

Sem vložte zadání Vaší práce.

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



Bakalářská práce

Aplikace pro vyhodnocení výsledků multikriteriálního vyhledávání na internetu

Miroslav Novotný

Vedoucí práce: Mgr. Monika Součková

23. dubna 2015

Poděkování

Rád bych poděkoval jak své vedoucí Monice Součkové tak zadavateli Vítu Skálovi za velmi přátelskou a klidnou atmosféru během celé práce. Dále své rodině, přítelkyni a přátelům za neoblomnou podporu.

Prohlášení

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

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

V Praze dne 23. dubna 2015

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2015 Miroslav Novotný. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Novotný, Miroslav. *Aplikace pro vyhodnocení výsledků multikriteriálního vyhledávání na internetu*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstrakt

Tato bakalářská práce se zabývá vývojem aplikace na vyhledávání, hodnocení a následnou vizualizaci vazeb mezi různými subjekty. Žádná podobná aplikace není volně dostupná. Vývoj proběhl v jazyce Java za použití Google Custom Search API, pro získávání výsledků hledání, a knihovny JUNG, pro vizualizaci síťových grafů. Aplikace byla dovedena do zdárného konce a funguje správně. Důležité vylepšení aplikace by spočívalo v lepším vyhledávacím algoritmu, který by vracel méně nerelevantních článků.

Klíčová slova vyhledávání po internetu, vizualizace vazeb, sociální vazby, Google API

Abstract

This Bachelor thesis deals with the development of an application for searching, evaluating and visualization of relations between different subjects. No such application is publicly available. The development was done in Java programming language using Google Custom Search API as a result retrieving API and JUNG library for graph visualization. The application was successfully finished and functions correctly. Important improvement lies in better search algorithm which would return less irrelevant articles.

Keywords internet search, relation visualization, social relations, Google API

Obsah

Úvod	1
Cíl práce	1
Struktura práce	1
Vývojové prostředí	3
1 Analýza	5
1.1 Požadavky	5
1.2 Forma aplikace	7
1.3 Jazyk implementace	8
2 Návrh	11
2.1 Vyhledávač	11
2.2 Vizualizace vazeb	15
3 Implementace	19
3.1 Vyhledávání	19
3.2 GUI	22
3.3 Jazyková lokalizace	24
4 Testování	27
4.1 Test použitelnosti	27
4.2 Výsledky testování	28
Závěr	29
Zhodnocení analýzy a návrhu	29
Zhodnocení implementace	29
Náměty pro další vývoj	29
Celkové zhodnocení	30
A Seznam použitých zkratek	31

B	Obsah přiloženého CD	33
C	Test použitelnosti: report zadavatele	35
	C.1 Sociální pavouk - výsledky testování	35
D	Uživatelská příručka	37
	D.1 Úvod	37
	D.2 Nutné nainstalovat	37
	D.3 Okno vyhledávání	37
	D.4 Okno vazeb	41
	D.5 Vypnutí programu	43
	Literatura	45

Seznam obrázků

1.1	Případ použití - funkční požadavky	6
1.2	Případ použití - rozpad případu použití: Práce s výsledkem vyhledávání	6
2.1	Výsledky přes CSE	14
2.2	Výsledky přes vyhledávač	15
2.3	Sociální pavouk	16
3.1	Ukázka: plocha hledání	23
3.2	Ukázka: plocha vazeb	23
D.1	Uživatelská příručka: okno hledání	38
D.2	Uživatelská příručka: okno detailu článku	40
D.3	Uživatelská příručka: okno vazeb	41

Seznam tabulek

2.1	Vrácené odpovědi od Google.com vs Google CSE	14
3.1	Nejčastější CSE status kódy a jejich význam	20

Úvod

Cíl práce

Tato bakalářská práce se zabývá vývojem aplikace pro vyhodnocení výsledků multikriteriálního vyhledávání na internetu. Práce byla vypracována ve spolupráci s Karlovou Univerzitou v rámci projektu Jak měřit sociální kapitál na úrovni obcí. Zadavatelem je pan PhDr. Ing. Vít Skála z fakulty sociálních věd. Výstupem práce je hotová aplikace s názvem Sociální pavouk. Název aplikace reflektuje skutečnost vyhledávání osob a fakt, že pavučina má podobný tvar jako graf sociálních struktur. V aplikaci je možné vyhledávat vztah jedna až N subjektů, pracovat s nalezenými články a nakonec zobrazit pavučinu, která znázorňuje spojitost skupin a sílu vztahů mezi účastníky se subjekty. Uživatelská příručka aplikace je v příloze D.

Z hledání na internetu vyplynulo, že aplikace splňující podobné požadavky či cíl není volně k dispozici.

Struktura práce

Práce se bude dělit do několika bloků, kde každý blok se zabývá jedním z aspektů vývoje aplikace. K vývoji aplikace byla použita agilní metodika z důvodu nutnosti dodání funkčního prototypu zadavateli co nejdříve.

Analýza

V první kapitole je rozebrána analýza problému a skládá se ze tří podkapitol. Tyto části jsou požadavky, forma aplikace a jazyk implementace.

Podkapitola požadavků řeší jaké všechny požadavky jsou na aplikaci vyžadovány. Zmíněné požadavky jsou funkční a nefunkční, kde funkční musí být nutně splněny pro naplnění cíle práce.

Podkapitola forma aplikace rozebírá, jestli je výhodnější aplikaci vyvíjet jako instalovatelný program na počítači nebo aplikaci umístit na webovou stránku.

Poslední podkapitola analýzy je jazyk implementace a rozebírá v jakém jazyce bude aplikace implementována. S ohledem na fakt, že byla jako forma vybrána desktopová aplikace, připadají v úvahě jazyky Java a C++.

Návrh

V druhé kapitole je rozebrán návrh klíčových částí aplikace a skládá se ze dvou podkapitol.

V první podkapitole návrhu je rozebrána volba vyhledávače. Výběr se konal mezi vyhledávačem Google a vyhledávačem Seznam.cz. Dále jsou v této podkapitole zmíněny všechny potřebné kroky ke zprovoznění vybraného vyhledávače ještě před samotnou implementací v Javě.

Druhá podkapitola návrhu hovoří o vizualizaci vazeb. Je zde popsána podoba grafů, které by měla aplikace generovat. Dále je zde popsána Java knihovna, která byla vybrána pro zobrazení daných grafů. Nakonec podkapitoly je uveden modelový příklad grafu a jeho možná interpretace.

Implementace

V třetí kapitole je rozebrána implementace klíčových částí aplikace a skládá se ze tří podkapitol.

V první podkapitole je rozebrána implementace zvoleného vyhledávání. Navíc je na konci podkapitoly zmíněn způsob, jak efektivně, za pomoci předpřipravených struktur, přidat vyhledávače další.

Druhá podkapitola hovoří o implementaci grafického uživatelského prostředí. Je zde přesně popsáno jaké všechny prvky jsou použity v daných panelech. Dále zde jsou ukázky podoby finální verze aplikace.

V třetí podkapitole je rozebrána jazyková lokalizace aplikace a její implementace. Na konci této podkapitoly je navíc popsáno, jak efektivně přidat podporu pro další jazyky bez nutnosti zásahu do již napsaného kódu.

Testování

Ve čtvrté a poslední kapitole před závěrem je rozebráno testování aplikace. Je zde rozebráno testování jak aplikace jako celek tak jejích částí. Mezi testované části patří např.: vnitřní logika aplikace, vyhledávač nebo grafické uživatelské prostředí. Dále je zde shrnut výsledek testu použitelnosti, který byl proveden se zadavatelem. Celé znění testu použitelnost lze najít v reportu zadavatele, který se nalézá v příloze C.

Vývojové prostředí

K vývoji aplikace bylo použito NetBeans IDE. NetBeans IDE je vývojové prostředí pro mnoho různých programovacích jazyků. Toto prostředí poskytuje nepřehledné množství funkcí zjednodušující práci programátora.

Analýza

Analýza je nejdůležitější částí správného projektu. Bez řádné analýzy situace, problémů a potřeb práce se snadno může stát, že práce selže. V této kapitole se nalézá analýza požadavků na aplikaci. Dále je zde zmíněna podoba aplikace a jaké prostředky a přístupy by měly být zvoleny k realizaci aplikace.

1.1 Požadavky

Je několik druhů požadavků na vyvíjený software. V této práci se zmíníme o funkčních a nefunkčních požadavcích. Požadavky byly sbírány během schůzek se zadavatelem práce a vedoucí práce. V čase se měnily a jejich finální forma je uvedena v této kapitole.

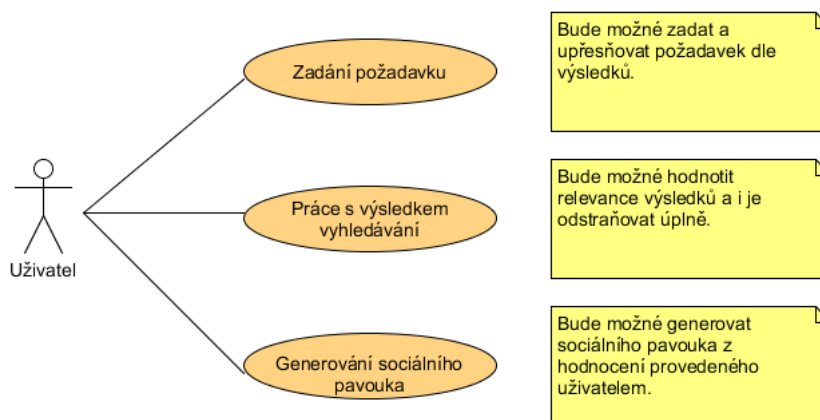
1.1.1 Funkční požadavky

Funkční požadavky jsou takové požadavky, které přímo plní cíle aplikace a to proč se vyvíjí. V pokynech na vypracování, které jsou součástí zadání bakalářské práce, jsou tyto požadavky velice dobře specifikovány.

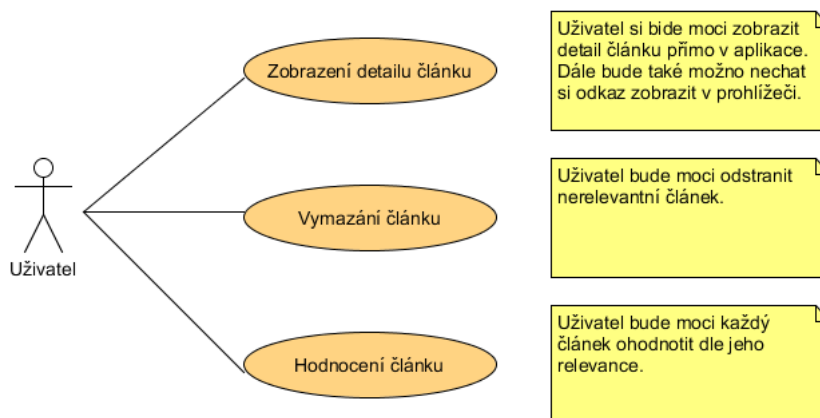
Zprv je nutné mít možnost zadat požadavek. Zadruhé po zobrazení výsledků musí být možné tyto výsledky ohodnotit. Naposled je potřeba aby bylo možné vygenerovat síťový graf vazeb hledaných subjektů. Všechny tyto požadavky, viz obrázek 1.1, jsou součástí cíle práce a každý musí být splněn .

V Use Case diagramu na obrázku 1.2 jsou jak zásadní cíle pro úspěch práce, tak nadstandardní funkce nevyplývající ze zadání práce. Tyto funkce navíc pomáhají uživateli k jednodušší práci s programem.

Zobrazení detailu článku je klíčové, bez detailu nelze článek číst a tím pádem ani rozumně ohodnotit. Vymazání článku je už nadstavbová funkce. Reálně by stačilo přiřadit článku nulovou relevanci, ale tento přístup by zhoršil orientaci mezi články, jelikož by uživatel zbytečně musel přeskakovat nerele-



Obrázek 1.1: Use Case - funkční požadavky



Obrázek 1.2: Use Case - rozpad případu použití: Práce s výsledkem vyhledávání

vantní články při projíždění. Hodnocení článku je opět zásadní pro uspokojení cílů práce.

1.1.2 Nefunkční požadavky

Nefunkční požadavky jsou takové požadavky, které přímo nevyplývají ze zadání, ale mají velký vliv na konečný stav aplikace. Nefunkční požadavky na vyvíjenou aplikaci byly následující.

- jednoduché grafické prostředí
- nízké požadavky na hardware

- nízké, ideálně žádné, periodické výdaje na placení kvót
- jednoduchá rozšiřitelnost
- možnost využití kdekoliv (kancelář v práci, doma ...)

1.2 Forma aplikace

Během schůzek se zadavatelem se ustálovala i forma aplikace. Ze začátku nebylo zřejmé, jestli aplikace bude vyvíjena jako webová a nebo desktopová. Tudíž ani nebylo jisté, v jakém jazyce by se měla vyvíjet. Zadavatel neměl žádný konkrétní požadavek v tomto ohledu.

Nejdříve byla vedena diskuze, zda bude aplikace webová nebo desktopová. Rozbory a závěr této diskuze jsou vypsány v následujících odstavcích.

1.2.1 Webová aplikace

Tento přístup tenkého klienta přináší spoustu výhod i nevýhod.

1.2.1.1 Výhody

- dostupnost : Aplikace je dostupná odkudkoliv a kdykoliv. Jediné období nedostupnosti může být při selhání poskytovatele hostingů.
- aktualizace : Uživatel nemusí nikdy aktualizovat aplikaci. Toto se provede na straně serveru a uživatel o tom ani neví, pokud ovšem není vizuálního charakteru.

1.2.1.2 Nevýhody

- cena : Server nebo hosting se musí zaplatit.
- autorizace a autentizace : Bylo by nutné k aplikaci omezit přístup. Aplikace je určena pro zadavatele a jeho tým a nikoho jiného.

1.2.2 Desktopová aplikace

Stejně jako webová aplikace má i desktopová aplikace svoje výhody a nevýhody.

1.2.2.1 Výhody

- cena : Desktopová aplikace nestojí nic navíc.
- autorizace a autentizace : Není nutná. Aplikace se nainstaluje přímo na počítač předem schválené osoby.

1. ANALÝZA

- dostupnost : Po instalaci je aplikace dostupná kdykoliv z daného počítače.

1.2.2.2 Nevýhody

- dostupnost : Aplikace je dostupná pouze z počítače kde je nainstalovaná. Na každé další zařízení se musí opět nainstalovat.
- aktualizace : Aktualizace se musí propagovat na každý počítač, kde je aplikace nainstalována.

1.2.3 Závěr

Se zadavatelem a vedoucí práce bylo dohodnuto, že nejlepší bude zvolit přístup desktopové aplikace. Odstraní se tím nutnost měsíčních plateb poskytovateli hostingu. Počet osob užívající aplikace není nijak vysoký, tím pádem problém aktualizace softwaru na každém stroji není nijak závažný. Projekt se zjednoduší i po implementační stránce, jelikož nebude nutné provádět autorizaci a autentizaci uživatele.

1.3 Jazyk implementace

Jazyk, v jakém se bude psát práce, byl řešen hned po rozhodnutí, že aplikace bude psána jako desktopová. Během konverzace o jazyce implementace vstaly dvě varianty. První varianta byla psát aplikaci v jazyce C++, druhá varianta byla jazyk Java. Rozbory a závěr této debaty jsou v následujících odstavcích.

1.3.1 C++

1.3.1.1 Výhody

- objektově orientovaný jazyk
- Rychlejší běh aplikace. Avšak aplikace vyvíjena touto prací není nijak výpočetně náročná, tzn. rozdíl vůči jazyku Java bude minimální.

1.3.1.2 Nevýhody

- nutnost kompilace nového programu pro každou platformu
- mnohem nižší počet dostupných knihoven
- mnoho GUI frameworků zabývajících se různými platformami

1.3.2 Java

1.3.2.1 Výhody

- objektově orientovaný jazyk
- multi-platformní
- velké množství hotových knihoven
- vlastní multi-platformní GUI framework

1.3.2.2 Nevýhody

- Běh aplikace je teoreticky pomalejší vzhledem k nutnosti interpretace pseudokódu. Ovšem po zvážení obsahu práce lze usoudit, že se tento fakt prakticky neprojeví.

1.3.3 Závěr

S ohledem na řečené výhody a nevýhody se ustanovilo, že bude aplikace vyvíjena v jazyku Java. Nevýhody jazyku Java jsou naprosto irelevantní oproti výhodám vzhledem k obsahu práce. Fakt, že aplikace psány v Javě jsou pomalejší než aplikace psané v C++, není dostačující pro přehodnocení rozhodnutí. Navíc stačí jeden kód a aplikace může fungovat na velmi mnoho různých platformách. Jazyk Java také poskytuje velmi mnoho nástrojů pro usnadnění a zrychlení vývoje aplikace.

Návrh

Pro naplnění požadavků 5 je zapotřebí implementovat dvě hlavní funkce. Zaprvé je nutné zprovoznit nějaký vyhledávač a napojit ho k aplikaci. Bez prohledávání internetu a vrácení článků nemůže aplikace naplnit cíle práce. Za druhé je nutné najít nějaký grafický nástroj, pravděpodobně knihovnu, která bude schopná vykreslovat grafy uvnitř Java aplikace. Ostatní cíle práce můžeme splnit použitím klasických prostředků jazyku Java.

2.1 Vyhledávač

Během rozhovorů s vedoucí a zadavatelem se ze začátku rozhodovalo mezi dvěma poskytovateli vyhledávání. Prvním byl Google s.r.o. a druhým byl Seznam a.s. Aspoň jeden z nich bylo potřeba implementovat pro splnění funkčního požadavku.

2.1.1 Google s.r.o.

Google s.r.o. je na trhu vyhledávačů od roku 1998 [10]. Od této doby neustále pracuje na zlepšení svých služeb a momentálně poskytuje jeden z nejlepších a nejpopulárnějších vyhledávačů na světě [12].

2.1.1.1 Výhody

- dlouholeté kvalitní služby
- dostupné jednoduché API [1]
- největší na světě

2.1.2 Seznam.cz a.s.

Seznam.cz a.s. je na trhu o něco déle než Google s.r.o., a to od roku 1996 [11]. Ovšem Seznam nezažil na české půdě takový růst jako Google na ame-

rické a je dnes podstatně menší. Mezi českými vyhledávači však stále patří k nejoblíbenějším [12].

2.1.2.1 Výhody

- dlouholeté kvalitní služby
- největší v ČR

2.1.2.2 Nevýhody

- veřejné API není dostupné

2.1.3 Závěr

API ke hledání přes Seznam.cz nebylo nalezeno a domluva s pracovníkem firmy neměla dobré výsledky. Implementace se jevila jako zbytečně komplikovaná a zdlouhavá. Z tohoto důvodu byl zvolen Google jako první vyhledávač k implementaci. Tato firma poskytuje jednoduché API, díky kterému tato aplikace získá základní funkcionalitu a začne naplňovat cíle práce. Navíc je Google nejpoužívanější na světě a tím pádem by pravděpodobně měl vracet dobré výsledky.

2.1.4 CSE

Google Custom Search provozovaná Google s.r.o. je API pro získávání výsledků hledání po internetu. Primárně slouží pro zakomponování Google vyhledávání do webové stránky, ale dá se využít i přímo v desktopové aplikaci. V základním tarifu je služba zcela zdarma. To ovšem s sebou přináší omezení maximálního počtu dotazů za den. Dokumentace je ke zhlédnutí na [1].

2.1.4.1 Vytvoření přístupu

- Registrace u Google.
- Přejít na stránku [2].
- Zadáme jméno vyhledávače a část webu, který chceme prohledávat. Dáme vytvořit.
- Přejdeme do nastavení.
- Nastavíme prohledávání celého webového prostoru.

Takto vytvořený Custom Search Engine má dvě specifika a tím jsou ID a klíč. Díky těmto dvěma údajům Google pozná kdo mu posílá dotaz. Google poskytuje zkušební online formulář na stránce [3].

2.1.4.2 Parametry a výstup CSE

- cx : id uživatele
- key : klíč uživatele
- q : dotaz pro vyhledávač

CSE má také spoustu volitelných parametrů, ovšem mnoho z nich je zaměřeno na obsah, jako třeba obrázky, který není pro tuto práci vůbec důležitý. Přehled všech parametrů lze nalézt na stránce [4].

Zmíněná stránka sama sobě odporuje v povinných parametrech. Na začátku jsou uvedeny všechny tři povinné parametry, ovšem v následující tabulce je pouze jeden povinný parametr a to q. Google odmítá vracet výsledky straně, která se mu řádně neidentifikuje, protože denní limit dotazů neidentifikovaného klienta je nula. Tudíž, povinné parametry jsou tři.

Pro práci s Google CSE se využívá stránka [5]. Tato stránka vrací odpověď ve formátu JSON. Google také nabízí jednoduchou Java knihovnu s názvem Gson [8], která umí převést JSON v textové formě do formy objektové, což značně usnadní práci s předaným výsledkem.

2.1.4.3 Využití

Klasické využití CSE je pro webovou stránku nějaké třetí strany. Stačí si do stránky vložit kus kódu, který Google vygeneruje po vytvoření CSE, a celá struktura stránek a veškerý obsah je prohledávatelný přímo z dané stránky. To zjednoduší a zrychlí práci uživatele se stránkou. Více na [7].

2.1.4.4 Příklad

Vyhledávání hesla Petr Nečas Andrej Babiš přes Sociálního pavouka a web Google.com ze dne 3.11.2014.

Z tabulky 2.1 shrnující obrázky 2.1 a 2.2 je očividné, že oba způsoby vrací úplně jiné výsledky. Pouze první vrácený výsledek se shoduje. Toto je způsobeno Google politikou. Pro hledání přes prohlížeč se Google snaží zlepšit výsledky informacemi, které o klientovi nasbíral. Ovšem při hledání přes CSE, který má sloužit jako vyhledávání na webových stránkách určených široké veřejnosti, se chová anonymně. Více je možno vyčíst na stránce [6].

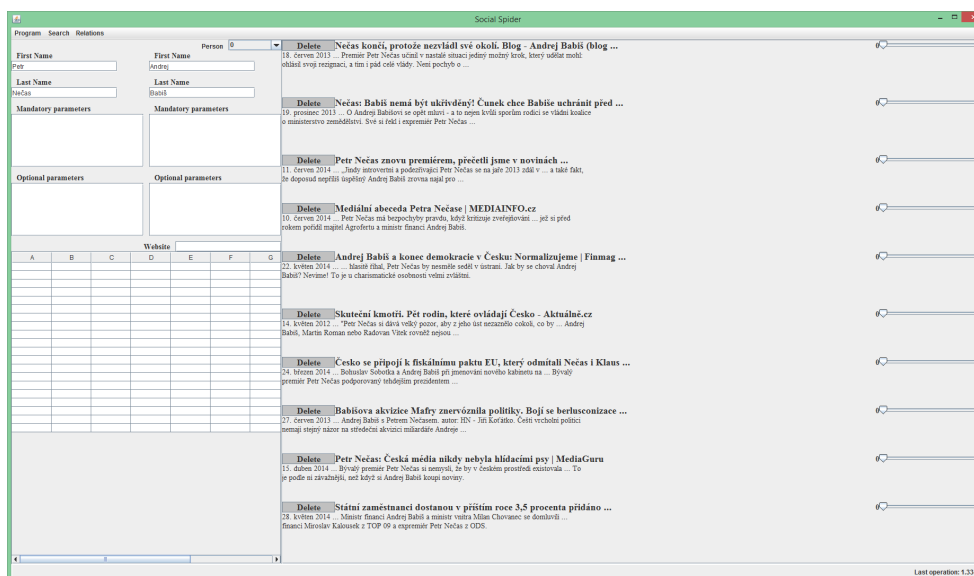
Momentální sofistikovanost vyhledávání na internetu spolu s dostupností zdrojů veřejných informací špatně pokrývají požadavky projektu Sociálního Pavouka. Velice často nastávají dvě mezní varianty.

- Mnoho výsledků, které mají nulovou výpovědní hodnotu
- Žádné výsledky

2. NÁVRH

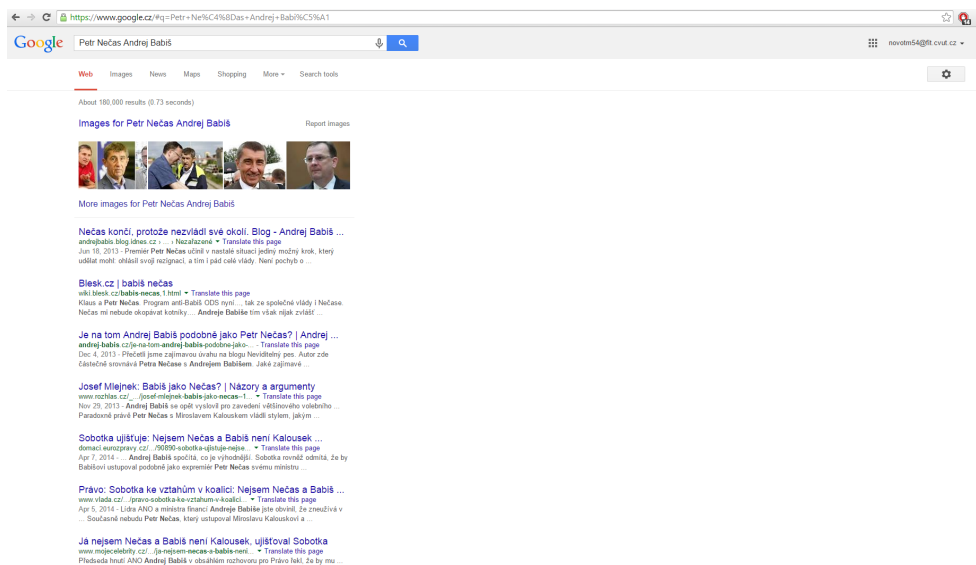
Tabulka 2.1: Vracené odpovědi od Google.com vs Google CSE

Google Custom Search 2.1	Google.com 2.2
Nečas končí, protože nezvládl své okolí...	Nečas končí, protože nezvládl své okolí ...
Nečas: Babiš nemá být ukřivděný. Čunek...	Blesk.cz babiš nečas
Petr Nečas znovu premiérem, přečetli jsme...	Je na tom Andrej Babiš podobně jako Petr ...
Mediální abeceda Petra Nečase ...	Josef Mlejnek: Babiš jako Nečas? Názory...
Andrej Babiš a konec demokracie v Česku ...	Sobotka ujišťuje: Nejsem Nečas a Babiš ...
Skuteční kmotři. Pět rodin, které ovládají ...	Právo: Sobotka ke vztahům v koalici: ...
Česko se připojí k fiskálnímu paktu EU, ...	Já nejsem Nečas a Babiš není Kalousek ...



Obrázek 2.1: Příklad: Výsledky přes CSE

Google buď nemá dostatečně inteligentní index a nebo, což je mnohem pravděpodobnější, dané zdroje informací na internetu prostě zatím neexistují. Inteligentnější vyhledávání a syntaktická analýza požadavku by ovšem významně vylepšila množinu pozitivních výsledků.



Obrázek 2.2: Příklad: Výsledky přes vyhledávač

2.2 Vizualizace vazeb

Existuje mnoho způsobů jak kreslit grafy jako např.: sloupcové, koláčové, síťové, funkce atd. V zadání Sociálního pavouka je potřeba nakreslit graf, kde je mnoho subjektů a mezi nimi nějaké vazby.

Co se týče kreslení různých vazeb mezi nějakou malou skupinou objektů, je velice dobré využít grafu síťového. Síťový graf umožňuje velice dobře zobrazit sounáležitost objektů a nebo jejich vazbu v jednoduchém a čistém prostředí.

- Barvy uzlů k tvoření skupin.
- Délky hran ke znázornění sounáležitosti nebo blízkosti spojených uzlů.
- Barva hran k důležitosti nebo síle dané vazby.

Ovšem pro větší soubory objektů se síťový graf stává nepřehledný a přeplněný.

Z jednoduchého důvodu čitelnosti grafu a velikosti souborů dat je v aplikaci použit síťový graf při kreslení vzájemných vazeb daných subjektů.

2.2.1 JUNG

Java Universal Network/Graph Framework je knihovna na grafovou vizualizaci v jazyce Java. JUNG je licencován pod BSD licenci. Dokumentaci je možno zhlédnout na [9].

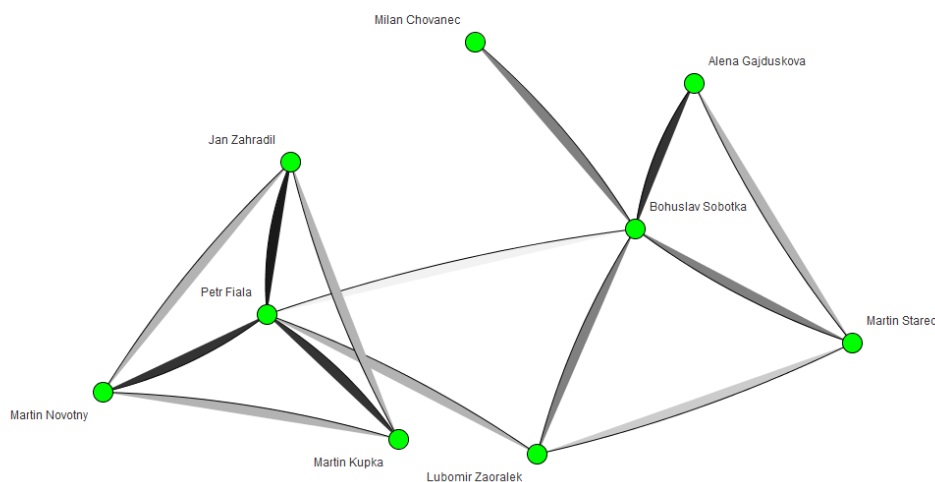
2. NÁVRH

JUNG implementuje velmi mnoho grafů a algoritmů nad těmito grafy. Pro tuto aplikaci bude stačit pouze velmi malá podmnožina ze všech možností knihovny.

- vizualizace grafu v panelu okna
- tvoření a editace neorientovaných grafů
- pohyby a posuny grafu nebo jeho částí

2.2.2 Příklad

Data 2.3 i interpretace jsou smyšlené a míněné pouze jako příklad.



Obrázek 2.3: Příklad: Sociální pavouk

Na grafu je devět lidí. Petr Fiala, Jan Zahradil, Martin Novotný, Martin Kupka, Lubomír Zaorálek, Milan Chovanec, Bohuslav Sobotka, Alena Gajdusková a Martin Starec. Na první pohled se graf dá rozčlenit na dva shluky.

První shluk zahrnuje Petra Fialu, Jana Zahradila, Martina Novotného a Martina Kupku. Petr Fiala má silnou vazbu ke zbývajícím třem. Členové skupiny mezi sebou mají slabší vztah, než je vztah každého se středem shluku, Petrem Fialou. Vysvětlení může být následující. Petr Fiala je předseda strany, zbývajících tři osoby jsou místopředsedové strany. Předseda strany musí komunikovat silně se svými místopředsedy, aby mohl správně delegovat povinnosti. Místopředsedové už mezi sebou tolik komunikovat nemusí.

Druhý shluk zahrnuje Bohuslava Sobotku, Milana Chovance, Alenu Gajduskovou, Martina Starce a Lubomíra Zaorálka. Opět středem shluku je předseda

strany, tentokrát Bohuslav Sobotka. Ovšem tento shluk není natolik centralizovaný jako shluk první a nachází se v něm chybějící vazby mezi různými články shluku. Toto se dá interpretovat momentálně nižší komunikací mezi danými subjekty. Zároveň vazby se středem shluku jsou jinak silné. Interpretace může být zatížení nebo vyšší produktivita daného článku.

Vazby vedoucí mezi shluky samotnými už nejsou tak silné jako vazby uvnitř shluku. Jednoduchá interpretace nižší komunikace mezi nestraníky a straníky.

Implementace

V této kapitole je možné se dočíst jak přesně byly různé aspekty aplikace implementovány. Každá podkapitola hovoří o daném aspektu. Jak již bylo zmíněno v kapitole 1, byla implementace provedena v jazyce Java. Navíc je zde rozebráno jak rozšířit určité aspekty aplikace za použití již naimplementovaných struktur bez potřeby měnit již existující kód.

3.1 Vyhledávání

Jako první bylo implementován vyhledávač. Jak bylo zmíněno v kapitole 2, byl jako vyhledávač zvolen Google CSE.

3.1.1 CSE

Je nutné vytvořit si CSE na stránkách Google jak bylo popsáno v 2. Bez platného ID a KEY Google nevrací žádné výsledky na zadaný dotaz. Pro získání výsledků hledání se využívá stránka [5].

```
URL url = new URL(askThis);
```

Vytvořený URL objekt poslouží k získání odpovědi. V proměnné askThis je stránka na komunikaci s Google serverem [5] a všechny potřebné parametry.

Navázání spojení se dělá přes metodu openConnection() na objektu třídy URL.

```
URLConnection connection =  
    (URLConnection) url.openConnection();
```

Nyní se ošetří možné nesrovnalosti. Google používá klasické HTTP stavové kódy. Nejčastější tři jsou uvedeny v tabulce 3.1.

Naposled stačí už jenom stáhnout odpověď ze vstupního proudu, který získáme přes metodu getInputStream() objektu connection.

```
try (BufferedReader reader = new BufferedReader(  

```

3. IMPLEMENTACE

Tabulka 3.1: Nejčastější CSE status kódy a jejich význam

Kód	Důvod
200	Vše proběhlo správně.
400	Špatně formulovaný dotaz.
403	Vyčerpán limit dotazů.

```
new InputStreamReader(
    connection.getInputStream(),
    STANDARD_HTTP_ENCODING)) {
    StringBuilder stringBuilder =
        new StringBuilder();
        while (true) {
            String line = reader.readLine();
            if (line == null) {
                break;
            }
            stringBuilder.append(line);
        }
}
```

S takto získanou odpovědí by se ovšem těžce pracovalo jelikož je v textové podobě tj. jeden dlouhý řetězec znaků. V tuto chvíli přichází na řadu knihovna Gson, která se dá získat na stránce [8]. Gson je licencován pod Apache License 2.0.

```
public class GoogleResultBean {
    private List<MinimalResultBean> items;

    ... getry & setry
}
```

```
public class MinimalResultBean {
    private String title;
    private String snippet;
    private String link;

    ... getry & setry
}
```

```
GoogleResultBean result = new Gson().
    fromJson(response,
        GoogleResultBean.class);
```

Pro načtení potřebných informací stačí pouze jedna řádka se správně definovanými třídami. Jména items, title, snippet a link nejsou náhodná. Tato jména se objevují v JSON odpovědi od Google. Tzn. při definici třídy stačí pojmenovat proměnné stejně jako pole, které chceme z JSON získat, a metoda fromJson(String, Class) v třídě Gson se už o vše postará. První parametr je řetězec v JSON formátu. Druhý parametr je třída, kterou má použít na uložení dat a proskenování řetězce.

3.1.2 Další vyhledávače

Aplikace je momentálně vybavena pouze jedním vyhledávačem, což je potenciální nevýhoda. Z tohoto důvodu je struktura implementace aplikace navržena pro jednoduché rozšíření o další vyhledávače bez potřeby modifikace již napsaného kódu.

3.1.2.1 Povinné údaje

- odkaz na stránku se článkem
- titulek článku
- textová ukázka článku

Z těchto údajů je naprosto nezbytný pouze odkaz na stránku se článkem. Ostatní můžou chybět. Obecně ale můžeme říct, že tyto údaje jsou pouze malá podmnožina toho, co nám každý lepší vyhledávač vrátí.

3.1.2.2 Přidání dalšího vyhledávače

Přidání dalšího vyhledávače do aplikace má dva kroky. První krok spočívá v implementaci nové třídy, která bude rozšiřovat již existující abstraktní třídu AbstractSearchEngine. Tato třída má pouze dvě abstraktní metody.

- `List<StandardResultBean> getResults(UserInputBean)` : Tato metoda vrací seznam výsledků poskytnutých vyhledávačem. Instance třídy UserInputBean obsahuje všechny řetězce k vyhledání, které uživatel zadal do aplikace.
- `List<StandardResultBean> getMoreResults()` : Tato metoda vrací pomyslnou další stránku výsledků již proběhlého předcházejícího vyhledávání.

Druhý krok spočívá v přidání nově implementovaného vyhledávače do menu vyhledávačů. To se provede metodou addSearchEngine(AbstractSearchEngine, String) na třídě MenuBar. První argument metody je instance nově implementovaného vyhledávače. Druhý argument je řetězec se jménem daného vyhledávače. Toto jméno se použije k zobrazení vyhledávače v menu aplikace.

Při správném dodržení zmíněných kroků je zaručena veškerá další funkcionálnita nad vyhledávači jako je procházení mezi výsledky různých dotazů a jiné.

3.2 GUI

Grafické prostředí aplikace je vytvořeno pomocí Java Swing. Celá aplikace má dvě proměnlivé plochy. Jedna plocha je určena pro hledání výsledků a práci s nimi. Druhá plocha je určená pro kreslení vazeb.

3.2.1 Plocha hledání

Plocha hledání 3.1 se rozděluje na několik částí. Vpravo je okno s výsledky hledání. Vlevo je panel, který nahoře má zadávací pole pro hledání, dole je pak tabulka pro zapisování poznámek.

Tabulka poznámek je Swingová komponenta `JTable`. Tato tabulka se dá libovolně zvětšit či zmenšit z menu.

Panel pro zadávání má dvě identické strany pro zadávání 1 a více osob. Jedno okno se pak skládá ze dvou `JTextField` a dvou `JTextArea`. Nad panelem zadávání je ještě `JComboBox` s listem dalších osob. Toto zaručuje možnost překlíkávání při vyšším počtu osob. Pod panelem je `JTextField` pro zadání stránky, kdybychom chtěli výsledky omezit pouze na tuto stránku.

Panel výsledků se skládá ze seznamu panelů. V každém tomto menším panelu je `JButton` pro smazání záznamu. `JSlider` pro hodnocení důležitosti odkazu. Titulek článku a ukázka z něj. Po kliknutí na titulek se místo panelu výsledků zobrazí detail článku.

3.2.2 Plocha vazeb

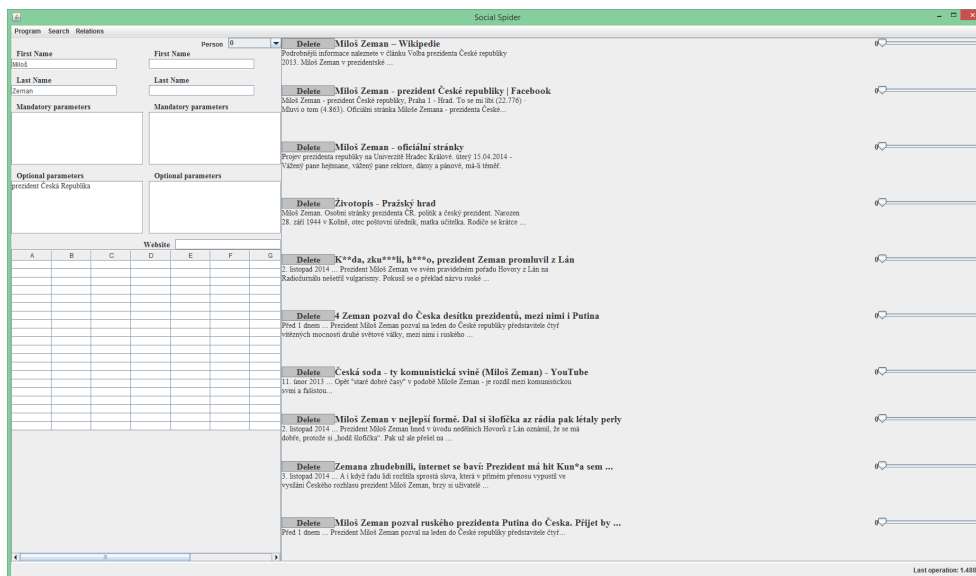
Plocha vazeb 3.2 se skládá z seznamu všech momentálně načtených vazeb v aplikaci. Panelu pro filtrování vazeb dle data. Následně pak panelu pro samotný obrázek vazeb.

Panel vazeb se skládá ze seznamu složených panelů. Každý menší panel má v sobě jméno vazby tj. lidí oddělení čárkou, podílejí se na vazbě. Dále `JButton`, který odznačí danou vazbu v grafu. A nakonec seznam `JRadioButton` pro označení dané vazby k zobrazení v grafu.

Panel filtru v sobě má dva `JSpinner` komponenty. Jeden pro počáteční datum a druhý pro koncové datum filtru. Dále pak dva `JButton` pro aktivaci a zrušení filtru.

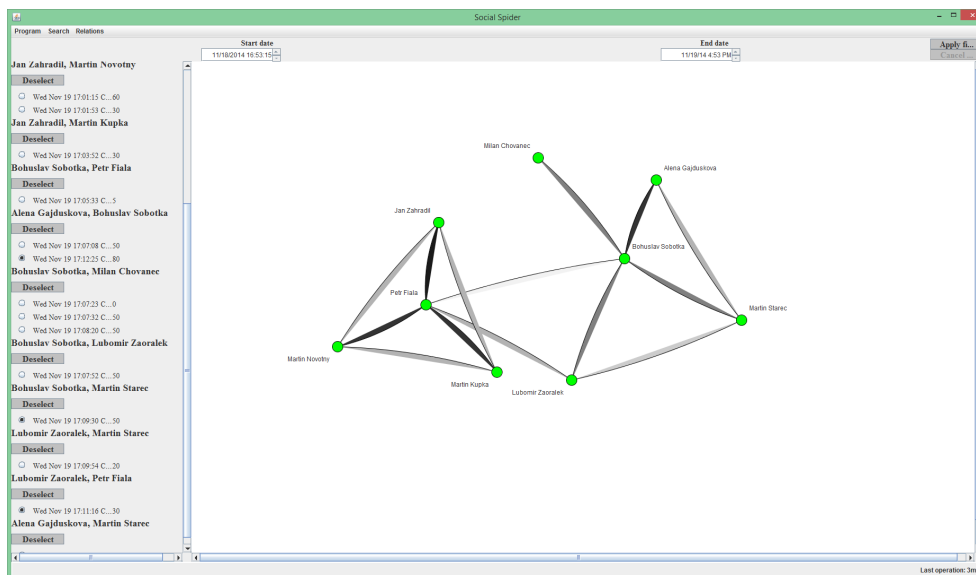
Nakonec panel grafu má v sobě jedině a to obrázek grafu. Vizualizace vazeb je zařízená knihovnou JUNG, která lze nalézt na [9]. Každý člověk je reprezentován uzlem. Vazby mezi lidmi jsou reprezentovány hranou mezi danými uzly. Čím je vazba silnější, tím je barva hrany tmavší a zároveň je hrana kratší.

3.2. GUI



Obrázek 3.1: Ukázka - plocha hledání

Tzn. lidi, kteří k sobě „mají blíž“ jsou reprezentovány uzly s krátkou hranou a naopak.



Obrázek 3.2: Ukázka - plocha vazeb

3.2.2.1 Třída pro změnu barvy hrany

Tato třída mění barvu hrany v grafu v závislosti na síle vazby mezi danými subjekty reprezentovanými krajními uzly hrany. Čím je vazba silnější, tím má hrana tmavší zbarvení.

```
new Transformer<GraphEdgeBean, Paint>() {
    @Override
    public Paint transform(GraphEdgeBean s) {
        float value = s.getRating();

        value = ( value * 100 ) / (
            ComponentCreator.SLIDER_MAX_VALUE -
            ComponentCreator.SLIDER_MIN_VALUE );
        value = ( 100 - value ) / 100;

        return Color.getHSBColor(0, 0, value);
    }
}
```

3.2.2.2 Třída pro změnu vzdálenosti uzlů

Tato třída mění vzdálenost uzlů v grafu v závislosti na síle vazby mezi danými uzly. Čím je vazba silnější, tím jsou k sobě uzly blíže.

```
new Transformer<GraphEdgeBean, Integer>() {
    @Override
    public Integer transform(GraphEdgeBean i) {
        return ( ComponentCreator.SLIDER_MAX_VALUE -
            ComponentCreator.SLIDER_MIN_VALUE -
            i.getRating() ) * 3 + 100;
    }
}
```

3.3 Jazyková lokalizace

Všechny potřebné věty a hesla pro každý podporovaný jazyk jsou uloženy v jednom souboru. Výchozí jazyk je v `messages.properties`. Každý další jazyk je v obdobném souboru, kde za `messages` je podtržítka a zkratka daného jazyku, např.: `messages_cs_CZ.properties`.

3.3.1 Třída Messages

Tato třída slouží k získávání řetězců překladu pro aplikace.


```

public enum Messages {
    Social_Spider , Close_Question , Exit_Confirmation ,
    atd . ;

public final String getMessage(Object ... args) {
    ResourceBundle bnd = ResourceBundle.getBundle(
        "org.cuni.fsv.socialspider.messages" );

    try {
        String tmpl = bnd.getString(name());

        return MessageFormat.format(tmpl, args);
    } catch( MissingResourceException |
        IllegalArgumentException ex) {
        Logger.getLogger(getClass().getName())
            .log( Level.WARNING, "", ex);

        return name().replace('_', ' ');
    }
}
}

```

Metoda vracející hesla najde jazykový balík. V tomto balíku pak najde hledanou větu, tu zformátuje dle argumentů a vrátí ji zpět dotazovanému. Pokud balík neexistuje nebo jsou argumenty špatně, metoda vrátí jméno enumu, na kterém byla metoda zavolána se změnou. Pro lepší čitelnost se před vrácením všechny podtržítka přepíší na mezeru.

3.3.2 Soubor messages*.properties

Takovéto soubory obsahují všechny potřebné řetězce s překlady pro aplikaci.

```

Social_Spider=Social Spider
Close_Question=Are you sure you want to exit?
Exit_Confirmation=Exit Confirmation
atd .

```

3.3.3 Rozšíření jazykové podpory

Pro přidání dalšího podporovaného jazyku stačí přidat *.properties soubor s překlady všech vět a hesel. Aplikace automaticky volí jazyk podle počítače, na kterém se nachází. Pokud jazyk není dostupný, zvolí výchozí jazyk tj. angličtinu.

Testování

Jediné testování aplikace kromě manuálního, které bylo prováděné neustále během vývoje, byl test použitelnosti. Ostatní testy jako Unit testy nejsou vhodné na práci, která se odehrává prakticky celá ve vykreslování grafických objektů na monitor uživatele. Proto testy typu správné sekvence zobrazení při proklikávání byly prováděny pouze manuálně.

Vnitřní logika, jako třeba ukládání obrázku grafu nebo dat byla také testována manuálně. Napsat test, který by zkontroloval existenci souboru je jednoduché, ale napsat test, který zkontroluje, jestli je obrázek uvnitř souboru grafově stejný by bylo více obtíží než užitku. Graf musí být dostatečně velký, ale ne moc. Zároveň může být jakkoliv pootečený, posunutý, popisky trochu jinde a jiné.

Jediné automatizované testování se v této práci hodí na třídu vyhledávače. Ovšem v rámci této práce, kdy celé grafické prostředí pracuje s výsledky vrácenými vyhledávačem, je zbytečné tyto testy psát, jelikož při prvním spuštění je naprosto jasné, jestli zadaný vyhledávač funguje tak jak má.

4.1 Test použitelnosti

Během vývoje aplikace byl proveden se zadavatelem test použitelnosti aplikace. Celý text zadavatelova reportu je možný zhlédnout v příloze C.

4.1.1 Platformy

Testování proběhlo na počítačích s různými operačními systémy. Na všech těchto systémech fungovala aplikace bez problémů.

4.1.2 Grafické prostředí

Splňuje všechny požadavky. Chyba se zadáváním specifické stránky vyhledávání opravena ve vyšší verzi, než měl zadavatel dostupnou během svého

4. TESTOVÁNÍ

testování.

4.1.3 Celkový výsledek testu použitelnosti

Aplikace splnila všechny vytyčené cíle. Avšak z důvodu velkého množství ne-relevantních odkazů aplikace nesplnila očekávání zadavatele.

4.2 Výsledky testování

Aplikace splňuje všechny požadavky práce, spňuje všechny funkční i nefunkční požadavky a tedy může být považována za úspěšnou.

Závěr

Zhodnocení analýzy a návrhu

Kapitoly 1 a 2 rozebírají jak by měla být práce vytvořena. I přes nedostatek času na začátku práce, který byl vyvolán nutností dodání fungujícího prototypu, dopadla jak analýza tak návrh úspěšně.

Zhodnocení implementace

Z faktu, že aplikace je hotová, funkční a splňuje všechny požadavky zadání se dá říct, že implementace, rozebíraná v kapitole 3, proběhla úspěšně.

Náměty pro další vývoj

Další vývoj aplikace může spočívat v implementaci některých z níže zmíněných funkcionalit.

- vyšší počet podporovaných vyhledávačů
- vyšší počet podporovaných jazyků
- vytvoření toolbarů pro jednodušší práci s aplikací
- vyšší konfigurovatelnost různých aspektů aplikace

Pro obsáhlejší vývoj aplikace je doporučena hlubší textová a obsahová analýza vrácených výsledků. Druhá možnost je implementace vlastního vyhledávače, který je uzpůsoben na hledání sociálních vazeb mediálně známých osob. Tímto by se daly vyřadit nerelevantní výsledky, což by mělo velmi pozitivní dopad na aplikaci, jelikož tento aspekt je hlavním důvodem nespokojenosti zadavatele.

Celkové zhodnocení

Cílem této práce bylo vytvořit aplikaci pro zadávání subjektů, hodnocení síly vazby subjektů na základě výsledků vrácených vyhledávačem a následnou vizualizaci vazeb. Všechny tyto cíle byly splněny.

Je nutné podotknout, že nebyla splněna očekávání zadavatele. Ve většině případech vyhledávač vrací nerelevantní odkazy a tudíž hodnocení článků je zcestné, tím pádem vygenerovaný pavouk nemá velkou výpovědní hodnotu. Toto je způsobeno mediální známostí hledaných osob a neuzpůsobením vyhledávání na sociální vazby subjektů. Následná nutnost ručního třídění nerelevantních článků je velice časově náročná a uživatelsky nepříjemná. Pro vylepšení aplikace by bylo potřeba implementace jiného vyhledávače, který pracuje na jiném principu než obecné webové vyhledávače jako je Google nebo Seznam.cz.

Seznam použitých zkratk

- API** Application Programming Interface - rozhraní pro využití aplikační funkcionality
- CSE** Custom Search Engine - název Google API pro vyhledávání
- GUI** Graphical User Interface - grafické uživatelské prostředí
- IDE** Integrated Development Environment - vývojové prostředí
- JSON** JavaScript Object Notation - způsob jak předat v jednoduchém textu strukturované objekty
- JUNG** Java Universal Network/Graph Framework - knihovna pro kreslení grafů v javě

Obsah přiloženého CD

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

Test použitelnosti: report zadavatele

C.1 Sociální pavouk - výsledky testování

C.1.1 Prostředí

Aplikace byla testovány na různých počítačích s různými operačními systémy: Window 7, Windows 8, MacOS. Na všech těchto platformách aplikace fungovala bez problémů.

C.1.2 Celkový dojem

Aplikace má příjemný vzhled, základní funkce fungují, vykreslování vazeb v grafickém prostředí je poutavé. Nefunguje, či funguje nespolehlivě vyhledávání na konkrétní zadané stránce (pole: „webová stránka“), není zřejmá funkce poznámek (tabulka v levé spodní části hlavního okna).

Celkově ale aplikace příliš nenaplnuje původní očekávání. Zejména z důvodu velkého množství nerelevantních odkazů, které se zobrazují ve výsledcích. Jedná se v podstatě o jednoduchou nadstavbu nad Google vyhledávačem, která pouze prostřednictvím formuláře umožňuje zadat hledaná slova bez nutnosti použití logických výrazů, což není velká přidaná hodnota. Možnost hodnotit výsledky pomocí jezdce v pravé části příliš přidanou hodnotu nezvyšuje, právě z důvodu enormního množství nerelevantních odkazů.

Smyslem aplikace bylo najít stránky, na kterých se vyskytují informace o dvou různých hledaných osobách. Přitom se vrací stránky, kde jsou informace jak o jedné nebo o druhé osobě a ty kde jsou informace o obou osobách, nejsou nijak přednostně zobrazeny.

C.1.3 Náměty na zlepšení funkcionality

Aby se funkčnost aplikace přiblížila původní představě, bylo by zřejmě nutné připojit určitou analýzu textu najitých odkazů. Ukazuje se, že pouze zobrazování výsledků, které na základě zadaných slov nabídne Google, není v praxi použitelná cesta. Aplikace by měla obsahovat vlastní modul, který bude procházet nalezené odkazy a na prvních místech zobrazovat ty, kde se budou zobrazovat všechna hledaná slova, zejména jména.

Uživatelská příručka

D.1 Úvod

Sociální pavouk je aplikace na vyhledávání subjektů a vizualizaci vazeb mezi nimi. Vytvořte si jednu nebo více skupin lidí a sledujte vývoj interakce v čase.

D.2 Nutné nainstalovat

Aplikace je napsána v jazyku Java a proto je pro spuštění nutnost mít na počítači nainstalováno Java Runtime Environment. JRE lze stáhnout ze stránek společnosti Oracle, např. <http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>.

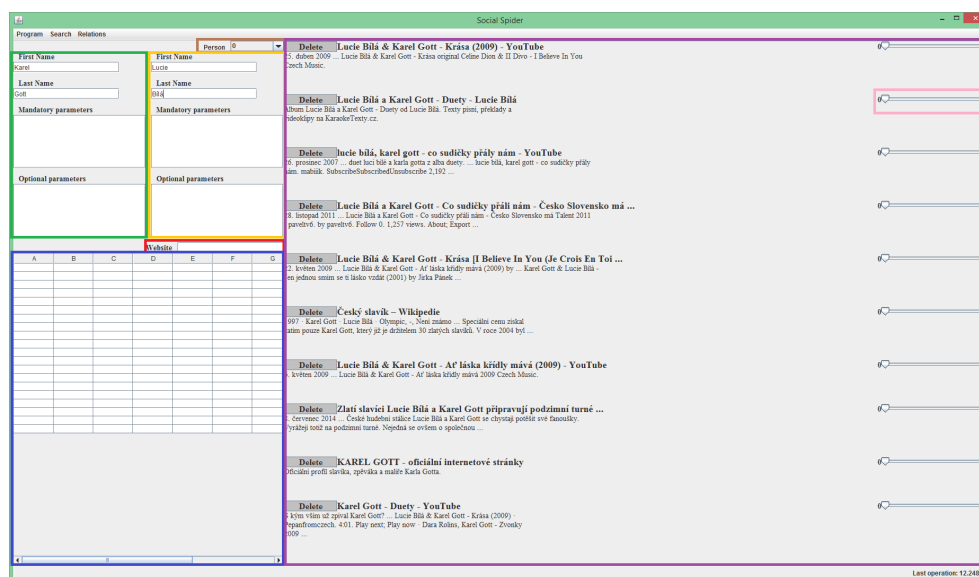
D.3 Okno vyhledávání

Po otevření aplikace se jako první ukáže okno vyhledávání D.1 , které je určené pro vyhledávání lidí a odkazů na články, ve kterých jsou zmíněni. Pokud se nacházíte na jiném okně, lze okno vyhledávání ukázat přes menu Program → Ukaž hledání.

D.3.1 Panel vyhledávání

Panel vyhledávání se nachází vlevo nahoře a dělí se na dvě hlavní části. Nalevo je formulář pro zadávání referenční osoby (na obrázku D.1 vyznačeno zeleně), ke které budete hledat vazby s dalšími lidmi. Tyto další osoby zadáte v okně pravém (na obrázku D.1 vyznačeno žlutě). Při vyhledávání je nejlepší zadat aspoň křestní jméno a příjmení hledané osoby. Moc málo parametrů může vést k irelevantním výsledkům. Na druhou stranu příliš mnoho parametrů může vyústit v situaci, kdy nebude nalezen jediný článek.

D. UŽIVATELSKÁ PŘÍRUČKA



Obrázek D.1: Uživatelská příručka: okno hledání

D.3.1.1 Přidání/odebrání osob v levém formuláři

Manipulace s počtem osob se dělá v menu Hledání → Formulář → Přidat/Odebrat osobu.

D.3.1.2 Překlikávání mezi osobami v levém formuláři

Přepínání mezi jednotlivými osobami levého formuláře lze docílit vybráním ze seznamu boxu, který se nachází přímo nad levým formulářem a je pojmenován Osoba (na obrázku D.1 vyznačeno hnědě).

D.3.1.3 Omezení výsledků na jednu stránku

Pro omezení výsledků hledání na jednu specifickou stránku stačí vyplnit pole pojmenované Stránce nacházející se pod levým formulářem (na obrázku D.1 vyznačeno červeně).

D.3.1.4 Spuštění vyhledávání

Vyhledávání se dělá přese menu Hledání → Hledej.

D.3.2 Panel poznámek

Panel poznámek (na obrázku D.1 vyznačeno modře) se nachází vlevo pod panelem vyhledávání a slouží k zapisování poznámek o lidech přímo do aplikace bez nutnosti překlíkávání mezi okny. Poznámky mají podoby tabulky a dají se

uložit ve formátu CSV, který se snadno importuje do moderních tabulkových procesorů jako je OpenOffice Calc nebo Microsoft Office Excel.

D.3.2.1 Uložení poznámek

Uložení poznámek je možné přes menu Hledání → Poznámky → Uložit poznámky.

D.3.2.2 Vyčištění poznámek

Vymazání všech momentálních poznámek lze provést přes menu Hledání → Poznámky → Smazat poznámky.

D.3.2.3 Přidání/odebrání sloupce/řádku

Manipulace s počtem řádků nebo sloupců se dělá přes menu Hledání → Poznámky → požadovaná akce.

D.3.3 Panel výsledků hledání

V tomto panelu (na obrázku D.1 vyznačeno fialově) se zobrazují výsledky hledání.

D.3.3.1 Vymazání nepotřebného výsledku

Vedle každého výsledku je tlačítko Vymazat pro odstranění daného výsledku.

D.3.3.2 Zobrazení detailu výsledku

Po kliknutí na titul článku se místo panelu výsledků zobrazí detail článku.

D.3.3.3 Ohodnocení výsledku

Článek se hodnotí posuvníkem na pravé straně vedle výsledku (na obrázku D.1 vyznačeno růžově).

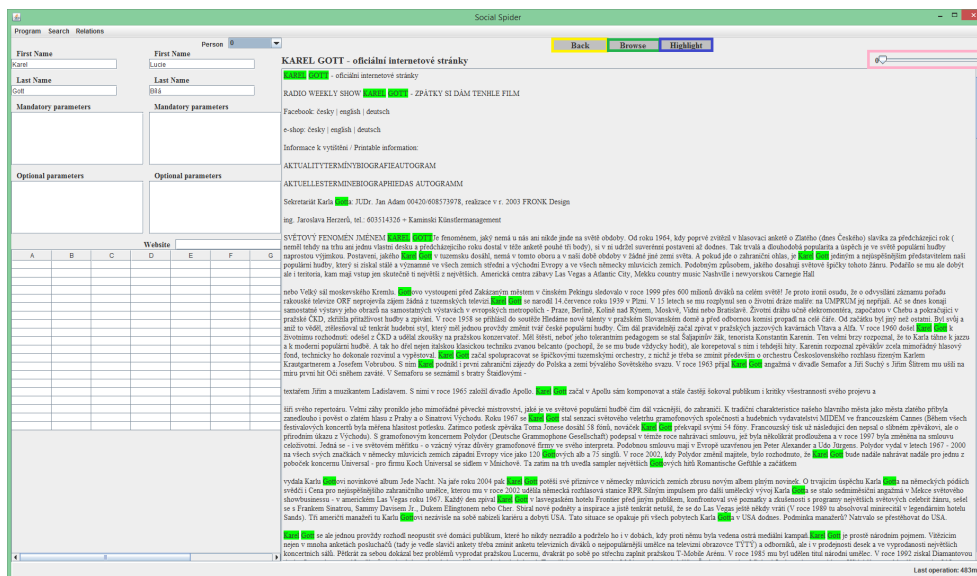
D.3.3.4 Získání více výsledků

Více výsledků je možno načíst přes menu Hledání → Více.

D.3.4 Panel detailu článku

Tento panel D.2 se zobrazí po kliknutí na titulek článku v panelu výsledků hledání.

D. UŽIVATELSKÁ PŘÍRUČKA



Obrázek D.2: Uživatelská příručka: okno detailu článku

D.3.4.1 Vrácení zpět na panel výsledků

Zobrazení panelu výsledku hledání lze dosáhnout kliknutím na tlačítko Zpět (na obrázku D.2 vyznačeno žlutě).

D.3.4.2 Otevření článku v prohlížeči

Zobrazení článku v prohlížeči lze dosáhnout kliknutím na tlačítko Prohlížet (na obrázku D.2 vyznačeno zeleně).

D.3.4.3 Zvýraznění hledaných hesel v článku

Zvýraznění hesel lze dosáhnout kliknutím na tlačítko Zvýraznit (na obrázku D.2 vyznačeno modře).

D.3.4.4 Ohodnocení výsledku

Článek se hodnotí posuvníkem vpravo nahore vedle titulku (na obrázku D.2 vyznačeno růžově).

D.3.5 Další funkce okna hledání

D.3.5.1 Volba vyhledávače

Vyhledávač lze zvolit v menu Hledání → Vyhledávač po kliknutí na příslušný vyhledávač.

D.3.5.2 Pohyb mezi různými vyhledáváními

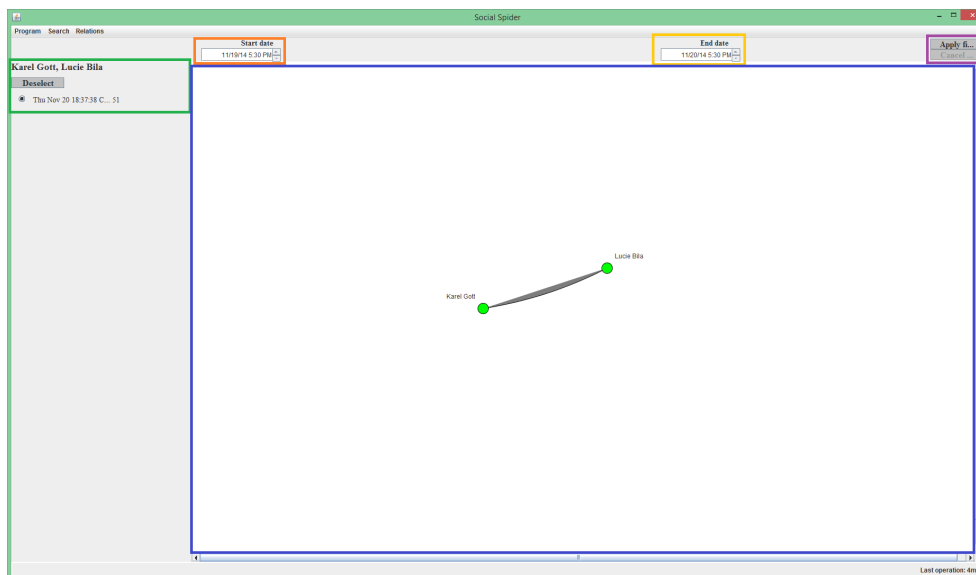
Předcházející a následující vyhledávání lze zobrazit přes menu Hledání → Další/Předcházející.

D.3.5.3 Uložení vazby/vazeb

Pro pracování s vazbami je nutné je prve uložit. To se provádí přes menu Hledání → Vazby → Uložit momentální/Uložit všechny.

D.4 Okno vazeb

Okno vazeb D.3 zobrazíte v menu Program → Ukaž vazby. V tomto okně můžete vytvořit grafy.



Obrázek D.3: Uživatelská příručka: okno vazeb

D.4.1 Panel vazeb

Panel vazeb se nachází vlevo (na obrázku D.3 vyznačeno zeleně). V tomto panelu jsou vazby pojmenovány dle lidí účastnících se vazby, čas pořízení vazby a její síla. Dále tlačítko pro odznačení vazby v grafu.

D.4.1.1 Uložení vazeb na počítač

Vazby se dají uložit na počítač skrz menu Vazby → Uložit vazby.

D.4.1.2 Načtení vazeb z počítače

Vazby se dají načíst z počítače skrz menu vazby → Načíst vazby.

D.4.1.3 Vyčištění panelu vazeb

Panel vazeb lze kompletně vyčistit (vazby uložené na počítači zůstanou netknuty) skrz menu vazby → Vyčistit vazby.

D.4.2 Panel filtru

Pokud máte pro všechny vazby mezi osobami příliš mnoho záznamů a hledáte určité datum, lze toho dosáhnout zapnutím filtru. Panel filtru se nachází nahoře nad panelem vazeb a panelem grafu.

D.4.2.1 Nastavení filtru

Vyplněním počátečního data (na obrázku D.3 vyznačeno oranžově) a koncového data (na obrázku D.3 vyznačeno žlutě) je filtr připraven k zapnutí.

D.4.2.2 Zapnutí/Zrušení filtru

Pro zapnutí/zrušení filtru stiskněte tlačítko Zapnout/Vypnout filtr (na obrázku D.3 vyznačeno fialově, horní/dolní tlačítko).

D.4.3 Panel grafu

V tomto panelu (na obrázku D.3 vyznačeno modře) se ukazuje sestavovaný graf.

D.4.3.1 Přidání vazby do grafu

Přidání vazby lze dosáhnout kliknutím na danou vazbu v panelu vazeb.

D.4.3.2 Odstranění vazby z grafu

Odstranění vazby lze dosáhnout kliknutím na tlačítko Odznačit u dané vazby v panelu vazeb.

D.4.3.3 Uložení obrázku grafu

Uložení grafu jako obrázek lze dosáhnout přes menu Vazby → Uložit obrázek grafu. Obrázek lze uložit ve formátech BMP, JPG, JPEG, PNG a GIF.

D.4.4 Další funkce vazeb

D.4.4.1 Označení všech vazeb

Přes menu Vazby → Označit vše lze označit první relaci v každé načtené vazbě.

D.4.4.2 Odoznačení všech vazeb.

Přes menu vazby → Odznačit vše lze odznačit vše a tím vyčistit graf.

D.4.4.3 Zapnutí myši na vybírání

Myš pro vybírání a chytání uzlů a hran lze zapnout skrz menu Vazby → Myš → Vybírací.

D.4.4.4 Zapnutí myši na transformace

Myš pro otáčení, zvětšování a pohyb grafu lze zapnout skrz menu Vazby → Myš → Transformační.

D.5 Vypnutí programu

Program lze vypnout klasicky vypnutím okna jako každá jiná aplikace nebo taky přes menu Program → Ukončit. Nezapomeňte si všechny vazby řádně uložit před vypnutím programu, jinak budou ztraceny.

Literatura

- [1] JSON/Atom Custom Search API: Introduction. [online]. [cit. 2014-11-28]. Dostupné z: <https://developers.google.com/custom-search/json-api/v1/introduction>
- [2] Custom Search: Creating CSE [online]. [cit. 2014-11-28]. Dostupné z: <https://www.google.com/cse/create/new>
- [3] Google APIs Explorer: Custom Search. [online]. [cit. 2014-11-28]. Dostupné z: <https://developers.google.com/apis-explorer/#p/customsearch/v1/search.cse.list>
- [4] CSE: list: Custom Search. [online]. [cit. 2014-11-28]. Dostupné z: <https://developers.google.com/custom-search/json-api/v1/reference/cse/list>
- [5] CSE HTTP Request site. [online]. [cit. 2014-11-28]. Dostupné z: <https://www.googleapis.com/customsearch/v1>
- [6] Custom Search vs Google.com: Custom Search Help. [online]. [cit. 2014-11-28]. Dostupné z: <https://support.google.com/customsearch/answer/70392?hl=en>
- [7] Custom Search: Google Developers. [online]. [cit. 2014-11-28]. Dostupné z: <https://developers.google.com/custom-search/>
- [8] Google Gson. [online]. [cit. 2014-11-28]. Dostupné z: <https://code.google.com/p/google-gson/>
- [9] JUNG: Java Universal Network/Graph Framework. [online]. [cit. 2014-11-28]. Dostupné z: <http://jung.sourceforge.net/>
- [10] Google: Our history in depth. [online]. [cit. 2014-11-30]. Dostupné z: <http://www.google.com/about/company/history/>.

LITERATURA

- [11] Seznam.cz: O společnosti Seznam.cz. [online]. [cit. 2014-11-30]. Dostupné z: <http://onas.seznam.cz/cz/o-firme/historie-firmy/>
- [12] Return Now | Resources: 2013 Search Engine Market Share By Country. [online]. [cit. 2014-11-30]. Dostupné z: <http://returnnonnow.com/internet-marketing-resources/2013-search-engine-market-share-by-country/>