Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16, Bratislava 4

Deep Search
Dokumentácia k tímovému projektu
(Inžinierske dielo)

Vedúci tímu: Ing. Nadežda Andrejčíkova, PhD.
Členovia tímu: Bc. Peter Berta, Bc. Matej Adamov, Bc. Michal Krempaský, Bc. Oliver Macko, Bc. Bronislava Pečíková
Akademický rok : 2017/2018

Obsah

1. Úvod

Informačná explózia so sebou prináša aj viacero problémov. Napriek tomu, že v dnešnej dobe si nemožno sťažovať na nedostatok informácií, máme často problém nájsť to, čo práve potrebujeme. Väčšina dokumentov je totiž v neštruktúrovanej podobe a získať z nich informácie typu: kto v danom období pôsobil v určitom regióne je prakticky nemožné. Fulltextové vyhľadávanie má jeden vážny nedostatok a to, že nezohľadňuje sémantiku daných kľúčových slov. V našom projekte sa zameriavame na spracovanie prirodzeného jazyka a stanovili sme si pomerne ambiciózny cieľ, ktorým je extrakcia štruktúrovaných dát z neštruktúrovaného textu so zachytením ich významu. Pričom zameriavať sa budeme najmä na životopisy a iné dokumenty, z ktorých budeme môcť extrahovať informácie o tom kde a kedy dané osoby študovali, prípadne pôsobili. Pokúsime sa tiež z textov získať vzťahy typu: kolega alebo spolužiak. Našou úlohou bude teda rozpoznávať a pokiaľ to bude možné aj jednoznačne identifikovať , entity typu osoba, korporácia , geografické jednotka, datácia a zároveň identifikovať udalosť štúdium, prípadne pôsobenie a v rámci nich vzťahy medzi týmito entitami. Cieľom je tieto údaje uložiť v štruktúrovanej podobe tak, aby bolo možné v nich vyhľadávať a získať informáciu o tom kto a kedy v danom mieste študoval, s kým sa mohol poznať a vytvárať tak aj virtuálne komunity napr. pre určité zameranie.

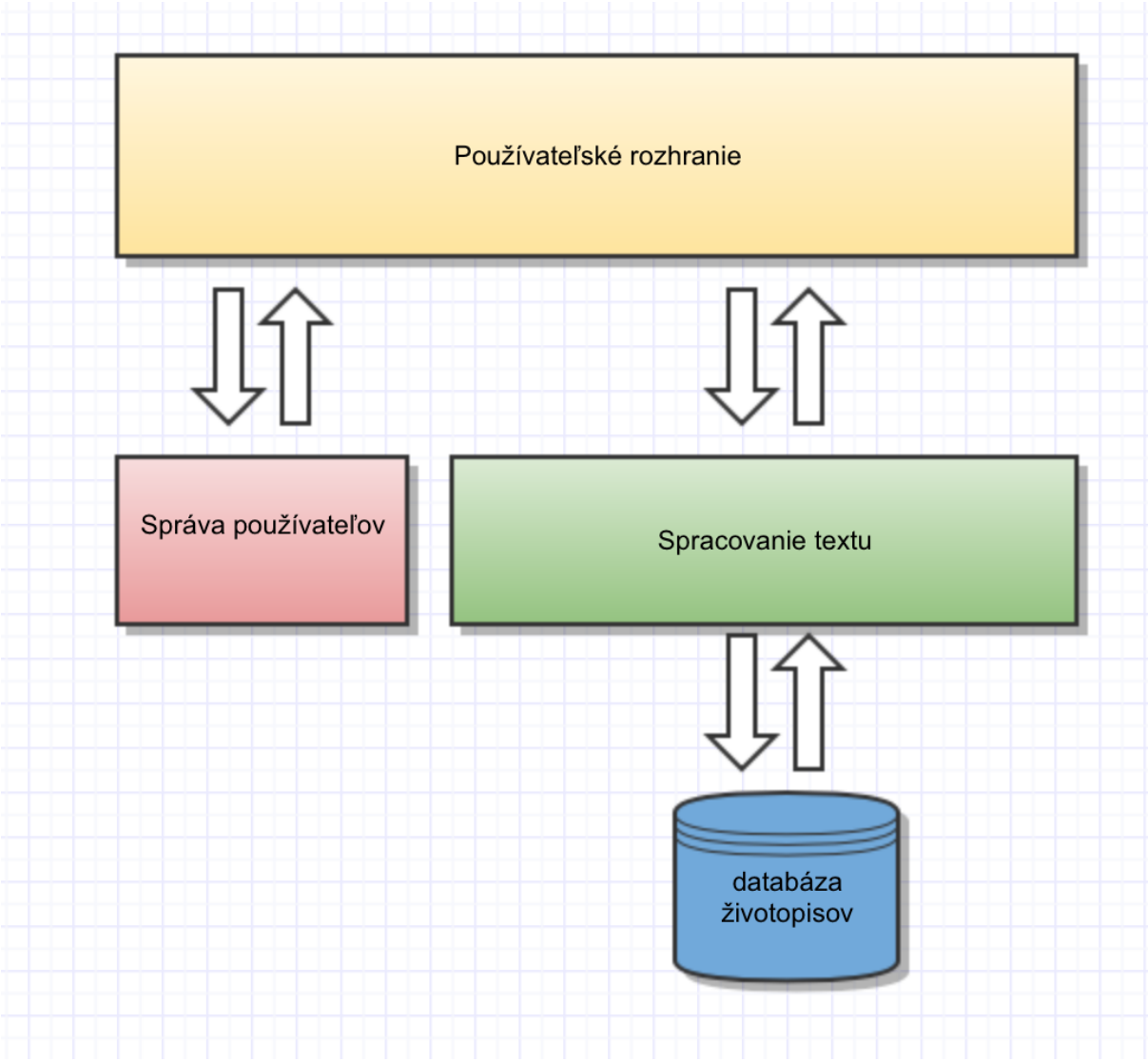
2. Globálne ciele projektu

Cieľom tohto semestra je vytvorenie databázy na základe neštruktúrovaných životopisov. Je potrebné identifikovať nasledovné polia:

- Narodenie – dátum a miesto
- Úmrtie – dátum a miesto
- Štúdium – dátum, miesto, profesori, spolužiaci

V ideálnom prípade by bolo vhodné taktiež vizualizovať súvislosti medzi osobnosťami pomocou grafu. Jednotlivé vrcholy budú predstavovať osobnosti a hrany budú predstavovať súvislosti medzi nimi.

3. Celkový pohľad na systém

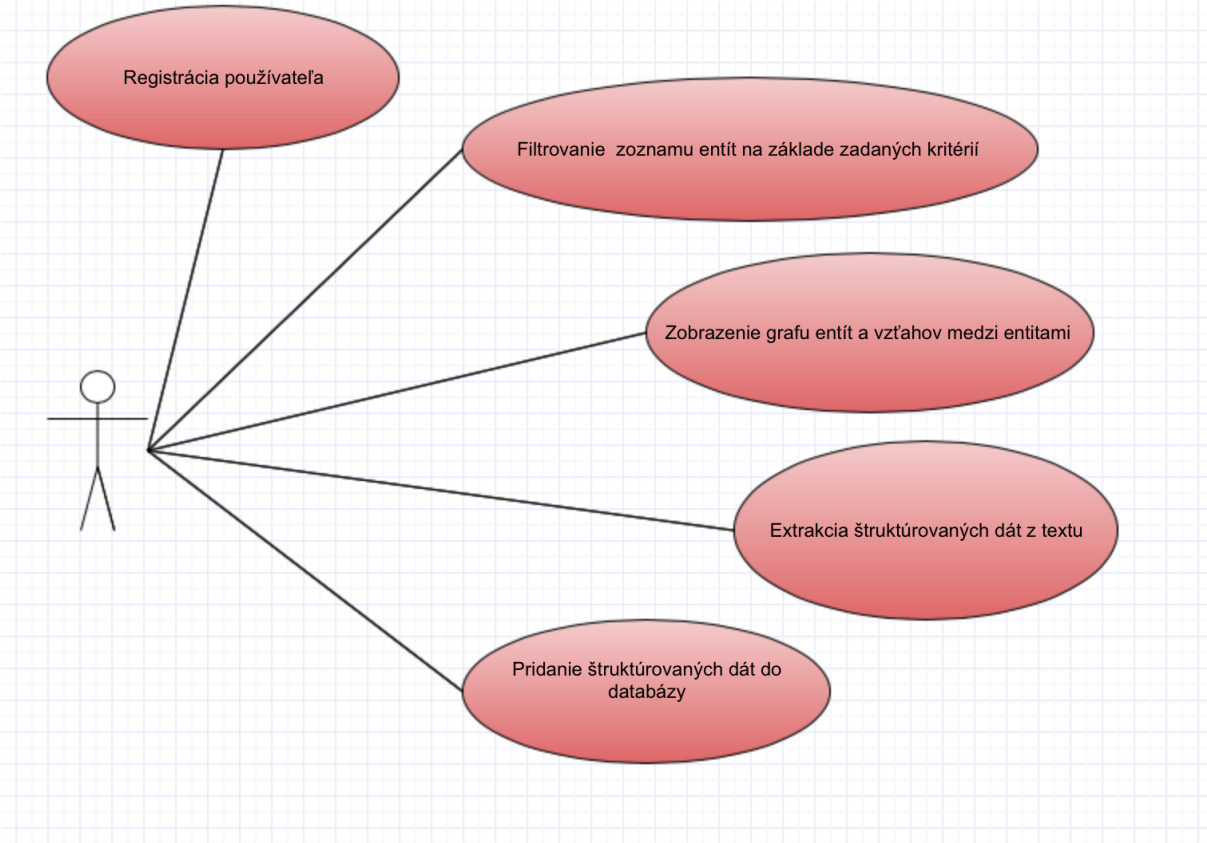


Obr. 1: Moduly systému

Systém pozostáva z troch modulov:

- používateľské rozhranie
- správa používateľov
- spracovanie textu

Používateľské rozhranie bude implementované prostredníctvom webovej aplikácie a bude ponúkať funkcionality ako sú vyhľadávanie v štruktúrovaných dátach, zobrazenie grafu vzťahov, pridanie životopisu… Správa používateľov bude riešiť prihlásenie, odhlásenie a registráciu používateľa. Modul spracovanie textu extrahuje zo životopisov údaje o štúdiu a zamestnaní a ukladá ich v štruktúrovanej podobe.



Obr. 2 diagram prípadov použitia

3.1. Modul analýzy vstupov

V rámci analýzy vstupných textov sme v druhom šprinte identifikovali kvalifikátory pre štúdium a zamestnanie. Manuálne sme spracovali okolo 100 životopisov, v ktorých sme zachytili rôzne vetné skladby v okolí identifikovaných kvalifikátorov. Z tejto množiny sme vytvorili univerzálnu štruktúru pre zachytenie výskytu kvalifikátorov spolu s pomenovanými entitami v ich okolí.

Pri zaznamenávaní pomenovaných entít sme použili dvojstupňovú kvalifikáciu pomenovaných entít pre český jazyk z článku “Czech Named Entity Corpus and SVM-based Recognizer”[1].

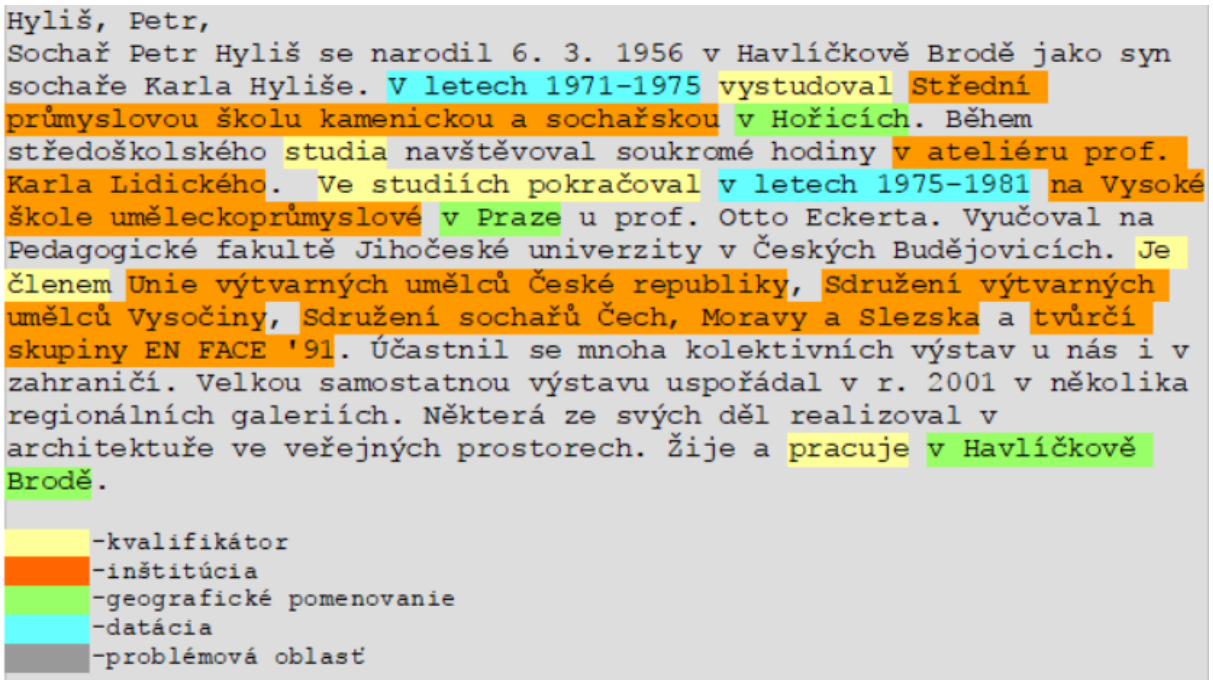
Skratky znázorňujú entity ktoré sa môžu nachádzať v okolí kvalifikátora, pričom čísla znázorňujú interval vzdialeností tejto entity od pozície kvalifikátora.

[1] KRAVALOVÁ, Jana; ŽABOKRTSKÝ, Zdeněk. Czech named entity corpus and SVM-based recognizer. In: Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration. Association for Computational Linguistics, 2009. p. 194-201

Štúdium	Zamestnanie
studovat <IC -1 +2> <GU +2> <TY -1 +2> vystudovat <IC -1 +2> <GU +2> <TY -1 +2> absolvovat <IC +1> <GU +2> <TY -1> absolvováni <IC +1> <GU> <TY> být žák <IC -1> <GU -2 +3> <TY -3 +4> studium <IC +2> <GU +1 +3> <TY -1 +1> nastoupit <IC +1> <GU +2> <TY -1> vzdeláni <IC +1> <GU +3> <TY +2>	pracovat <IC +2> <GU +2> <TY -1> být pracovník <IC> <GU> <TY> <IF +1> působit <IC -1 +1> <GU +3> <TY -1 +1> <IF> člen <IC +1> <GU +2> <TY -1> <IF> zakladatel <IC +1> <GU -1 +1> <TY> <IF> spoluzakladatel <IC +1> <GU +2> <TY +3> <IF> zamestnanec <IC> <GU> <TY -1> <IF +1>

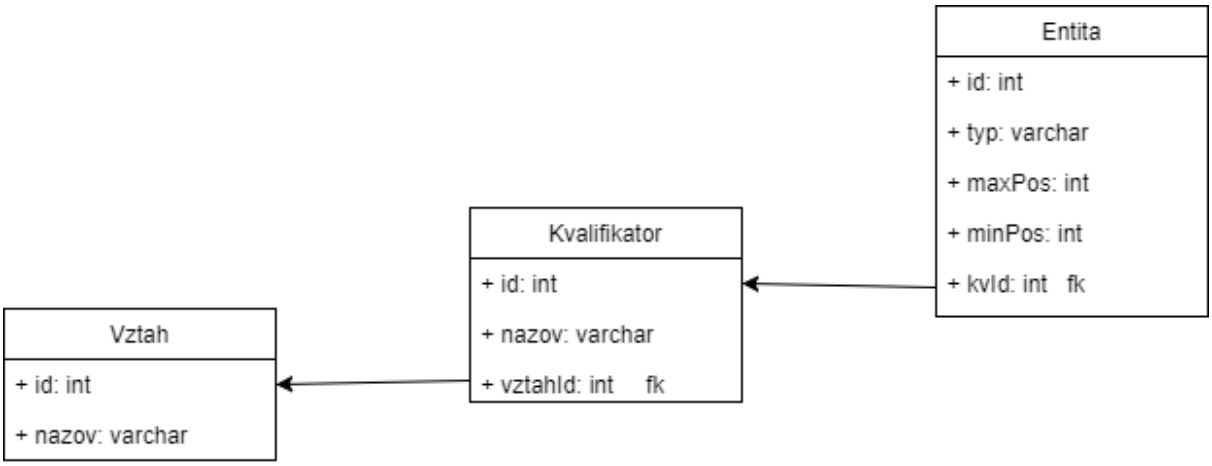
Tab. 1. Kvalifikátory

Túto štruktúru sme manuálne simulovali nad množinou nových životopisov a výsledky sme v nich farebne vizualizovali. Táto simulácia dosiahla 80% úspešnosť v rámci zachytenia kvalifikátora definovanou štruktúrou.



Obr. 3: Simulácia

Vzhľadom na výsledky simulácie tejto štruktúry, sme navrhli model relačnej databázy, ktorý sme následne implementovali, a naplnili doposiaľ identifikovanými kvalifikátormi.



Obr. 4: Relačný model

3.1.1. Výber databázy pre ukladanie neštruktúrovaných dát

Pre ukladanie neštruktúrovaných dát sme zvažovali technológie *ElasticSearch*, *Postgre* a ukladanie v klasických textových súboroch.

3.1.2. Výber databázy pre ukladanie štruktúrovaných dát

Zvažovali sme *Neo4j* a *Cayley*, *Cache*, objektové databázy.

Štruktúrované dáta získane zo spracovania životopisov je potrebné v nejakej forme ukladať. K dispozícii sú grafové databázy, objektové alebo klasické relačné databázy. Výhodou grafových databáz je, že vedia dobre reprezentovať vzťahy medzi objektami, čo by bolo v našom prípade užitočné pre ukladanie vzťahov ako sú kolega, zamestnanec, spolužiak …. Z grafových databáz sme zvažovali najmä *Neo4j* a *Cayley*.

3.1.2.1. Neo4J

V dnešnej dobe asi jedna z najpopulárnejších *opensource* databáz vyvíjaná v jazyku JAVA. Vďaka dobrému *toolingu* pre analýzu a vizualizáciu grafov má dobré využitie v telekomunikačných službách, napríklad pri detekcii *fraudov*, sieťových a IT operáciách atď.

Výhody:

- dlho žijúci projekt s veľkou komunitou
- klient v mnohých jazykoch
- *end to end control* - napr. nie je závislá na externých úložiskách ako *Cayley*
- *RESTful api*

Nevýhody:

- nie je *cloud based* (nemusí sa vždy jednať o nevýhodu)
- nepodporuje *shardovanie*
- nie je zadarmo pre komerčné použitie

3.1.2.2. Cayley

Cayley je neoficiálny *opensource* produkt od spoločnosti *Google* vyvíjaný v jazyku *Go*, inšpirovaný grafovou databázou *Freebase*. Cieľom databázy *Cayley* je vytvorenie tzv. *toolboxu* pre pracovanie s prepojenými resp. grafovými dátami (sémantický web, sociálne sieťe, atď.).

Výhody:

- podporuje viac *query* jazykov ako napr. *Gizmo*, *GraphQL* alebo *MLQ*
- zabudovaný *query* editor a vizualizér
- *RESTful API*
- podpora viacerých *backend* úložísk: kvl (*Bolt*, *LevelDB*), *NoSQL* (*MongoDB*), *SQL* (*Postgres*, *CocroachDB*, *MySQL*)
- modulárny design (jednoduché rozširovanie s novými jazykmi a *backend* riešeniami)
- veľký dôraz na rýchlosť

Nevýhody:

- pomerne mladý projekt - ku dnešnému dňu má iba 1130 *commitov* a 59 prispievateľov
- nemá úplne kompletnú dokumentáciu
- *storage* rieši v externých databázach (môže byť aj výhoda)

3.2. Modul predspracovania textu

3.2.1. Požiadavky

Na to aby sme vedeli získavať z textu poznatky, ktoré by umožnili pochopenie kontextu vety, potrebujeme vytvoriť niekoľko nástrojov, ktoré umožnia zmysluplné dopytovanie nad textom.

Pre tento modul vznikli požiadavky na základné spracovanie neštruktúrovaného textu životopisov do viet a slov. Pre tieto účely potrebujeme niekoľko metód, ktoré by vedeli následne spracovávať text a vytvárať množinu poznatkov o slovách vo vetách.

- Tokenizácia - text potrebujeme vo forme tokenov - samostatných slov
- Lematizácia - potrebujeme určiť základný tvar slov aby sme mohli vyhľadávať kvalifikátory
- Morfologické značky a slovné druhy- potrebujeme zistiť pre každé slovo tieto značky a určiť slovný druh aby sme vedeli lepšie identifikovať entity

3.2.2. Analýza prístupov cez jednotlivé služby

Pre väčšinu jazykov existuje podpora prístupov k slovníkom obsahujúcim tieto vedomosti na úrovni webových služieb alebo voľne dostupných knižníc.

3.2.2.1. NLTK

Natural Language Tool Kit - je knižnica v *Pythone*, primárne robená pre účely spracovávania anglického textu. Obsahuje v sebe mnoho slovníkov a funkcionalít, ktoré vedia generovať tagy, lemy, identifikovať slovné druhy alebo aj tokenizovať. Pre riešenie rozdeľovania slov vyhovuje z tohto slovníka najviac podpora tokenizácie textu pre rôzne jazyky.

3.2.2.2. Lemmagen

Je český voľne dostupný morfologický slovník a značkovač určený pre spracovávanie českého textu. Väčšina služieb tejto knižnice vykonáva morfologickú analýzu nad vloženým textom a vie vrátiť morfologické značky, tokeny a lematizované formy slov na základe natrénovaného lingvistického modelu.

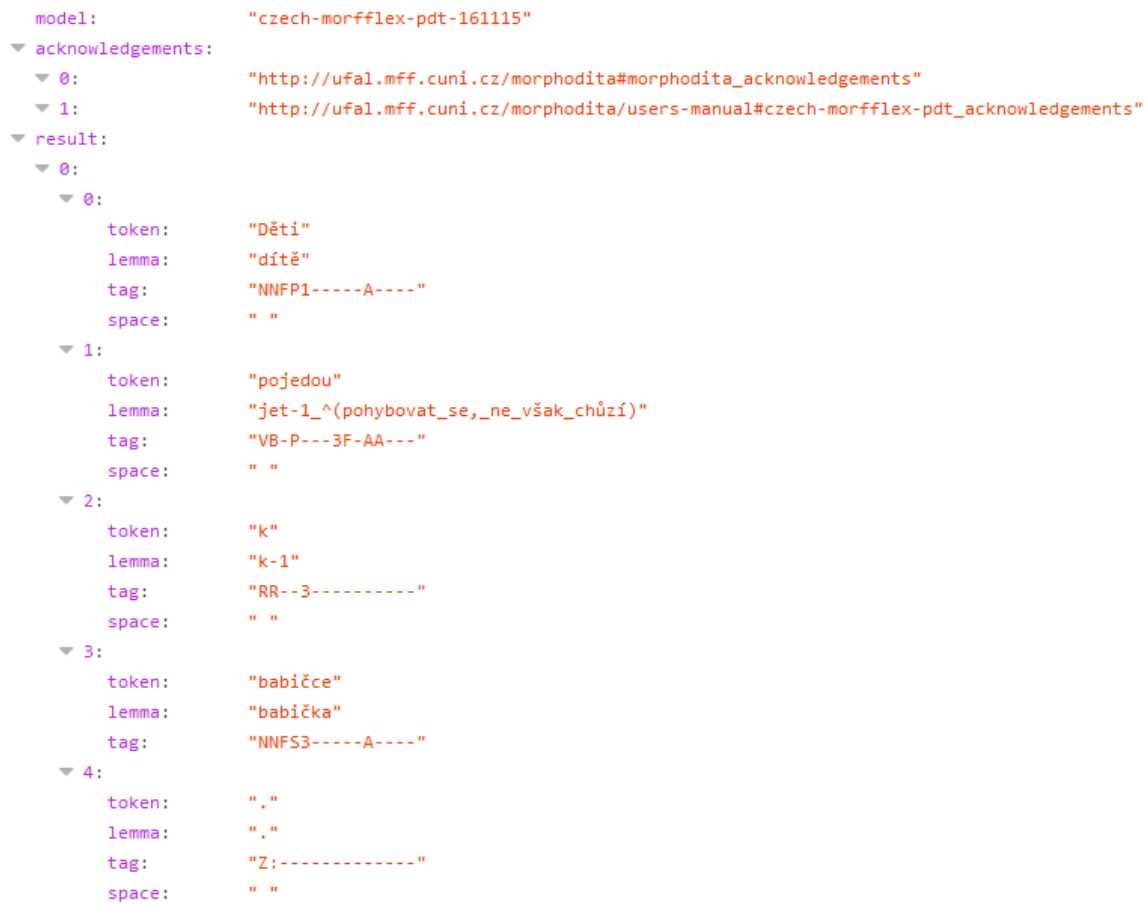
Ponúka webové rozhrania s REST API ako aj knižnicu v C++ na lokálne použitie. Príklad použitia na webe je zobrazený na obrázku 1.

Obr. 5: Príklad použitia na webe

3.2.2.3. MorphoDita

Táto webová služba je dostupná aj cez rozhranie REST API, ktoré nám umožňuje na základe metód *tokenize*, *analyze* a *tag* vrátiť nad poslaným textom výstup vo forme JSON typu kde je každé slovo tokenizované s pridelenou *lemou* a morfologickou značkou.

Obr. 6: Príklad použitia metódy *analyze*



Obr. 7: Príklad použitia metódy tag

3.2.2.4. KonText

Je to rozhranie, ktoré umožňuje pristupovať ku slovníkom alebo korpusom českého jazyka. Každý z týchto korpusov reprezentuje vedeckú prácu výskumu akademikov českých univerzít. Tie následne poskytujú niekoľko operácií nad svojimi korpusmi. Nevýhoda tohto prístupu je, že nevieme nájsť knižnicu alebo prístup, ktorý by nám jednoducho umožnil lokálne sa dopytovať nad týmto rozhraním. Niektoré operácie dokonca vyžadujú prihlasovanie sa na webovej stránke. Tento koncept nateraz berieme ako nevýhodu a skúsime nájsť alternatívne prístupy.

3.2.2.5. Text.fiit

Je to stránka výskumu fakulty FIIT, ktorá sa zaoberá spracovaním slovenského jazyka. Na tejto stránke máme na výber postačujúce metódy tokenizácie, lematizácie, morfologických značiek a určovania slovných druhov. Tieto metódy sú prístupné vo forme webových služieb a tento prístup berieme ako vhodný pre spracovávanie slovenského jazyka.

3.2.3. Návrh

Z analýzy vyplýva možnosť použitia viacerých prístupov. Najideálnejšie je použitie morfodity na účely spracovania textu. Keďže ale odhadujeme rôznu postupnosť procesov spracovávania textu, je teda vhodné navrhnúť volanie týchto služieb v samostatných pod moduloch. To nám umožní vyššiu flexibilitu pri definovaní procesu spracovávania textu, kde môžeme implementovať rôzne adaptéri upravujúce vstupy a výstupy. Pre tento účel sa javí vhodné použitie architektonického štýlu dátovody a filtre, ktorý umožňuje rýchlu adaptáciu alebo aj konfiguráciu procesu spracovávania textu.

3.2.4. Implementácia

Realizácia návrhu prebehla v duchu architektonického štýlu kde sme vytvorili spracovanie vo forme vstupných dát ako "slovník" a následne aplikujeme metódu, ktorá sa chová ako filter a spracuje text. Tento výstup následne použije pre vstup nasledujúceho filtra. Tento proces vieme konfigurovať.

Implementovali sme aj prístup k rozhraniu MorphoDity, kde je implementácia služieb volajúcich REST API, ktoré vrátia spracovaný JSON vo forme ako je vidieť v analýze.

Implementovali sme aj prototyp pre dopytovanie nad textom v databáze *elasticsearch*. V tomto prototype sme použili plugin *Lemmagen*, ktorý umožňuje dopytovanie nad textom v lematizovanej forme. Tento prístup je vhodný na identifikáciu životopisov obsahujúcich kvalifikátory čo nám otvára prístup vytvorenia schémy uloženia životopisov v kombinácii s výstupom *MorphoDity*.

Nakoľko pre naše účely by nepostačoval automaticky vygenerovaný *mapping* vytvorili sme index a nastavili *mapping* a *analyzer* nasledovným spôsobom:

```
PUT /cv_cvicny
{
  "settings": {
    "index": {
      "analysis": {
        "filter": {
          "lemmagen_filter_en": {
            "type": "lemmagen",
            "lexicon": "cs"
          }
        },
        "analyzer": {
          "lemmagen_lowercase_en": {
            "type": "custom",
            "tokenizer": "uax_url_email",
            "filter": [
              "lemmagen_filter_en",
              "lowercase"
            ]
          }
        }
      }
    }
  },
  "mappings": {
    "cv": {
      "properties": {
        "text": {
          "type": "text",
          "analyzer": "lemmagen_lowercase_en"
        }
      }
    }
  }
}
```

Obr. 8: Mapping

Po vytvorení indexu a nastavenia *mappingu* a *analyzera* sme do indexu pridali dokumenty a následne sme sa mohli dopytovať dát. Príklad *query*, ktorá vyhľadáva životopisy obsahujúce kvalifikátor "študovať" je uvedený nižšie:

```
GET /cv/_search?pretty
{
  "query": {
    "bool": {
      "must": [
        {
          "match": {
            "text": "studovat"
          }
        },
        {
          "match_phrase": {
            "text": "studovat na"
          }
        }
      ]
    }
  },
  "highlight": {
    "fields": {
      "text": {"fragment_size" : 151}
    }
  },
  "_source": [
    "name"
  ],

  "size": 20
}
```

Obr. 9: Query

Príklad výstupu z uvedenej query:


```
        "_source": {
          "name": "Vokálek, Josef"
        },
        "highlight": {
          "text": [
            "Josef Vokálek se narodil  3. 1. 1887 v Čakovicích, zemřel 30. 6. 1969 v
Sedlci. Malíř a profesor v Praze. <em>Studoval</em> <em>na</em> pražské akademii
výtvarných",
            " umění u prof. Krattnera, v r. 1919 <em>studoval</em> <em>na</em> École
des Beaux Arts v Paříži. Věnoval se krajinářství i figurální malbě, preferoval
techniku pastelu."
          ]
        }
```


Obr. 10: Výstup z query

3.3. Modul správa používateľov

Naša aplikácia nezahŕňa len vyhľadávanie nad štruktúrovanými dátami, ale aj samotné pridávanie nových textov na spracovanie naším systémom. Potrebujeme teda zabezpečiť, aby sa nám tam nedostali nesprávne a nepravdivé dáta. Toto dosiahneme tým, že zavedieme jednoduchú správu používateľov.

Definovali sme si 3 základné používateľské role:

- Admin

  Admin je používateľ, ktorý spravuje systém. Prideľuje práva registrovaným používateľom a validuje registráciu.

- Anonymný používateľ

  Anonymný používateľ je používateľ systému, ktorý si chce vyhľadať nejakú konkrétnu informáciu v už spracovaných štruktúrovaných dátach. Takýto používateľ nemá práva na vkladanie vlastných dát do systému na spracovanie.

- Registrovaný používateľ

  Registrovaný používateľ je taký používateľ, ktorý sa zaregistroval do systému, bola mu schválená registrácia adminom a následne sa prihlásil do systému. Takto prihlásený používateľ, ktorému boli nastavené práva adminom, dokáže do systému vložiť vlastné dáta, ktoré náš systém následne spracuje do štruktúrovanej podoby.

Registrácia používateľa prebieha pomocou jednoduchého formulára kde používateľ vyplní svoje meno, priezvisko, emailovú adresu a zvolí si používateľské meno a prihlasovacie heslo. Vyplnené dáta systém spracuje a vytvorí používateľovi záznam v databáze používateľov. Admin používateľovi registráciu buď schváli alebo zamietne. Pri prihlasovaní používateľ vyplní používateľské meno a heslo, systém skontroluje či existuje registrovaný používateľ s danými prihlasovacími údajmi a na základe výsledku vyhodnotí prihlásenie ako úspešné alebo neúspešné.

3.4. Modul používateľské rozhranie

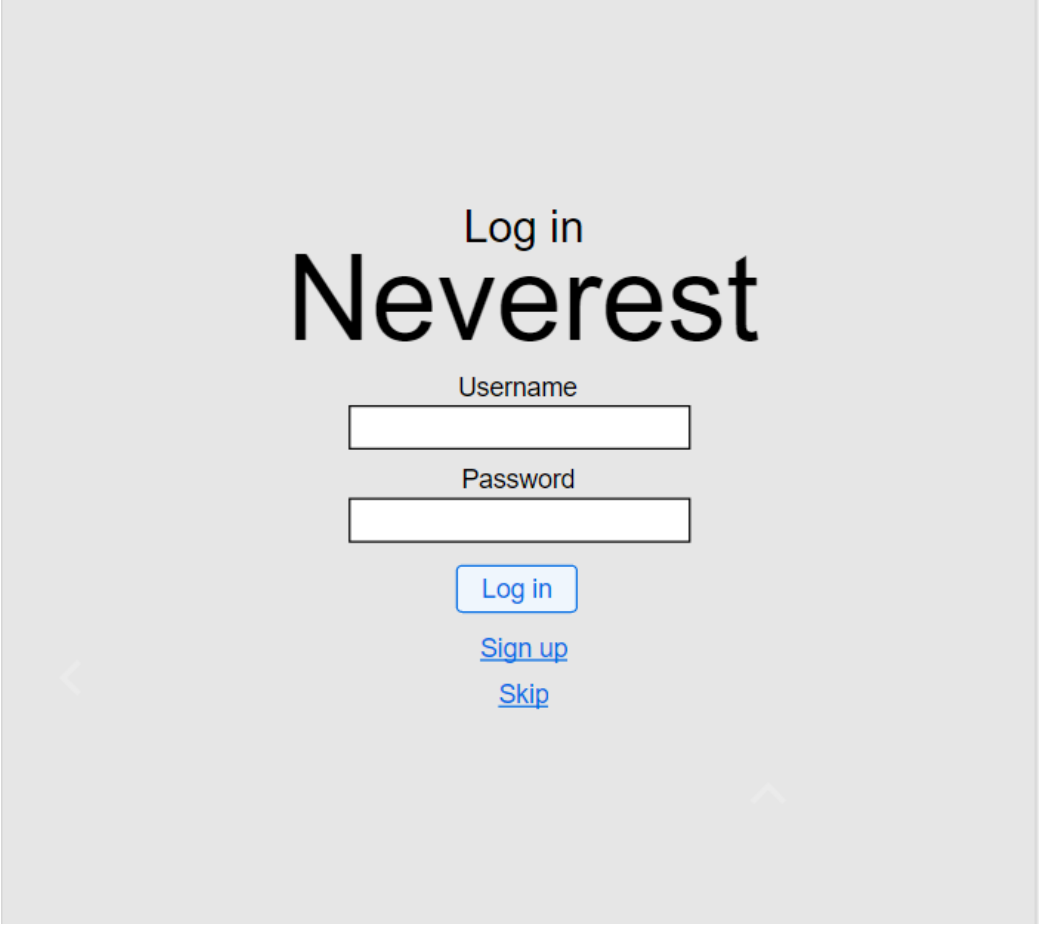
Systém bude sprístupnený používateľom v podobe webového rozhrania. Toto rozhranie bude ponúkať používateľom funkcionalitu medzi ktorú patri:

- vyhľadávanie informácii o zadaných entitách
- filtrovanie zoznamu entít na základe zadaných kritérií
- spracovanie a zobrazenie vlastného textu do štruktúrovanej podoby
- pridanie štruktúrovaných dát do databázy

Každá funkcionalita je dostupná na základe typu účtu pod ktorým je používateľ prihlásený. Preto je potrebné vytvorenie obrazoviek na prihlasovanie a registráciu, ktoré budú spolu s modulom správy používateľov poskytovať možnosť prihlasovania a vedenia účtu.

3.4.1. Prihlásenie

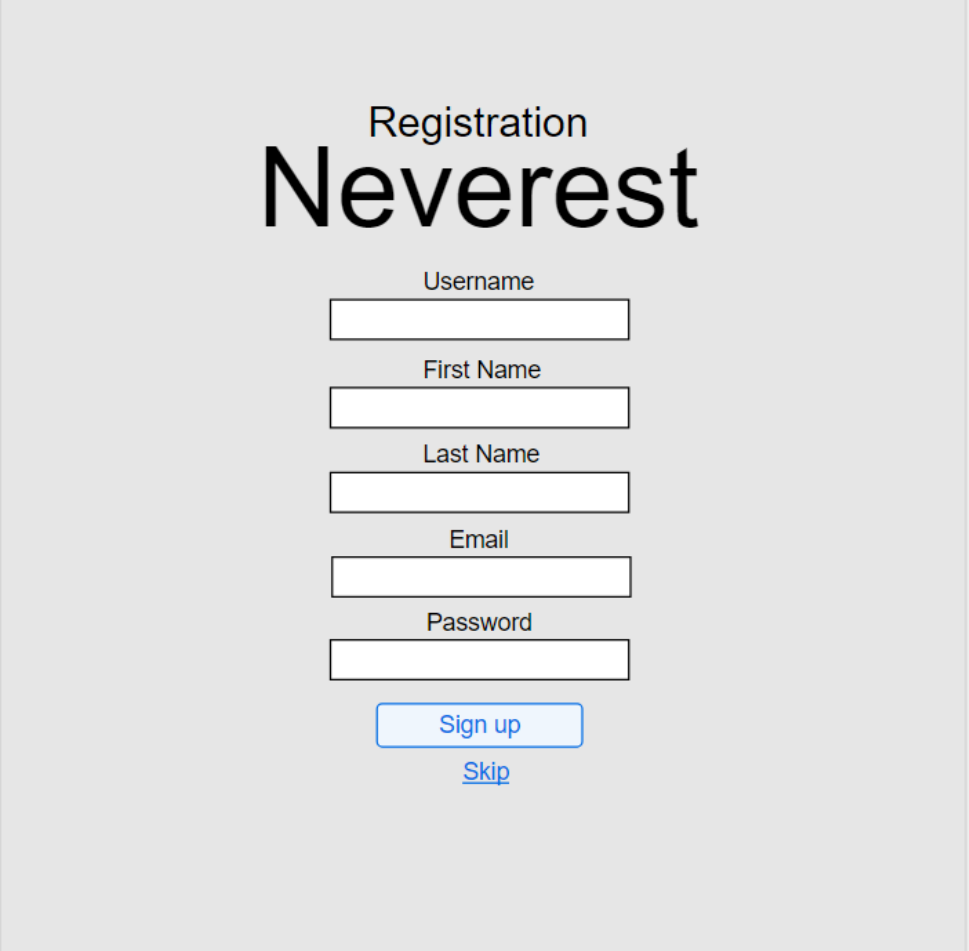
V obrazovke prihlasovania používateľ zadá svoje používateľské meno a heslo a pomocou tlačidla „Log in“ sa prihlási do systému. Nižšie pod týmto tlačidlom je dostupná registrácia nových používateľov, a taktiež preskočenie prihlásenia pre anonymných používateľov, ktorým je následne poskytnutá obmedzená funkcionalita systému.



Obr. 11: Obrazovka prihlásenia

3.4.2. Registrácia

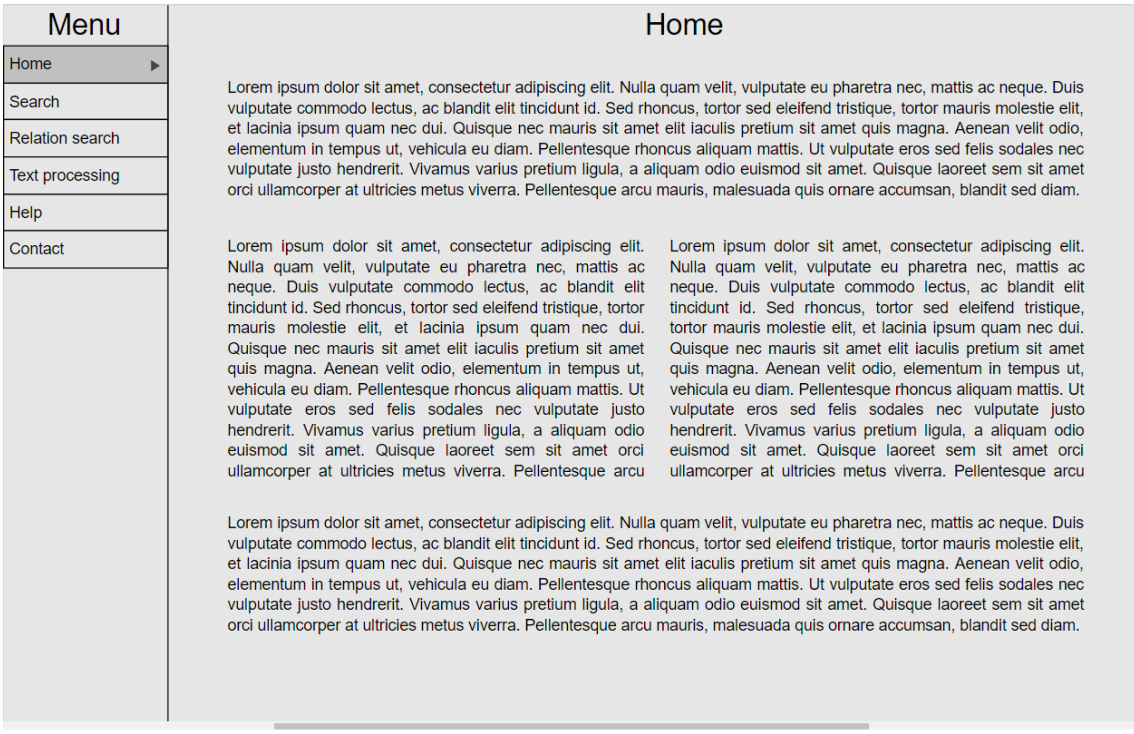
Registrácia používateľa sa vykonáva na základe zadaného používateľského mena, hesla a emailu. Používateľ do formulára vyplní tieto údaje, a odošle registráciu tlačidlom „Sign up“. V prípade, že si používateľ registráciu rozmyslí, má možnosť ju preskočiť a pristúpiť k systému anonymne s obmedzenou funkcionalitou. Na tento úkon slúži odkaz „Skip“.



Obr. 12: Obrazovka registrácie

3.4.3. Hlavná obrazovka

Hlavná obrazovka je vertikálne rozdelená na dva panely. V ľavom paneli sa nachádza menu na vyber požadovaných funkcií. Tie sa následne zobrazujú v pravom paneli.



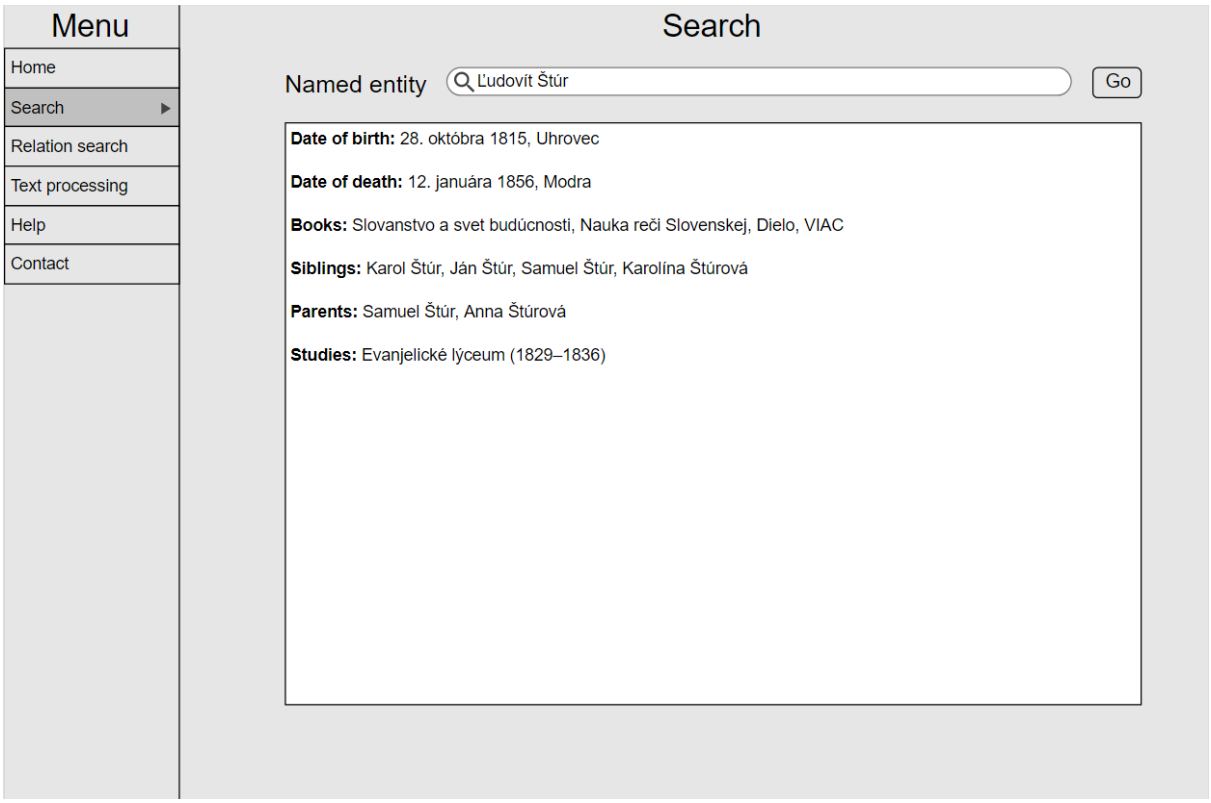
Obr. 13: Hlavná obrazovka

3.4.4. Vyhľadávanie

Funkcionalita vyhľadávania je v podobe jednoduchého vyhľadávacieho poľa v databáze na základe kľúčových slov zadaných používateľom. Používateľovi sa zobrazí zoznam výsledkov, v ktorom si následne môže zobraziť detailne informácie o požadovaných entitách.

3.4.5. Vyhľadávanie vzťahov

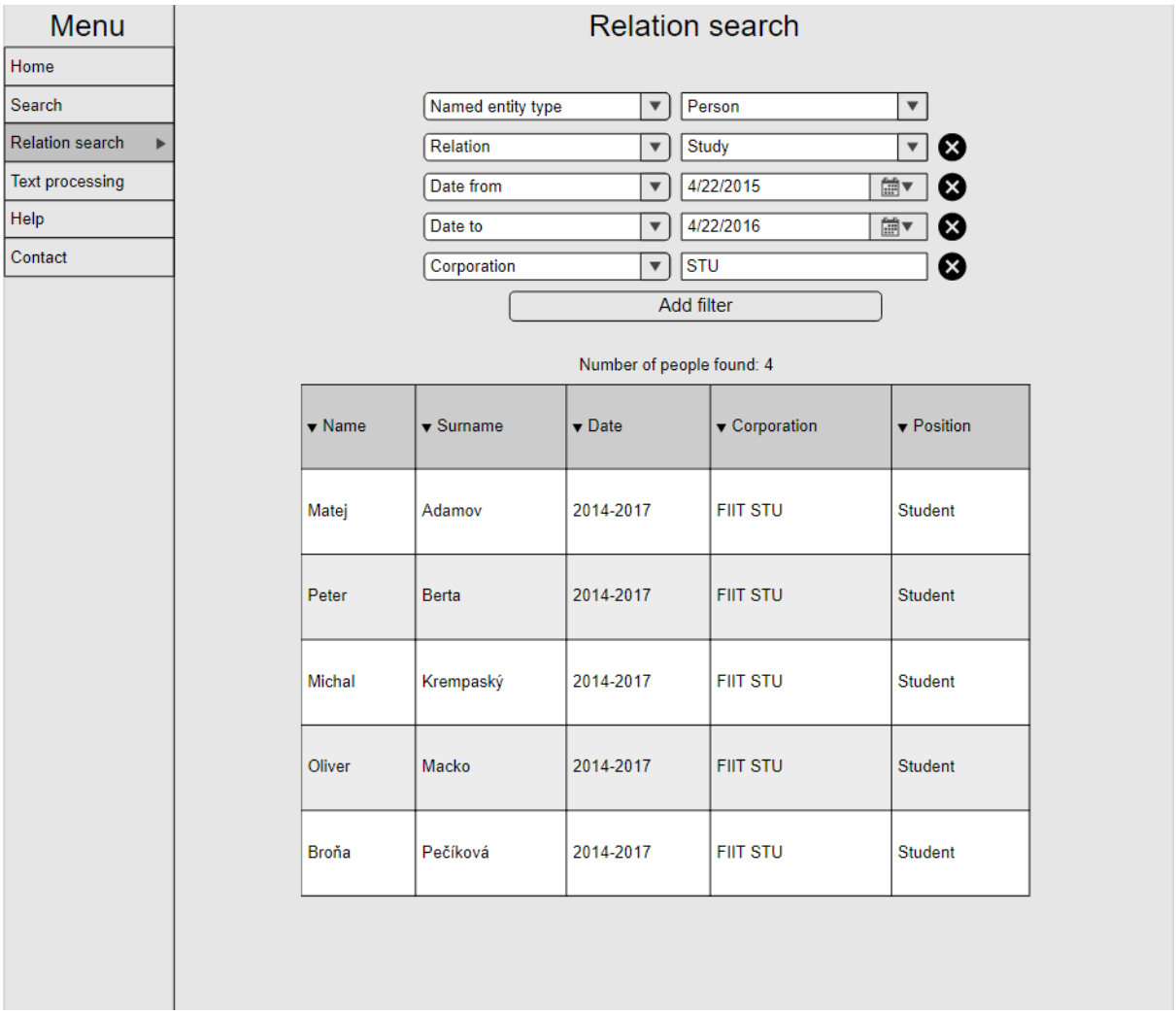
Pri vyhľadávaní vzťahov si používateľ zadáva viacero kritérií vyhľadávania. Tieto kritéria sú v požívateľskom rozhraní reprezentované filtrami, ktoré je možné pridávať a odoberať podľa potreby. V týchto filtroch používateľ vie definovať časovú os, korporáciu, miesto, vzťah, osobu, podľa toho čo vyhľadáva. Výsledkom tohto vyhľadávania je kolekcia všetkých záznamov, ktoré zodpovedajú zadaným kritériám.



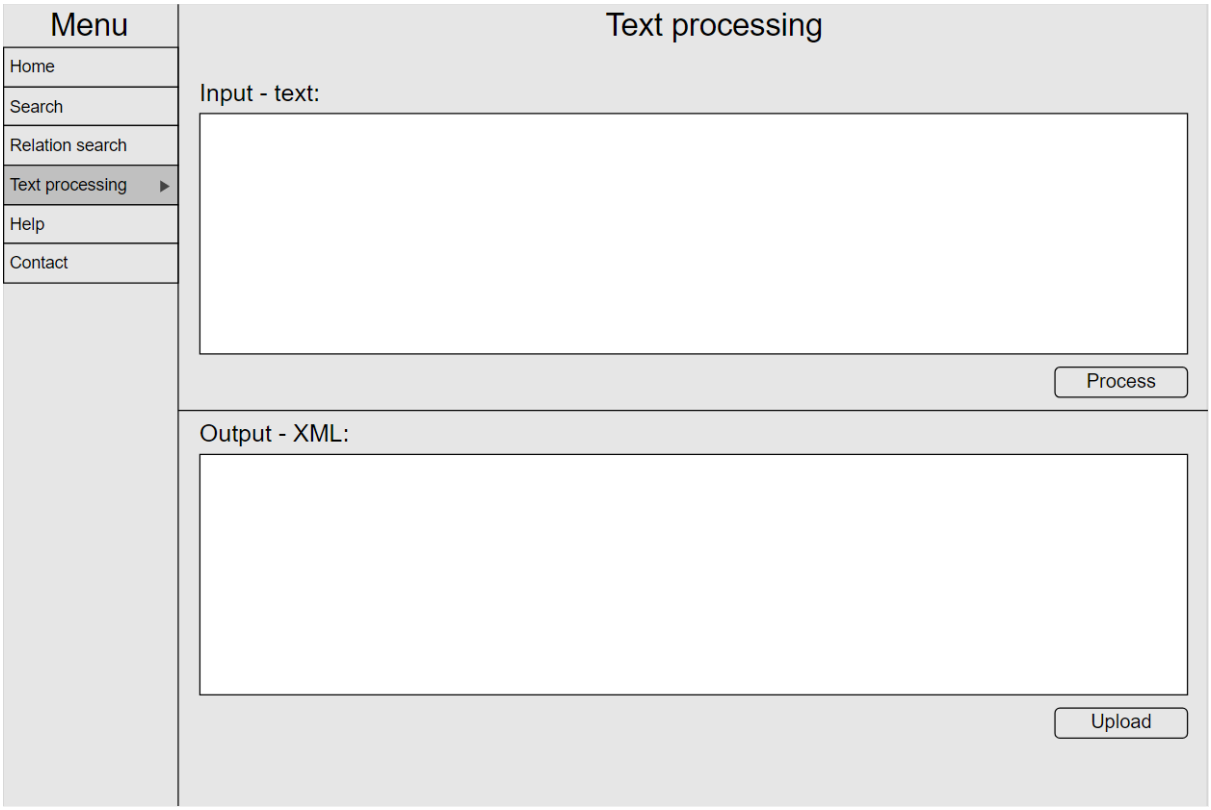
Obr. 14: Obrazovka vyhľadávania

3.4.6. Spracovanie textu

Spracovanie textu je nástroj webového rozhrania, ktorý umožňuje používateľovi z neštruktúrovaného textu životopisu vytiahnuť informácie o štúdiu. Výsledné informácie sú vo formáte XML. Na spracovanie textu slúži tlačidlo „Process“. Tlačidlo „Upload“ slúži na vloženie štruktúrovaných dát do databázy. Táto možnosť je dostupná iba používateľom s potrebnými právami prístupu.



Obr. 15: Vyhľadávanie podľa vzťahov



Obr. 16: Obrazovka spracovania textu