehcache.xml*. Nastavenie spočíva v určení maximálneho počtu uložených elementov a maximálnej dobe uloženia elementu v cache.

4. Implementácia

- **personResponseCache** – udržiava výsledky vyhľadávania zamestnancov podľa zadaného vyhľadávacieho výrazu (10 min.).
- **departmentResponseCache** – udržiava výsledky vyhľadávania pracovísk podľa vyhľadávacieho výrazu (10 min.).
- **udbEmployeeCache** – udržiava výsledky volania detailných informácií o zamestnancovi z UDB (5 min.).
- **udbDepartmentCache** – udržiava výsledky volania detailných informácií o pracoviskách z UDB (5 min.).

4.2 Klientska aplikácia

Na vytvorenie single-page aplikácie som zvolil JavaScriptový framework AngularJS (18), vďaka ktorému je možné oddeliť zobrazovaciu logiku od aplikačnej logiky použitím návrhového vzoru MVC (25). Aplikačnú logiku je taktiež možné rozdeliť na viacero modulov, vďaka ktorým je možné udržať štruktúru klientskej aplikácie.

4.2.1 Zobrazovacia logika

Na zobrazovanie dát sa používa HTML obohatený o funkcionality AngularJS, ktorá pomocou definovaných atribútov (tzv. directive) umožňuje pracovať s elementami DOM. Spolu s data-bindingom tak vytvára templatovací jazyk na vytváranie finálneho HTML, ktorý sa následne zobrazí v aplikácii. Každá stránka mojej aplikácie má vlastnú template, ktorá sa zobrazuje na predom definovanej URL aplikácie. Súčasťou tejto URL sú predávané dáta, pomocou ktorých je zobrazená stránka. Súčasťou každej URL sú hodnoty **searchFilter** (zvolená hodnota typu vyhľadávania), **searchValue** (hodnota vyhľadávania). Ďalšími hodnotami sú **username** (užívateľské meno zamestnanca) a **ids** (identifikačné čísla publikácií).

- **searchField.html** je template, obsahujúca vyhľadávacie pole a je vnorená v každej ďalšej template, nakoľko má byť súčasťou každej stránky aplikácie.
- **search.html** je template, ktorá zobrazuje výsledky vyhľadávania.
URL zobrazenia
/result/:searchFilter/:searchValue
- **personDetail.html** je template, ktorá zobrazuje profilovú stránku zamestnanca.
URL zobrazenia
/person/:searchFilter/:searchValue/:username
- **publicationDetail.html** je template, ktorá zobrazuje podrobné informácie o zvolených publikáciách.
URL zobrazenia
/publications/:searchFilter/:searchValue/:ids

4.2.2 Aplikačná logika

Hlavnou úlohou aplikačnej logiky je komunikácia s webovou službou, smerovanie aplikácie medzi jej jednotlivými stránkami, úprava zobrazovaných dát a ďalšie funkcionality ako je napríklad stránkovanie. Dáta zobrazované zobrazovacou logikou sa ukladajú do modelu, ktorý je v AngularJS reprezentovaný premennou `$scope`. Aplikácia je implementovaná z týchto modulov:

- **expertiseSearch.services** – tento modul je zodpovedný za doťahovanie dát z webovej služby pomocou AJAX volaní. Definuje prevolávané URI na získavanie dát a prípadne ukladá informáciu o neúspešnom volaní.
- **expertiseSearch.pagination** – tento modul definuje objekty používané pri stránkovaní.
- **expertiseSearch.controller** – tento modul implementuje controller, ktorý sa stará o naplňovanie modelu dátami na zobrazenie a vytvára funkcie používané zobrazovacou logikou ako je prechod na inú stránku aplikácie alebo zmena stránky vyhľadanych výsledkov.
- **expertiseSearch.filters** – tento modul poskytuje implementáciu filtrov, ktoré upravujú zadané dáta, ako je napríklad filter na zobrazenie dátumu v správnom formáte.

5 Testovanie

Testovanie je veľmi dôležitou súčasťou pri tvorbe aplikácií, ktoré dopomáha k identifikácii chýb systému, či už počas implementácie, alebo pri následných úpravách kódu, akou môže byť napríklad refactoring. V rámci testovania som sa zameril najmä na unit testing (viď 5.1), ktorý slúži na automatické testovanie a overovanie funkčnosti implementácie webovej služby. Pre účely testovania klientskej časti aplikácie bolo nutné vytvoriť si vlastné testovacie dátové sady (viď 5.3). V rámci tohto projektu prebehlo iba neformálne testovanie z užívateľmi aplikácie. Počas tohto testovania bola webová služba nasadená na lokálnom GlassFish serveri.

5.1 Unit testing

Vďaka navrhutej štruktúre kódu a správnym rozdelením funkcionality do jednotlivých vrstiev bolo možné pokryť väčštinovú časť webovej služby automatickými testami. Na tieto účely slúži JUnit framework (viď 5.1.1), ktorý je možné obohatiť o rozširujúci Mockito framework (viď 5.1.2). Webová služba je dokopy pokrytá sto dvadsiatimi deviatimi testami.

5.1.1 JUnit (26)

JUnit je jednoduchý a najpoužívanejší framework, ktorý slúži na tvorbu opakovateľných testov. Tieto testy je možné spustiť ručne, no sú taktiež automaticky spúšťané pri každej kompilácii aplikácie.

5.1.2 Mockito (27)

Vďaka návrhu aplikácie a rozdelenia funkcionality do vrstiev môžem u vyšších vrstiev predpokladať funkčnosť vrstiev nižších a sfaľšovať (mockovať) dáta, ktoré majú poskytovať. K vytvoreniu takýchto testov slúži spomínaný framework Mockito.

5.1.3 Spring testing

Pri testovaní DAO vrstvy, ktorá komunikuje s internou databázou som využil testovanie pomocou Spring frameworku. Ten poskytuje možnosť rollbacku transakcie, čo znamená, že stav databázy sa po vykonaní testu nezmení. Všetky úpravy dát v internej databáze, či už vznik nových, alebo úprava stávajúcich sa po vykonaní testu zrušia.

5.2 FindBugs

FindBugs je program, ktorý používa statickú analýzu na vyhľadanie potencionálnych chýb v javovskom kóde. Na indentifikáciu chýb tento open source program operuje nad skompilovaným javovským kódom. Report z analýzy je možné nahliadnúť na CD.

5.3 Testovacie dátové sady

Za dobu tvorby tejto práce som nedostal prístupy ani k jednému z potrebných externých dátových zdrojov. Bohužiaľ, niekde zlyhala komunikácia a na odoslané maily som nedostal žiadne reakcie. Nakoľko som chcel udržať štruktúru aplikácie, bol som nútený vytvoriť si vlastné externé zdroje a aplikačné rozhrania k nim. Keďže sa o komunikáciu z externými dátovými zdrojmi stará integračná vrstva, pri implementácii s reálnymi zdrojmi stačí iba zmena implementácie integračných rozhraní a aplikácia bude plne funkčná.

5.3.1 VVVS Mock API

Pri tvorbe VVVS Mock API som vychádzal z dokumentácie reálneho API, čím som smeroval k cieľu vytvoriť čo najviac kompatibilné aplikačné rozhranie. Vďaka takémuto návrhu som sa snažil minimalizovať nutnosť zmien, ktoré by súviseli z implementáciou reálnej integrácie. Dáta, ktoré mock API ponúka som získal vďaka aplikácii RIV (1). Táto aplikácia ponúka možnosť exportu vyhľadaných výsledkov do XLS, z ktorých som časť spracoval (50 publikácií) a vytvoril si z nich vlastné dáta. Počty bodov pre publikácie boli vytvorené podľa jednoduchého algoritmu, kde sa v podstate dá povedať že ich hodnoty sú náhodné. Účasť práce autorov na publikáciách je rozdelená rovným dielom. Export použitý na vytvorenie testovacích dát sa nachádza v resource aplikácie. Tieto dáta sú importované pomocou služby ImportService.

5.3.2 UDB Mock API

Na poskytovanie dát z UDB som si taktiež vytvoril aplikačné rozhranie, nakoľko ako bolo spomenuté k UDB neexistuje žiadna dokumentácia. Dáta poskytované z UDB Mock API boli ručne vytvorené podľa získaných autorov z exportu dát aplikácie pre Radu pre výskum, vývoj a inovácie. Títo autori boli rozdelení do troch pracovísk, ktorých informácie boli získané pomocou webovej aplikácie UDB.

5.4 Testovanie synchronizačných úloh

Keďže synchronizačné úlohy sa spúšťajú vo vopred definovaných intervaloch, vytvoril som čisto pre testovacie účely volanie na aplikačné rozhranie, pomocou ktorého je možné vyvolať spustenie jednotlivých synchronizačných úloh. Pri tomto ručnom spúšťaní synchronizačných úloh je nutné dávať pozor na vytvorené cache (viď 4.1.9), vďaka ktorým sa môže zmena prejaviť s časovým meškaním.

- **GET /syncAll** – spustí synchronizáciu všetkých publikácií.
- **GET /syncNew** – spustí synchronizáciu nových publikácií.
- **GET /syncEmployee** – spustí synchronizáciu zamestnancov FEL.

5.5 Testovanie klientskej aplikácie

Testovanie klientskej aplikácie je obmedzené kvôli menšiemu množstvu testovacích dát. Je teda vysoko pravdepodobné, že pri viacerých náhodne zadaných výrazoch nenájde žiadne výsledky. Je teda nutné použiť primitívne kľúčové slová (napríklad „3D“, „routing“, „framework“ a pod.) prípadne na otestovanie stránkovania použiť dvojslabičný výraz, ktorý je pravdepodobne súčasťou viacerých publikácií (napr. „is“).

5.6 Výsledok testovania

V rámci testovania aplikácie som plne pokryl testovanie webovej aplikácie, až na integračnú vrstvu, ktorú nebolo možné otestovať voči reálnym dátam. Klientska časť aplikácie bola neformálne otestovaná niekoľkými užívateľmi, ktorý sa zaujímali o funkcionálnosť aplikácie.

6 Záver

6.1 Splnenie cieľov

Úlohou tejto bakalárskej práce bolo navrhnuť a implementovať intuitívnu a ľahko ovládateľnú aplikáciu na vyhľadávanie zamestnancov a pracovísk fakulty elektrotechnickej na základe ich odbornosti.

Analýzou súčasného stavu aplikáciu v rámci ČVUT som ukázal, že neexistuje žiadna podobná aplikácia s požadovanou funkcionalitou. Každá z analyzovaných aplikácií chýbala väčšina základných požiadaviek. Žiadna z aplikácií, ktoré umožňovali vyhľadávanie publikácií nedokázala priamo zobraziť kontaktné údaje autorov. To užívateľa donúti k použitiu inej aplikácie na vyhľadanie zamestnancov, pomocou ktorej dokáže zistiť dané kontaktné údaje. Jedinou aplikáciou, ktorá umožňuje zobrazovanie kontaktných údajov spolu z publikačnou činnosťou zamestnanca je novo vznikajúca aplikácia fakulty informačných technológií, ktorá ale nepodporuje možnosť vyhľadávania podľa publikačnej činnosti.

Návrhom a následnou implementáciou aplikácia splnila všetky požiadavky nutné na správne určovanie odbornosti u zamestnancov a pracovísk. Vďaka rozdeleniu aplikácie na webovú službu a klientsku aplikáciu som dosiahol možnosť jednoduchého nasadenia na akúkoľvek webovú stránku. Ak je webová služba nasadená na serveri, stačí do webovej stránky načítať JS súbor ovládajúci aplikáciu a vložiť doslova pár riadkov HTML kódu na miesto v stránke, kde sa má aplikácia zobraziť. Design aplikácie (CSS súbor) je možné kedykoľvek zmeniť bez straty akejkoľvek funkcionality.

Na základe kladnej reakcie iných študentov na opísanú funkcionalitu aplikácie si dovoľím povedať, že táto aplikácia by mohla mať veľký úspech. Tento predpoklad sa mi potvrdil aj počas neformálneho testovania s užívateľmi, ktorí ohodnotili aplikáciu ako veľmi nenáročnú na používanie a dostatočne intuitívnu.

6.2 Možné vylepšenia

Aktuálne navrhnutá aplikácia využíva dáta na určovanie odbornosti iba z jedného dátového zdroja. Ak by sa aplikácia napojila na ďalšie dátové zdroje, z ktorých by dokázala počítať odbornosť, mohla by sa zvýšiť presnosť výpočtu odbornosti a vďaka tomu aj poskytovanie presnejších výsledkov vyhľadávania. V aplikácií by taktiež mohla existovať prihlasovacia časť pre zamestnancov, vďaka ktorej by mal každý zamestnanec možnosť editácie svojho profilu. Mohol by si napríklad ručne pridať svoju odbornosť pomocou kľúčových slov (podobne ako funguje LinkedIn (28)), alebo by taktiež mohol mať možnosť označiť si svoje publikácie, s ktorými je stotožnený a podľa ktorých sa chce reprezentovať. Nie vždy musí byť každá publikácia reprezentatívna. Vedúci pracovísk, prípadne poverené osoby by mali rovnakú možnosť úpravy profilu svojho pracoviska.

6.3 Budúcnosť aplikácie

Ak by nápad a samotný návrh aplikácie mal rovnaký úspech u vedeckých pracovníkov a zamestnancov fakulty elektrotechnickej ako mal u študentov, ktorým táto práca bola prezentovaná, mohla by sa táto aplikácia stať súčasťou fakultných webových stránok, prípadne by sa mohla stať základom pre výslednú aplikáciu, ktorá by sa nasadzovala do produkcie.

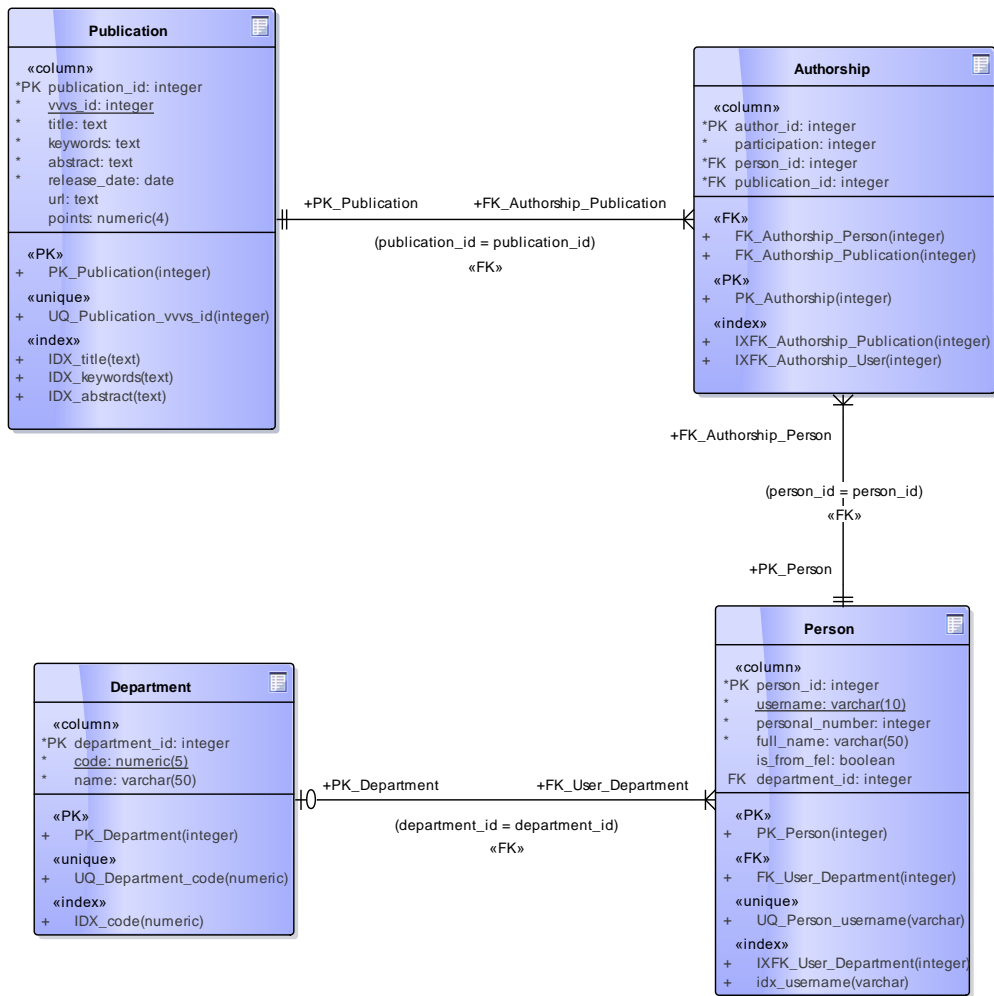
7 Referencie

1. **Rada pro výzkum, vývoj a inovace.** Rejstřík informací o výsledcích a Informace o předávání údajů | Výzkum a vývoj v ČR. [Online] [Datum: 4. Január 2016]
2. **InfoScience Praha s.r.o.** Vyhledání výsledků - data IS VaVal. [Online] [Datum: 7. Január 2016]
3. **FEL, ČVUT.** ČVUT - FEL - K13136 - Publikace - 2015 - ČVUT - Fakulta elektrotechnická. [Online] [Datum: 7. Január 2016]
<http://www.fel.cvut.cz/cz/13136/publication/2015html>.
4. **ČVUT, VIC.** Vstup do aplikací VVVS a EZOP. [Online] [Datum: 25. December 2015]
<http://www.vvvs.cvut.cz/>.
5. **VIC, ČVUT.** Jednoduché vyhledávání | VVVS - ČVUT v Praze. [Online] [Datum: 7. Január 2016] <https://www.vvvs.cvut.cz/search/>.
6. **ČVUT, VIC.** VVVS - Hlavní stránka aplikace VVVS. [Online] [Datum: 7. Január 2016]
https://www.vvvs.cvut.cz/aplikace/menu_ltd.php.
7. **ČVUT, FIT.** Akademičtí pracovníci - IS FIT ČVUT. [Online] [Datum: 7. Január 2016]
<https://is.fit.cvut.cz/web/webfit/kontakty/zamestnanci#/>.
8. **Stadler, Jan.** Wiki - VVVSapi - Projekt KOSapi. [Online] [Datum: 26. December 2015] <https://kosapi.fit.cvut.cz/projects/vvvsapi/wiki>.
9. **Wikipedie.** LDAP. [Online] [Datum: 26. December 2015]
<https://cs.wikipedia.org/wiki/LDAP>.
10. **Jirůtka, Jakub.** Usermap API (Main.usermap-api). [Online] [Datum: 28. December 2015] <https://rozvoj.fit.cvut.cz/Main/usermap-api>.
11. **Wikipedie.** Index (databáze). [Online] [Datum: 28. December 2015]
[https://cs.wikipedia.org/wiki/Index_\(datab%C3%A1ze\)](https://cs.wikipedia.org/wiki/Index_(datab%C3%A1ze)).
12. **The PostgreSQL Global Development Group.** PostgreSQL: Documentation: 9.1. [Online] [Datum: 28. December 2015]
<http://www.postgresql.org/docs/9.1/static/index.html>.
13. **Wikipedie.** GET. [Online] [Datum: 29. December 2015]
<https://cs.wikipedia.org/wiki/GET>.
14. **Wikipedie.** JavaScript Object Notation. [Online] [Datum: 29. December 2015]
https://cs.wikipedia.org/wiki/JavaScript_Object_Notation.
15. **Wikipedie.** AJAX. [Online] [Datum: 29. December 2015]
<https://cs.wikipedia.org/wiki/AJAX>.
16. **Wikipedie.** Single-page application. [Online] [Datum: 29. December 2015]
https://en.wikipedia.org/wiki/Single-page_application.
17. **Wikipedie.** Comparison of layout engines (HTML5). [Online] [Datum: 29. December 2015]
[https://en.wikipedia.org/wiki/Comparison_of_layout_engines_\(HTML5\)#APIs](https://en.wikipedia.org/wiki/Comparison_of_layout_engines_(HTML5)#APIs).
18. **Google.** AngularJS — Superheroic JavaScript MVW Framework. [Online] [Datum: 29. December 2015] <https://angularjs.org/>.
19. **The Apache Software Foundation.** Apache Maven Project. [Online] [Datum: 29. December 2015] <https://maven.apache.org/>.
20. **Pivotal Software, Inc.** Spring Framework. [Online] [Datum: 29. December 2015]
<http://projects.spring.io/spring-framework/>.
21. **Oracle Corporation.** Jersey. [Online] [Datum: 29. December 2015]
<https://jersey.java.net/>.

7. Referencie

22. **Redhat.** Hibernate. [Online] [Datum: 29. December 2015] <http://hibernate.org/>.
23. **Terracotta, Inc.** Quartz Scheduler | Documentation. [Online] [Datum: 28. December 2015] <http://www.quartz-scheduler.org/documentation/quartz-1.x/tutorials/crontrigger>.
24. **Terracotta, Inc.** Ehcache. [Online] [Datum: 30. December 2015] <http://www.ehcache.org/>.
25. **Wikipedie.** Model-view-controller. [Online] [Datum: 29. December 2015] <https://cs.wikipedia.org/wiki/Model-view-controller>.
26. **JUnit.** JUnit - About. [Online] [Datum: 3. Január 2016] <http://junit.org/index.html>.
27. **Wikipedia.** Mockito. [Online] [Datum: 3. Január 2016] <https://en.wikipedia.org/wiki/Mockito>.
28. **LinkedIn.** Největší profesní síť na světě | LinkedIn. [Online] [Datum: 7. Janár 2016] <https://cz.linkedin.com/>.

A. Expertise search databázový model



B. Obsah priloženého CD

CD obsahuje zdrojové kódy aplikácie Expertise search, klientskej aplikácie a testovacích aplikácií VVVS Mock a UDB Mock. Priložené sú taktiež skompilované kódy aplikácií, inicializačné skripty na vytvorenie internej a externých databáz a report analýzy z FindBugs.

Štruktúra priloženého CD:

- source
 - client application
 - expertiseSearch – v tomto priečinku sa nachádzajú zdrojové súbory ku klientskej aplikácii
 - fel_data – tento priečinok obsahuje skopírované súbory na zobrazenie stránok fakulty elektrotechnickej
 - index.html – zobrazovaná stránka, v ktorej je nasadená klientska aplikácia
 - expertise_search – tento priečinok obsahuje zdrojové kódy serverovej časti aplikácie Expertise search
 - VVVS mock – tent priečinok obsahuje zdrojové kódy k testovaciemu rozhraniu pre VVVS dáta
 - src/main/resources/dataToImport.xlsx – exportované dáta z RIV, použité na vytvorenie testovacích dát
 - UDB mock – tento priečinok obsahuje zdrojové kódy k testovaciemu rozhraniu pre UDB dáta
- compiled – tento priečinok obsahuje skompilované kódy aplikácií
- db scripts – tento priečinok obsahuje databázové skripty na vytvorenie jednotlivých databáz a na ich naplnenie testovacími dátami
- FindBugs – obsahuje HTML súbor s výsledkom analýzy aplikácie Expertise search
- kucharo2_bak.pdf – elektronická verzia práce
- installation_guide.txt – inštalačný návod na rozbehovanie aplikácie
- frameworks&licences.txt – zoznam použitých frameworkov s ich verziami a licenciami