VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SLUŽBA PRO OVĚŘENÍ SPOLEHLIVOSTI A PEČLIVOSTI ČESKÝCH ADVOKÁTŮ

A SERVICE FOR VERIFICATION OF CZECH ATTORNEYS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. RADIM JÍLEK**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. IGOR SZÖKE, Ph.D.**
SUPERVISOR

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií Akademický rok 2016/2017

Zadání diplomové práce

Řešitel: **Jílek Radim, Bc.**
Obor: Počítačová grafika a multimédia
Téma: **Služba pro ověření spolehlivosti a pečlivosti českých advokátů**
A Service for Verification of Czech Attorneys
Kategorie: Softwarové inženýrství

Pokyny:

1. Seznamte se s existujícími nástroji pro vzdálenou interakci s webem a OCR převod. Nastudujte techniky hledání v textu a metody porovnávání textu.
2. Navrhněte způsob hodnocení spolehlivosti a kvality advokáta, dále pak systém pro automatické získávání a vyhodnocení relevantních veřejných dat z webových stránek soudů.
3. Navržený systém implementujte včetně vyhledávání jmen advokátů v textu rozhodnutí soudu.
4. Vyhodnoťte spolehlivost převodu OCR, použité techniky hodnocení advokáta, vyhledávání a porovnávání textu. Vše proveďte na reálných datech.
5. Zhodnoťte výsledky a navrhněte směry dalšího vývoje.
6. Vyrobte A2 plakátek a cca 30 vteřinové video prezentující výsledky vaší práce.

Literatura:

- Podle pokynů školitele

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2 a část bodu 3 ze zadání.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Szőke Igor, Ing., Ph.D.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2016
Datum odevzdání: 24. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Tato práce se zabývá návrhem a implementací internetové služby, která umožňuje objektivně posoudit a ověřit spolehlivost a pečlivost českých advokátů, a to na základě veřejně dostupných dat několika soudů. Cílem práce je vytvořit a zprovoznit tuto službu. Výsledkem práce jsou programy zajišťující dílčí úkony při realizaci tohoto záměru.

Abstract

This thesis deals with the design and implementation of the Internet service, which allows to objectively assess and verify the reliability and diligence of Czech lawyers based on publicly available data of several courts. The aim of the thesis is to create and put into operation this service. The result of the work are the programs that provide partial actions in the realization of this intention.

Klíčová slova

advokáti, internetová služba, stahování dat, scraping, crawler, Python, Nejvyšší správní soud, Ústavní soud, Česká advokátní komora, Levenshteinova vzdálenost

Keywords

lawyers, internet service, download data, scraping, crawler, Python, Supreme administrative court, Constitutional court, The czech bar association, Levenshtein distance

Citace

JÍLEK, Radim. *Služba pro ověření spolehlivosti a pečlivosti českých advokátů*. Brno, 2017. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Igor Szöke, Ph.D.

Služba pro ověření spolehlivosti a pečlivosti českých advokátů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Igora Szöke, Ph.D. Další informace a konzultace mi poskytli Mgr. et Mgr. Tereza Papouškova, RNDr. Jan Papoušek a Mgr. Jan Drábek. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Radim Jílek
21. května 2017

Poděkování

Chtěl bych poděkovat panu Ing. Igorovi Szökemu, Ph.D., že se ujal vedení mé práce a za vlídný přístup během konzultací. Největší dík bych chtěl vyjádřit Mgr. et Mgr. Tereze Papouškové, bez jejíž myšlenky by tato práce nevznikla. Nemalý dík patří i mým spolupracujícím kolegům, kteří se společně se mnou podíleli a podílejí na vývoji aplikace popisované v této práci. Jmenovitě jsou to Mgr. Jan Drábek, Tomáš Vejpustek a RNDr. Jan Papoušek.

Obsah

Kapitola 1

Úvod

Trendem současné doby je porovnávání nejrůznějších produktů, služeb a informací. Je tedy stále důležitější mít nástroje, které jsou schopné toto porovnávání zajistit. Základním předpokladem je mít k dispozici dostatek relevantních informací. Zdrojem těchto informací mohou být jak osobní zkušenosti uživatelů/zákazníků, tak i veřejně dostupné informace různých institucí. Hodnocení uživatelů bývají často velmi subjektivní.

Tato práce se zabývá návrhem a tvorbou internetové služby, která umožňuje ověřit spolehlivost a pečlivost českých advokátů na základě informací o konečných rozhodnutích soudů. V oblasti advokacie se totiž často setkáváme s tím, že klient může být nespokojen ne proto, že poskytovaná služba nebyla kvalitní a podle jeho představ, ale proto, že i s kvalitním advokátem klient svůj spor prohrál. Subjektivní hodnocení advokátů jsou tedy často ovlivněna právě výsledkem sporu a neposkytují skutečnou informaci o práci, přístupu a pečlivosti daného advokáta. Česká justice má v návaznosti na zákony České republiky striktně definovány formální požadavky pro dokumenty podávané na soudy. Onu pečlivost advokáta lze tedy vyčíst z oficiálních dokumentů, které jsou soudy vypracovány jako reakce na podané žádosti.

Na poli českého internetu není k dispozici služba, která by poskytovala objektivní informace o všech advokátech bez rozdílu. O prvotním impulsu pro vznik tohoto projektu, za účelem zaplnění této mezery na internetu, pojednává následující kapitola (2). V této části je také nastíněn základní koncept a představa, jak by výsledná služba mohla fungovat a jaká data jsou potřeba pro její zprovoznění. Zprovoznění služby je týmový projekt, a protože pouze část řešení je předmětem této diplomové práce, je v této části dále uvedeno rozdělení kompetencí/práce v rámci týmu.

Kapitola (3) představuje konkrétní informace, ke kterým soudy umožňují přístup veřejnosti na svých webových stránkách. U popisu stěžejních informací je popsán i jejich význam pro návrh služby. Dále je v této kapitole uvedeno několik příkladů podobných internetových služeb, které ovšem nejsou zcela objektivní. Hledisko hodnocení advokátů se totiž liší podle toho, jestli je jeho práce kvalitní, tzn. udělá, vše co má, správně, ne zbytečně apod. a dalším hlediskem je, zda spor klient vyhrál či ne. Není tedy možné kvalitu advokátů hodnotit jen podle počtu nebo podílu vyhraných sporů, ale také podle počtu podání, která neobsahují chyby, které jsou důvodem k odmítnutí či neprojednávání daného podání. Popsané existující služby se staly inspirací pro to, jaké další informace o advokátech by mohly budoucí uživatele služby zajímat.

Provoz takové služby vyžaduje vybudování obsáhlé databáze případů všech zpracovávaných soudů a informací o advokátech. Rozložení informací v databázi a její návrh popisuje kapitola (4), která přibližuje vztahy mezi získávanými daty. Nalezneme zde dále popis

funkcionality jednotlivých částí a řídícího systému celé služby. V neposlední řadě je zde představen mechanismus hodnocení dokumentů ve vztahu k advokátovi.

Další kapitola (5) popisuje proces analýzy požadavků zpracovávaných webových stránek. Nabízí popis několika uvažovaných knihoven pro realizaci stahování dat ze sledovaných stránek a způsob výběru knihovny, která odpovídá nejlépe stanoveným požadavkům. Současně jsou zde představeny nástroje a programy usnadňující samotný proces vývoje. Závěr kapitoly se zabývá možnostmi převodu netextových dokumentů (PDF, obrázky) do textové reprezentace metodou OCR.

Kapitola (6) popisuje postup realizace a implementace jednotlivých součástí systému a ukazuje problematická místa, se kterými jsem se při realizaci služby setkal. V rámci textu jsou popsány konkrétní problémy a myšlenky, které provázely proces vývoje až k nalezení výsledného řešení.

Předposlední kapitola (7) shrnuje práci celého týmu a ilustruje ji na ukázkách rozhraní. U každé části rozhraní je pak popsáno, jaké informace a které části systému jsou v pozadí zobrazeného rozhraní. Dále jsou představeny možnosti, které služba poskytuje uživateli. Je představena také část administračního rozhraní, které umožňuje ruční ohodnocování, a nebo korekci automatického ohodnocení. Na závěr kapitoly je zařazeno shrnutí, které popisuje technologie použité u jednotlivých částí systému a jejich návaznost.

V závěru práce (8)je shrnut průběh vývoje celé služby a nastíněny možnosti dalšího rozšíření, na konci textu pak úvaha o možnostech jejího využití služby pro širší veřejnost.

Tato práce navazuje na semestrální projekt, v jehož průběhu vznikl základní obsah kapitol 2, 3, 4, 5, které byly v průběhu vypracovávání diplomové práce rozšířeny a upraveny.

Kapitola 2

Idea

Tato práce nevznikla jako většina ostatních prací, výběrem tématu vypisovaného některým z vyučujících na škole, ale vyplynula z požadavku Mgr. et Mgr. Terezy Papouškové na vytvoření datasetu[1], který obsahuje jména advokátů spojená s označeními případů, v nichž podávali návrh k Ústavnímu soudu. Tento dataset je podkladem pro zpracování její disertační práce, která se bude zabývat „nastolováním veřejné agendy Ústavním soudem a rolí advokátů v tomto procesu“. Uvědomila si, že neexistuje žádná možnost ověřit si (online), jak si určitý advokát vedl v minulosti, zda se chová profesionálně a je při předkládání podání pečlivý. Tedy ne ověřit si jeho „úspěšnost“ - počet vyhraných a prohraných sporů. Z toho vznikla myšlenka na vytvoření služby podchycující údaje ze všech zveřejňovaných soudních rozhodnutí. V současné době zveřejňují a zpřístupňují kompletní rozhodnutí jen Nejvyšší soud, Nejvyšší správní soud a Ústavní soud. Z rozhodnutí ostatních soudů jsou zveřejňována pouze některá rozhodnutí. Tato práce zpracovává jen rozhodnutí o kasačních stížnostech, rozhodnutí o dovolání a rozhodnutí o ústavních stížnostech, které musí stěžovatelé povinně podávat skrze advokáty. [12, §105] [10, §30] [11, §241] [13, §265]
V této kapitole je popsán cíl práce a základní schéma, jak by měla celá služba fungovat.

2.1 Záměr

Jak již bylo řečeno, na poli českého internetu dosud neexistuje žádná služba, která by poskytovala objektivní zpětnou vazbu o kvalitě služeb českých advokátů. O jejich kvalitě vypovídá poctivý přístup k danému případu a pečlivost při vypracování potřebných podání. V neposlední řadě pak i spokojenost klientů s jejich službami a přístupem.

Například v oblasti zdravotnictví existuje služba `znamylekar.cz`, která umožňuje na základě jména, popřípadě působiště, vyhledat lékaře dané specializace a bližší informace o něm. Součástí bližších informací mohou být i komentáře uživatelů. Podobně funguje i server `nejremeslnici.cz`.

Záměrem této práce je tedy navrhnout, vytvořit a zprovoznit obdobnou službu, která by na základě veřejně dostupných dat vyhodnotila kvalitu a nedostatky podání jednotlivých advokátů. Služba bude objektivní ke všem advokátům, jelikož bude založena jen na volně dostupných datech a interpretaci výsledku rozhodnutí. Veškeré výsledky rozhodnutí jsou veřejné[2], ale jen nejvíce sledované soudy tyto informace zveřejňují na svých webových

[1] soubor dat

[2] Je možné o ně požádat žádostí o informaci na základě zákona č. 106/1999 Sb., zákon o svobodném přístupu k informacím

stránkách. Konkrétně se jedná o tyto soudy: Nejvyšší správní soud, Ústavní soud, Nejvyšší soud.

Navrhovaná služba bude tedy zahrnovat jen advokáty působící u těchto soudů a to jen ve výše specifikovaných řízeních. Měřítkem kvality je splnění formálních požadavků, jejichž nedodržení může vést k tomu, že se soud podáním nezabývá meritorně[3], ale odmítne jej jen z důvodu formálních nedostatků. Tím trpí jak daný klient, tak i justice jako celek (i na špatně napsaná podání musí soudy reagovat). Možnost ověřit si spolehlivost a pečlivost určitého advokáta, který předkládá podání soudu, může výrazně ovlivnit, zda si klient (uživatel) zvolí pro svoji kauzu daného advokáta. Výběrem kvalitních advokátů by mohlo dojít i ke snížení množství žádostí podávaných k soudům a tím k odlehčení soudů.

2.2 Jak to bude celé fungovat?

Veřejná data potřebná pro uvedený záměr je možné najít na webových stránkách výše uvedených soudů a stránkách České advokátní komory. Pro získání dat bude potřeba na stránkách jednotlivých soudů vyhledávat případy podle data a typu podání, který je určen přidělenou spisovou značkou. Soudy toto vyhledávání umožňují, půjde tedy o vhodné a automatické vyplnění kritérií vyhledávacích formulářů. Následně se zobrazí stránka s výsledky. Tyto výsledky jsou zobrazeny obvykle ve formě tabulek a jsou stránkovány. Informace, relevantní pro další zpracování (metadata), bude tedy nutné z těchto tabulek extrahovat a uložit v jednotném formátu. Tímto způsobem musíme zpracovat všechny stránky výsledků. Stahování a extrahování dat bude probíhat prostřednictvím skriptů, které se budou spouštět automatizovaně a pravidelně. Než bude spuštěno pravidelné stahování, bude ještě potřeba stáhnout dosavadní databázi rozhodnutí soudů za období několika let. Doba se liší u jednotlivých soudů v závislosti na době jejich vzniku, anebo roku zveřejnění digitalizovaných dokumentů.

Takto získané informace budou následně uloženy do databáze, a to včetně dokumentu, který obsahuje znění rozhodnutí soudu. Na základě uložených informací o rozhodnutích budeme dále schopni nad touto databází provést hromadné ohodnocení dokumentů ve vztahu k advokátovi. Tj. zjistíme, zda výsledek rozhodnutí je ve prospěch advokáta nebo zda poukazuje na nějaký nedostatek způsobený nedbalostí advokáta. Dále se pokusíme z dokumentů získat jména advokátů, kteří v nich figurují a přiřadit je advokátům, kteří jsou uvedeni v seznamu advokátů České advokátní komory (reálným osobám).

Nad takto provázanými daty už bude možné zobrazovat statistiky o množství podání daného advokáta a kvalitě jeho služeb. Stejně tak bude možné uživateli poskytnout možnost vyhledávání advokáta podle jména a příjmení, případně dalších identifikačních údajů dostupných v databázi České advokátní komory. Nejdůležitější části vývoje jsou zachyceny na obrázku 2.1.

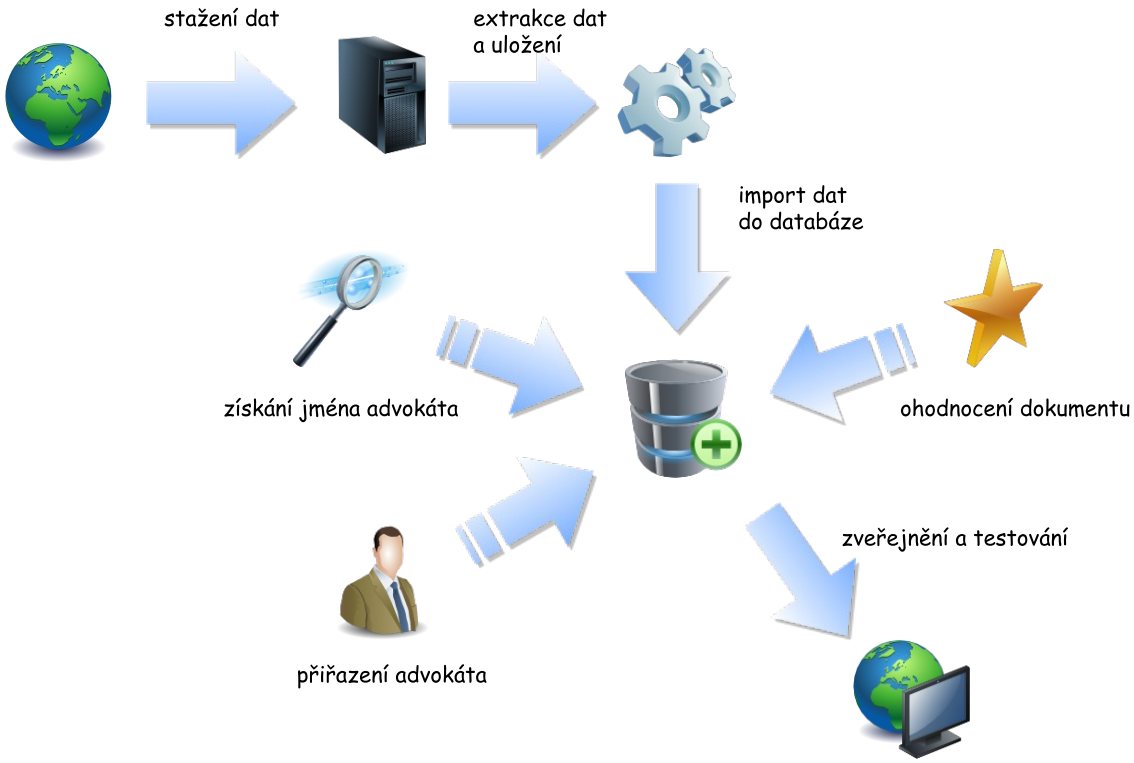
Výsledná služba bude realizována jako webová stránka s vyhledávacím formulářem pro zadání celého jména (či alespoň části jména) požadovaného advokáta nebo jeho IČ. U detailu vybraného advokáta budou uvedeny jeho kontaktní a identifikační údaje. Dále pak seznam případů, které daný advokát řešil napříč všemi sledovanými soudy, a jejich výsledků, společně s odkazy na originální dokumenty. Navíc bude u daného advokáta zobrazený počet správně a chybně vypracovaných podání.

[3] co do obsahu

2.3 Rozdělení práce v týmu

Za účelem realizace projektu byl založen občanský spolek *DATOS - data o spravedlnosti, z. s.*, který sdružuje osoby zapojené do vývoje aplikace/služby. V rámci tohoto týmu je rozdělena práce na jednotlivých částech projektu takto:

- Mgr. et Mgr. Tereza Papoušková, RNDr. Jan Papoušek
  - vedení týmu
  - komunikace se soudy
  - finanční podpora
- Bc. Radim Jílek – autor této diplomové práce
  - návrh a implementace způsobu stažení dat (crawler)
    - Nejvyšší správní soud
    - Ústavní soud
    - Česká advokátní komora
  - návrh a implementace ohodnocení dokumentů
    - Nejvyšší správní soud
    - Ústavní soud
  - návrh a implementace způsobu přiřazování advokátů
    - Nejvyšší správní soud
    - Ústavní soud
  - návrh databáze
- Mgr. Jan Drábek
  - návrh a implementace způsobu stažení dat (crawler) - Nejvyšší soud
  - správa a úprava databáze
  - import dat do databáze
  - návrh a implementace ohodnocení dokumentů - Nejvyšší soud
  - návrh a implementace způsobu přiřazování advokátů - Nejvyšší soud
  - návrh a implementace webu - backend, REST API
  - administrace systému
- Tomáš Vejpustek
  - návrh a implementace webu - frontend



Obrázek 2.1: Schéma vývoje

Obrázek 2.1 zobrazuje dekompozici problému, která plyne ze záměru a popisu funkčnosti navrhované služby. Zároveň popisuje tok dat v průběhu zpracování celé aplikace. V centrální části obrázku je pak znázorněno, jaké operace/procesy budou probíhat nad získanými daty uloženými v databázi. Ve spodní části obrázku je následně naznačeno, že získaná agregovaná data budou využita pro prezentování zjištěných informací koncovému uživateli.

Kapitola 3

Teoretický rozbor

V této kapitole jsou představeny podobné služby, které poskytují informace o advokátech a někdy i o jejich hodnocení. V další části této kapitoly jsou představeny konkrétní údaje, které poskytují soudy a je přiblížen jejich význam pro následující zpracování.

3.1 Podobné služby

Níže je uvedeno s popisem několik webových portálů poskytujících služby podobného charakteru jako tato práce. Tyto portály byly inspirací při návrhu aplikace.

3.1.1 dTest: Databáze advokátů[1]

Tato databáze advokátů popisuje advokáty a jejich zaměření v rámci práva. Lze zde nalézt i další informace, které jsou ovšem běžně dostupné na webu České advokátní komory. Není překvapením, že na tomto webu (dtest.cz) jsou uvedeni jen někteří advokáti, a to na základě jejich registrace na tomto webu. Kromě jména v ní lze vyhledávat podle oboru (specializace) a kraje působnosti advokáta.

3.1.2 Katalog právníků[2]

Tato databáze obsahuje základní informace nejen o advokátech či advokátních kancelářích, ale například i o odhadcích a notářích. Navíc obsahuje popis služeb subjektu, který je zřejmě ponechán v rukou samotného vlastníka. Dále zde najdeme například otevírací dobu a doplňující informace o umístění. Tento web je představován jako prezentace a propagace právních služeb. Samozřejmostí je vyhledávání podle jména, dále lze vyhledávat podle oboru, poskytovaných služeb a kraje působnosti subjektu. I tato databáze obsahuje pouze záznamy subjektů registrovaných na tomto webu.

3.1.3 Advokáti - právníci s hodnocením klientů[3]

Tento web se od předchozích liší tím, že není zaměřen přímo na vyhledávání advokátů, ale různých profesí, služeb a firem. Na rozdíl od předchozích databází neumožňuje vyhledávání podle jména. Obsahuje kategorii *advokáti*, která zahrnuje advokátní kanceláře, právní

[1] dTest: Databáze advokátů - Nezávislé testy, víc než jen recenze, `https://www.dtest.cz/advokati`

[2] Právník, advokátní kancelář, právní ochrana, právní poradenství, evropské právo - katalog-pravniku.cz, `http://www.katalog-pravniku.cz`

[3] Advokáti - právníci s hodnocením klientů, `https://pravo-finance.sluzby.cz/advokati-pravni-sluzby`

služby, právníky, seznam advokátů, právo obchodní, pracovní, rodinné a trestní. Dále lze vybrat lokalitu působení daného subjektu. Výsledkem vyhledávání je seznam s odkazy na webové prezentace subjektů (vizitek), kde jsou uvedeny základní kontaktní informace a obvykle lze nalézt i záložku hodnocení. Ta obsahuje hodnocení formou hvězdiček a často i slovní komentář klienta. Provozovatelé webu uvádějí, že dané hodnocení je ověřeno a uvedeno s podpisem hodnotitele.

3.1.4 Otvorené Súdy[4]

Tento rozsáhlý portál se zabývá otázkou slovenského soudnictví a jeho transparentností. Obsahuje souhrn veřejně dostupných či zpřístupněných informací na jednom místě a ukazuje provázanost jednotlivých dat. Je zaměřen na jednotlivé soudce, soudní rozhodnutí a aktivity jednotlivých soudů. Umožňuje vyhledávání jednotlivých případů podle spisové značky, soudců, soudů, právních oblastí, forem rozhodnutí, data apod.

Poslední zmíněný web má nejblíže k záměru této práce. I když tento portál poskytuje mnohem komplexnější a ucelenější informace, jeho myšlenka a přínos jsou inspirací pro zamýšlenou službu. Ovšem v České republice není zavedení takového portálu dosud možné, jelikož naše legislativa (zatím) neukládá soudům povinnost zveřejňovat svá rozhodnutí, natož formou otevřených dat. Tato služba by se tak mohla stát pomyslnou „první vlaštovkou“.

Ukázky rozhraní a možností vyhledávání výše popsaných serverů jsou k nalezení v příloze.

3.2 Význam dat

Hledání informací pomocí vyhledávacích formulářů na webových stránkách soudů umožňuje vyhledávání hned podle několika kritérií. Je to například vyhledávání podle data rozhodnutí, spisové značky či textu obsaženého v rozhodnutí soudu. Tato kriteria lze kombinovat. V této práci budu využívat vyhledávání podle spisové značky (její části) a data rozhodnutí.

3.2.1 Nejvyšší správní soud[5]

U tohoto soudu jsou zpracovávána konečná rozhodnutí o kasačních stížnostech (rejstříky As, Ads, Afs, Ans, Aos, Aps, Ars, Azs) podaných po 1. 1. 2006.

V samotných výsledcích vyhledávání budou obsaženy tyto informace, z nichž některé budou využity pro další zpracování.

Jednací číslo – jedná se o jednoznačný identifikátor dokumentu rozhodnutí soudu, který ve svém základu obsahuje identifikaci případu, ke kterému náleží, tzv. spisovou značku. Např. 9 Ad 20/2013 - 38, kde část před pomlčkou je spisová značka a číslo za ní je určení daného dokumentu ve spisu. Spisová značka samotná je pak složena z:

- označení soudního oddělení - `9`
- rejstříkové značky - `Ad`
- pořadí věci u soudu - `20`
- roku - `2013`

[4] Otvorené Súdy - Transparency International Slovensko, `https://otvorenesudy.sk`

[5] informace jsou čerpány z návodu pro vyhledávání na `http://nssoud.cz`

Forma/Způsob rozhodnutí – poskytuje informaci o formě a výsledku rozhodnutí. Tento údaj je pro hodnocení advokátů stěžejní. Např. Rozsudek, Usnesení...

Datum rozhodnutí – jedná se o datum rozhodnutí, které je uvedené na dokumentu.

Účastníci řízení – zde je v případě účasti právnických osob vypsán seznam účastníků soudního řízení. Jména fyzických osob jsou anonymizována a ve výsledcích se vůbec neobjevují.

Soud – název soudu, u kterého bylo dané řízení vedeno.

Odkazy – spadá sem hned několik odkazů reprezentovaných konkrétními ikonami (viz obrázek 3.1), které odkazují na doplňující informace k případu:

- informace o řízení
- právní věty
- anotace
- sbírka NSS
- anonymizovaná/zkrácená verze rozhodnutí

V případě odkazu na rozhodnutí je tedy možné dostat se k samotnému textu rozhodnutí.

Prejudikatura[6] – zde se objevuje spisová značka dokumentu, ke kterému se musí přihlížet při posuzování případu, který na něj odkazuje. Může se jednat o spisovou značku jiného soudu.

Všechny uvedené informace shrnuje obrázek 3.1, který mimo jiné ukazuje i konkrétní podobu výsledků vyhledávání u Nejvyššího správního soudu.

Obrázek 3.1: Ukázka výsledků vyhledávání u Nejvyššího správního soudu

3.2.2 Ústavní soud[7]

U tohoto soudu jsou zpracovávána konečná rozhodnutí o všech návrzích podaných od počátku činnosti soudu[8] do 31. 12. 2006 (v tomto období nebyla nijak rozlišována rozhodnutí o ústavních stížnostech a rozhodnutí vydaná v jiných typech řízení).

Všechny následující informace nejsou obsaženy v samotných výsledcích vyhledávání, jako tomu bylo v předchozím případě, ale jsou zobrazeny až v kartě určitého záznamu (detailu případu). Na ten se lze dostat odkazem přes spisovou značku uvedenou ve výsledcích vyhledávání.

Identifikátor evropské judikatury (European Case Law Identifier - ECLI) – je celoevropský systém jednotné identifikace, je citací judikatury a struktury metadat, která se vztahují ke každému dílčímu rozhodnutí soudu členského státu EU. ECLI kód umožňuje citovat judikaturu Ústavního soudu v celoevropském formátu.

Název soudu – název soudu, u kterého bylo dané řízení vedeno.

Soudce zpravodaj – soudce zpravodaj je zpracovatelem příslušného spisu a zpravidla i autorem textu rozhodnutí. Je prvním soudcem, který čte zaevidovaný spis, a je na něm, jaký bude další osud návrhu (může rozhodnout o jeho okamžitém odmítnutí, pořadí projednávání apod.). Neodmítl-li návrh, předkládá soudce zpravodaj senátu či plénu Ústavního soudu návrh k projednávání.

Spisová značka – jedná se o identifikátor daného případu vedeného u Ústavního soudu. Např. III.ÚS 874/17
Spisová značka samotná je pak složena z:

- označení soudního oddělení: III.
- rejstříkové značky: ÚS
- pořadového čísla případu v daném roce: 874
- roku zaevidování: 2017

Populární název – název charakterizující nálezy a publikovaná rozhodnutí, který výstižně označuje věcný obsah projednávané kauzy.

Datum rozhodnutí – datum uvedené na příslušném dokumentu vydaném Ústavním soudem.

Datum vyhlášení – datum veřejného vyhlášení nálezu Ústavního soudu.

Datum podání – datum, kdy byl daný návrh podán a zaevidován.

Datum zpřístupnění – datum, od kterého je text rozhodnutí zveřejněn ve veřejné databázi Ústavního soudu.

Forma rozhodnutí – poskytuje informaci o formě rozhodnutí. U Ústavního soudu jsou to: Nález, Usnesení, Stanovisko pléna.

Typ řízení – označuje typ podaného návrhu, a tedy sděluje, co je předmětem navazujícího řízení. Po roce 2006 bylo zavedeno 15 typů řízení.

[7] čerpáno z oficiálních stránek Ústavního soudu `http://usoud.cz`

[8] Vznik Ústavního soudu České republiky je spjat s rokem vzniku České republiky (1993)

Typ výroku – text, který stručně vyjadřuje podstatu rozhodnutí. Jedná se o stěžejní údaj k ohodnocení dokumentu ve vztahu k advokátovi.

Paralelní citace (Sbírka zákonů) – odkaz na příslušnou právní normu uveřejňující rozhodnutí ve sbírce zákonů.

Paralelní citace (Sbírka nálezů a usnesení) – odkaz na konkrétní nález či usnesení uveřejněné ve SbNU Ústavního soudu.

Navrhovatel – jedná se o určení subjektu, dle číselníku, který podává návrh. Například: soud, skupina poslanců, politická strana, ministerstvo, zastupitelstvo obce, fyzická osoba...

Dotčený orgán – je-li případu účasten orgán veřejné moci, je zde uveden typ a případně jméno zodpovědné osoby.

Význam – charakterizuje význam (důležitost) rozhodnutí na škále 1 (často nálezy rušící právní předpis) až 4 (často rozhodnutí o odmítnutí návrhu).

Napadený akt – typ aktu, který navrhovatel napadá. Například v případě ústavních stížností jde zpravidla o rozhodnutí soudu.

Dotčené ústavní zákony a mezinárodní smlouvy – odkazy na ústavní zákony a mezinárodní smlouvy, které se úzce vztahují k danému rozhodnutí.

Ostatní dotčené předpisy – odkazy na další právní předpisy, které se úzce vztahují k danému rozhodnutí.

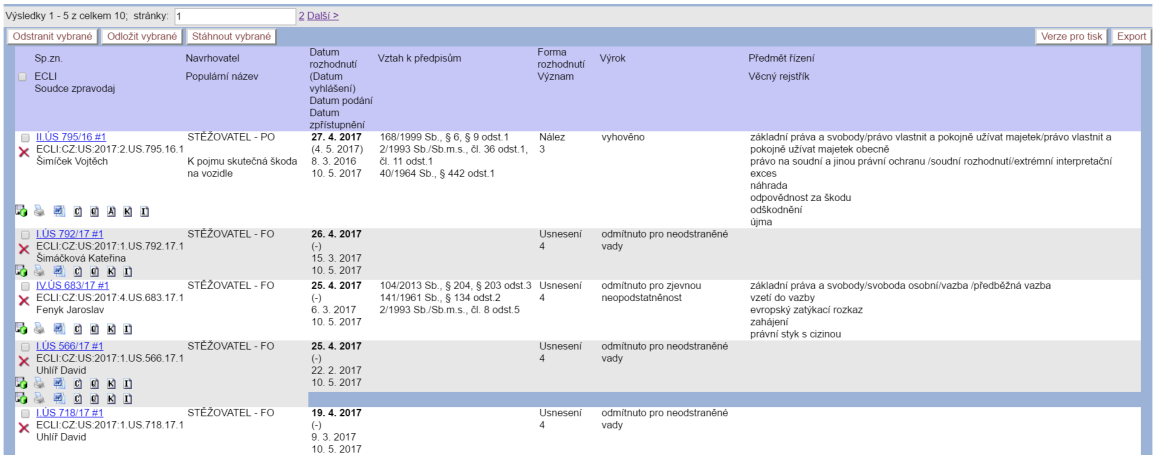
Odlišné stanovisko – jméno soudce (popřípadě soudců), který nesouhlasil s výrokem či odůvodněním daného rozhodnutí.

Předmět řízení – uvádí pojmy z předem definovaného číselníku, které charakterizují předmět řízení. (Oproti Věcnému rejstříku spíše než podstatu případu charakterizuje navrhovatelovu argumentaci, tj. důvod, proč se na Ústavní soud obracel.)

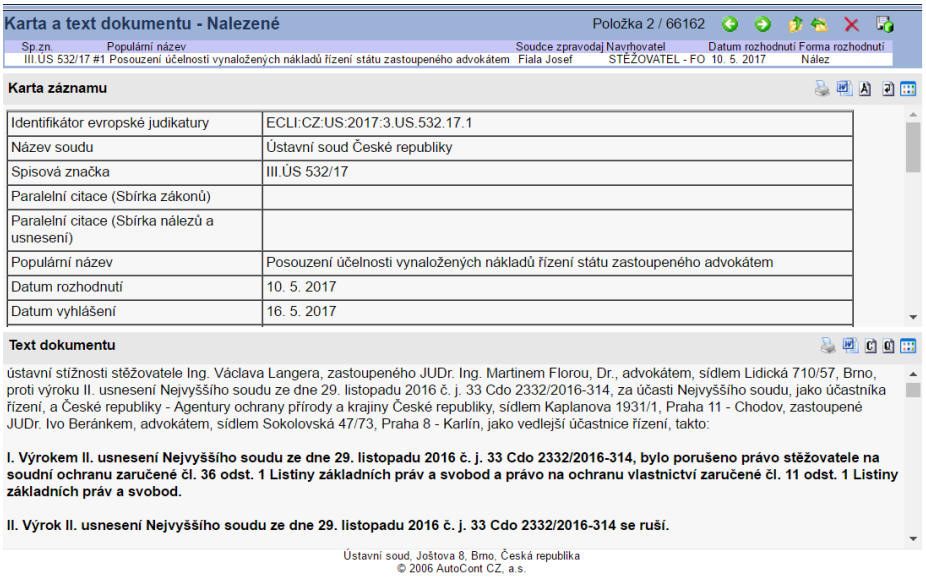
Věcný rejstřík – uvádí pojmy z předem definovaného číselníku, které charakterizují, čeho se daný případ týkal. (Oproti Předmětu řízení podstatu případu charakterizuje konkrétněji.)

URL adresa – na této adrese je dostupný text samotného rozhodnutí bez dalších doplňujících informací, týkajících se případu.

Obrázek 3.2 ukazuje stránku s výsledky vyhledávání, kde jsou vidět některé z výše popsaných údajů. Na obrázku 3.3 je pak zobrazen detail karty případu.



Obrázek 3.2: Ukázka výsledků vyhledávání u Ústavního soudu



Obrázek 3.3: Karta s detailem případu

3.2.3 Česká advokátní komora[9]

Níže uvedené informace nejsou, podobně jako u Ústavního soudu, uvedené přímo ve výsledcích vyhledávání a pro jejich zobrazení je nutné přejít na stránku detailu skrze jméno advokáta.

Jméno – jméno a příjmení advokáta (zapsané velkými písmeny) společně se všemi užívanými tituly.

Evidenční číslo – evidenční číslo advokáta, pod kterým je veden v seznamu České advokátní komory.

IČ – identifikační číslo osoby advokáta.

[9] čerpáno ze stránek České advokátní komory, `http://www.cak.cz/`

ID datové schránky – identifikátor datové schránky advokáta.

Stav – stav činnosti advokáta – aktivní, pozastavená činnost, vyškrtnut.

Ustanovení ex-offo – informace, zda je advokát v seznamu přidělovaných právních zástupců.

Zaměření – soupis zaměření právních oblastí působení advokáta.

Jazyk – seznam jazyků, které advokát ovládá na takové úrovni, aby v nich mohl psát podání k soudu a argumentovat.

Kontakty – odkaz na webové stránky a dále seznam emailových adres, uvedený v zabezpečeném formátu, kde je symbol @ nahrazen obrázkem stejného symbolu.

Zaměstnaní advokáti – seznam advokátů (s odkazy na jejich detail), které daný advokát zaměstnává.

Koncipienti – seznam koncipientů (s odkazy na jejich detail), které daný advokát zaměstnává.

Trvale spolupracuje s firmou – seznam firem, se kterými advokát v minulosti spolupracoval či aktuálně spolupracuje. Prioritně je uvedena firma/advokátní kancelář, kde advokát působí.

V další části detailu je pak seznam informací, které se týkají firmy, kde advokát působí.

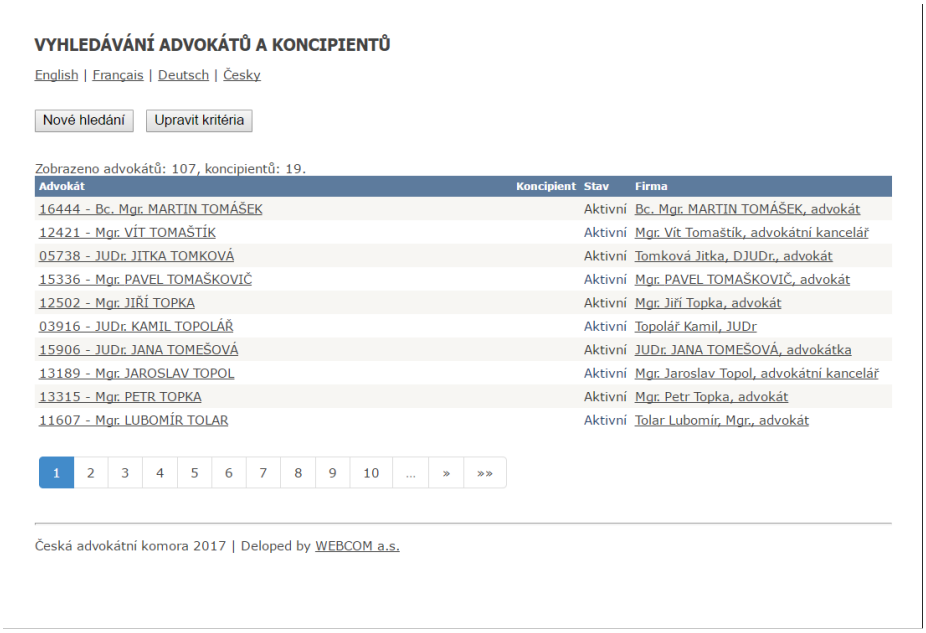
Název – jméno firmy s odkazem na detailnější informace o ní.

IČ – identifikační číslo firmy

Adresa – adresa firmy, včetně orientační a popisného čísla, ulice, města/obce a PSČ

Kontakty – kontakty na firmu, mezi nimi odkaz na webové stránky, seznam emailů, telefonních a fax. čísel.

Obrázek 3.4 zobrazuje stránku s výsledky vyhledávání, kde jsou vidět některé z výše popsaných údajů. Na obrázku 3.5 je pak zobrazena karta s detailem advokáta.



VYHLEDÁVÁNÍ ADVOKÁTŮ A KONCIPIENTŮ

English | Français | Deutsch | Česky

Nové hledání | Upravit kritéria

Zobrazeno advokátů: 107, koncipientů: 19.

Advokát	Koncipient	Stav	Firma
16444 - Bc. Mgr. MARTIN TOMÁŠEK		Aktivní	Bc. Mgr. MARTIN TOMÁŠEK, advokát
12421 - Mgr. VÍT TOMAŠTÍK		Aktivní	Mgr. Vít Tomaštík, advokátní kancelář
05738 - JUDr. JITKA TOMKOVÁ		Aktivní	Tomková Jitka, DJUDr., advokát
15336 - Mgr. PAVEL TOMAŠKOVIČ		Aktivní	Mgr. PAVEL TOMAŠKOVIČ, advokát
12502 - Mgr. JIŘÍ TOPKA		Aktivní	Mgr. Jiří Topka, advokát
03916 - JUDr. KAMIL TOPOLÁŘ		Aktivní	Topolář Kamil, JUDr
15906 - JUDr. JANA TOMEŠOVÁ		Aktivní	JUDr. JANA TOMEŠOVÁ, advokátka
13189 - Mgr. JAROSLAV TOPOL		Aktivní	Mgr. Jaroslav Topol, advokátní kancelář
13315 - Mgr. PETR TOPKA		Aktivní	Mgr. Petr Topka, advokát
11607 - Mgr. LUBOMÍR TOLAR		Aktivní	Tolar Lubomír, Mgr., advokát

1 2 3 4 5 6 7 8 9 10 ... » »»

Česká advokátní komora 2017 | Deloped by WEBCOM a.s.

Obrázek 3.4: Výsledky vyhledávání na stránkách České advokátní komory

Advokát

Jméno	**13189 - Mgr. JAROSLAV TOPOL**
Evidenční číslo	13189
IČ	71346414
ID datové schránky	7uv2nqu
Stav	Aktivní
Způsob výkonu advokacie	
Ustanovení ex-offo	☑

Zaměření
01 generální praxe
25 ochrana osobnosti
39 správní právo
18 insolvenční právo
63 rozhodčí řízení

Jazyk
anglický

Kontakty

www	www.ak-topol.cz
email	info@ak-topol.cz

Zaměstnaní advokáti

Koncipienti
41281 - Mgr. MARTIN MROVEC
42259 - Mgr. TOMÁŠ BRANDEJSKÝ

Trvale spolupracuje s firmou
Mgr. Jaroslav Topol, advokátní kancelář

Ostatní činnosti advokáta

Firma

Název	**Mgr. Jaroslav Topol, advokátní kancelář**
IČ	71346414
Adresa	Na Zlatnici 301/2
	14700 Praha

Kontakty

www	www.ak-topol.cz
email	info@ak-topol.cz
Telefon	+420222263841
Mobil	
Fax	

Obrázek 3.5: Karta s detailem advokáta

Kapitola 4

Neformální návrh

V této kapitole jsou neformálně popsané jednotlivé části systému a jejich význam. Dále je zde uvedeno několik pojmů, které přibližují konkrétní způsob zpracování. Cílem práce je vytvořit službu, která bude zahrnovat informace o případech Nejvyššího správního soudu, Nejvyššího soudu a Ústavního soudu a na základě těchto informací zpřístupňovat statistiky o kvalitě práce českých advokátů. Struktura a obsah zpřístupněných dat jednotlivých institucí se natolik liší, že bude potřeba vytvořit specifický systém srovnávání dat. Na základě rozdělení práce v týmu, jak bylo uvedeno v podkapitole 2.3, se budu v dalším textu práce zabývat zpracováváním údajů Nejvyššího správního soudu, Ústavního soudu a České advokátní komory.

4.1 Použité pojmy

V této části je definován pojem, který souvisí se získáváním dat z webových stránek a jehož označení je použito dále v textu. Další pojem definuje metodu, která je použita pro porovnávání textu.

4.1.1 Získávání dat z webových stránek

Často nazývané též „web scraping“, „web harvesting“, „web data extraction“, „web spidering“, „web wrapping“ a nebo „web crawling“, je sada technik, které se používají pro automatizované získávání informací z webových stránek. Pro samotné získávání dat je využito robotů, což jsou specializované programy/skripty, které navštěvují a procházejí webové stránky a stahují z nich data (dále crawler). [6, 3]

4.1.2 Levenshteinova vzdálenost

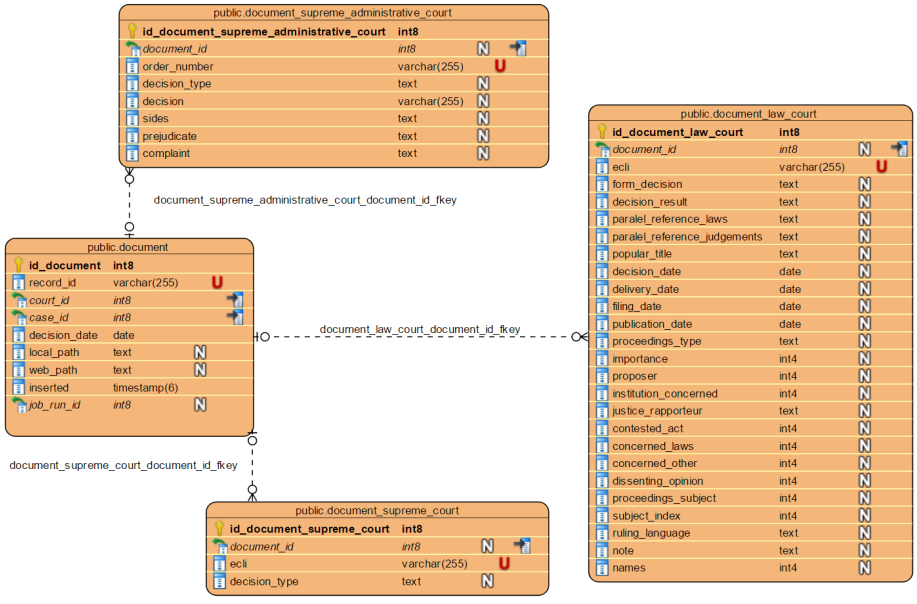
Levenshteinova vzdálenost je vzdálenost dvou řetězců, která je definována jako minimální počet operací potřebných k tomu, aby se jeden řetězec změnil na druhý. Tato metoda připouští tři operace, kterými lze editovat text. Jsou to přidání libovolného znaku, vypuštění libovolného znaku a nebo záměna libovolného znaku za jakýkoliv jiný znak. Levenshteinovou vzdáleností mezi dvěma řetězci je pak nejmenší počet těchto operací nutný k převedení jednoho řetězce na druhý. Levenshteinova vzdálenost tedy vyjadřuje podobnost, respektive rozdílnost, dvou řetězců. [8, 5]

4.2 Databáze

Níže jsou popsány jednotlivé logické části databáze a jejich vzájemné vztahy. Stručně jsou zde představeny údaje, které jsou do databáze ukládány.

4.2.1 Dokumenty

Databáze obsahuje záznamy o všech dokumentech, které byly zveřejněny a staženy. Informace o každém dokumentu lze rozdělit do dvou kategorii. První z nich jsou obecné informace, které jsou pro všechny dokumenty všech soudů společné. Patří mezi ně datum rozhodnutí, příslušnost k soudu a k případu, URL adresa směřující na originální dokument na webu soudu a též cesta vedoucí k lokální kopii dokumentu. Dále pak provozní informace jako jsou datum vložení záznamu a označení funkční části systému, která vložení provedla. Druhou kategorií jsou informace, které jsou specifické pro daný soud. Informace jsou různé, a proto je pro každý soud vytvořena zvláštní tabulka. Shrneme-li údaje napříč soudy do výčtu, obsahují jednotlivé tabulky tyto informace: unikátní identifikátor dokumentu, informaci o formě konečného rozhodnutí soudu a nakonec výsledek samotného rozhodnutí. Obě naposledy zmíněné informace lze jinak vyčíst z textu samotného rozhodnutí. Výše uvedené je tedy vždy uloženo ve 2 tabulkách (`document` a dále `document_law_court` nebo `document_supreme_administrative_court` nebo `document_supreme_court`), a to v závislosti na daném soudu. S tabulkou `document` je propojena tabulka `case`, která obsahuje označení případu spisovou značkou, příslušnost k soudu a speciální data obsahující jména advokátů příslušících k tomuto případu (v textové formě). Informace o případech a k nim náležících dokumentech je tedy uložena vždy v rámci tří tabulek. Jednotlivé tabulky dokumentů, a atributy v nich uložené, znázorňuje obrázek 4.1.

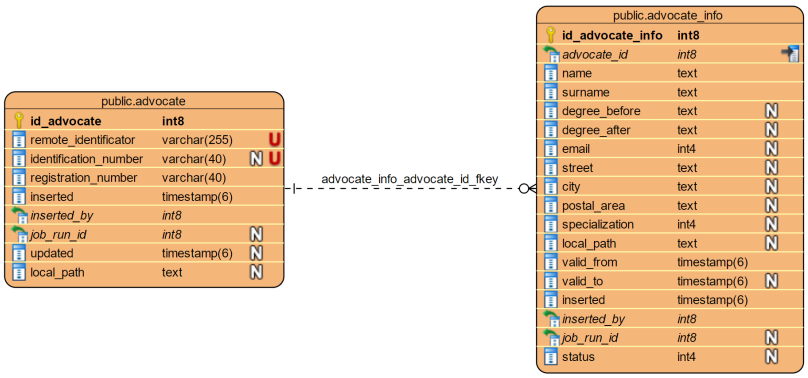


Obrázek 4.1: Část databáze týkající se uložených dokumentů

4.2.2 Advokáti

Další část databáze tvoří informace o samotných advokátech, které jsou, stejně jako dokumenty, uloženy do více tabulek. První je tabulka `advocate`, která uchovává informace

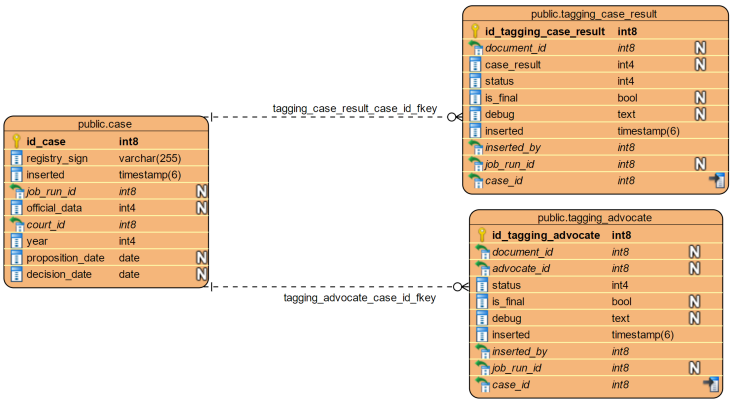
určující osobu advokáta. Najdeme zde evidenční číslo advokáta, pod kterým je veden v seznamu České advokátní komory, dále pak IČ a nakonec unikátní identifikátor (hash), který je součástí URL adresy na webu České advokátní komory. Do další tabulky `advocate_info` jsou ukládány sady informací získaných z profilů jednotlivých advokátů na webu České advokátní komory. Zde nalezneme jméno a příjmení advokáta společně se všemi tituly, které uvedl při zápisu do seznamu České advokátní komory. Dále pak informaci o působišti advokáta, jako jsou město, ulice a PSČ. V neposlední řadě pak samozřejmě kontaktní email, případně seznam specializací a stav provádění činnosti daného advokáta (aktivní, neaktivní, pozastavená činnost, vyškrtnut). A opět cestu k lokálnímu souboru, který obsahuje všechny výše uvedené informace ve formátu `HTML`. Jelikož část uvedených informací se může v průběhu času měnit (změna příjmení po svatbě, změna sídla advokátní kanceláře, přidání emailové adresy, rozšíření specializace a vzdělání...), je každý záznam opatřen intervalem platnosti dat. V databázi jsou tedy uloženy všechny změny, a to vždy jako nový záznam, obsahující celou sadu informací. V naší databázi je ke každému advokátovi vedena historie změn v databázi České advokátní komory. Tyto změny jsou pochopitelně evidovány až od zahájení prací na tomto projektu. Jednotlivé tabulky, a atributy v nich obsažené, ilustruje obrázek 4.2.



Obrázek 4.2: Část databáze obsahující informace o advokátech

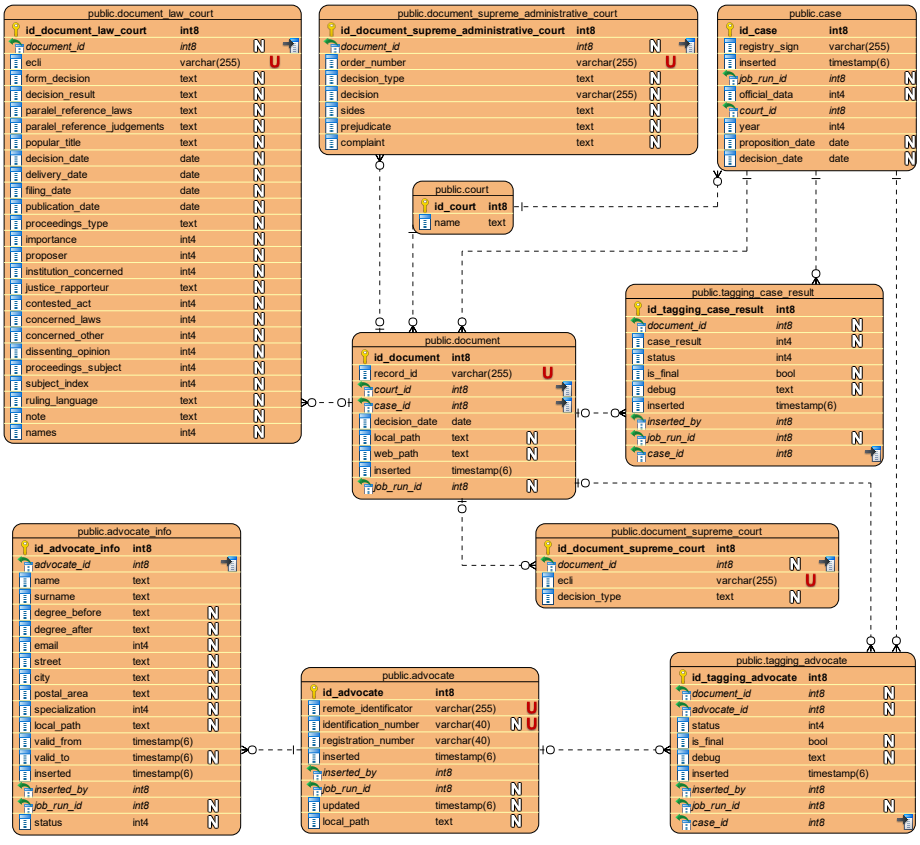
4.2.3 Výsledky procesů

Poslední část databáze uchovává výsledky procesů ohodnocení dokumentů a přiřazení advokáta případu. Tabulka `tagging_case_result` konkrétně uchovává ohodnocení dokumentu ve vztahu k advokátovi (podrobně vysvětleno níže), dále pak stav průběhu samotného ohodnocování daného dokumentu, odkaz na dokument a případ, ke kterému přísluší. Nakonec jsou zde uložené pomocné údaje pro ověření správnosti automatického ohodnocení a příznak, který určuje, zda už byla správnost ohodnocení zkontrolována. Podobně tabulka `tagging_advocate` zachycuje průběh a výsledek procesu přiřazování advokáta k případu. Zcela logicky je zde tedy uveden odkaz na advokáta, kterému byl případ automaticky přiřazen (byl-li jednoznačně určen), opět konečný stav samotného procesu přiřazování, pomocné údaje pro ověření správnosti a související příznak. Všechny údaje uvedené v tomto odstavci pak budou umožňovat a ulehčovat ruční opravy či ruční ohodnocení a přiřazení advokáta ve sporných situacích. Tabulky, které ukládají získané výsledky procesů, a jejich vazbu na případ, znázorňuje obrázek 4.3.



Obrázek 4.3: Část databáze ukládající výsledky procesů

Celkový pohled na databázi a vazby v ní obsažené ilustruje obrázek 4.4[1].



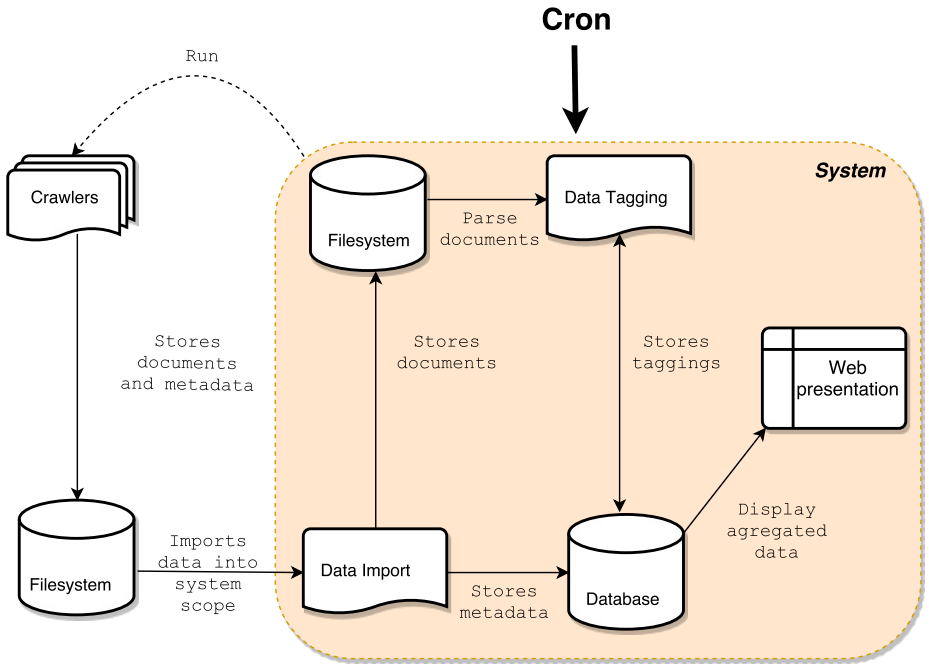
Obrázek 4.4: Návrh uvažované databáze

[1]Pro větší přehlednost jsou v obrázku vynechány servisní tabulky (user, migrations, jobs...)

4.3 Funkce hlavního systému

Jelikož větší část služby bude fungovat automatizovaně, chtěl jsem, aby existovala kontrola nad jednotlivými prováděnými akcemi a jejich výsledky. V systému je proto pro každou funkční část vytvořena uživatelská role, která identifikuje, která část systému je zodpovědná za vložená data. Existence systému také usnadňuje samotné spouštění jednotlivých částí systému, a to tak, že všechny prováděné skripty využívají jedno společné rozhraní pro spuštění. Zároveň je tento systém funkční logikou celé služby.

Za tímto účelem bude ke každé tabulce v databázi přidán údaj o části systému, který je zodpovědný za vložená data a též informace o tom, ve kterém běhu systému byla tato data vložena. Navíc zde bude informace i o běhu samotného skriptu, jako jsou návratový kód, čas a trvání běhu skriptu, případně další informační, debugovací či chybové hlášky. V případě výskytu chyby tak nebude nutné hledat její původ napříč celým systémem, ale bude již jednoznačně určeno, ve které funkční části systému se chyba vyskytla. A navíc bude možné dotčená data identifikovat a v případě nutnosti manuálně opravit. Jednotlivé části systému a tok dat ilustruje obrázek 4.5.



Obrázek 4.5: Architektura systému s dílčími částmi systému

4.4 Popis struktury crawleru

Aby bylo možné získat potřebné informace z webových stránek soudů, je třeba, aby každý crawler obsahoval několik funkcí, které budou zajišťovat potřebnou funkcionalitu. Jedná se o vyplnění vyhledávacího formuláře na určené webové adrese. Mezi dalšími pak průchod napříč stránkami výsledků a uložení každého dokumentu do souboru `HTML`. V neposlední řadě musí každý crawler extrahovat potřebné informace ze struktury `HTML` stránky a to

podle kritérií specifických pro daný soud. A nakonec uložit získané informace do souboru CSV[2] pro další zpracování.

4.5 Ohodnocení dokumentu ve vztahu k advokátovi

Samotný způsob hodnocení advokáta vychází z posouzení výsledku a formy rozhodnutí soudu. Existují čtyři možnosti ohodnocení: [12, §46 a následující] [11, §243c a následující] [10, §43] [13, §265i a následující]

- positive - toto ohodnocení znamená, že dokument je v pořádku, advokát se ve svém jednání a povinnostech nedopustil žádného závažného nedostatku, který by byl důvodem pro odmítnutí podání.
- negative - takto bude dokument ohodnocen, pokud advokát například napsal špatně či nedostatečně vyargumentoval podání, které na soud posílal. Nebo také nedodal všechny nutné podklady. Jinými slovy, chyba se stala na straně advokáta.
- neutral - toto hodnocení obdrží dokumenty, které byly pozastaveny z nějakých jiných důvodů[3], než pochybením na straně advokáta (například klient se rozhodl vzít stížnost zpět).
- unknown - jen podle výsledku rozhodnutí soudu není možné automaticky určit, o který z výše uvedených případů se jedná. Je tedy nutný manuální zásah administrátora.

Mezi podmínkami pro ohodnocení dokumentů u různých soudů jsou drobné nuance. Pro označení dokumentu jako `positive` u Nejvyššího správního soudu je dostačující, aby forma rozhodnutí byla `Rozsudek`. Dále, je-li forma rozhodnutí `Usnesení`, záleží i na způsobu rozhodnutí. V případě, že obsahuje frázi `odmítnuto`, je tento dokument ohodnocen jako `negative`. Pokud by tuto frázi neobsahoval a obsahoval by místo ní `zastaveno`, byl by tento dokument označený jako `neutral`.

```
if($type == $specificForm)
  return RESULT_POSITIVE;
elseif(
  !contains($decision, DECISION_RESULT_NEGATIVE) &&
  contains($decision, DECISION_RESULT_NEUTRAL)
)
  return RESULT_NEUTRAL;
elseif(contains($decision, DECISION_RESULT_NEGATIVE))
  return RESULT_NEGATIVE;
else
  return RESULT_UNKNOWN;
```

Listing 4.1: Ukázka obecného kódu zajišťujícího ohodnocení dokumentu

[2] Comma-separated values

[3] k zastavení řízení dojde i v případě, že není včas zaplacen soudní poplatek (což může a nemusí být chyba advokáta)

Kapitola 5

Formální návrh řešení

Cílem práce je vytvořit službu, která bude využívat veřejné informace získané z webových stránek jednotlivých soudů. Proto je potřeba zvolit podle požadavků na funkcionalitu vhodné nástroje pro práci s webovými stránkami a jejich obsahem. Jak už bylo uvedeno v kapitole 4, zabývám se zpracováním dat Nejvyššího správního soudu, Ústavního soudu a České advokátní komory.

5.1 Analýza požadavků

Než jsem přistoupil k volbě vhodného nástroje, musel jsem provést analýzu požadavků funkcionality na zpracovávaných webových stránkách. Základním požadavkem byla stabilita, možnost běhu na serveru (bez nutnosti internetového prohlížeče). Dále pak existence modulu pro Python.

Nejvyšší správní soud[1]

- interakce s formulářem
- vykonávání JavaScriptu
- vyhledávání odkazu podle textu
- použití CSS selektorů pro výběr elementů
- omezení načítaného obsahu webu (obrázky, JS, Flash...)

Ústavní soud[2]

- interakce s formulářem
- vyhledávání odkazu podle textu
- použití CSS selektorů pro výběr elementů

[1] rozšířená verze formuláře pro vyhledávání je dostupná na adrese:
`http://nssoud.cz/main0Col.aspx?cls=JudikaturaBasicSearch&pageSource=0`

[2] samotný vyhledávací formulář je umístěn na adrese:
`http://nalus.usoud.cz/Search/Search.aspx`

Česká advokátní komora[3]

- interakce s formulářem
- vyhledávání odkazu podle textu
- použití CSS selektorů pro výběr elementů

5.1.1 Uvažované nástroje

Selenium

Selenium je multiplatformní nástroj určený speciálně pro automatizované testování webových aplikací. Selenium je napsáno v jazyce Java, ale poskytuje též API[4] rozhraní pro různé programovací jazyky. Samotný nástroj je tvořen z několika vzájemně se doplňujících komponent:[1]

- Selenium IDE - prostředí pro tvorbu testovacích scénářů pro internetový prohlížeč Mozilla Firefox
- Selenium RC (Remote Control) - systém client/server, který umožňuje tvorbu automatizovaných testů v široké škále programovacích jazyků a pro více internetových prohlížečů
- Selenium WebDriver - jedná se o nový a jednodušší způsob vytváření testů než v Selenium RC, který umožňuje spouštět a kontrolovat prohlížeče separátně, bez nutnosti mít spuštěný Selenium Server (součást Selenium RC)
- SeleniumGrid - komponenta, která umožňuje spouštění testů na několika prohlížečích současně

Mechanize

Mechanize je knihovna implementovaná v jazyce Python, jejímž využitím lze automatizovaně interagovat s webovými stránkami. Jádro knihovny je postaveno na modulu, který je implementován v jazyce Perl. Knihovna umožňuje práci s cookies, vyplňování a odesílání formulářů s využitím CSS selektorů, následování odkazů, dodržování zásad robots.txt a práci s historií prohlížení.[4]

Ghost.py[5]

Ghosty.py je WebKit[6] klient napsaný v jazyce Python. Podobně jako předchozí nástroje umožňuje vzdálenou interakci s webovými stránkami. Konkrétně umožňuje práci s formuláři (vyplňování a odesílání), vykonávání JavaScriptu, pořizování screenshotů, dále umožňuje omezení načítaných zdrojů dané webové stránky (obrázky, JavaScript soubory, Flash animace...). Poskytuje také sadu funkcí, které umožňují vyčkat například na:

- zobrazení konkrétního elementu na stránce

[3] formulář pro vyhledání advokáta se nachází na adrese: `http://vyhledavac.cak.cz/`

[4] Application Programming Interface

[5] `http://ghost-py.readthedocs.io/en/latest/, http://jeanphix.me/Ghost.py/`

[6] vykreslovací jádro prohlížeče a framework, základ webových prohlížečů napříč operačními systémy

- úplné načtení stránky
- načtení stránky po kliknutí na určitý element

5.1.2 Volba nástroje

Vyzkoušel jsem všechny zmíněné nástroje pro získání dat ze stránek Nejvyššího správního soudu. Již při automatizovaném vyplňování vyhledávacího formuláře jsem narazil u některých nástrojů na nedostatky spojené s výběrem hodnoty ze seznamu (`select`). Ukázalo se, že například `Mechanize` vyžaduje zadání hodnoty atributu `value` dané možnosti a nestačí pouze zadání textu, který je v této volbě zobrazen. Po vyplnění formuláře jsem se zaměřil na procházení stránkami výsledků, kde jsem potřeboval následovat odkaz, který je určen číslem následující stránky. Konkrétně v případě těchto stránek je po kliknutí na odkaz vyvolána funkce JavaScriptu, která se postará o přechod na konkrétní stránku. Je tedy výhodnější zavolat tuto funkci přímo, než vyhledávat element ve stránce a na ten kliknout. Efekt bude stejný. Potřeboval jsem tedy, aby nástroj podporoval vykonávání JavaScriptu, což vyloučilo jinak dostačující nástroj `Mechanize`.

Po tomto prvotním experimentu bylo na čase vyzkoušet zpracování (pouze projití stránek) všech výsledků vyhledávání. V této fázi jsem ověřoval stabilitu nástroje při velkém počtu požadavků na cílový server. Lokálně spuštěný server nástroje `Selenium RC` k mému překvapení nestačil zpracovávat všechny požadavky na server a jeho odpovědi. Běh skriptu se postupně zpomaloval, až nakonec lokální server spadl. Naproti tomu skript využívající nástroje `Ghost.py` zvládl zpracovat všechny požadavky a odpovědi serveru a skončil korektně. Je sice pravda, že vykonávání skriptu trvalo delší dobu, ale vzhledem k množství procházených stránek, a prvotní/experimentální verzi skriptu, je to pochopitelné. Na základě výše popsané analýzy a výsledku zkušebního testování jsem se rozhodl využít pro implementaci crawleru `Ghost.py`.

5.2 Použité nástroje

Při volbě dalších nástrojů jsem vycházel z provedené analýzy a zkušeností s nástroji používanými pro zpracování webového obsahu při školních projektech v jazyce Python.

5.2.1 Knihovny a moduly jazyka Python

Virtualenv

Virtualenv je nástroj pro vytváření separovaných Python prostředí. Umožňuje instalaci a správu Python modulů pro každý projekt, bez potřeby instalace modulů do hlavního prostředí Pythonu použitého operačního systému. Zlepšuje přehlednost a pomáhá udržovat v projektu jen ty moduly, které jsou opravdu použity.[7]

BeautifulSoup

Jedná se o modul, který umožňuje snadné parsování `HTML/XML` dokumentů. Poskytuje dále rozhraní pro průchod DOM[8] a manipulaci s jednotlivými elementy. V této práci se jedná o stěžejní modul, díky kterému lze parsovat stažené dokumenty a extrahovat z nich potřebná data a to již bez potřeby aktivního internetového klienta (offline).[7]

[7] čerpáno ze stránek `https://virtualenv.pypa.io/en/stable/`
[8] Document Object Model

Pandas

Knihovna primárně určená pro analýzu tabulkových dat. Umožňuje pracovat efektivně a rychle s velkým množstvím dat.[9]

5.2.2 Utility

curl – curl je nástroj příkazové řádky pro transfer dat z nebo na server (upload/download), který využívá a podporuje přenosové protokoly `HTTP, HTTPS, FILE, FTP, FTPS` a mnoho dalších. Funkčnost zajišťuje knihovna `libcurl`, kterou lze použít napříč programovacími jazyky, například pro Python, PHP, JAVU.[10]

cron – cron je nástroj umožňující naplánování spouštění vlastních úloh (příkazů, skriptů), které mají být vykonávány opakovaně v určitých časových intervalech (denně, týdně, každou hodinu...). Jeho použití je základem pro autonomní, automatizované, pravidelné provádění přesně definované činnosti.

pdftotext – jedná se o nástroj příkazové řádky, který umožňuje převod dokumentu PDF do prostého textu. Extrahuje data z textové vrstvy vložené do souboru PDF. Tento nástroj je dostupný napříč operačními systémy, získal si oblibu u Linuxových distribucí, kde je obsažen již v základním balíčku nástrojů. Na jeho základech byly vybudovány i další nástroje, jako jsou Xpdf nebo Poppler.

composer – composer je multiplatformní nástroj určený pro správu závislostí jazyka PHP (knihoven a dalších zdrojů). Umožňuje snadnou kontrolu nad knihovnami a jejich závislostmi napříč projektem. Udržuje knihovny aktuální a poskytuje rozhraní, které umožňuje udržet konzistenci projektu při práci v týmu.[11]

npm – obdobným nástrojem jako `composer` je pro jazyk Javascript `npm`. Jedná se o základního správce balíčků pro prostředí `Node.js`. Pomáhá udržovat závislosti mezi poskytovanými balíčky a zajišťuje kontrolu jejich aktualizací.[12]

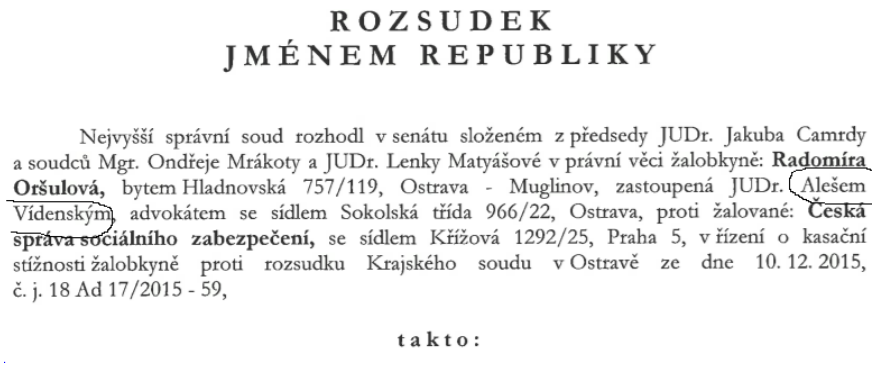
Git – git je open-source, multiplatformní, distribuovaný, hojně využívaný systém pro správu verzí. Jeho používání výrazně usnadňuje vývoj softwaru v týmu. Git využívá mnoho světových špiček při vývoji svého softwaru. Git se také stal synonymem pro open-source projekty a v současné době existuje mnoho serverů, které využívání Gitu nabízejí. Pro správu zdrojových kódů této práce bylo konkrétně využito služby `https://github.com/`.[2]

5.3 Extrakce textu rozhodnutí

Jelikož texty rozhodnutí Nejvyššího správního soudu jsou dostupné jen ve formátu PDF, je nutné z nich samotný text rozhodnutí extrahovat. V případě, že se jedná o PDF s textovou vrstvou, není tento export náročný. Ovšem v případě, že zmíněné PDF obsahuje jen naskenované stránky originálního rozhodnutí, je nutné využít pro získání textu metod OCR[13].

[9] čerpáno z oficiální dokummentace `http://pandas.pydata.org/pandas-docs/stable/`
[10] zdrojem informací je web `https://curl.haxx.se/`
[11] čerpáno z oficiálních stránek projektu - `https://getcomposer.org/`
[12] čerpáno z oficiálních stránek projektu - `https://www.npmjs.com/`
[13] Optical Character Recognition

Pro vyhledávání jmen advokátů je potřeba pouze první stránka dokumentu, na které jsou v hlavičce definovány jednotlivé strany případu a jejich právní zastoupení. Hlavičku takového dokumentu s vyznačeným jménem právního zástupce žalobce zobrazuje obrázek 5.1.



ROZSUDEK
JMÉNEM REPUBLIKY

Nejvyšší správní soud rozhodl v senátu složeném z předsedy JUDr. Jakuba Camrdy a soudců Mgr. Ondřeje Mrákoty a JUDr. Lenky Matyášové v právní věci žalobkyně: **Radomíra Oršulová,** bytem Hladnovská 757/119, Ostrava - Muglinov, zastoupená JUDr. Alešem Vídenským, advokátem se sídlem Sokolská třída 966/22, Ostrava, proti žalované: **Česká správa sociálního zabezpečení,** se sídlem Křížová 1292/25, Praha 5, v řízení o kasační stížnosti žalobkyně proti rozsudku Krajského soudu v Ostravě ze dne 10. 12. 2015, č. j. 18 Ad 17/2015 - 59,

takto:

Obrázek 5.1: Ukázka hlavičky textu rozhodnutí Nejvyššího správního soudu

Prvotní úvahou bylo využít k převodu některý ze specializovaných serverů, který službu OCR převodu poskytuje.

5.3.1 Online převod

U nalezených serverů byl základní problém v tom, že ne všechny služby poskytovaly převod PDF dokumentů, případně neumožňovaly převod pouze první strany. Bylo tedy nutné převést první stránku PDF dokumentu na JPEG. Dalším problémem byla například nutnost vyplnit captcha[14] ověření před každým převodem, což znemožňuje využití takové služby pro automatizovaný převod. Některé servery poskytující službu převodu OCR dokumentu byly ovšem na takové úrovni, že poskytovaly vlastní API, čímž podstatně ulehčily opakované a automatizované použití. U těchto serverů jsem ovšem narazil na jiný problém. Některé servery umožňovaly využít API bezplatně, jeho využití však bylo limitováno maximálním počtem požadavků za měsíc. Vezmeme-li v úvahu množství dokumentů, které je potřeba zpracovat (databáze za období několika let obsahuje zhruba 40 000 záznamů), potřebovali bychom jednorázově (nebo aspoň v rozumné době) provést srovnatelné množství požadavků. Převod pravidelně získávaných dokumentů už by se pravděpodobně do měsíčního limitu vešel. Pro převedení dokumentů stávající databáze by tedy bylo potřeba využít některého z placených tarifů.

Server	Formát dokumentu	Free	API	Cena
ABBYY Cloud OCR SDK	BMP, PNG, JPEG, PDF...	Ano, 100/měsíc	Ano	$839.99
Free Online OCR	JPEG, PNG, PDF...	Ano, 100/den	Ano	-
Free OCR API	PDF, obrázky	Ano, 25000/měsíc[15]	Ano	$49.95
OCR WEB SERVICE	PDF, TIFF, JPEG, PNG...	Ano, 25/den	Ano	$299.95

Tabulka 5.1: Přehled zkoumaných serverů poskytujících API pro OCR převod dokumentů

[14] completely automated public Turing test to tell computers and humans apart
[15] maximálně 500/den

5.3.2 Lokální převod

Další možností je využít některé z volně dostupných utilit, které lze spustit na serveru prostřednictvím terminálu.

Tesseract – teseract je engine pro optické rozpoznávání znaků (OCR), který byl původně vyvíjen společností Hewlett-Packard. Byl publikován jako hlavní část vědeckého výzkumu v rámci doktorského studia v laboratořích Hewlett-Packard. Později byl Tesseract uvolněn jako open-source pod Apache License. Na jeho vývoji se v současné době podílí společnost Google. V repozitáři `tesseract-ocr`[16] jsou dostupná trénovací data a natrénované modely pro mnoho světových jazyků (včetně češtiny). Na tomto enginu je postavena většina komerčních i nekomerčních aplikací.[9]

Gocr – gocr je program pro optické rozpoznávání znaků (vyvíjený pod licencí GPL), který lze použit z příkazové řádky. Jako vstup přijímá formát `PNM, PGM, PBM, PPM, PCX` dále pak `PNG, JPG, JPEG, TIFF, GIF` a `BMP`. Rozpoznaný text vypisuje na standardní výstup. Svými parametry spuštění umožňuje korekci metod použitých pro preprocessing i samotné rozpoznávání. Například určení úrovně šedé, míry zašumnění obrazu nebo velikosti mezer mezi slovy.[17]

Jak bylo naznačeno v předchozím textu, možnosti vzdáleného převodu nejsou dobře dostupné, jedná-li se o velký počet dokumentů, které je potřeba převést. Za účelem převodu bylo tedy vybráno lokální řešení, konkrétně `teseract`, protože poskytuje již natrénované modely pro český jazyk.
Při experimentování s tímto programem se vyskytovaly u převedeného textu tyto vady:

- záměna písmene
- sloučení písmen
- vynechání písmene
- chybný převod či záměna interpunkčních znamének
- generování mezer

Jelikož bylo plánováno nad takto získaným textem provádět vyhledávání specifické fráze – pro identifikování místa výskytu jména advokáta, byla kvalita výsledného textu nutnou podmínkou. Pro vyhledávání v tomto textu bylo použito regulárních výrazů, v důsledku toho byla jména advokátů získaná z těchto textů většinou zkomolená a nebo neúplná.

Během doby, která byla věnována úpravám stávajícího postupu tak, aby poskytoval lepší výsledky, se naskytla možnost získávat informace o advokátech působících u daných řízení přímo od příslušných soudů. Měli bychom tak k dispozici jméno advokáta a spisovou značku případu, u kterého působil. Jelikož je informace o jméně advokáta daného případu stěžejní, přistoupili jsme na variantu využít data dodaná soudy. Tyto informace jsou sice zpoplatněny a na jejich dodání budeme čekat, ale jsou relevantní a ověřené.

[16] https://github.com/tesseract-ocr
[17] čerpáno z manuálové stránky `man gocr`

Kapitola 6

Implementace, realizace

Tato kapitola dokumentuje můj postup práce při realizaci dle návrhu a dále problémy, které se objevily v průběhu implementace jednotlivých částí systému. Problémy jsou zde podrobně popsány a následně je uvedeno, jak jsem přistoupil k jejich řešení v rámci samotné implementace. Největší část popisu je věnována implementaci vlastních crawlerů (viz 4.1.1), jelikož zde se objevilo nejvíce problémů k řešení.

6.1 Průchod mezi stránkami výsledku - NSS

Jelikož stránkování výsledku u tohoto soudu neposkytuje jednoduchý element pro přechod na následující stránku výsledků („další“) a ani neprovádí přechod pomocí argumentů v adresovém řádku, musel jsem implementovat způsob, který bude procházet stránky postupně. Podle celkového počtu záznamů v daném rejstříku (viz Spisová značka) a množství zobrazených záznamů (volitelný v nabídce formuláře) na stránce jsem spočítal počet stránek, který je potřebný pro průchod všemi záznamy.

Dalším krokem bylo nalezení konkrétního elementu, který by umožnil přechod na požadovanou stránku. Navigace stránkování výsledků obsahuje 24 elementů (číslovaných od 0) s tím, že první i poslední jsou absolutními odkazy na první a poslední stránku výsledků. Moje prvotní představa o dosazování čísla požadované stránky do `id` některého elementu byla platná do té doby, dokud byl počet stránek všech výsledků menší než 23. V tomto případě byly zobrazeny všechny prvky navigace a `id` postupovalo logicky (první|1-22|poslední) jak dokládá obrázek 6.1.

« První 1 2 3 4 5 6 7 8 9 [10] 11 12 13 14 15 16 17 18 19 20 21 22 Poslední »
Zobrazeno 91-100 z 533

Obrázek 6.1: Zobrazení chování navigace do stránky 12 (číslo stránky odpovídá `id` elementu)

Jakmile byl celkový počet stránek výsledků vyšší a číslo vybrané stránky převyšovalo 12, označení aktuální stránky se v navigaci přesunulo na pozici 12 (`id=11`), což odpovídá prostřednímu prvku navigace, rozložení navigace v tomto případě lze vidět na obrázku 6.2.

« První 2 3 4 5 6 7 8 9 10 11 [12] 13 14 15 16 17 18 19 20 21 22 23 Poslední »
Zobrazeno 111-120 z 533

Obrázek 6.2: Chování navigace od stránky 12 - dochází k posunu číslování celé navigace

Dále pak bylo `id` elementu konstantní (`id=12`) opět až do doby, kdy zbývalo do konce 10 a méně stránek výsledků. Chování navigace u posledních stránek ilustruje obrázek 6.3. Hodnota, která se přičetla k základu 12 je závislá na vzdálenosti požadované stránky od konce procházení. K tomu jsem využil pole s hodnotami 0-10 a přístup prostřednictvím záporného indexu, což Python podporuje.

Obrázek 6.3: Zobrazení chování navigace ke konci průchodu

Po výpočtu `id` požadovaného elementu následovalo jeho stisknutí, to ovšem vyvolalo jen Javascriptovou funkci, jejíž jedním z parametrů bylo právě `id` elementu. Ve výsledku je tedy jednodušší volat přímo tuto funkci, než vyhledávat prvek na stránce a následně spustit akci navázanou na klik myší.

K případu, kdy je stránek více než 22 se dostaneme jen zřídka, jelikož při pravidelném týdenním stahování nejsou přírůstky tak veliké. Jinak tomu je v případě stažení celé prvotní databáze případů od doby existence soudu a zveřejnění digitalizovaných dokumentů.

Při přechodu na jiný rejstřík se pak web choval neočekávaně a přistoupil na stránku se stejným číslem, na kterém skončil průchod výsledků předchozího rejstříku, a když to nebylo možné, byla zobrazena poslední stránka výsledků nového rejstříku. Pro korektní zpracování všech výsledků jsem implementoval chování, při kterém se přechází po změně rejstříku vždy na první stranu výsledků.

6.2 Stažení HTML dokumentů a extrakce dat - NSS

U Nejvyššího správního soudu jsou všechny nejdůležitější informace uvedeny již v souhrnu výsledků a není tedy v současné době potřeba přecházet na stránku s detailnějšími informacemi o případu. Je možné stáhnout celou stránku s výsledky a následně z ní najednou extrahovat informace týkající se několika případů. Jednotlivé buňky tabulky, do které jsou uspořádány informace o jednotlivých případech – námi požadované informace, neposkytují žádné upřesňující identifikátory či použité třídy stylů `CSS`. Pro nalezení požadovaných elementů je tedy využito konkrétní cesty ve struktuře `DOM`. Použitá knihovna BeautifulSoup (5.2.1) poskytuje metodu pro získání textu z `HTML` elementu. Ve své podstatě odstraňuje z obsahu elementu všechny `HTML` tagy. Nezvládá tedy úplně správně práci s víceřádkovým textem, který je řádkován pomocí `HTML` elementu `<br>` a dochází zde ke slévání textu. Bylo tedy nutné implementovat rozdělení víceřádkového textu tak, aby byl text korektní a čitelný. Kupříkladu informace o formě a způsobu rozhodnutí je případem víceřádkového textu, kdy první řádek obsahuje informaci o formě rozhodnutí a následující řádek pak informaci o konkrétním způsobu rozhodnutí. Vlastního dělení víceřádkového textu jsem pak ještě využil v případě získávání účastníků řízení.

Všechny extrahované informace jsou následně ukládány do souboru CSV, který obsahuje sloupce společné pro všechny soudy a dále několik sloupců, které jsou specifické pro tento daný soud.

6.3 Ohodnocení případu

Tento proces již pracuje s daty uloženými v námi vytvářené databázi, forma, kterou tato data mají, je tedy předem známa. Základním úkolem tohoto procesu je najít mezi dokumenty, které souvisí s případem, ten poslední (nejnovější) a v případě, že jej lze ohodnotit, provést jeho ohodnocení. Jak již bylo nastíněno v podkapitole 4.5, mezi podmínkami pro jednotlivé soudy jsou drobné nuance. Jedná se o rozdíly ve frázích, které označují formu rozhodnutí.

Je-li forma rozhodnutí „Nález“ (pro Ústavní soud) a nebo „Rozsudek“ (pro Nejvyšší správní soud) znamená to, že advokát se ve svém jednání a povinnostech nedopustil žádného závažného nedostatku, který by byl důvodem pro odmítnutí podání. A proto bude dokument s touto formou rozhodnutí ohodnocen jako `possitive`.

Další formou rozhodnutí je pak „Usnesení“ (společné pro Ústavní i Nejvyšší správní soud), kdy je pro ohodnocení potřeba podívat se i na konkrétní způsob rozhodnutí. Způsobů rozhodnutí existuje mnoho, ale pro účely hodnocení je důležitý fakt, že každý způsob rozhodnutí, který obsahuje frázi „odmítnuto“ naznačuje pochybení na straně advokáta, který chybně napsal či nedostatečně vyargumentoval podání, které posílal na příslušný soud. Konkrétní důvod, pro který bylo podání odmítnuto je pak možné zjistit po pečlivém prostudování textu samotného dokumentu - v případě Ústavního soudu je pak důvod uveden přímo v textu způsobu podání. V případě přítomnosti fráze „odmítnuto“ pak je příslušný dokument označen jako `negative`.

Nakonec existují i případy, které jsou pozastaveny z jiných důvodů, než je pochybení na straně advokáta, těchto důvodů existuje opět velké množství, ale pro lepší představu můžeme uvést např. stáhnutí stížnosti ze strany klienta, nezaplacení soudních poplatků či úmrtí stěžovatele. Tyto případy se vyznačují přítomností fráze „zastaveno“ ve způsobu rozhodnutí. Jelikož v těchto případech není zastavení řízení v důsledku přímého přičinění advokáta, je pak tento konkrétní dokument označen jako `neutral`.

Aby byly pokryty všechny varianty získaných dat, jsou veškeré další formy rozhodnutí jednotlivých soudů označeny jako `unknown`. Jedná se o různá stanoviska pléna a další formy rozhodnutí (např. Jinak), kde není možné jen podle formy a způsobu rozhodnutí automaticky rozhodnout, zda se jedná o některý z výše uvedených případů. Pro ohodnocení těchto dokumentů je pak potřeba zásahu administrátora, který má kvalifikaci v oblasti práva a dokáže text rozhodnutí správně interpretovat a vyvodit z něj příslušné závěry.

I když oblast porozumění textu počítačem je v současné době na vysoké úrovni, obávám se, že i při dnešních možnostech není správné pochopení netriviálního právního dokumentu možné. Prioritu a podmínky jednotlivých ohodnocení lze shrnout jako:

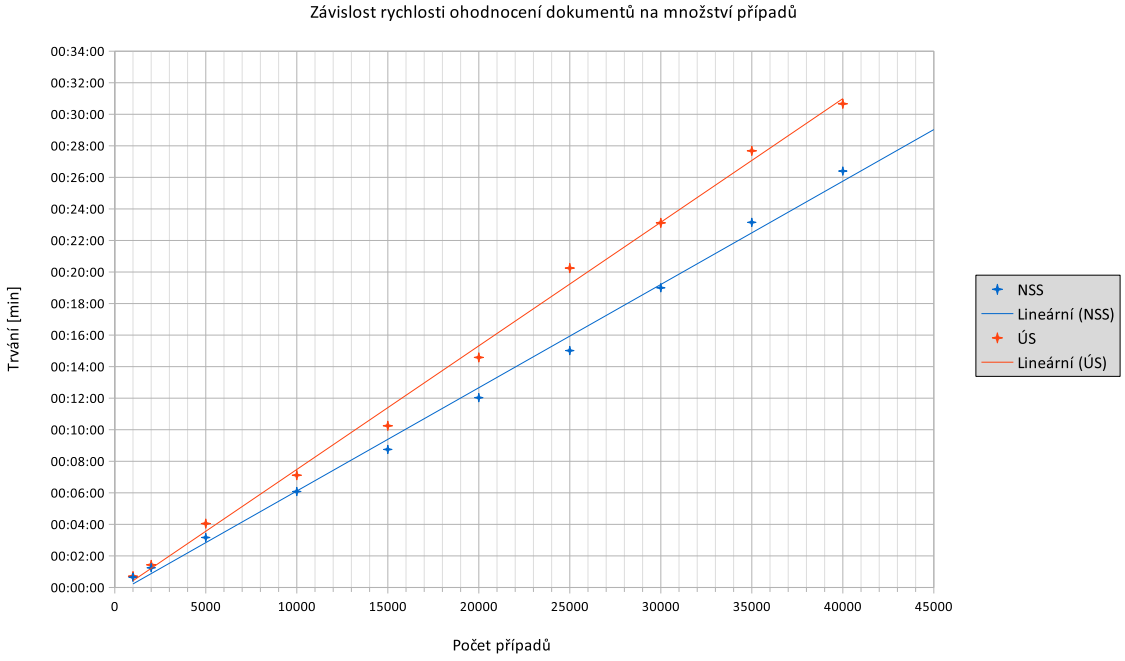
1. Rozsudek/Nález -> `possitive`
2. Usnesení, odmítnuto -> `negative`
3. Usnesení, zastaveno -> `neutral`
4. ostatní formy a způsoby rozhodnutí -> `unknown`

Do procesu hodnocení jsou vždy zahrnuty všechny dokumenty. V případě, že se hodnocení dokumentu shoduje s tím, které již je v databázi uloženo, nové hodnocení se neukládá. V opačném případě je hodnocení uloženo. Jelikož postupem času bude databáze větší a obsáhlejší, zavedli jsme mechanismus, který nebude opakovaně zahrnovat do tohoto procesu ohodnocené dokumenty, kterým byl nastaven příznak `is_final`. Tímto příznakem jsou

označeny všechny ohodnocené dokumenty, které již prošly ruční validací. Zavedení tohoto opatření povede v konečném důsledku k urychlení celého procesu ohodnocování dokumentů.

6.3.1 Doba trvání

Doba trvání celého procesu se liší podle množství zpracovávaných dokumentů a množství již zpracovaných, a v databázi uložených, ohodnocení. U Nejvyššího správního soudu se doba zpracování pohybuje od 7 do 10 minut, přičemž je zpracováváno cca 36000 případů. U Ústavního soudu je pak doba zpracování v rozmezí od 25 do 34 minut. V tomto případě je zpracováváno přibližně 66000 soudních kauz. Dobu trvání a její tendenci pro jednotlivé soudy znázorňuje graf na obrázku 6.4.



Obrázek 6.4: Graf zobrazující lineární závislost trvání procesu ohodnocení dokumentu na množství případů

6.4 Vyplnění vyhledávacího formuláře - ÚS

Stejně jako v předchozím případě, i u tohoto soudu bylo potřeba před zpracováním samotných výsledků vyplnit vyhledávací formulář požadovanými kritérii. Ovšem vyplnění některých údajů bylo poněkud složitější.

U případů, které byly podány až po roce 2006, začal soud rozlišovat typ řízení. Pro vytvoření naší služby jsme potřebovali typ řízení nazvaný „O ústavních stížnostech“. Pole umožňující filtrování právě podle tohoto typu však umožňovalo pouze vyplnění prostřednictvím připravené komponenty obsahující zaškrtávací seznam jednotlivých typů řízení. Tato komponenta se ovšem otvírá jako další okno prohlížeče aktivované voláním Javascriptové funkce. Jelikož mnou zvolený nástroj pro interakci s webem nepodporuje více oken, byl jsem nucen tuto situaci řešit jinak. Vypozoroval jsem, že volba vybraného typu řízení je před-

vyplněna po opětovném navštívení stránky, pravděpodobně za využití cookies[1]. Dále jsem zjistil, že samotná komponenta pro výběr typu řízení je jen další stránka webu. Provedl jsem tedy nejdříve přechod na URL komponenty se seznamem pro výběr, provedl výběr a následně přešel na stránku s vyhledávacím formulářem, kde již byla předvyplněna volba, kterou jsem provedl na stránce externí komponenty.

Následně bylo potřeba vyplnit datum, případně volbu pro vyhledání přírůstků pouze za několik dní. Dále formulář obsahuje volbu pro počet záznamů na stránce, přičemž její využití může výrazně omezit počet procházených stránek výsledků. Možnost zobrazení nabývá hodnot 10, 20, 40 a 80 záznamů na stránku. Využití většího množství záznamů na stránku mělo ovšem za následek výrazné prodloužení doby potřebné k načtení stránky. Při 80 záznamech na stránku to činí téměř 5s, zatímco při 20 záznamech jen necelé 2s.

Ukázalo se ale, že stránka s výsledky neposkytuje všechny potřebné informace o případu, jako tomu bylo u Nejvyššího správního soudu. V tomto případě je potřeba přejít na stránku s detailem případu, kde jsou obsaženy požadované informace.

Prvotní úvahou bylo extrahovat ze stránky výsledku vyhledávání URL adresy jednotlivých záznamů a dále použít pro stažení stránky například `curl` a získat tak potřebné podklady pro extrakci požadovaných dat. Ukázalo se, že stažená data obsahovala zdrojový kód vyhledávacího formuláře a to i přesto, že byla zadána URL ke konkrétnímu případu. Tuto skutečnost jsem ověřoval ruční kontrolou URL adres a zobrazovaného obsahu. Pří zadání adresy byl prohlížeč nejprve přesměrován na stránku s vyhledávacím formulářem. Po opakovaném zadání adresy se zobrazil detail případu, ale část s textem stále chyběla. Vyhledal jsem tedy tento případ pomocí ECLI[2] a zobrazil si jeho detail. K mému překvapení se na stránce detailu část s textem rozhodnutí nacházela. Usoudil jsem, že popsané chování zřejmě souvisí s odesláním formuláře a faktem, že jsou stránky Ústavního soudu postaveny na technologii ASP.NET. Řešení tohoto problému vyžaduje nízkoúrovňový přístup a dobrou znalost frameworku .NET. [3] Dříve, než jsem započal studium této technologie, hledal jsem jiný přístup pro dosažení požadovaných výsledků. Vzhledem k výše uvedenému bylo tedy potřeba simulovat chování běžného uživatele a zajistit tak korektní chování aplikace.

6.4.1 Vyzkoušené varianty

Přechod mezi detaily

Na stránce detailu případu se nachází několik ovládacích prvků, s jejichž pomocí by mohlo být procházení všemi případy výsledků jednodušší. Detail obsahuje navigaci, která umožňuje přechod na další a předchozí případ. Oba elementy mají dokonce své vlastní `id`, takže jejich identifikace a nalezení ve struktuře `HTML` stránky je snadné. Implementoval jsem tedy cyklus, který v přítomnosti elementu `GotoNextId` na něj klikne a přejde na detail následujícího případu.

Při zkoušení na reálných datech jsem ovšem došel ke znepokojivému zjištění, a to, že tento způsob přecházení mezi detaily případů je funkční jen do té doby, dokud je pořadí případu v celkových výsledcích menší než 1000. Dále již není možné načíst další záznam a dojde

[1] datové soubory malého rozměru, které jsou při návštěvě webové stránky bez aktivního jednání ze strany uživatele ukládány do prohlížeče zařízení, kterým je v daný okamžik připojen k internetu `https://www.epravo.cz/top/clanky/pravni-povaha-cookies-98982.html`

[2] Identifikátor evropské judikatury (European Case Law Identifier - ECLI)

[3] součást .NET Frameworku firmy Microsoft pro tvorbu webových aplikací a služeb – umožňuje tvorbu webových aplikací v jazyce C#

k přesměrování na poslední zobrazený záznam. Při samotném přecházení mezi jednotlivými záznamy se nemění žádné parametry v URL adrese a je tedy obtížné identifikovat tento problém. Vypozoroval jsem, že čas potřebný k přechodu mezi záznamy souvisí s množstvím záznamů zobrazených na jedné stránce vyhledávání.

Návrat na stránku s výsledky

Dalším způsobem, jak projít a zpracovat všechny výsledky, bylo přejít vždy na stránku detailu, tuto uložit a vrátit se na stránku s výsledky. Znamenalo to ale vyhledat na stránce s výsledky vždy následující element s odkazem vedoucím na detail stránky, na ten kliknout a v průběhu zpracování jedné stránky výsledků se opakovaně vracet na původní stránku s výsledky.

Jelikož jednotlivé elementy s odkazy na detail nejsou nijak jednoznačně identifikovatelné, získal jsem ze stránky všechny odkazy a hledal ty, jejichž `CSS` třída odpovídala `CSS` třídě sudého (`resultData0`) nebo lichého (`resultData1`) řádku. Jelikož objekt elementu knihovny BeautifulSoup (získaný parsováním stránky) a objekt knihovny `Ghost.py` (elementu pro interakci) nebyly vzájemně kompatibilní, musel jsem přistoupit k vyhledání elementu pro interakci na základě atributu `href`, jehož hodnotu relativního odkazu jsem získal při parsování stránky.

I přes získání seznamu všech potřebných odkazů, a tím pádem i potřebných elementů, bylo potřeba po každém návratu na stránku s výsledky vyhledat opětovně následující element, protože reference na dříve vyhledaný element již zanikla. Vezmeme-li v potaz 80 záznamů na stránce s výsledky, pak by bylo nutné provést 80 kliknutí, respektive přechodů na jinou stránku a dalších 80 přechodů zpět na stránku s výsledky. Pozorováním a experimentováním jsem zjistil, že přechod na stránku detailu zabere v průměru 1,2 sekundy a návrat na stránku s výsledky pak v průměru 800 milisekund. To znamená 160 sekund na průchod jediné stránky s výsledky. Dále pak průchod mezi jednotlivými stránkami, díky velkému počtu výsledků na stránku, činí v průměru dalších 5 sekund.

Při zkoumání této možnosti jsem navíc vypozoroval, že v tomto způsobu provedení se mění parametry v URL adrese detailu, a to konkrétně argument `id`, který jednoznačně určoval daný případ. Tento parametr jsem tedy využil jako název souboru pro ukládaný `HTML` soubor, přičemž jsem se vyhnul nutnosti parsování stránky detailu opakovaně (nyní a při extrakci dat). V tomto případě pak bylo také nutné implementovat přechod na další stránku, přičemž stránka s výsledky obsahuje prvek, který umožňuje přechod na další stránku, ten však jen změní URL adresu, ve které je parametrem právě číslo následující stránky. Přechod na další stránku prostřednictvím interakce s tímto elementem, či přechod přímo na URL adresu, trvá stejnou dobu. Ovšem otevření vlastní URL adresy separátně poskytuje větší kontrolu nad průchodem výsledky a minimalizuje operace vyhledání prvku ve struktuře HTML stránky.

Při zpracování velkého množství stránek se navíc objevily výpadky, a tak jsem implementoval mechanismus, který dokáže navázat na předchozí nedokončené stahování a to právě podle poslední zpracované stránky, jejíž číslo se vždy po zpracování uloží do souboru `current_page.ini`. Existence tohoto souboru se ověřuje při každém spuštění a číslo v něm uložené se pak použije jako počáteční číslo stránky s výsledky, odkud pokračuje stahování. Při korektně dokončeném stahování je pak tento soubor smazán.

Přímý přístup na detail

Výše popsané řešení bylo funkční a tedy použitelné, jen se vzrůstajícím počtem stránek výsledků se doba průchodu výrazně prodlužovala. Hledal jsem tedy způsob, jak průchod urychlit.

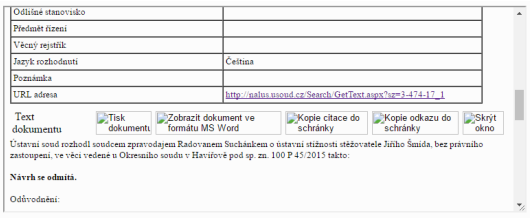
Uvědomil jsem si, že na stránku s výsledky se vždy vracím s tím, že na ní jen vyhledám následující element s odkazem, jehož adresu již znám. Napadlo mě tedy rovnou přejít na onu adresu detailu (vytvoření absolutního odkazu není složité) a tímto způsobem navštívit všechny detaily na dané stránce výsledků. Po jejich vyčerpání pak přejít na další stránku a celý postup opakovat. Jak bylo zmíněno výše, přechod na další stránku je opět otevření URL adresy, kde je číslo požadované stránky argumentem. V konečném důsledku se tedy interakce se stránkou s výsledky omezila na vyhledání adres odkazů vedoucích na detail případu. Tímto přístupem jsem ušetřil čas, který byl dříve potřebný na opětovný návrat ze stránky detailu případu na stránku s výsledky.

Původní čas činil cca 800 milisekund na jeden případ. Úspora vzniklá popsaným opatřením přinesla urychlení o cca 38%, což činí v konečném důsledku, při 80 záznamech na stránku, 64 sekund na jednu stránku s výsledky oproti původní variantě.

Potlačení inline Javascriptu a CSS

V rámci stažení dokumentu s detailem bylo provedeno odstranění inline `CSS` stylu, který způsoboval špatnou čitelnost dokumentu - v dokumentu bylo zobrazeno příliš mnoho posuvníků (scrollbar), protože tabulky s údaji měly velikost uzpůsobenou rozměrům okna prohlížeče. Při načtení stránky pak byla, prostřednictvím argumentu `onload` v `HTML` elementu `<body>`, volána Javascriptová funkce, která velikost těchto tabulek dynamicky změnila, čímž došlo k přepsání původních hodnot `CSS` stylu. Na malém zařízení tak byla čitelnost stránek problematická. Zpracování takto vizuálně nečitelných stránek není žádným problémem. Úpravy stránky jsou prováděny za účelem zobrazení náhledu dokumentu v administračním rozhraní pomocí `HTML` tagu `<iframe>`. Ten obvykle zaujímá jen část stránky a jeho omezený rozměr by se bez popsaného zásahu promítl do velikosti jednotlivých elementů zobrazované stránky a způsobil by zmenšení prvků pod únosnou mez. Porovnání varianty původního dokumentu a upraveného dokumentu potlačením inline `CSS` a Javascriptu znázorňuje obrázek 6.5.

(a) Původní dokument při malé velikosti okna



(b) Upravený dokument zobrazený v administračním rozhraní, jako iframe

Obrázek 6.5: Rozdíl v zobrazení dokumentu před a po provedené úpravě

6.5 Přiřazení případu advokátovi

Následující text popisuje problém získávání jmen z textu rozhodnutí soudu a použité postupy. Dále je představen proces filtrování dat a mechanismus přiřazování případů advokátovi.

6.5.1 Jména advokátů

Postup získávání jmen potřebných pro proces přiřazování byl nastíněn v kapitole 5. Zdrojem pro vyhledávání je text, který je získán převodem PDF na text za pomoci technologie OCR. Takto získaný text vykazoval vady, mezi které patřily například záměna písmene, sloučení písmen, vynechání písmene, chybný převod či záměna interpunkčních znamének. Vyhledávání nad takto zkomoleným textem nebylo nikterak jednoduché a využití regulárního výrazu, který nad korektním textem fungoval obstojně, bylo s úpravami možné. Řetězce vyhovující upravenému regulárnímu výrazu nebyly ale vždy požadovaným obsahem. Připočteme-li ještě vlastní chybovost původních textů - například chybějící interpunkce u titulů, různé způsoby oddělování jednotlivých titulů, používání zkratkovitých výrazů či překlepy, byla získaná data neúplná nebo zcela chybná. Získané řetězce navíc byly v sedmém pádě, což byl tvar, který sice vyplýval z kontextu textu, ale pro samotné porovnávání nebyl vhodný. Další otázkou tedy byla lemmatizace získaných řetězců do tvaru prvního pádu. Lemmatizace samotná je velmi složitý problém, vyžadující obsáhlou databázi jmen a příjmení a jejich nejrůznějších tvarů. Hledal jsem, zda v prostředí internetu neexistuje nějaká služba, která by problém transformace jména z jednoho pádu do jiného řešila. Nalezl jsem web Skloňování jmen[4], který poskytuje převod jména z 1. pádu na oslovení v 5. pádu a disponuje vlastním API. Vznesl jsem tedy dotaz, zda poskytují i mnou požadovaný převod ze 7. pádu na první. Odpověď byla bohužel negativní.

Další možností, která se nabízela, pak bylo hledání částečné shody v textu. Druhým textem, určeným k porovnávání, bylo jméno a příjmení advokáta, které jsem získal stažením a extrakcí dat z webových stránek České advokátní komory (viz 6.6). V průběhu řešení tohoto problému se ale objevila možnost získávat jména advokátů přímo od soudu (a ve správném tvaru), což se vzhledem k vývoji dalších částí systému jevilo jako nejrozumnější řešení alespoň do doby, než bude implementován celý systém a bude řádně odladěn. Zdržení vzniklé řešením samotného problému lemmatizace by mohlo mít za následek nedostatek času k vývoji a provázání zbylých součástí systému.

Data dodaná soudy jsou při importu do databáze situována do sloupce `official_data` v tabulce nesoucí informace o případu (`case`). Soudy nám sice poskytly informaci o právním zastoupení stěžovatele u případu, ale ne vždy se jednalo o jméno advokáta. Kupříkladu u Nejvyššího správního soudu byla tato data částmi textů rozhodnutí, která v sobě obsahovala informaci o právním zastoupení, ale ne jen konkrétní jméno. V dodaném dokumentu sloupec „advokát“ obsahoval i názvy advokátních a dalších kanceláří (auditorských či daňových). Dále pak názvy firem a institucí, jména zaměstnanců s právním vzděláním, či informaci, že stěžovatel je sám advokátem - ovšem jméno stěžovatele samotného už chybělo. Bylo tedy potřeba vyfiltrovat dodaná data tak, aby zbyly pouze informace obsahující jméno. Dalším filtrem, který byl použit pro zpřesnění dat, je vyřazení případů, kde dodaná data obsahovala více jmen, a to z toho důvodu, že nelze jednoznačně určit, kterému advokátovi má být výsledek řízení přiřazen. Toto pravidlo je dále aplikováno i na advokátní kanceláře, a to bez ohledu na to, zda se v jejich názvu vyskytuje jméno advokáta (nebo

[4] `http://www.sklonovani-jmen.cz`

jména) či ne. Pro toto filtrování bylo využito vyhledání určitých frází v dodaných textech. Fráze označují 3 typy případů.

- „sama“, „sám“ - značí, že osoba sama je advokátem
- „§“ - značí, že osoba má sama právnické vzdělání dle § 105/2 s. ř. s.[5]
- „s.r.o.“, „a.s.“, „o.s.“, „v.o.s.“ - jedná se o advokátní kancelář, firmu, sdružení apod.

Jména z dat obsahující více jmen jsou rozlišována, jednotlivá jména jsou z nich získávána, ale nejsou použita jako vstupy pro proces samotného porovnávání. Jejich případné zařazení do tohoto procesu tak bude v budoucnosti velice snadné.

6.5.2 Hledání shody

Získaná jména zbývalo provázat s příslušnou osobou skutečného advokáta, která je reprezentována záznamem v seznamu České advokátní komory. Samotné porovnávání jmen probíhá na třech úrovních. První úrovní je přesná shoda jména z rozhodnutí soudu, včetně titulů před jménem, jsou-li uvedeny, s řetězcem, který je tvořen údaji získanými z databáze České advokátní komory. Konkrétně tituly před jménem, křestním jménem a příjmením. Pokud toto porovnání nepřináší výsledek, dochází na druhou úroveň porovnávání, kde je ze získaného textu extrahováno pouze jméno a příjmení. Tato dvojice je dále porovnávána s obdobnou dvojicí získanou z plné varianty jména uvedené v seznam České advokátní komory (tedy bez titulů). Samotné porovnávání probíhá na základě Levenshteinovy vzdálenosti (4.1.2). Hlavně proto, aby byla vyloučena pozitivní shoda nalezená jen na základě pořadí advokáta v seznamu. Pro každé jméno advokáta je tak vypočtena tato vzdálenost a příslušné jméno advokáta je uloženo do pole odpovídajícího právě této vzdálenosti, a to s patřičným využitím mezní hodnoty vzdálenosti. Ze seznamu s nejlepší vzdáleností je pak vybrán konečný advokát, případně je navrženo několik nejpravděpodobnějších kandidátů, je-li v seznamu více shod. Objevují se ale i případy, kdy se v dodaném textu vyskytuje jméno a příjmení v opačném pořadí. Další úrovní je tedy porovnávání takového jména s řetězcem obsahujícím křestní jméno a příjmení v reversním pořadí (opět se nepředpokládá využití titulů). U dat získaných z Ústavního soudu se nezřídka vyskytuje i případ, kdy data obsahují jen informaci o příjmení. Zavedl jsem tedy pro tento soud další úroveň porovnávání, a to jen na základě příjmení. Všechna jména z databáze České advokátní komory jsou vybrána tak, aby byla minimalizována náhodná shoda v případě vícero držitelů stejného jména. Ze všech jmen advokátů jsou proto vybrána ta, kde trojice atributů - titul před jménem, křestní jméno a příjmení - jsou unikátní napříč celým spektrem. Během vývoje algoritmu jsem vyzkoušel několik možných variant implementace, jejich myšlenka a vývoj algoritmu je nastíněn níže.

6.5.3 Vývoj algoritmu

Prostý cyklus

K problému průchodu, všemi případy a všemi advokáty, za účelem jejich přiřazení, jsem zpočátku přistupoval logicky a velmi všeobecně. Neměl jsem představu, jaká bude doba průchodu. Implementoval jsem tedy průchod daty ve dvou vnořených cyklech, kdy první procházel přes všechny případy a filtroval jména a speciální případy z dat dodaných soudy

[5] soudní řád správní

(viz 6.5.1). Druhý cyklus iteroval přes všechny advokáty a hledal shodu mezi jménem získaným z rozhodnutí soudu a jménem získaným ze seznamu České advokátní komory. Každá nalezená shoda byla dále připravena pro vložení do databáze a po ukončení obou cyklů bylo v transakci provedeno vložení do databáze. Trvání celého procesu se v konečném důsledku pohybovalo v řádech hodin (závislé na počtu případů daného soudu). Konkrétně pak:

- Nejvyšší správní soud - přibližně 5h 15min
- Ústavní soud - cca 12h 15min

Tato doba trvání procesu byla nepřijatelná a vedla tak k dalším úvahám, jak algoritmus průchodu a porovnávání vylepšit.

Průběžné mazání

Onou myšlenkou, která by mohla proces celého průchodu zefektivnit, bylo přehodit pořadí cyklů a zavést mechanismus redukce velikosti procházených dat. To jest, v prvním cyklu procházet přes advokáty a v druhém pak iterovat přes jednotlivé případy. To nám umožňuje projít a nalézt v jedné iteraci vnitřního cyklu všechny případy náležící danému advokátovi. Budeme-li totiž procházet všechny případy a hledat shodu pro dalšího advokáta, naprosto nesmyslně budeme prohledávat znovu případy, které jsme v některé z předchozích iterací přiřadili advokátu jinému. Abychom redukovali množství případů, které musíme pro nalezení shody projít, vyloučíme všechny již přiřazené případy z dalšího prohledávání. Abychom nemanipulovali se strukturou pole příliš často, zavedl jsem mechanismus, který si ukládá indexy pole přiřazených případů a v okamžiku, kdy je jejich počet, po ukončení průchodu vnitřního cyklu, větší než 100, dojde k vymazání všech indexovaných položek z pole. Jelikož není možné předvídat, kterému advokátovi bude daný případ přiřazen, je možné, že požadovaná úspora času se projeví až v pozdější fázi průchodu vnějším cyklem (advokáty), kdy bude například jednomu advokátovi přiřazeno významně větší množství případů. S tímto přístupem byla doba trvání stále v řádech hodin, úspora času však byla zaznamenána hlavně při větším množství případů, tedy u Ústavního soudu.

- Nejvyšší správní soud - zhruba 4h 20min
- Ústavní soud - cca 8h 30min

Prioritní shoda

V předchozích případech jsem použil seznam advokátů řazený abecedně podle příjmení. Při každém průchodu tak bylo pořadí advokátů vždy stejné, což mohlo mít za následek to, že shoda nastala až ke konci pole. Experimentoval jsem i s náhodným pořadím advokátů, ale to přineslo problém opakovaného dotazování na databázi u každého případu. Ani náhodné pořadí neřešilo plně zmíněný problém. Napadlo mě tedy určit pořadí advokátů podle míry četnosti výskytu. Celá změna implementace znamenala návrat k myšlence průchodu obou cyklů v pořadí případy, advokáti. Ovšem s tím rozdílem, že při nalezení shody byl daný advokát přesunut na první pozici seznamu, což zvyšovalo pravděpodobnost shody v dřívější iteraci cyklu. Postupným porovnáváním se tedy tvořil jakýsi žebříček nejčastějších advokátů. Aby nedocházelo k přesunům prvků na stejnou pozici, implementoval jsem mechanismus, který kontroloval, zda aktuální pozice advokáta je menší než 100. V případě, že by byla vyšší, advokát by byl přesunut na začátek seznamu, jinak by setrval na své aktuální pozici. Tím se omezil žebříček na 100 nejčastějších advokátů, přičemž byl průběžně aktualizován. Tato metoda přinesla oproti předchozím variantám výraznou úsporu času.

- Nejvyšší správní soud - zhruba 2h
- Ústavní soud - cca 1h 30min

Tato doba trvání by již byla přijatelná, hlavně s ohledem na to, že samotný proces přiřazování se spouští též pravidelně. Žádné další modifikace zmíněného postupu již nepřinesly lepší výsledky.

Hledání s pamětí

Všechny výše zmíněné přístupy byly implementovány v jazyce PHP, a to hlavně z toho důvodu, že mohly využívat `ORM`[6] a s ním spjaté `Services`[7], což zapouzdřovalo celou implementaci. Po konzultaci získaného výsledku s vedoucím mé práce jsme dospěli k názoru, že oním zpomalujícím faktorem je samotné PHP. Přece jen se jedná o operace nad řetězci a ty zvládá lépe například jazyk Python. Během reimplementace dosavadního algoritmu z PHP do Pythonu jsem se znovu zamyslel nad mechanismem prioritizace. Uvědomil jsem si, že případy, kdy je advokátovi opakovaně přiřazen nějaký případ vlastně znamená, že daný text se opětovně porovnává s jménem advokáta z České advokátní komory. Celý proces by se dal zjednodušit tak, že výsledek provedeného porovnání by se pro daný text uložil a při dalším prohledávání by se nejprve zkontrolovalo, zda již neexistuje výsledek pro tento text. To vedlo ke vzniku slovníku[8], který vždy obsahoval jako klíč daný text určený k porovnávání a jeho hodnotou byl výsledek nalezené shody (tj. přiřazený advokát). Stejného principu lze využít i pro nenalezení shody textu se jménem advokáta. Zavedení tohoto principu omezuje samotné provádění procesu porovnání. Použití popsaného postupu přineslo urychlení o více než 90%. Nelze jednoznačně říci, zda za tak výraznou úsporu může použití jazyka Python či aplikace popsané myšlenky, v konečném důsledku je důležité, že se doba běhu výrazně snížila.

- Nejvyšší správní soud - zhruba 7 min
- Ústavní soud - cca 15 min

Dosažené výsledky v jednotlivých fázích vývoje algoritmu shrnuje tabulka 6.1.

Varianta	Nejvyšší správní soud (39 500 případů)	Ústavní soud (67 500 případů)
Prostý cyklus	5h 15 min	12h 15 min
Průběžné mazání	4h 20 min	8h 30 min
Prioritní shoda	2h	1h 30 min
Hledání s pamětí	7 min	15 min

Tabulka 6.1: Shrnutí výsledků jednotlivých představených přístupů

Do celého procesu přiřazování jsou vždy zahrnuty jen ty případy, jejichž informace o automaticky přiřazeném advokátovi nebyla dosud zkontrolována a potvrzena ručním zpracováním. Tedy ty případy, kde příznak `is_final` nenabývá hodnoty `true`. Při průběžném

[6] Object/Relational Mapping-konverze dat mezi relační databází a objektově orientovaným programovacím jazykem

[7] kolekce metod pracujících s daným objektem

[8] datová struktura jazyka Python (`dict`)

zpracovávání výsledků přiřazení a jejich kontrole se tak postupem času bude snižovat počet případů vstupujících do tohoto procesu. Největší zvrat nastane v okamžiku, až budou takto zkontrolována a potvrzena data týkající se minulosti, kde se dá předpokládat, že v daných případech již nenastanou žádné změny.

6.6 Získání dat z České advokátní komory

Formulář na stránkách České advokátní komory, určený pro vyhledávání, umožňuje najít informace o advokátovi, ale i o jednotlivých koncipientech. Problémem ovšem je, že v něm nelze určit, zda vyhledáváme advokáta a nebo právě koncipienta. Tedy v okamžiku, kdy odešleme formulář bez jakýchkoliv údajů, zobrazí se nám seznam všech advokátů a koncipientů. Seznam s výsledky je samozřejmě stránkován. Podobně jako u stránek Nejvyššího správního soudu ale neposkytuje žádný prvek pro přechod na další stranu, za to ale poskytuje velice přímočarý způsob přechodu na určitou stránku. Parametr čísla stránky je totiž součástí volání javascriptové funkce `__doPostBack()`, která je vyvolána pro přechod na libovolnou stránku vyhledávání. Tabulka s výsledky obsahuje 4 sloupce, první s názvem *advokát*, kde je uvedeno jméno a odkaz na detail daného advokáta. Druhý sloupec nese název *koncipient*, kde se obdobně nachází jméno a odkaz na detail daného koncipienta, dále pak sloupec *stav*, který vyjadřuje stav činnosti advokáta (aktivní, neaktivní, pozastavená činnost, vyškrtnut). Poslední sloupec pak zobrazuje firmu, kde daný advokát či koncipient působí, pochopitelně společně s odkazem na detail firmy. Logicky jsem očekával, že jsou nejdříve řazeni advokáti a až dále koncipienti. Při postupném průchodu několika desítkami stránek jsem ale zjistil, že koncipienti jsou řazeni převážně v prostřední části výsledků. Nebylo tedy dostačující určit počet procházených stránek z informace o celkovém počtu advokátů, ale bylo potřeba projít všechny stránky výsledků a vyhledávat pouze výskyt hodnot ve sloupci advokát, respektive hledat specifický tvar odkazu. Procházení jednotlivými stránkami je tedy opět řízeno cyklem, kde je celkový počet stránek vypočten součtem z informace o počtu zobrazených advokátů a koncipientů.

Konkrétní odkaz, vedoucí na detail advokáta, má formát, jehož příklad je uveden dále: `/Units/_Search/Details/detailAdvokat.aspx?id=`,
kde parametr `id` obsahoval hash hodnotu, která byla pro daného advokáta specifická. Na každé stránce s výsledky tedy byly vyhledány odkazy vyhovující tomuto formátu. Následně bylo potřeba detaily advokátů na těchto odkazech stáhnout. Zpočátku se nabízel stejný postup, jako byl použit u Ústavního soudu - tedy přechod na detail, stažení a přechod na další stránku - nicméně tento postup se ukázal být pomalým. Napadlo mě, že vzhledem k použitému parametru `id` v adrese odkazované stránky by mohly být stránky statické (persistentní). Ukázalo se, že tomu tak skutečně je a přechod na stránku s detailem je možný i bez předchozího vyhledávání. To mě navedlo na myšlenku prostého stažení `HTML` obsahu odkazu bez nutnosti přechodu na stránku s detailem. Upravil jsem tedy způsob procházení tak, že se všechny odkazy v průběhu průchodu stránkami výsledků ukládaly a následně byly všechny postupně staženy. Jako ideální identifikátor, a zároveň název souboru, posloužil parametr `id`. Tento přístup také umožnil stahovat jen nové záznamy (za předpokladu, že staré zůstaly v původní složce adresářové struktury). Ovšem zavedení požadavku na udržování historie změn v záznamech advokátů pak možnost stahování pouhých přírůstků vyloučilo a jsou tedy vždy stahovány všechny záznamy.

6.6.1 Extrakce dat

Celá stránka s detailem advokáta je koncipována jako tabulka. Nejsou v ní použity žádné `CSS` třídy a identifikátory, které by usnadnily vyhledání požadovaných informací. Objevuje se zde minimálně jeden údaj, a to seznam specializací, který je napříč advokáty proměnlivý. Jednotlivé specializace jsou ale uvedeny jako samostatné řádky tabulky. Nelze tak spoléhat na pevné pořadí ostatních informací, uvedených v řádcích tabulky. Každý požadovaný údaj je umístěn v buňce tabulky, kde předchozí buňka (horizontálně) obsahuje popisek, který říká, co je na tomto řádku umístěno. Je tedy nejdříve vyhledán tento popisek a následně s ním sousedící buňka. Pro hledání informací, které jsou uvedené na více řádcích, je pak využito navigace v `DOM` stromě, která umožňuje přistupovat k rodičovskému prvku (`parent`), prvkům sousedícím (`next/previous`) a nakonec k vedlejším prvkům na stejné úrovni (`siblings`). Údaje o emailových adresách jsou pak získávány z odkazů formátu `javascript:window.location='mailto:'` následovaných textem emailové schránky s rozdělením na uživatelské jméno, zavináč a název domény, například `'martina.pesakova'+'@'+'advokati-kpv.cz'`. Přímo zobrazené emailové adresy obsahovaly místo symbolu @ obrázek. Největším problémem bylo rozdělit nalezené jméno advokáta na jednotlivé části:

- tituly uvedené před jménem
- křestní jméno
- příjmení
- tituly uvedené za jménem

Tento problém umocňoval fakt, že celé jméno bylo uvedené velkými písmeny a tedy nebyla možnost využít způsobu detekce podle velkých počátečních písmen jména i příjmení. Pro určení souvisejícího, tedy že další části řetězce jsou tituly před jménem a za ním, bylo stěžejní určit, co je jméno a kde se ve zkoumaném řetězci vyskytuje. Celý proces hledání jsem postavil na úpravě kopie původního řetězce. Prvotní fází bylo rozložit celý řetězec na jednotlivé elementy a postupnými kontrolami vyřadit nežádoucí celky. Prvním předpokladem byl správný tvar zápisů titulů před i za jménem. Tedy, že tituly před jménem se neoddělují čárkou a v případě titulu, jehož označení se píše za jménem se za jménem píše vždy čárka a následující tituly jsou též odděleny čárkou[9]. Na základě tohoto pravidla bylo možné eliminovat části obsažené za čárkou s tím, že se určitě nejedná o jméno. Dále byly odstraněny všechny elementy, které obsahovaly tečku – lze předpokládat, že se jedná zkratku titulu. Ovšem na poli advokacie není žádnou výjimkou titul získaný v zahraniční. A jelikož o způsobu zápisu některých těchto titulů panují neshody, nelze se spoléhat na zápis těchto titulů, který obsahuje tečky. Mezi tyto tituly patří například *MBA, BA, BPA, BBA, MPA, DBA*. Nezřídka se též advokáti vzdělávají v různých dalších oblastech, aby mohli rozšířit svoje zaměření a vyhovovali tak potřebám více klientům. Objevují se tedy i případy použití spojky *et*, kterou lze užít právě i pro rozlišení stejného titulu z různých oblastí. Pro tyto případy bylo tedy zavedeno porovnávání elementu s určitým řetězcem. Po odstranění i těchto titulů pak již výsledný seznam obsahoval pouze jméno a příjmení. Tímto způsobem je možné zpracovat i jména, která jsou složena z více křestních jmen i více příjmení, což není u advokátů zapsaných u České advokátní komory tak vzácné, jelikož jsou zde zapsáni

[9]Internetová jazyková příručka, Ústav pro jazyk český Akademie věd ČR, v. v. i. `http://prirucka.ujc.cas.cz/?id=782`

často i cizinci. Poslední fází bylo najití získaného jména a příjmení v textu a tím určení počátečního/koncového indexu vymezujícího oblast, kde jsou obsaženy tituly.

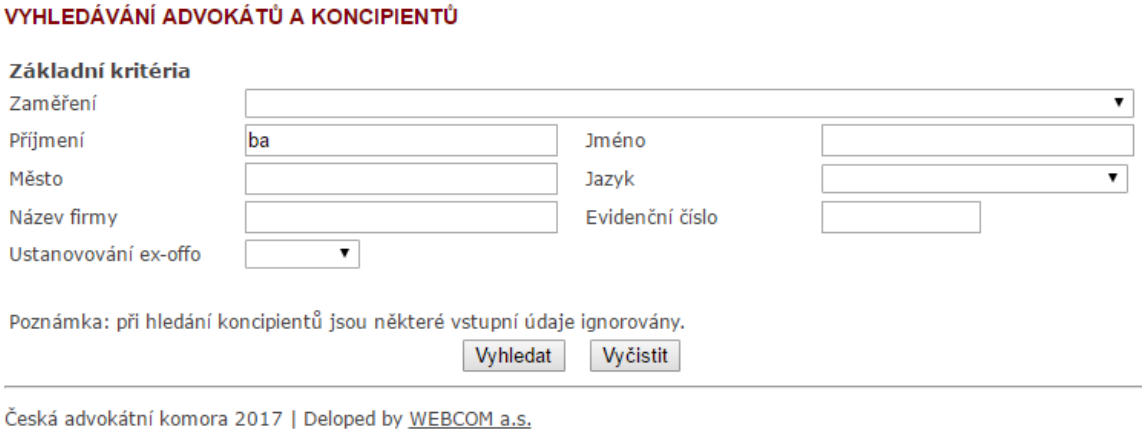
6.6.2 Inovace stránek České advokátní komory

Ve fázi dokončování textu této práce došlo k inovacím stránek České advokátní komory, které přinesly změny ve struktuře stránek, navigace a další omezení. Společně s těmito změnami se na webu České advokátní komory objevilo toto varování:

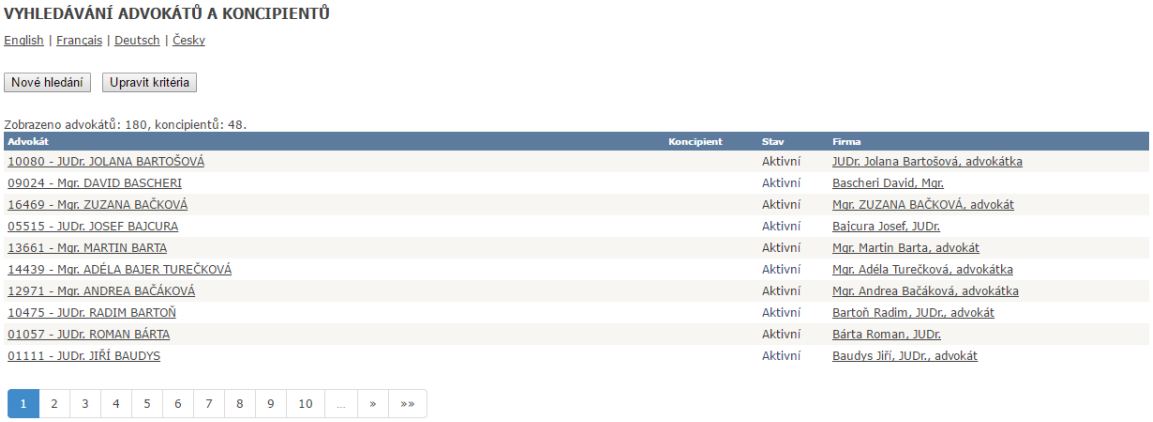
„Vážení,
Seznam advokátů není v tuto chvíli plně funkční. Omlouváme se za vzniklé technické potíže, na jejichž odstranění intenzívně pracujeme. V případě nutnosti ověření si informací nás kontaktujte telefonicky na telefonních číslech:
273 193 223
273 193 220.
Děkujeme Vám za pochopení.“

Dříve popsaný způsob řešení tedy přestal být plně funkční a bylo jej třeba upravit tak, aby reagoval na nové rozhraní. Zásadní změnou bylo to, že formulář odeslaný bez jakýchkoliv údajů již negeneroval kompletní seznam advokátů a koncipientů, ale pouze 500 advokátů (neseřazených podle abecedy). A to ještě s omezením zobrazení 10 advokátů na jednu stránku vyhledávání. Navíc se ale objevil element pro přechod na další stránku. Ten obsahoval odkaz, kde číslo stránky a počet záznamů byly argumenty URL adresy. Úpravy průchodu mezi jednotlivými stránkami výsledku by tak nevyžadovaly mnoho úprav. Abych se zbavil omezení zobrazení pouze 500 advokátů, experimentoval jsem s parametrem `pageSize`, ovšem zadání hodnoty převyšující 500 nemělo kýžený efekt. Více advokátů se nezobrazilo, ale počet stránek výsledků se omezil na 1. Dále jsem experimentoval s vyhledáváním podle příjmení, ale i zde bylo omezení na 500 advokátů, i když bylo zřejmé, že advokátů, jejichž příjmení začíná určitým písmenem je více (stačilo srovnat počet záznamů s předchozími daty). Průchod abecedou tedy vyřešil případy, kdy bylo advokátů méně než 500 a společně s parametrem `pageSize` byla odbourána nutnost průchodu stránkami výsledků. Řešením případů, kdy je výsledků zobrazeno 500, bylo rozdělení intervalu příjmení. Oba případy byly dohromady implementovány pomocí rekurze, kdy podmínkou pro rekurzivní volání je počet zobrazených záznamů. Je-li tedy záznamů 500 a více, je v dalším volání této funkce rozšířen řetězec o další písmeno z abecedy a je použit pro nové hledání. V průběhu tak dochází k postupnému „hádání“ příjmení - příklad vyplnění formuláře pro dělení intervalu ilustruje obrázek 6.6. Pro případ, že vyhledávání bude neúspěšné a nebudou zobrazeny žádné záznamy, je zavedena kontrola na výskyt prvku s určitou `CSS` třídou, uvnitř kterého jsou zobrazeny jednotlivé řádky výsledků.

V seznamu výsledků se změnil formát odkazů vedoucích na detail advokáta, který je teď ve tvaru - `/Contact/Details/`*`id`*, kde `id` je opět unikátní hash identifikátor, který je ale jiný než v předchozím případě. Stránky s detailem advokáta jsou stále statické a pro jejich stahování je tedy možné použít stávající řešení. U samotné extrakce dat pak nastala jen změna ve formátu uváděného jména, který navíc obsahuje evidenční číslo. To je uvedeno před jménem a je odděleno pomlčkou. Citelnou změnou je omezení databáze pouze na advokáty ve stavu aktivní, nelze ale předjímat, zda se jedná o konečný stav databáze.



Obrázek 6.6: Vyhledávání v průběhu postupného průchodu (dělení intervalu písmene b)



Obrázek 6.7: Výsledky vyhledávání pro „ba“, první stránka výsledků

Obrázky 6.6, 6.7 ukazují proces vyhledávání advokáta pomocí mechanismu dělení intervalu, v tomto případě písmene „b“. Zároveň je vidět seznam výsledků na upraveném webu České advokátní komory. Výsledky vyhledávání nejsou řazeny abecedně ani podle evidenčního čísla advokáta. Stránky jsou stále v údržbě, takže je možné, že toto chování není konečným stavem.

6.7 Pravidelné spouštění

Popsané části byly navrženy jako součást většího systému, který pravidelně získává a aktualizuje data, respektive obsah databáze. Každý popsaný skript je spouštěn v určitou dobu za pomoci `cronu` (viz 5.2.2). Příklad aktuálně použitého plánu je představen na ukázce 6.1. Zobrazený plán popisuje jen část týkající se jednoho soudu, konkrétně Nejvyššího správního soudu.

Postupně je zde vidět příprava prostředí pro spuštění crawleru a spuštění PHP skriptu, který spouští samotný crawler s požadovanými parametry. Dále pak volání skriptu pro import získaných dat do databáze. V rámci tohoto skriptu dojde k vymazání obsahu destinace, která je parametrem spuštění skriptu, a složka je pak připravena pro další běh crawleru. Následující úloha spouští proces ohodnocení dokumentů. A nakonec se provádí příprava prostředí a následné spuštění skriptu, který zajišťuje přiřazení případů advokátům. Každá úloha je naplánována s dostatečnou rezervou tak, aby nebyla další úloha spuštěna v době běhu jiného procesu. Podrobný plán, zahrnující získání a import dat ze všech soudů a České advokátní komory, je obsažen v příloze.

```
## Devel cron jobs

# Every tuesday at 3:42 run crawler for NSS
42 3 * * 2 export WORKON_HOME=/usr/local/share/.virtualenvs &&
    ↪ source /usr/bin/virtualenvwrapper.sh && workon staging-crawler
    ↪ -nss && php /home/cestiadvokati.cz/web-devel/www/index.php app
    ↪ :nss-crawler /home/cestiadvokati.cz/crawlers-devel/nss >/dev/
    ↪ null 2>&1; deactivate

# Every wednesday at 3:42 run import of crawler NSS data
42 3 * * 3 php /home/cestiadvokati.cz/web-devel/www/index.php app:
    ↪ import-documents nss /home/cestiadvokati.cz/crawlers-devel/nss
    ↪ /result >/dev/null 2>&1

# Every friday at 4:00 run tagging results of NSS
0 4 * * 5 php -d memory_limit=2048M /home/cestiadvokati.cz/web-devel
    ↪ /www/index.php app:tag-results nss

# Ever friday at 17:00 run tagging advocates of NSS
0 17 * * 5 export WORKON_HOME=/usr/local/share/.virtualenvs &&
    ↪ source /usr/bin/virtualenvwrapper.sh && workon advocate-tagger
    ↪  && php /home/cestiadvokati.cz/web-devel/www/index.php app:tag
    ↪ -advocates nss >/dev/null 2>&1; deactivate
```

Listing 6.1: Plán spouštění jednotlivých součástí systému

Kapitola 7

Výsledky

V této kapitole je představen výsledek celé práce. Pro lepší názornost a pochopení návazností jsou v této kapitole stručně popsány a odkazovány i části, které nebyly předmětem řešení této diplomové práce, jak bylo uvedeno v kapitole 2 v sekci Rozdělení práce v týmu.

7.1 Představení aplikace

Na obrázku 7.1 je výřez z hlavní stránky webu[1], kde může návštěvník vyhledat požadovaného advokáta podle jména a nebo zadáním jeho IČ. Toto vyhledávání je možné právě díky informacím, které byly získány ze stránek České advokátní komory (viz 6.6).

Obrázek 7.1: Ukázka vyhledávacího formuláře naší služby

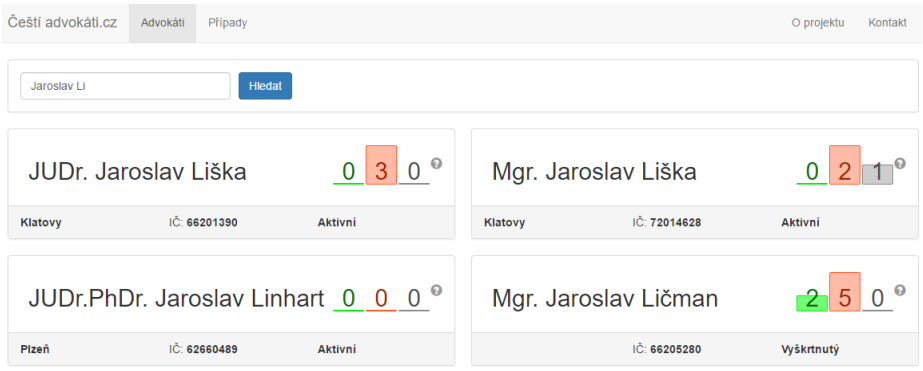
Po zadání jména jsou návštěvníkovi vyhledáni všichni advokáti, kteří vyhovují zadanému jménu. Aby bylo možné advokáty rozlišit, jsou v přehledu zobrazeny doplňující informace o advokátovi. Patří mezi ně celé jméno včetně titulů (nejnovější informace získané z webu České advokátní komory), IČ, město, kde má advokát sídlo a stav činnosti advokáta. Dále je zde zobrazen náhled podílu konečných meritorních[2], nemeritorních[3] rozhodnutí a rozhodnutí o zastavení řízení[4]. Příklad výsledků vyhledávání ilustruje obrázek 7.2, kde je pro vyhledávání použité neúplné jméno.

[1] v době vzniku této práce není web dosud veřejný, ukázku lze nalézt na adrese `https://devel.cestiadvokati.cz/`

[2] to znamená, že se jeho podáním soud alespoň zčásti co do obsahu zabýval

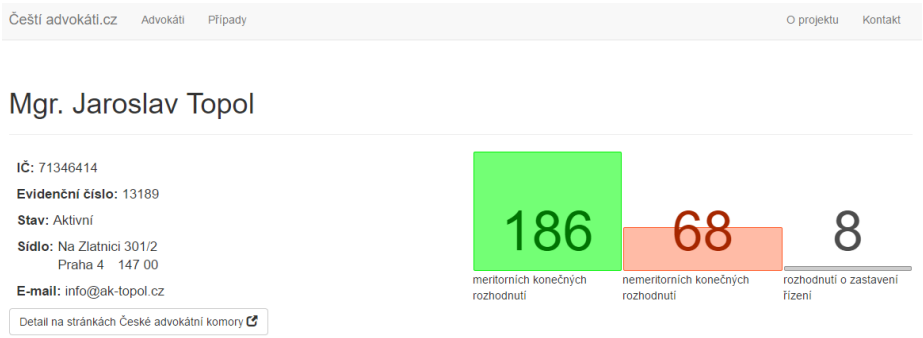
[3] to znamená, že se jeho podáním soud co do obsahu vůbec nezabýval

[4] to znamená, že podání bylo pravděpodobně vzato zpět



Obrázek 7.2: Ukázka výsledků vyhledávání (navržené shody)

Po výběru požadovaného advokáta se návštěvníkovi zobrazí detailní karta advokáta, která je členěna do tří částí. První částí obsahuje podrobnější informace o advokátovi - IČ, evidenční číslo, stav, konkrétní adresu sídla, kontaktní email a odkaz na detail advokáta na stránkách České advokátní komory. V druhé části je vidět podíl jednotlivých rozhodnutí, jak byl popsán v předchozím odstavci textu (meritorní, nemeritorní, zastavené). Ten je doplněn o grafické znázornění a konkrétní čísla. Tyto údaje jsou součtem informace o přiřazení případu advokátovi a ohodnocením dokumentu (positive, negative, neutral). Informace byly získány při procesu ohodnocování dokumentů (viz 6.3) a přiřazování případů (viz 6.5). První a druhou část stránky s detailem advokáta zobrazuje obrázek 7.3[5].

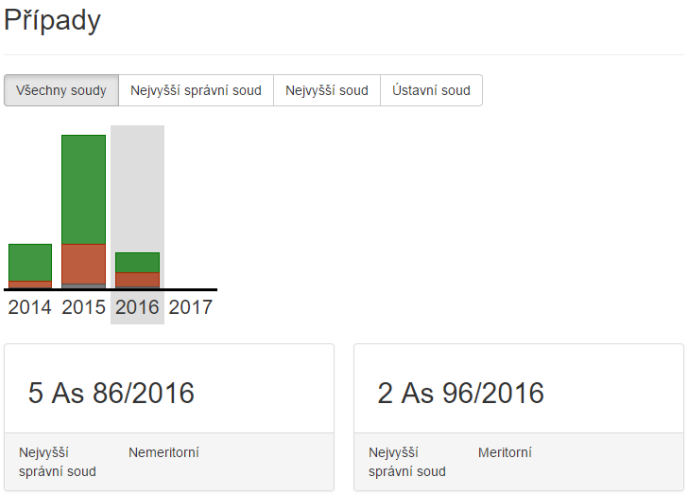


Obrázek 7.3: Detail advokáta - podrobné informace a statistika

Třetí část stránky s detailem advokáta je věnována samotným případům, je zde výpis spisových značek všech případů, které byly advokátovi přiřazeny. U každého případu je zobrazen i konečný typ rozhodnutí a příslušnost k soudu. Pro lepší orientaci či zkoumání jen určitých případů jsou poskytnuty uživateli filtry, kterými lze zobrazit data jen z určitého soudu. Dále je v této části graf, který zobrazuje podíl jednotlivých konečných rozhodnutí během daného roku. Zároveň tento graf slouží jako filtr pro požadovaný rok a nebo typ konečného rozhodnutí. Popsanou část karty detailu advokáta ilustruje obrázek 7.4[6].

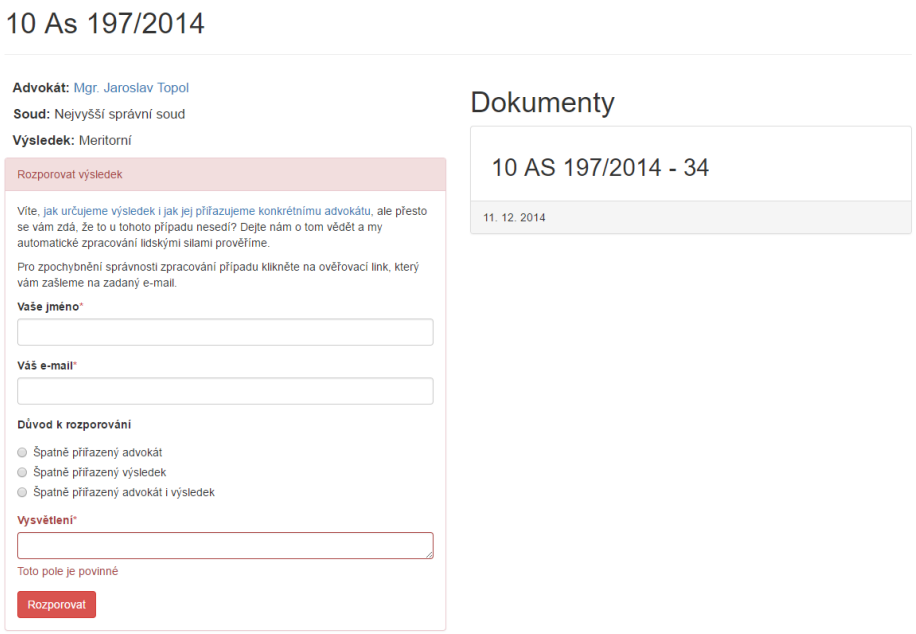
[5] pro lepší ilustraci byl zvolen detail jiného advokáta, než který byl použit v předchozích případech

[6] Na obrázku je zobrazena jen část případů z důvodu velkého množství případů tohoto advokáta



Obrázek 7.4: Detail advokáta - případy a filtry

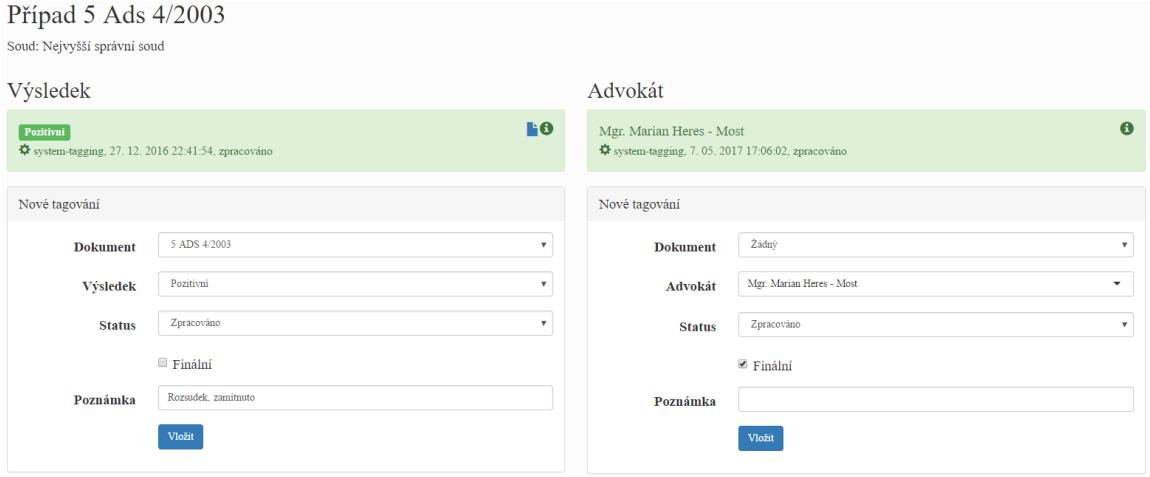
Zajímá-li návštěvníka konkrétní případ, je mu po výběru daného případu zobrazen detail případu. Zde je uveden opět advokát, kterému byl případ přiřazen a odkaz na jeho detail, dále je zde typ konečného rozhodnutí a informace, ke kterému soudu případ přísluší. V pravé části je pak seznam dokumentů, které souvisí s daným případem. U takového dokumentu je uvedeno datum rozhodnutí a odkaz na původní dokument ve spisovně daného soudu. Na této stránce se nachází tlačítko *rozporovat výsledek*, které umožňuje návštěvníkovi zpochybnit výsledek konečného rozhodnutí či přiřazení případu advokátovi. Většinou tento formulář využijí účastníci řízení nebo přímo advokát, který v daném procesu vystupoval, a v okrajových případech pak pozorní návštěvníci, kteří budou zkoumat texty rozhodnutí daných případů. Popsanou část ilustruje obrázek 7.5.



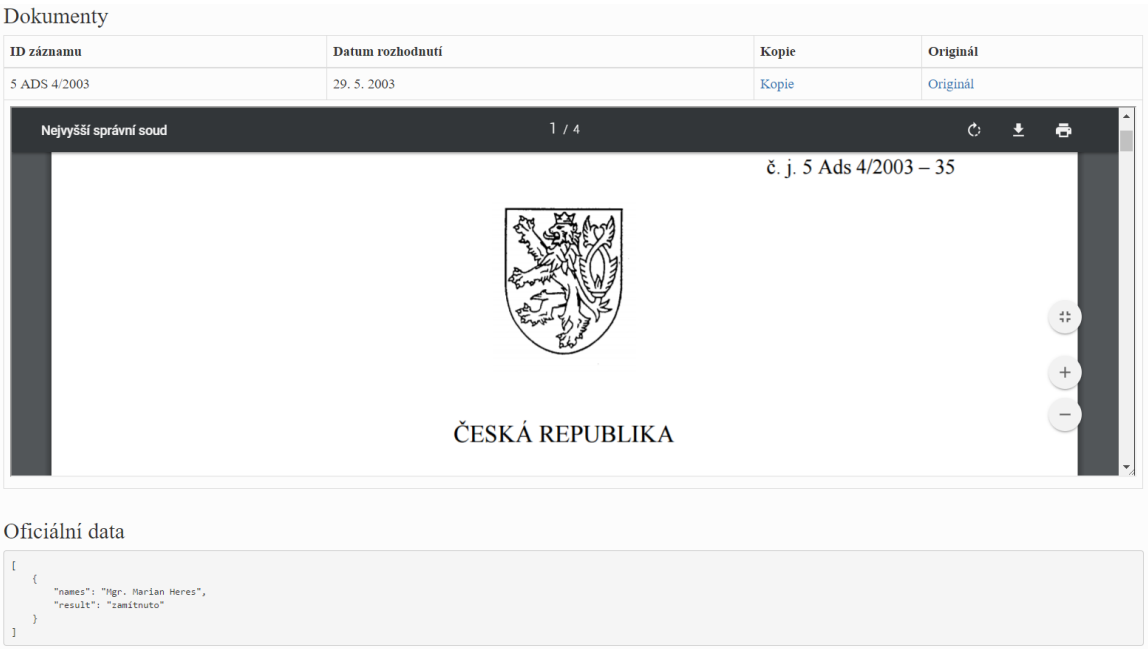
Obrázek 7.5: Detail případu - související dokumenty a rozporování výsledku

Využití popsaného formuláře indikuje stav rozporování v administrační sekci (možnosti filtrování a seznam případů zobrazuje obrázek 7.6), kde je mimo jiné obsaženo rozhraní pro ruční korekce a zadávání ohodnocení a přiřazování advokáta případům. Toto rozhraní (obrázek 7.7) zobrazuje výsledky přiřazené a určené automatickým zpracováním a dále pomocné debugovací informace (vstupy a výstupy procesů), které vznikly za běhu procesu. Ty usnadňují ruční korekci. Pokud poskytnuté informace nejsou dostatečné, jsou zde zobrazeny ještě náhledy dokumentů s textem rozhodnutí a data získaná od soudů. Obě popsané části administračního rozhraní shrnují obrázky 7.6, 7.7 a 7.8.

Obrázek 7.6: Administrační sekce pro ruční korekturu - souhrn



Obrázek 7.7: Rozhraní pro korekci případu v administrační sekci



Obrázek 7.8: Pomocné údaje pro korekci případu

7.2 Vyhodnocení

V současné době jsou (až na ojedinělé výjimky) již všechna rozhodnutí soudů veřejně dostupná online, nebylo tomu tak ale vždy. Rozsah zpracovávaných konečných rozhodnutí se proto pro jednotlivé soudy liší. Pro připomenutí (3.2.1, 3.2.2):

Nejvyšší správní soud Konečná rozhodnutí o kasačních stížnostech (rejstříky As, Ads, Afs, Ans, Aos, Aps, Ars, Azs) podaných po 1. 1. 2006.

Ústavní soud Konečná rozhodnutí o všech návrzích podaných od počátku činnosti soudu do 31. 12. 2006 (v tomto období nebyla rozlišována rozhodnutí o ústavních stížnostech a rozhodnutí vydaná v jiných typech řízení). Z rozhodnutí o návrzích podaných po 1. 1. 2007 jsou již zahrnuta jen konečná rozhodnutí o ústavních stížnostech.

Nejvyšší soud Konečná rozhodnutí o dovoláních v občanskoprávních věcech (rejstříky Cdo, Cdon, ICdo, NSČR, Odo, Odon) podaných po 1. 1. 2001 a v trestněprávních věcech (rejstřík Tdo) podaných po 1. 1. 2002.

Následující tabulka (7.1) zobrazujte celkové statistiky jednotlivých soudů. U každého soudu jsou vzaty v úvahu jen ty roky, kdy jsou data jednoznačná a máme k dispozici informace od soudů zahrnující daný rok. Poslední informace od soudů zahrnují všechny advokáty do konce roku 2015. Proto se rozsah vyhodnocených let u jednotlivých soudů liší, konkrétně:

- Nejvyšší správní soud – konečná rozhodnutí o kasačních stížnostech podaná v letech 2006–2015
- Ústavní soud – rozhodnutí o ústavních stížnostech podaná v období 2007–2015
- Nejvyšší soud – rozhodnutí o dovolání z let 2001–2015

Soud	Celkový počet případů	Počet ohodnocených případů	Počet přiřazených případů	Počet přiřazených a ohodnocených případů	Přiřazených a ohodnocených případů
Nejvyšší správní soud	27353	26240	17542	16690	61,02 %
Ústavní soud	34431	33763	26419	25886	75,18 %
Nejvyšší soud	76900	67715	35764	34671	45,09 %

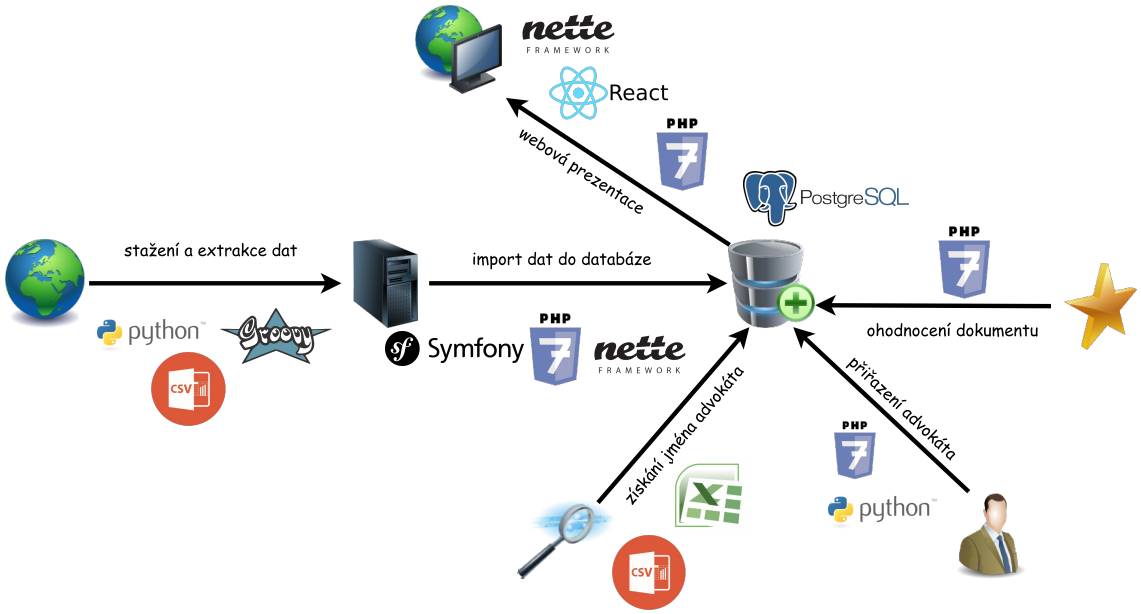
Tabulka 7.1: Celkový přehled případů jednotlivých soudů

Tabulka ukazuje úspěšnost přiřazení a ohodnocení jednotlivých případů u všech tří sledovaných soudů. Důvod, proč nejsou všechny případy uvedené jako ohodnocené je rozepsán na straně 22 a 31. Důvody, proč nejsou všechny případy přiřazeny advokátovi, jsou popsány v rámci implementace tohoto procesu (viz 6.5).

Případy, u kterých bylo současně provedeno ohodnocení a přiřazení, jsou zobrazeny ve sloupci „Počet přiřazených a ohodnocených případů“, v posledním sloupci je pak vyjádřen jejich procentuální[7] podíl na celkovém počtu případů. A právě tyto případy jsou zdrojem pro zobrazené grafy v detailu advokáta na stránkách naší aplikace (viz obrázek 7.3 a 7.4).

7.3 Souhrn technologií

Následující obrázek (7.9) znázorňuje, jaké technologie jsou použity pro výslednou realizaci služby. Dále ukazuje, které technologie, formáty a jazyky se podílejí na určitých částech systému, jak jsou tyto části provázány a s jakými daty pracují.



Obrázek 7.9: Přehled použitých technologií během vývoje

[7] uvedené hodnoty byly zaokrouhleny na 2 desetinná místa

Kapitola 8

Závěr

Cílem mé diplomové práce bylo navrhnout, vytvořit, implementovat a zprovoznit webovou službu, která na základě veřejně dostupných dat objektivně hodnotí kvalitu a nedostatky podání jednotlivých českých advokátů u Nejvyššího soudu, Nejvyššího správního soudu a Ústavního soudu.

Nezbytnou součástí bylo seznámení se s nástroji pro vzdálenou, automatizovanou interakci s webovými stránkami. Dále pak analýza struktury a použitých technologií na stránkách sledovaných soudů. Následně prostudování a vyzkoušení přístupů pro získávání informací ze strukturovaných dat na těchto stránkách při implementaci crawlerů, které obstarávají získání požadovaných informací. Získaná data byla exportována do souboru CSV a následně uložena do navržené databáze, která poskytuje vazby mezi dokumenty a advokáty, které odpovídají reálnému světu a reálným osobám. Na získaných datech bylo následně provedeno ohodnocení dokumentů, které odráží požadovanou pečlivost advokáta.

Po prvotních snahách a experimentech s převody dokumentů pomocí techniky OCR se tento problém ukázal být složitější, než bylo původním předpokladem, a protože se naskytla možnost získání dat z textů rozhodnutí od samotných soudů, bylo přistoupeno k této variantě. Tato data byla nezbytná pro další pokračování vývoje aplikace. Nad databází obsahující jména advokátů působících u daných případů a informací o osobě konkrétního advokáta ze seznam České advokátní komory bylo možné provést přiřazení případu advokátovi, k čemuž bylo využito mechanismu porovnávání textu založeném na využití Levenshteinovy vzdálenosti.

V průběhu práce se vyskytly problémy související se změnou struktury a navigačních prvků na stránkách České advokátní komory. Jelikož jsou všechny crawlery implementovány na míru webovým stránkám jednotlivých institucí, je možné, že změny související s úpravami na nové rozhraní mohou postihnout kterýkoliv ze sledovaných webů, respektive crawlerů. Po dobu úprav pak nebudou získávány nejnovější informace pravidelně. I v takovém případě ale bude služba poskytovat dostatečně objektivní informace, neboť podíl takto opožděně získaných dat na celkovém počtu případů je zanedbatelný.

V době dokončování této práce je služba téměř připravena na ostrý provoz a zveřejnění. Úspěšnost přiřazených a ohodnocených případů se v současné době pohybuje v rozmezí 65-75 % u mnou zpracovávaných soudů. Hlavní překážkou spuštění je čekání na aktuální informace od soudů, aby služba již od počátku své existence prezentovala nejnovější údaje. V budoucnu by mohla aplikace poskytovat prostor advokátům pro komentování daných případů a umožnit jim tak vysvětlit, proč došlo na jejich straně k pochybení.

V nedaleké budoucnosti se plánuje i možnost rozšíření vyhledávání podle měst, advokátních kanceláří, případně i jednotlivých specializací advokátů. Přibude-li někdy možnost

získávání dat z dalších soudů (například krajských), mohla by se aplikace postupem času stát vítaným pomocníkem pro nejrůznější organizace a společnosti, které pomáhají občanům s řešením právních problémů, i pro řadové občany.

V současné době jsou členové spolku *DATOS - data o spravedlnosti, z. s.* rozhodnuti poskytovat takto získané informace formou otevřených dat pro další užití na stránkách projektu – `www.cestiadvokati.cz`

Literatura

[1] Avasarala, S.: *Selenium WebDriver practical guide : interactively automate web applications using Selenium WebDriver*. Birmingham, UK: Packt Pub, 2014, ISBN 1782168850.
URL http://freepdf-books.com/selenium-webdriver-practical-guide

[2] Chacon, S.: *Pro Git*. Praha: CZ. NIC, 2009, ISBN 978-80-904248-1-4.

[3] Gupta, S.; Kaiser, G.: Extracting Content from Accessible Web Pages. In *Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A)*, W4A '05, New York, NY, USA: ACM, 2005, ISBN 1-59593-219-4, s. 26–30, doi:10.1145/1061811.1061816.
URL http://doi.acm.org/10.1145/1061811.1061816

[4] Lawson, R.: *Web Scraping with Python : scrape data from any website with the power of Python*. Birmingham: Packt Publishing, 2015, ISBN 1782164367.
URL http://zempirians.com/ebooks/Richard%20Lawson-Web%20Scraping%20with%20Python-Packt%20Publishing%20(2015).pdf

[5] Levenshtein, V. I.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, ročník 10, Únor 1966: str. 707.

[6] Mitchell, R.: *Web Scraping with Python: Collecting Data from the Modern Web*. O'Reilly Media, Inc., první vydání, 2015, ISBN 1491910291, 9781491910290.
URL https://books.google.cz/books?id=7z_fCQAAQBAJ&lpg=PP1&hl=cs&pg=PT28#v=onepage&q&f=false

[7] Nair, V.: *Getting Started with Beautiful Soup*. Birmingham, UK: Packt Publishing, 2014, ISBN 978-1783289554.
URL https://goo.gl/qOR22r

[8] Navarro, G.: A Guided Tour to Approximate String Matching. *ACM Comput. Surv.*, ročník 33, č. 1, Březen 2001: s. 31–88, ISSN 0360-0300, doi:10.1145/375360.375365.
URL http://doi.acm.org/10.1145/375360.375365

[9] Smith, R.: An Overview of the Tesseract OCR Engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, ročník 2, Sept 2007, ISSN 1520-5363, s. 629–633, doi:10.1109/ICDAR.2007.4376991.
URL
https://github.com/tesseract-ocr/docs/blob/master/tesseracticdar2007.pdf

[10] *Zákon č. 182/1993 Sb., zákon o Ústavním soudu*. 1993.

[11] *Zákon č. 99/1963 Sb., občanský soudní řád.* 1963.

[12] *Zákon č. 150/2002 Sb., soudní řád správní.* 2003.

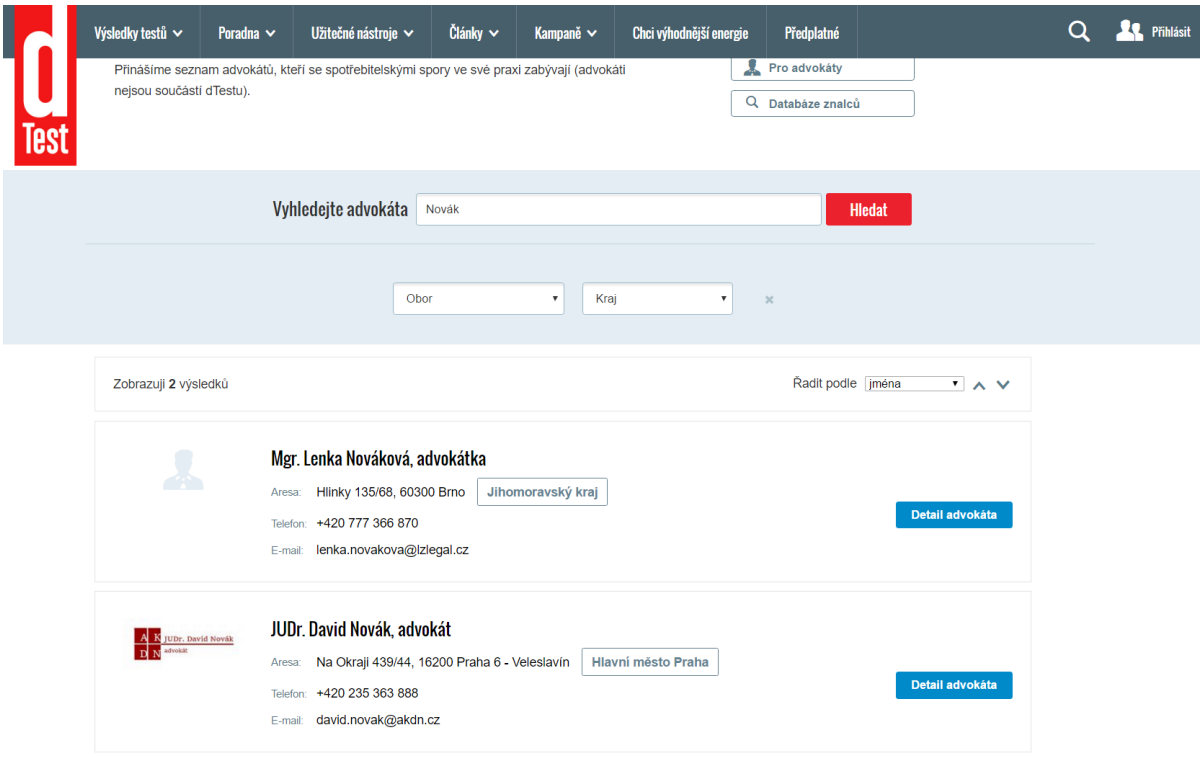
[13] *Zákon č. 141/1961 Sb., trestní řád.* 1961.

Přílohy

Seznam příloh

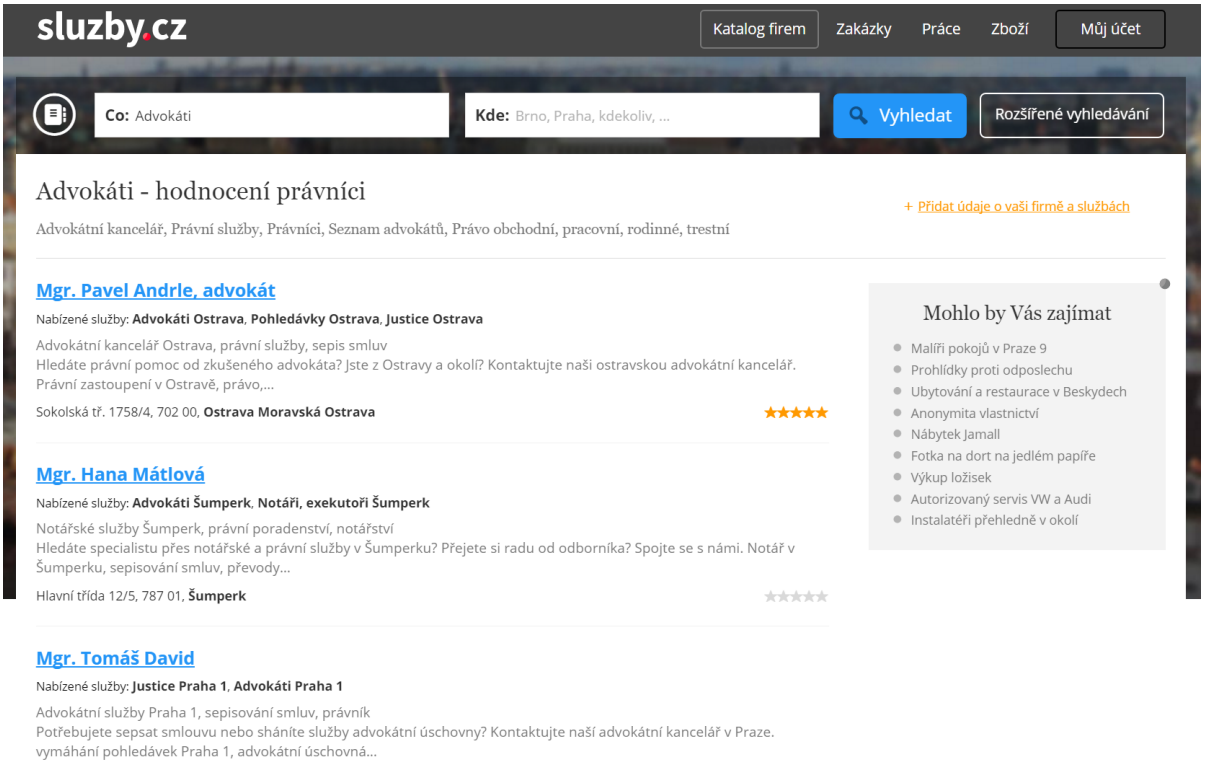
Příloha A

Ukázky webů podobných služeb

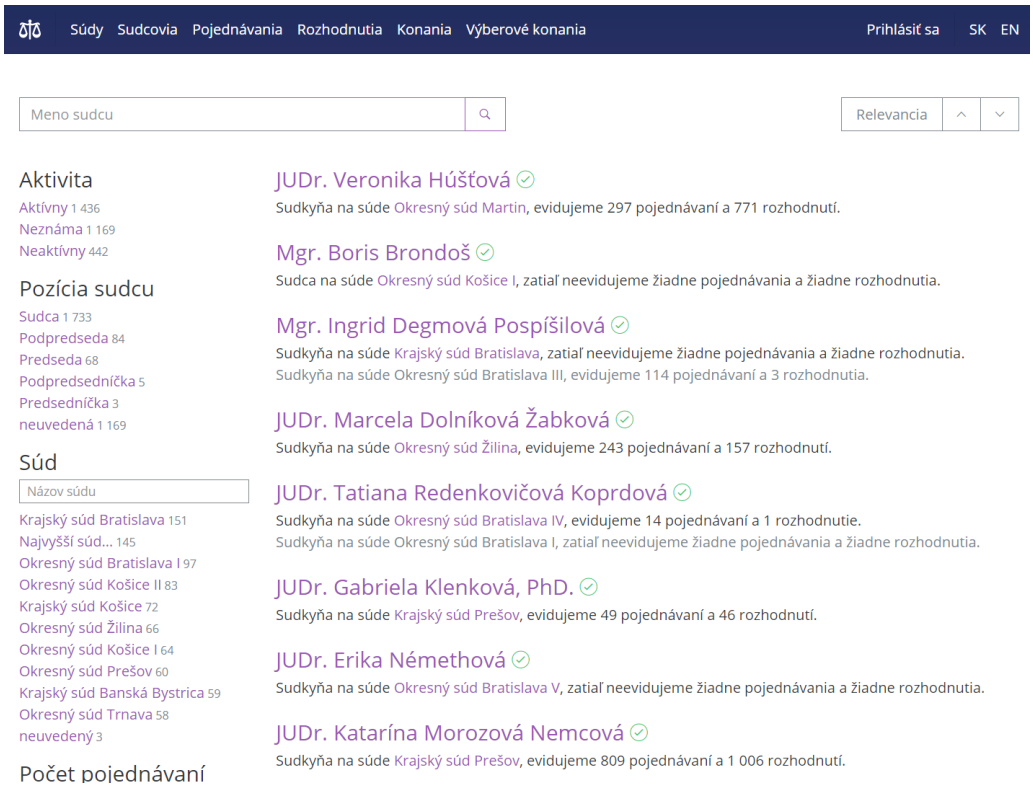


Obrázek A.1: Ukázka webu dtest.cz

Obrázek A.2: Detail advokáta na webu katalog-pravniku.cz



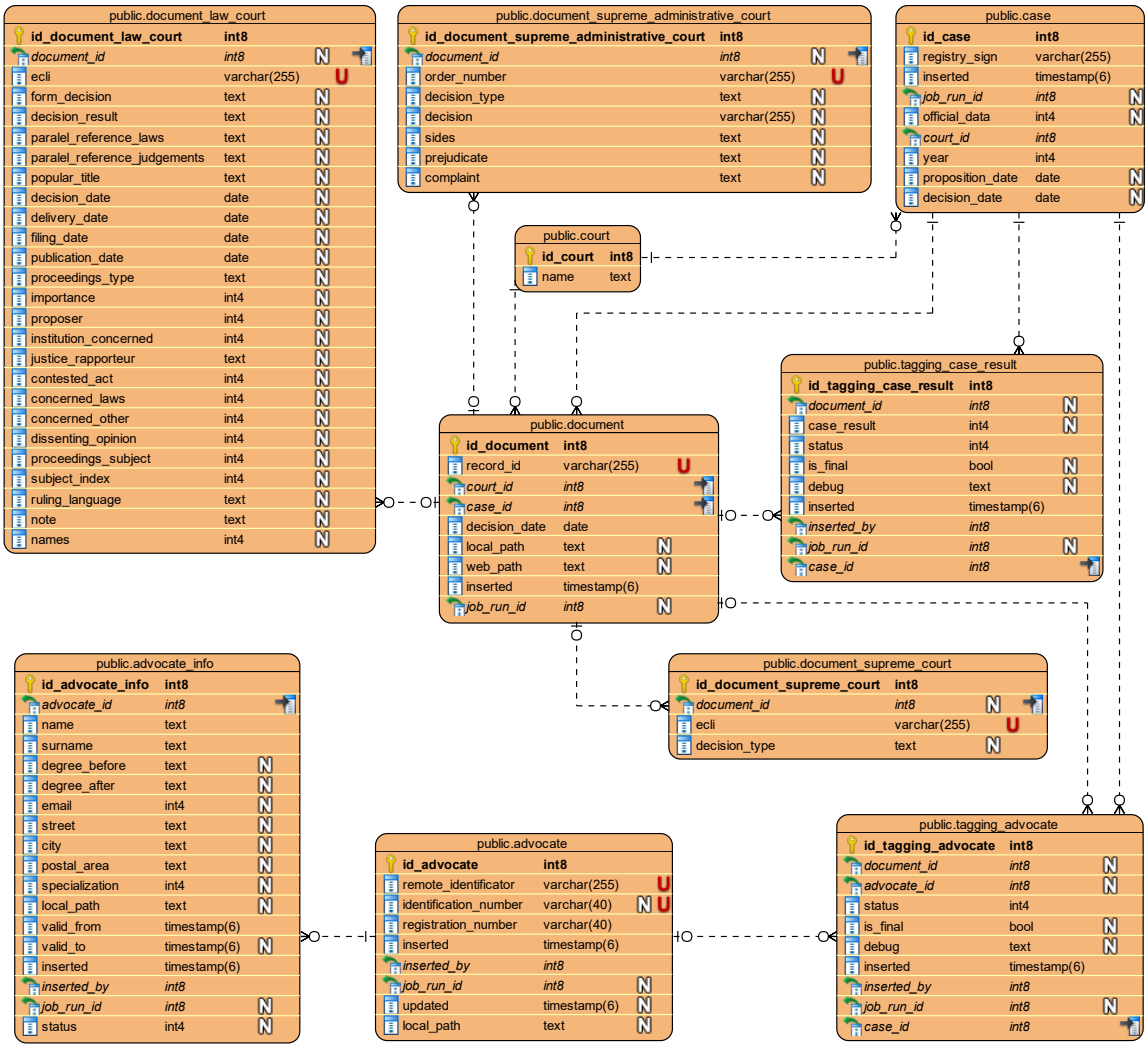
Obrázek A.3: Ukázka vyhledávání na serveru sluzby.cz



Obrázek A.4: Ukázka zobrazení výsledků vyhledávání na webu otvorenesudy.sk

Příloha B

ER diagram navržené databáze



Obrázek B.1: Návrh uvažované databáze

Příloha C

Plán pravidelného spouštění

```
# Every tuesday at 3:24 run crawler for NS
24 3 * * 2 LANG=cs_CZ.UTF-8 php /home/cestiadvokati.cz/web-devel/www
    ↪ /index.php app:ns-crawler /home/cestiadvokati.cz/crawlers-
    ↪ devel/ns >/dev/null 2>&1
# Every wednesday at 3:24 run import of crawler NS data
24 3 * * 3 php /home/cestiadvokati.cz/web-devel/www/index.php app:
    ↪ import-documents ns /home/cestiadvokati.cz/crawlers-devel/ns/
    ↪ result >/dev/null 2>&1
# Every thursday at 3:24 run tagging results of NS
24 3 * * 4 php -d memory_limit=2048M /home/cestiadvokati.cz/web-
    ↪ devel/www/index.php app:ns-result-tagger >/dev/null 2>&1
# Every friday at 3:24 run tagging advocates of NS
24 3 * * 5 php -d memory_limit=2048M /home/cestiadvokati.cz/web-
    ↪ devel/www/index.php app:ns-advocate-tagger >/dev/null 2>&1
```

Listing C.1: Nejvyšší soud

```
# Every tuesday at 3:42 run crawler for NSS
42 3 * * 2 export WORKON_HOME=/usr/local/share/.virtualenvs &&
    ↪ source /usr/bin/virtualenvwrapper.sh && workon staging-crawler
    ↪ -nss && php /home/cestiadvokati.cz/web-devel/www/index.php app
    ↪ :nss-crawler /home/cestiadvokati.cz/crawlers-devel/nss >/dev/
    ↪ null 2>&1; deactivate
# Every wednesday at 3:42 run import of crawler NSS data
42 3 * * 3 php /home/cestiadvokati.cz/web-devel/www/index.php app:
    ↪ import-documents nss /home/cestiadvokati.cz/crawlers-devel/nss
    ↪ /result >/dev/null 2>&1
# Every Friday at 4:00 run tagging results of NSS
0 4 * * 5 php -d memory_limit=2048M /home/cestiadvokati.cz/web-devel
    ↪ /www/index.php app:tag-results nss
# Ever friday at 17:00 run tagging advocates of NSS
0 17 * * 5 export WORKON_HOME=/usr/local/share/.virtualenvs &&
    ↪ source /usr/bin/virtualenvwrapper.sh && workon advocate-tagger
    ↪  && php /home/cestiadvokati.cz/web-devel/www/index.php app:tag
    ↪ -advocates nss >/dev/null 2>&1; deactivate
```

Listing C.2: Nejvyšší správní soud

```
# Every tuesday at 4:42 run crawler for US
42 4 * * 2 export WORKON_HOME=/usr/local/share/.virtualenvs &&
   ↪ source /usr/bin/virtualenvwrapper.sh && workon staging-crawler
   ↪ -us && php /home/cestiadvokati.cz/web-devel/www/index.php app:
   ↪ us-crawler /home/cestiadvokati.cz/crawlers-devel/us >/dev/null
   ↪  2>&1; deactivate
# Every wednesday at 4:42 run import of crawler US data
42 4 * * 3 php /home/cestiadvokati.cz/web-devel/www/index.php app:
   ↪ import-documents us /home/cestiadvokati.cz/crawlers-devel/us/
   ↪ result >/dev/null 2>&1
# Every Friday at 5:00 run tagging results of US
0 5 * * 5 php -d memory_limit=2048M /home/cestiadvokati.cz/web-devel
   ↪ /www/index.php app:tag-results us
# Ever friday at 19:00 run tagging advocates of US
0 19 * * 5 export WORKON_HOME=/usr/local/share/.virtualenvs &&
   ↪ source /usr/bin/virtualenvwrapper.sh && workon advocate-tagger
   ↪  && php /home/cestiadvokati.cz/web-devel/www/index.php app:tag
   ↪ -advocates us >/dev/null 2>&1; deactivate
```

Listing C.3: Ústavní soud

```
# Every monday at 2:22 run crawler for CAK
22 2 * * 1 export WORKON_HOME=/usr/local/share/.virtualenvs &&
   ↪ source /usr/bin/virtualenvwrapper.sh && workon staging-crawler
   ↪ -cak && php /home/cestiadvokati.cz/web-devel/www/index.php app
   ↪ :cak-crawler /home/cestiadvokati.cz/crawlers-devel/cak >/dev/
   ↪ null 2>&1; deactivate
# Every tuesday at 2:22 run import of CAK data
22 2 * * 2 php -d memory_limit=1024M /home/cestiadvokati.cz/web-
   ↪ devel/www/index.php app:import-advocates /home/cestiadvokati.
   ↪ cz/crawlers-devel/cak/result/ >/dev/null 2>&1
```

Listing C.4: Česká advokátní komora

Příloha D

Obsah CD

`source` – adresář, ve kterém jsou obsaženy zdrojové kódy aplikace

- `app` – funkční logika aplikace a nastavení
  - `app/Commands/TagResults.php` – PHP skript pro ohodnocení dokumentů
- `externals` – složka se skripty, které jsou externě volány z jednotlivých příkazů
  - `us_crawler.py` – skript v jazyce Python zajišťující získání dat z Ústavního soudu
  - `nss_crawler.py` – skript v jazyce Python zajišťující získání dat z Nejvyššího správního soudu
  - `cak_crawler.py` – skript v jazyce Python zajišťující získání dat z České advokátní komory
  - `tagger.py` – skript v jazyce Python zajišťující přiřazení případů advokátům
- `README.md` - soubor obsahující pokyny k instalaci a spuštění

`sample` – ukázka souborů s daty získanými od soudů[1]

`doc` – adresář, kde je umístěna dokumentace skriptů jazyka Python

`tex` – adresář se zdrojovými kódy technické zprávy + použité obrázky

`promo` – adresář obsahující plakát a video prezentující výsledky práce

[1] všechna získaná data nelze volně šířit, jelikož byla získána pro účel tohoto konkrétního projektu (dohoda se soudy)