FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Portál srovnávače faktur se smlouvami z registru smluv
Student:	Matěj Adamec
Vedoucí:	Ing. Lucie Svitáková
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2020/21

Pokyny pro vypracování

Cílem práce je navrhnout metody pro mapování faktur na příslušné smlouvy z registru smluv a vytvořit webovou aplikaci, která bude tyto výsledky prezentovat a upozorní na případné nesrovnalosti. Pro stažení faktur ministerstev může být využit nástroj Laboratoře otevřených dat nebo rovněž Hlídač státu.

Detailní pokyny:

1. Proveďte analýzu faktur, které ministerstva poskytují otevřeně (různé faktury mohou obsahovat různé informace).
2. Získejte potřebná data faktur a smluv podle popisu výše.
3. Navrhněte metody, pomocí kterých propojíte faktury se smlouvami, na které se vážou (využití variabilního symbolu, popisu faktury, atd. v souvislosti s bodem jedna).
4. Tyto metody implementujte.
5. Vytvořte webový portál, který bude tyto výsledky zobrazovat. Zároveň bude upozorňovat na nesrovnalosti, jako je například součet vyfakturovaných částek převyšující smluvní hodnotu, a základní statistiky.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 26. prosince 2019

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Bakalářská práce

Portál srovnávače faktur se smlouvami z registru smluv

Adamec Matěj

Katedra softwarového inženýrství
Vedoucí práce: Ing. Svitáková Lucie

4. června 2020

Poděkování

Rád bych poděkoval vedoucí práce Ing. Lucii Svitákové, za cenné rady a připomínky při tvorbě této práce. Také bych chtěl poděkovat Ing. Markovi Sušickému z OpenDataLab, který mi dal rady ohledně návrhu a implementace. Poděkování patří také rodině a kamarádům, kteří mně byli oporou.

Prohlášení

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

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 4. června 2020 .....

Odkaz na tuto práci

Adamec, Matěj. *Portál srovnávače faktur se smlouvami z registru smluv.* Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.

Abstrakt

Ministerstva České republiky uzavírají smlouvy s dodavateli a následně vznikají faktury, které jsou vypláceny. O obou položkách jsou těmito ministerstvy zveřejňovány informace. Avšak ucelený přehled o tom, jaké faktury patří k určité smlouvě neexistuje a propojení položek je často nadlidský výkon.

Tato bakalářská práce se zabývá návrhem metod pro mapování faktur na uzavřené smlouvy, které tato ministerstva v České republice zveřejňují. Dále se také věnuje implementaci těchto metod a tvorbě webové aplikace, která bude výsledky prezentovat a upozorní na případné nesrovnalosti.

Aby bylo možné vytvořit spojení, byly popsány jednotlivé atributy a vztahy, které byly použity při mapování. Následně byly navrhnuty a implementovány tři aplikace. První, která extrahuje informace o fakturách a smlouvách z dostupných zdrojů. Druhá aplikace mapuje faktury na smlouvy na základě definovaných testů. A třetí, webová aplikace, která tyto výsledky prezentuje a upozorňuje na případné nesrovnalosti. Přínosem této práce je nový pohled na nakládání státu s veřejnými prostředky a lepší kontrola státních institucí.

Klíčová slova otevřená data, faktury, registr smluv, ministerstva České republiky, mapování, webový portál, prezentace výsledků

Abstract

The Ministries of the Czech republic enter into contracts with suppliers and subsequently issue invoices that are paid out. The ministries publish information on both of these items. However, a comprehensive overview of which invoices belong to a particular contract does not exist, and linking them is often hard.

Therefore, this bachelor's thesis deals with a design of methods for mapping invoices to contracts, published by ministries in the Czech Republic. Part of this thesis is also an implementation of these methods and the creation of a web application that presents the results and points out any discrepancies.

In order to create a connection, the individual attributes and relationships that can be used in mapping are described. Subsequently, three applications are designed and implemented. The first one extracts information about invoices and contracts from available sources and stores them in a database for further processing. The second application maps invoices to contracts based on defined tests. The third one, a web application, presents the mapping results and points out contracts where discrepancies have been identified. This work's contribution is a new perspective on the management of public funds and better control of state institutions.

Keywords opendata, invoices, registr smluv, ministry of the Czech republic, mapping, web portal, presentation of the results

Obsah

Seznam obrázků

Seznam výpisů kódů

Úvod

Ministerstva musí od 1. 7. 2016 podle zákona č. 340/2015 Sb. zveřejňovat uzavřené smlouvy v registru uveřejněny.[1] Dále dle zákona č. 106/1999 Sb. jsou státní instituce povinny poskytnout informace o nakládání s veřejnými financemi.[2] Tyto zákony umožňují občanům kontrolovat, jak stát a jeho instituce nakládají s finančními zdroji.

Běžný občan může o data požádat nebo si je stáhnout z již dostupných zdrojů a zkontrolovat jednotlivé položky. Informace o fakturách a smlouvách nejsou zveřejněna ve formátu, který dovoluje jednoduše určit, které položky k sobě patří a které ne. Propojení dat je obtížnější také díky tomu, že ve většině případů data neobsahují přímou vazbu, která by jednoznačně určila, ke které položce patří. Tento problém nemá triviální řešení.

Proto je v dnešní době nutné mít program, který dokáže tyto vazby vytvořit. Díky nim bude možné veřejné instituce lépe kontrolovat a poukazovat na plýtvání veřejnými prostředky. Tyto vazby mohou být vytvořeny pomocí informací, které smlouvy a faktury obsahují a mají je společné. Například podle předmětu smlouvy, variabilního symbolu a částky. Každá informace může být využita k potvrzení nebo naopak k vyvrácení spojení.

Cílem práce je navrhnout systém, který vytvoří vazby mezi fakturami zveřejňovanými ministerstvy České republiky a smlouvami zveřejněnými v Registru smluv. K tomu, aby systém mohl fungovat správně, budou navrženy metody pro mapování faktur na příslušné smlouvy. Pro účel zobrazení výsledků mapování bude vytvořena webová aplikace, která bude tyto výsledky prezentovat a upozorní na případné nesrovnalosti.

Aby systém mohl vazby vytvořit, budou analyzována data, která ministerstva zveřejňují, a také definovány způsoby, které bude možné využít ke stáhnutí a zpracování dat. Rovněž budou určeny způsoby, které jsou pro dosažení výsledku nejvýhodnější.

Po analýze dat vznikne systém, který tato data extrahuje a následně použije metody, které na základě analýzy dat vytvoří spojení mezi fakturami

ministerstev ČR a smlouvami zveřejněnými v Registru smluv.

Posledním cílem této práce je vytvoření webového portálu, který bude tyto výsledky zobrazovat. Zároveň bude upozorňovat na nesrovnalosti, jako je například součet vyfakturovaných částek převyšující smluvní hodnotu, a základní statistiky, které je možné z dat vyvodit.

Proto se v první kapitole zaměřuji na analýzu informací o fakturách a smlouvách ministerstev České republiky, která jsou veřejně dostupná. Jsou zde popsány jednotlivé atributy zveřejňovaných informací.

V druhé kapitole řeším návrh metod pro mapování faktur na smlouvy. Za tím účelem jsou popsány vztahy, které budou použity při mapování. Dále jsou také popsány navržené modely, které budou použity při zpracování dat.

Třetí část se věnuje implementaci tří aplikací. První aplikace slouží ke stažení potřebných dat. Druhá aplikace, které za použití navržené metody vytvoří spojení mezi fakturami a smlouvami. A třetí webová aplikace, která prezentuje data a výsledky párování.

Důvodem pro vypracování této práce je můj zájem o práci s reálnými daty a možnost vytvoření programu, který může pomoci s odhalováním podvodů a kontrolou státu.

Kapitola 1

Analýza

1.1 Zákon o registru smluv

Zákon č. 340/2015 Sb. ze dne 24. 11. 2015, kterému se také říká Zákon o registru smluv, ukládá subjektům státní správy a samosprávy, ale také organizacím a firmám jimi zřízených a vlastněných, povinnost uveřejňovat své smlouvy v Registru smluv. Mezi subjekty, kterých se zákon týká, patří státní úřady, kraje, větší obce, státní příspěvkové organizace a fondy, veřejné výzkumné instituce, veřejné vysoké školy a firmy, ve kterých má stát, kraj nebo obec většinový podíl. Tyto instituce musí zveřejňovat smlouvy, jejichž plnění přesahuje částku 50 000 Kč bez DPH. Tato smlouva musí být nahrána ve strojově čitelné podobě, což mi dovoluje data následně analyzovat a využít při kontrole.[5]

1.2 Zákon o svobodném přístupu k informacím

Právo na přístup k informacím je uvedeno již v Listině práv a svobod, kde je v 17. článku uvedeno „Svoboda projevu a právo na informace jsou zaručeny. [...] Svobodu projevu a právo vyhledávat a šířit informace lze omezit zákonem, jde-li o opatření v demokratické společnosti nezbytná pro ochranu práv a svobod druhých, bezpečnost státu, veřejnou bezpečnost, ochranu veřejného zdraví a mravnosti.“ a „Státní orgány a orgány územní samosprávy jsou povinny přiměřeným způsobem poskytovat informace o své činnosti. Podmínky a provedení stanoví zákon.“[6]

Jana Fabíková popisuje ve své práci zákon o svobodném přístupu k informací takto: „Zákon č. 106/1999 Sb., o svobodném přístupu k informacím ve znění pozdějších předpisů jako obecně právní norma zajišťuje právo veřejnosti na informace. Povinné subjekty jsou tímto zákonem zavázány především k tomu, aby zveřejňovaly základní a standardní informace o své činnosti tak, aby byly všeobecně přístupné.

Současná právní úprava svobodného přístupu k informacím reaguje na vývoj legislativy v českém a evropském právu, realizuje nově kladené požadavky, získané zkušenosti z vlastní realizace zákona a propojuje režim správního řízení.“[7]

1.3 Existující projekty

Tato sekce se věnuje projektům nebo programům, které mají podobné cíle, jako tato bakalářská práce. Jedná se o projekty, které se snaží veřejnosti přiblížit nakládání státu s finančními zdroji nebo se zabývají propojením vystavených účetních dokladů se smlouvami/objednávkami.

1.3.1 Hlídač státu

Jedním ze serverů, které kontrolují státní zakázky a další aktivity je Hlídač státu. Tento server se zaměřuje na využívání otevřených dat, analýzu jednotlivých datasetů a zveřejňování informací o aktivitách státu a politiků. Projekt Hlídače státu vznikl v roce 2017 a od té doby vzniklo několik dílčích projektů, které se zaměřují na kontrolu částí státní správy a zvýšení kontroly veřejných prostředků ze strany občanů. Součástí Hlídače státu jsou například Hlídač smluv (Kontrola dat v Registru smluv a provázání dat, které obsahuje), Hlídač webů (nezávislý monitoring IT služeb státní správy), Hlídač osob a politiků (Informace o politicích, sponzorech politických stran a osob navázaných na politiky) a další.[11]

Projekt Hlídače státu (konkrétněji Hlídače smluv) je podobný této práci v tom, že si klade za cíl upozornit na smlouvy, které jsou zveřejněny v Registru smluv je u nich podezření na plýtvání a zneužití moci v úřadech. Hlavní rozdíl této práce od Hlídače státu je ten, že tato práce má za cíl vytvoření vazeb mezi fakturami a smlouvami a na základě těchto vazeb upozornit na možné plýtvání prostředky nebo na podezřelé zakázky. Hlídač státu tyto vazby nevyužívá a kontrolu provádí na základě dat samotné smlouvy.

1.3.2 Supervizor

Je aplikace Ministerstva financí vytvořená za účelem vizualizace datasetů zveřejněných Ministerstvem financí a dalších organizací, které se do projektu chtějí připojit.[12] Tato aplikace zobrazuje jednotlivé rozpočtové položky, které je možné rozdělit na jednotlivé dodavatele, a na faktury s nimi spojené. Tento projekt má částečně společný cíl, náklady ministerstva umožňuje veřejnosti prohlížet a kontrolovat.

Od této práce se liší tím, že Supervizor Ministerstva financí pouze zobrazuje datasety se zveřejněnými fakturami, ale nesnaží se účetní doklady propojit se smlouvami a objednávkami.

1.3.3 Microsoft Dynamics 365 Finance

Dalšími projekty, které se zabývají spojováním faktur a objednávek jsou účetní programy, které mohou firmy využívat pro své účetnictví. Jedním z nich je například Microsoft Dynamics 365 Finance, který se zaměřuje na správu účetnictví středních a velkých organizací. V tomto programu je také funkcionalita, která poskytuje uživateli možnost automatického proplácení faktur, které společnost obdrží od svých dodavatelů. Při tomto procesu jsou faktury spojovány s objednávkami, a pokud splňují některá pravidla, jsou automaticky proplaceny. Pokud kritéria nesplňují, jsou označeny a vyžadují manuální kontrolu.[13]

Tento program poskytuje několik druhů spojování, kde je porovnávána celková cena, množství a částka za jednu položku. V případě dat, která jsou dostupná, je možné využít podobný postup, ale v mnoha případech tato pravidla nebudou dostačující a budu muset využít další atributy, které smlouvy a faktury obsahují. Proto tento program není možné použít.

1.4 Faktury a smlouvy

Faktura je účetní doklad, který musí podle zákona č. 563/1991 Sb. (Zákon o účetnictví) splňovat následující parametry:

(a) označení účetního dokladu

(b) obsah účetního případu a jeho účastníky

(c) peněžní částku nebo informaci o ceně za měrnou jednotku a vyjádření množství

(d) okamžik vyhotovení účetního dokladu

(e) okamžik uskutečnění účetního případu, není-li shodný s okamžikem podle písmene d),

(f) podpisový záznam podle § 33a odst. 4 osoby odpovědné za účetní případ a podpisový záznam osoby odpovědné za jeho zaúčtování [8]

Tato definice se vztahuje na účetní doklady obecně a určuje povinné údaje, které vystavená faktura musí obsahovat.

Smlouva může být definována takto: „Smlouva není nic jiného, než jeden ze způsobů, jakým vznikají závazky (obligace). Smlouva je dvoustranný či vícestranný právní úkon, jímž si smluvní strany sjednávají určitá práva a povinnosti. Jednoduše řečeno jde o dohodu, která zakládá právní vztah mezi stranami smlouvy.“[9]

Existuje několik druhů smluv (např. nájemní smlouva, smlouva o dílo, ...) a není definován přesný tvar a atributy, které musí každá smlouva obsahovat. Hlavním zdrojem smluv, který bude v této práci využíván, je Registr

smluv podle Zákona o registru smluv (předpis č. 340/2015 Sb.), který spravuje Ministerstvo vnitra České republiky a v metodické příručce k Registru smluv jsou povinné údaje popsány takto: „Povinnými metadaty jsou identifikace smluvních stran, vymezení předmětu smlouvy a datum uzavření smlouvy. Povinným metadatem je také cena, a pokud jí smlouva neobsahuje, tak hodnota předmětu smlouvy, lze-li ji určit.“ [10]

Faktury a smlouvy jsou dva základní objekty, se kterými v této práci budu pracovat. Konkrétně mě zajímají smlouvy, které uzavřelo ministerstvo jako jedna ze smluvních stran. A také faktury, které byly ministerstvu zaslány, byly proplaceny a jsou spojené s některou z uzavřených smluv.

1.5 Popis zdrojů dat

První částí analýzy je popis zdrojů dat, které je možné použít pro extrahování uzavřených smluv a vystavených faktur. Každý ze zdrojů má své výhody a nevýhody, a proto je důležité je analyzovat a vybrat ten, který bude nejlepší. Dostupné jsou následující zdroje:

- Národní katalog otevřených dat
- Webové stránky ministerstev
- Databáze Opendata
- API Hlídače státu
- Registr smluv

Všechna ministerstva musí podle Zákona o svobodném přístupu k informacím (č. 106/1999 Sb.) zveřejňovat nebo poskytnout na základě žádosti faktury, které ministerstvo vyplatilo. Zákon ale nespecifikuje přesný formát, ve kterém jsou data poskytována. Pouze specifikuje, že musí být zveřejněn ve strojově čitelném formátu, který popisuje takto: „Strojově čitelným formátem se pro účely tohoto zákona rozumí formát datového souboru s takovou strukturou, která umožňuje programovému vybavení snadno nalézt, rozpoznat a získat z tohoto datového souboru konkrétní informace, včetně jednotlivých údajů a jejich vnitřní struktury.“ [2]

1.5.1 Národní katalog otevřených dat

Národní katalog otevřených dat, nebo také NKOD, je server vytvořený Ministerstvem vnitra ČR ve spolupráci s Fakultou informatiky a statistiky Vysoké školy ekonomické v Praze. Tento server si klade za cíl vytvořit jedno místo, na kterém bude možné dohledat všechny datové sady, které jsou zveřejňovány veřejnou správou. V otevřených datech je možné vyhledávat a stahovat soubory, které jsou zde uveřejněny. K souborům jsou uvedeny podmínky, za

kterých mohou být využívány a metadata, která katalog uchovává, jako například čas zveřejnění, periodicita a další. Data jsou uložena přímo v katalogu nebo je zde uveden odkaz na úložiště instituce, která data zveřejnila. V současné době je v katalogu již zveřejněno přes 130 tisíc datových sad od 39 poskytovatelů.[3]

Pří vyhledání datasetů s fakturami je možné zjistit, že pouze některá ministerstva je v katalogu publikují. Mezi tato ministerstva patří: ministerstvo dopravy, ministerstvo zdravotnictví, ministerstvo životního prostředí, ministerstvo pro místní rozvoj, ministerstvo financí a ministerstvo obrany.

Poté jsou zde ministerstva, která pro uveřejnění nevyužívají NKOD, ale zveřejňují informace o fakturách na svých webových stránkách. Těmi jsou například ministerstvo průmyslu a obchodu, ministerstvo kultury a ministerstvo spravedlnosti.

Ostatní ministerstva je nikde nepublikují a poskytnou je pouze na základě žádosti. Těmi jsou ministerstvo vnitra, ministerstvo práce a sociálních věcí, ministerstvo školství, mládeže a tělovýchovy, ministerstvo zemědělství a ministerstvo zahraničních věcí.

1.5.2 CKAN API a webové stránky ministerstev

Některá ministerstva, která zveřejňují data na svých webových stránkách, šla o krok dál a implementovala rozhraní CKAN API, které usnadňuje přístup ke katalogům s otevřenými daty. Data mohou být poté připojena do národního katalogu otevřených dat.

CKAN API je opensource webová platforma pro publikaci a vyhledávání dat. Platforma umožňuje zveřejňování dat pomocí API a pomocí rozhraní je možné vyhledávat v datech, které instituce zveřejňila. CKAN API je určen pro státy, orgány veřejné správy, samosprávy a podniky. Tato platforma je využívána po celém světe například Evropskou Unií, Velkou Británií a nebo také Brazílií.[4]

V této práci se věnuji získání dat, a proto se zaměřím na API pro vyhledávání a získání datasetů, které mě zajímají a můžu je využít při párování dat. CKAN API využívá JSON jako formát pro data, které vrací jako odpověď na dotaz. Rozhraní poskytuje dvě hlavní funkce a to package_list a package_show.

První funkce package_list umožňuje vypsání všech datasetů, které jsou dostupné. Identifikátory mohou být poté využity k získání dat pomocí druhé funkce package_show. Například API Ministerstva financí po dotázání se na adresu `https://data.mfcr.cz/cs/api/3/action/package_list` vrátí JSON, který obsahuje 3 části. Jak je vidět v části kódu 1.1, první prvek s názvem „help“ obsahuje informaci o datech a dále i parametrech, které je možné využít pro snížení počtu výsledků. Druhá část „success“ obsahuje pravdivostní hodnotu true, která značí, že výsledky se povedlo získat. A třetí část „result“ obsahuje identifikátory jednotlivých datasetů.

```
{
  "help":"Return a~list of the names of the site's
  datasets (packages)...",
  "success":true,
  "result":[
      "zakladni-sazby-zahranicniho-stravneho",
      "seznam-vladnich-instituci-v-cr",
      "seznam-objednavek-ministerstva-financi-cr",
      "statistika-zadosti-podle-zakona-1061999-sb",
      "kontroly-v-oblasti-eet",
      ...
      "prehled-faktur-ministerstva-financi-cr",
      ...
      ,"report-provoznich-nakladu-mf"
  ]
}
```


Výpis kódu 1.1: JSON soubor vrácený po dotázání na dostupné datové sady ministerstva financí

Druhá funkce CKAN API je package_show, která požaduje jeden parametr, který identifikuje datovou sadu. Výsledek vyhledávání pomocí funkce package_list obsahoval identifikátor „prehled-faktur-ministerstva-financi-cr“, jehož dataset podle názvu obsahuje informace o fakturách, které ministerstvo financí ČR přijalo. Pokud se tedy dotáži na adresu `https://data.mfcr.cz/cs/api/3/action/package_show?id=prehled-faktur-ministerstva-financi-cr` je vrácen JSON, který obsahuje informace o datasetu. Jak je v kódu 1.2 vidět, dataset obsahuje metadata, která obsahují informace jako například jméno datasetu, kdy a za jakých podmínek byl dataset zveřejněn a další. Dále také obsahuje část pojmenovanou „resources“, která obsahuje informace i souborech, které dataset tvoří. Odkaz pro stažení souboru je pod položkou `url`. Tato adresa odkazuje na soubor, který je možné stáhnout a data z něj extrahovat.

Některá ministerstva, jako například ministerstvo průmyslu a obchodu, zveřejňují data na svých webových stránkách. Z rozcestníku nebo pomocí vyhledávání je možné najít požadovaný dataset a pomocí odkazu ho stáhnout. Získání dat z těchto stránek nemusí být vždy snadné, především pokud url, které odkazují na jednotlivé soubory, nemají společný tvar, ve kterém se mění pouze časové údaje.

Data uveřejněná v katalogu nebo na stránkách ministerstva jsou zveřejňována v různých formátech (csv, xml, xls, xlsx) a různě strukturované. Každé ministerstvo také rozhoduje, jak často bude informace zveřejňovat (měsíčně, jednou za půl roku, ročně). Zpracování dat také komplikuje fakt, že tvar, ve

```
{
  "help":"",
  "success":true,
  "result":{
    "id":"c8ecdac8-922f-4014-bd0f-fc12a7e659fa",
    "name":"prehled-faktur-ministerstva-financi-cr",
    "title":"Přehled faktur Ministerstva financí ČR",
    "publisher_name":"Ministerstvo financí české republiky",
    "publisher_uri":"http://www.mfcr.cz",
    "maintainer_email":"otevrenadata@mfcr.cz",
    "ruian_code":"1",
    "ruian_type":"ST",
    "license_title":"Jedná se o˜volné dílo.",
    "license_link":"https://portal.gov.cz/portal/ostatni/
                    volny-pristup-k-ds.html",
    "notes":"...",
    "url":"https://data.mfcr.cz/cs/dataset/
           prehled-faktur-ministerstva-financi-cr",
    "state":"Active",
    "metadata_created":"2015-01-15T18:45:12+01:00",
    "metadata_modified":"2020-03-06T18:44:06+01:00",
    "type":"dataset",
    "resources":[
      {
        "id":"649107b9-e2d8-4b47-92c5-024c6a200c0a",
        "url":"https://opendata.mfcr.cz/exports/faktury/csv",
        "description":"<p>Bez fondu privatizace<p>",
        "format":"csv",
        "mimetype":"",
        "state":"Active",
        "name":"Uhrazené faktury MF",
        "size":"",
        "created":"2019-11-13T11:29:55+01:00",
        "last_modified":"2020-03-06T16:28:01+01:00"
      }, {
        ...
      }
    ]
  }
}
```


Výpis kódu 1.2: Odpověď package_show funkci

kterém jsou data zveřejňována, se může měnit. Změnit se může pojmenování sloupců nebo položek v souborech, ale také kódování znaků. U souborů ve formátu csv jsou důležité oddělovače a způsob označení souvislého textu. Tyto prvky se u jednotlivých souborů mohou také změnit. Na webu otevřených dat jsou uveřejněny doporučení, jak data zveřejňovat, ale tyto pravidla dodržují jen některé instutuce.

Většina souborů obsahuje informaci o účastnících (jméno, IČO, název firmy), označení faktury (číselné nebo textové), suma (v Kč nebo s uvedenou měnou), datum vystavení a datum zaplacení. Ostatní údaje některá ministerstva nezveřejňují. Například variabilní symbol, číslo položky rozpočtu, identifikátor smlouvy nebo předmět plnění.

Některá ministerstva zveřejňují spolu s fakturami také export smluv. Tato práce využívá jako hlavní zdroj smluv Registr smluv, ale tyto exporty mohou obsahovat informace, které nemohou být uveřejněny v Registru Smluv, protože to rozhraní neumožňuje. Příkladem takové informace může být datum platnosti smlouvy, které mně může při mapování velice pomoci, protože omezí množství možných párování, ale zároveň tato informace není v Registru smluv obsažena. Některé exporty dokonce obsahují přímé spojení mezi fakturou a smlouvou, a proto by mohlo být užitečné tyto údaje využít.

Webové stránky ministerstev nebo Národní katalog otevřených dat představují základní zdroj informací, ale parsování dat může být náročné. Proto by mohlo být výhodné využití nástrojů, které získání dat ulehčí.

1.5.3 OpenDataLab a Opendata aplikace

OpenDataLab je projekt, který vznikl v říjnu roku 2018 ve spolupráci Fakulty informačních technologií Českého vysokého učení technického v Praze a společnosti Profinit EU, která se zaměřuje na poradenství, návrh, optimalizaci a vývoj softwaru. Tato laboratoř se zabývá využitím opendat, která jsou zveřejňována státními institucemi a dalšími veřejnými subjekty.[14]

Program s názvem Opendata [15] je jedním z programů, který vznikl v rámci projektu OpenDataLabu a je zaměřen na extrahování dat z datasetů, která ministerstva České republiky zveřejňují. Aplikace je napsána v jazyce Java a její instalace a spuštění je snadné. Aplikace po spuštění začne stahovat jednotlivé soubory, data transformuje do objektů, které jsou následně nahrané do předem specifikované databáze. Aplikace je vytvořena tak, aby extrahovala většinu údajů, které se v souborech objevují, ale bohužel ne všechny. Některé údaje jsou vynechány, a to například údaje, které přímo specifikují, ke které smlouvě faktura patří.

1.5.4 Hlídač státu API

Hlídač státu krom upozorňování na konkrétní zakázky poskytuje také API, pomocí kterého je možné se dotázat na konkrétní smlouvu nebo exportovat

větší množství smluv. API Hlídače státu také poskytuje data v definovaném tvaru, ale některé klíčové informace, které by bylo možné použít při párování, neobsahuje. Dotazování na velké množství dat v malém časovém úseku by mohlo být náročné pro poskytovatele, a proto by bylo nutné rozložit dotazování na delší časový úsek tak, aby nezatěžoval servery.

1.5.5 Popis dostupných dat - Faktury

Zde je popis jednotlivých položek, které se v souborech s fakturami ministerstev objevují. Některé položky jsou zveřejňovány pouze jedním ministerstvem a některé mohou být kombinací dvou dalších.

Identifikátor faktury dodavatele: Je označení faktury dodavatelem. Jedná se pravděpodobně o účetní identifikátor. Ne vždy je uveden ve formě prefixu a čísla, někdy je uveden jen v podobě čísla. Může být unikátní pro dané ministerstvo nebo pouze pro daný soubor nebo časové období.

Označení dokladu: Jedná se o prefix účetního označení faktury. Skládá se ze 1-3 znaků, které mohou rozlišovat zaplacené faktury například od zálohovaných. Prefix může být například v podobě jednoho písmenka F, FAK nebo FD1. Společně s číslem dokladu může tvořit celý účetní identifikátor.

Číslo dokladu: Je číselné označení faktury. Spolu s označením dokladu může tvořit celý účetní identifikátor. Ministerstva neoznačují číslo dokladu jednoznačně, takže v souborech, které poskytují, je více údajů, které je možné za číslo dokladu považovat.

IČO ministerstva: Je identifikační číslo ministerstva. Je uvedeno jen v několika souborech, ale je možné ho odvodit podle zdroje dat.

Název ministerstva: Je textový název ministerstva.

IČO dodavatele: Je identifikační číslo dodavatele. Je uvedeno téměř vždy.

Název dodavatele: Textový název dodavatele.

Celková částka: Je částka, kterou ministerstvo zaplatilo. Pokud není dostupná informace o měně, je uvedena v Korunách českých. Někdy je uvedena jako celé číslo. Někdy také jako číslo na dvě desetinná místa. Některá ministerstva také oddělují tisíce mezerou. V některých souborech jsou uvedeny dva sloupce s částkou a to jeden pro částku v Kč a druhý pro částku v cizí měně. V takovém případě je uveden také sloupec s měnou.

Částka za položku: Je částka za jednu nakoupenou položku. Může se lišit od celkové částky v případě, že se faktura vztahuje na několik položek.

Částka bez DPH: Je zaplacená částka po odečtení DPH.

Měna: Je měna, ve které byla částka zaplacena. Někdy je uvedena v podobě dvou písmen (KČ) a někdy v podobě tří (CZK, EUR).

Účel platby: Je textový popis účelu platby. Někdy je uveden pouze heslovitě. Jindy je popis delší a obsahuje informace, které mohou pomoci s párováním. Může se jednat o produkt nebo dokonce číslo smlouvy. Je zde také možnost testování, které by zjistilo, zda se části věty neshodují s předmětem nebo popisem smlouvy.

Variabilní symbol: Je číselné označení platby nebo několika plateb. Jedná se o číslo, které se skládá maximálně z deseti číslic. Může se jednat o náhodně vygenerované číslo, a nebo také o číslo faktury nebo smlouvy, ke které patří.

Datum přijetí: Je datum, kdy ministerstvo obdrželo fakturu. Datum přijetí by mělo být větší nebo stejné jako datum vystavení. Dále by toto datum by mělo být stejné nebo nižší než je datum úhrady a datum splatnosti. Nemá jednotný formát. Uveden je rok, měsíc a den.

Datum splatnosti: Je datum, do kterého má ministerstvo fakturu zaplatit. Mělo by být pozdější, než je datum vystavení, přijetí a úhrady. Nemá jednotný formát. Uveden je rok, měsíc a den.

Datum úhrady: Je datum, kdy ministerstvo fakturu zaplatilo. Mělo by být dřívější, než datum splatnosti. Dále by mělo být pozdější nebo stejné jako datum vystavení a přijetí. Nemá jednotný formát. Uveden je rok, měsíc a den.

Datum vystavení: Je datum, kdy byla faktura vystavena dodavatelem. Mělo by být dřívější, než je datum přijetí, úhrady a splatnosti. Nemá jednotný formát. Uveden je rok, měsíc a den. Tento údaj je možné využít v případě, že ministerstvo spolupracuje s dodavatelem delší dobu. Toto datum by mělo být mezi datem vytvoření a datem platnosti u smluv na dobu určitou.

Kód rozpočtové položky: Je číselný identifikátor rozpočtové položky, ke které faktura patří. Je uveden především u novějších faktur. Tento kód by mohl být použit pro omezení možných navázání v případě že ministerstvo vyplatilo více faktur jednomu dodavateli na různé rozpočtové položky.

Název rozpočtové položky: Je textový název položky v rozpočtu ministerstva.

Identifikátor smlouvy: Je identifikátor smlouvy, na kterou je faktura vystavena. Jedná se o číselný nebo textový řetězec, který odkazuje na identifikátor smlouvy. Vě většíně souborů není uveden.

1.5.6 Popis dostupných dat - Smlouvy

Zde je popis údajů o smlouvě, které je možné získat pomocí API Registru smluv.

Identifikátor smlouvy: Je číselný identifikátor smlouvy v Registru smluv. Nejčastěji se jedná číslo skládající se ze 7 číslic, ale může být i kratší nebo delší. Pro každou smlouvu je unikátní. Ale v případě, že má smlouva více verzí, zůstává tento identifikátor stejný. Proto je lepší používat identifikátor verze.

Identifikátor verze: Je číselný identifikátor verze v Registru smluv. Pomocí tohoto čísla je možné odvodit webovou stránku, na které jsou informace o smlouvě zobrazeny. Tento identifikátor je unikátní v celém systému a identifikuje jednoznačně smlouvu a její verzi. Nejčastěji se jedná číslo skládající se ze 7 číslic, ale může být kratší i delší.

Odkaz: Je URL webové stránky, na které jsou informace o smlouvě zobrazené. Jedná se o zřetězení „`https://smlouvy.gov.cz/smlouva/`“ a identifikátoru verze.

Datum zveřejnění: Je datum ve formátu ISO 8601, které označuje okamžik zveřejnění smlouvy v registru.

Datová schránka subjektu: Je identifikátor datové schránky subjektu. Skládá se ze 7 znaků.

Název subjektu: Je textový název subjektu.

IČO subjektu: Je identifikační číslo subjektu. Skládá se z 8–mi číslic. Některá starší čísla mohou mít méně cifer, ale jsou doplněny na začátku o nuly. v Registru smluv se objevují plné i neúplné.

Adresa subjektu: Je adresa, na které je zadavatel registrován. Je uvedena ulice, číslo popisné, poštovní směrovací číslo, město a stát.

Útvar: Je název útvaru, odboru nebo části subjektu.

Označení plátce: Je označení, zda je subjekt plátce. Jedná se o hodnotu „0“ nebo „1“, kde „1“ znamená, že subjekt je příjemce a hodnota „0“ znamená, že subjekt není příjemce. Toto označení je v mnoha případech prázdné nebo stejné jako atribut označení dodavatele.

Název smluvní strany: Je textový název dodavatele. Některé smlouvy v Registru smluv mají v tomto atributu uvedený odbor, adresu nebo další informace.

IČO smluvní strany: Je identifikační číslo dodavatele. Skládá se z 8–mi číslic. Některá starší čísla mohou mít méně cifer, ale jsou doplněny na začátku o nuly. v Registru smluv se objevují plné i neúplné.

Adresa smluvní strany: Je adresa, na které je zadavatel registrován. Je uvedena ulice, číslo popisné, poštovní směrovací číslo, město a zkratka státu.

Označení příjemce: Je označení, zda je smluvní strana stranou příjemce. Jedná se o hodnotu „0“ nebo „1“, kde „1“ znamená, že smluvní strana je příjemce služby a hodnota „0“ znamená, že je dodavatelem. Toto označení je v mnoha případech prázdné nebo stejné jako atribut označení dodavatele.

Předmět: Je textový popis předmětu smlouvy.

Datum uzavření: Je datum, kdy byla smlouva uzavřena. Je uveden ve formátu ”dd-mm-yyyy”. Datum je nižší než datum zveřejnění.

Číslo smlouvy: Je textový identifikátor smlouvy, který se skládá z několika částí oddělených lomítkem. Tento identifikátor je přiřazen ministerstvem.

Schválil: Je jméno osoby, která smlouvu schválila. Obsahuje také titul osoby.

Hodnota bez DPH: Je částka, na kterou byla smlouva uzavřena bez započítání DPH. Uvedena v Kč.

Hodnota včetně DPH: Je částka, na kterou byla smlouva uzavřena se započítaným DPH. Uvedena je v Kč. Jedná se o částku větší než je hodnota bez DPH.

Navázaný záznam: Je identifikátor smlouvy, ke které je tato smlouva navázána.

Platný záznam: Hodnota „0“ nebo „1“, která označuje, zda je tento záznam platný. Pokud má jedna smlouva více verzí, příznakem 1 je označena pouze nejnovější verze smlouvy. Ostatní jsou označeny 0. Pokud je smlouva označena jako nevalidní a neexistuje smlouva se stejným číslem, které je validní, tak byla tato smlouva znepřístupněna na základě požadavku subjektu. Taková smlouva se zobrazí v exportu, který poskytuje Registr smluv, ale není možné ji zobrazit ve webové verzi.[1]

Registr smluv neobsahuje informaci o datu ukončení, ale je někdy obsažen v souborech, které ministerstva zveřejňují. A proto tento údaj budu uvádět.

1.6 Popis chyb nebo anomálií ve smlouvách

Informace o smlouvách v registru smluv mohou být uvedeny ve tvaru, ve kterém by je uživatel nečekal. Může se jednat validní způsob, který je doporučen v metodice Registru Smluv, nebo také o chybu ve zveřejněných datech. Některé z těchto anomálií jsou zde popsány.

1.6.1 Název subjektů

Názvy subjektů ve smlouvách v Registru smluv často obsahují informace, které by měly být zařazeny v jiném atributu nebo mají pouze popisný efekt:

Označení země: Název subjektu obsahuje, krom jména instituce, také název země.

Například „Česká republika - Ministerstvo obrany“ nebo „ČR - Ministerstvo obrany“

Názvy útvarů a částí subjektu: Informace o útvaru nebo části instituce je uvedena ve nazvu subjektu a ne v přímo části, která je pro to určena.

Například „ČR Ministerstvo obrany - Vojenský útvar v 4312“

Adresy: V nazvu je uvedena adresa instituce.

Například „Ministerstvo zemědělství, Těšnov 65/17, 110 00 Praha 1 - Nové Město“

Jména osob: V názvu se objevují jména osob, které smlouvu uzavíraly.

Například „ČR - Ministerstvo kultury; PhDr. Ilja Šmíd; Maltézské nám. 1; 118 11 Praha 1; Univerzita Karlova; doc. Mirjam Friedová; Ph.D., Ovocný trh 5; 116 36 Praha 1; Národní památkový ústav; Ing. arch. Naděžda Goryczková; Valdštejnské náměstí 3; 118 01 Praha 1;“

Jméno systému: V názvu subjektu je uvedeno jméno systému, ve kterém je smlouva uveřejněna. Například „Registr smluv (Ministerstvo vnitra)“

1.6.2 Chybějící IČO u subjektu

Pro párování smluv a faktur je primárně použito IČO ministerstva a IČO dodavatele. Proto je důležité, aby smlouvy tyto údaje obsahovaly. Registr smluv ale obsahuje smlouvy, které IČO dodavatele neobsahují. Tento stav může nastat u fyzické osoby, které nejedná v rámci své podnikatelské činnosti. Ale u ostatních subjektů by mělo být vyplněno IČO nebo datová schránka.

Příkladem může být smlouva mezi Ministerstvem dopravy České republiky a akciovou společností „Veletrhy Brno a.s.“ s předmětem „Objednávka - Motosalon 2019“. IČO dodavatele je možné najít například pomocí webového

portálu `rejstrik.penize.cz` a dohledat, že tato akciová společnost má IČO 25582518.

Některé IČO by bylo možné doplnit pomocí dotazů na stránky, jako je například `https://or.justice.cz`, kde je možné na základě jména vyhledat informace o právnické osobě.

1.6.3 Záporná cena

Cena uvedená ve smlouvě může být záporná v případě, že se jedná o dodatek ke smlouvě. v tom případě upravuje hodnotu smlouvy, na kterou je navázána.

1.6.4 Neuvedená cena

Zákon uvádí cenu jako jeden z povinných údajů, ale také specifikuje výjimky, u kterých cena uvedena být nemusí. Cena nemusí být uvedena u smluv, které jsou na dobu neurčitou a není možné odhadnout, jak velkou částku bude nutné zaplatit. Nebo také u smluv, kde je cena považována za obchodní tajemství.

1.6.5 Neidentifikovatelný subjekt

Registr smluv obsahuje také smlouvy, u kterých jsou místo jména a dalších atributů dodavatele uvedeny znaky „x“. Jedná se o případ, kdy smlouva obsahuje osobní údaje, obchodní nebo bankovní tajemství.

1.6.6 Prodloužení smlouvy

Některé smlouvy jsou po ukončení doby platnosti prodlouženy a i tento fakt by měl být zaznamenán v Registru Smluv. Existuje doporučený postup při zadávání prodloužení smlouvy, podle metodiky k Registru Smluv. Měl by být vytvořen nový záznam, stejně jako kdyby byla vytvořena nová smlouva, a ten provázat se smlouvou, která je prodlužována. Záznamy, které smlouvu prodlužují, nemusí mít nutně stejné smluvní strany (stejné atributy).

Například smlouva mezi ministerstvem financí a akciovou společností ABF, a.s. s předmětem smlouvy „Nájemní smlouva - pronájem pozemků v k.ú, Letňany (AVISme 2016000625)“.

Tato smlouva je poté dvakrát prodloužena, ale už s jiným subjektem (jiné údaje, ale ve skutečnosti se jedná o stejnou firmu).

Nejdříve je prodloužena s „PVA EXPO, a.s.“ dne 27. 3. 2018. a poté se stejnou akciovou společností dne 4. 3. 2019.

1.6.7 Neplatné smlouvy

V metodickém návodu k aplikaci zákona o Registru smluv je uvedeno následující:

„Registr smluv je zřízen zejména pro účely evidování uzavřených smluv. Není možné z něj zjistit, zda je uveřejněná smlouva platná či nikoliv, neboť

uveřejněná smlouva mohla být např. vypovězena nebo chyběla vůle k jejímu uzavření a nejedná se tak vůbec o smlouvu (tyto skutečnosti není potřeba promítat v registru smluv).“[10]

Z důvodu, který je uveden výše, není možné zjistit, zda je smlouva neplatná, a proto není možné tyto smlouvy vyfiltrovat a nezahrnovat mezi potenciální smlouvy, ke kterým faktura může patřit.

Kapitola 2

Návrh

V této části je popsán návrh aplikace, pro mapování faktur na uzavřené smlouvy. Popisuji zde atributy a vztahy, které budou použity při vytváření spojení. Následně popisuji také třídy, které jsou použity při implementaci a jsou zde uvedeny návrhy vzhledu webového portálu.

2.1 Atributy a vztahy, které je možné využít při párování

V této části detailněji popíšu atributy faktur a smluv, které mohou být využity při párování faktur na smlouvy. Společně s atributy zde popisuju i vztahy a jak je možné je využít.

2.1.1 Identifikační čísla

IČO ministerstva a IČO dodavatele je jeden z hlavních atributů, který využiji. Tento atribut je pro mne velmi důležitý, protože je možné pomocí něj omezit množinu všech spojení, která k jedné faktuře mohou existovat. Když je smlouva vytvořená, obsahuje IČO ministerstva a IČO dodavatele. Následně faktury, které jsou s touto smlouvou spojené, obsahují stejná identifikační čísla. Proto je možné vytvořit pravidla, díky kterým budou brány v úvahu pouze smlouvy a faktury, které patří ke stejnému ministerstvu a dodavateli.

Tento údaj může v některých případech chybět, a proto je potřeba využít další atributy k nalezení možných spojení. V některých případech by mohlo být možné IČO dohledat na základě jména nebo dalších údajů.

Identifikační číslo je možné použít pro vytvoření spojení. Tento údaj má dva formáty, ve kterých se uvádí, že je potřeba ho upravit nebo nastavit pravidla tak, aby s dvojím formátem počítaly. Jeden formát je řetězec, který se skládá z osmi číslic, kde první znaky mohou být tvořeny nulami. Druhý formát neobsahuje úvodní nuly a obsahuje pouze znaky od první nenulové číslice až

do konce. Pro ulehčení je možné již při načítání dat identifikační čísla, která mají méně než osm znaků, doplnit o úvodní nuly.

2.1.2 Název subjektu a název ministerstva

Název je pro uživatele atribut, který je při spojování použit jako první. Při vytváření spojení ale není vždy nejlepší. Pokud bychom ho využili při definování pravidel, v nichž by se využíval název, bylo by nutné počítat s překlepy a různými derivacemi slov, které se v tomto atributu objevují. Proto je místo názvu při identifikaci možné využít IČO, u kterého není nutné řešit možné chyby a přebytečná slova.

Ale v případě, že IČO u záznamu chybí, je tento údaj jeden z klíčových při dohledávání společných záznamů. V takovém případě je pomocí názvu možné dohledat IČO v některé z databází nebo na webovém serveru, které tyto informace obsahují.

Jak název, tak i IČO ministerstva nebo dodavatele je možné využít při používání externích systémů, které mohou být využity při získávání dodatečných informací, které jsou poté použity při testování spojení.

2.1.3 Časové údaje smlouvy a faktury

Obecně je u smluv možné mluvit o následujících časových údajích:

- Datum uzavření smlouvy
- Začátek platnosti smlouvy
- Konec platnosti smlouvy

U faktur je časových údajů více:

- Datum vystavení faktury
- Datum přijetí faktury
- Datum zaplacení faktury
- Datum splatnosti faktury

Mezi těmito časovými údaji jsou vztahy, které je možné využít k výběru nejlepší možné kombinace. Časové údaje smlouvy jsou v následujícím pořadí:

Uzavření smlouvy $\leq$ Začátek platnosti $\leq$ Konec platnosti smlouvy

Údaje spojené s fakturou jsou seřazeny následovně:

Datum vystavení $\leq$ Přijetí faktury $\leq$ Zaplacení faktury $\leq$ Splatnost faktury

Tyto vztahy platí v ideálním případě. Například datum zaplacení faktury nemusí být nutně před datem splatnosti. Ale ve většině případů tento vztah platí, a proto si tento vztah dovolím využívat. Dále se také nestává, že datum začátku platnosti smlouvy je stejné jako datum ukončení smlouvy. Ale zrušení této rovnosti by nijak nesnížil počet potenciálních spojení a pro jednoduchost toto pravidlo bude využito.

Toto jsou ale vztahy mezi časovými údaji uvnitř jednoho objektu. Při párování je využito kombinace těchto pravidel.

Pokud je smlouva uzavřená a je platná, mohou na smlouvu vznikat faktury. Ve chvíli, kdy je faktura vystavena dodavatelem, mělo by platit, že datum vystavení faktury je pozdější, než datum uzavření smlouvy a datum začátku platnosti smlouvy. Stejný vztah vůči datu uzavření a platnosti smlouvy by měl platit u data přijetí, zaplacení a splatnosti faktury.

Na základě zmíněných pravidel by mělo platit:

Datum uzavření smlouvy $\leq$ Datum platnosti smlouvy $\leq$ Datum vystavení faktury $\leq$ Datum přijetí faktury $\leq$ Datum zaplacení faktury $\leq$ Datum splatnosti faktury

Při párování budou porovnávány především položky ze smlouvy oproti položkám ve faktuře a je dobré pravidla kontrolovat odděleně, protože některá data mohou chybět.

Sada pravidel by tedy vypadala následovně:

- Datum uzavření smlouvy $\leq$ Datum vystavení faktury
- Datum uzavření smlouvy $\leq$ Datum přijetí faktury
- Datum uzavření smlouvy $\leq$ Datum zaplacení faktury
- Datum uzavření smlouvy $\leq$ Datum splatnosti faktury
- Datum začátku platnosti smlouvy $\leq$ Datum vystavení faktury
- Datum začátku platnosti smlouvy $\leq$ Datum přijetí faktury
- Datum začátku platnosti smlouvy $\leq$ Datum zaplacení faktury
- Datum začátku platnosti smlouvy $\leq$ Datum splatnosti faktury

Podobná pravidla je možné uvést i pro datum ukončení platnosti smlouvy. Každý z časových atributů účetního dokladu by měl být časově zasazen před ukončením smlouvy. I v tomto případě mohou nastat výjimky, a proto toto kritérium bude bráno pouze jako pomocné.

Příkladem, kdy toto pravilo může pomoci, je například faktura zveřejněná ministerstvem financí ČR, kde je dodavatelem akciová společnost BISNODE ČESKÁ REPUBLIKA, A. S. a časové údaje jsou následující:

- Datum vystavení faktury: 4. 1. 2010
- Datum přijetí faktury: Prázdné
- Datum zaplacení faktury: 1. 2. 2010
- Datum splatnosti faktury: 5. 1. 2010

K této faktuře je možné za pomoci identifikačních čísel ministerstva a dodavatele navázat 2 smlouvy, které mají datum uzavření následující:

- 27. 6. 2005
- 30. 12. 2019

Pokud porovnám data faktury s daty smluv, vyjde najevo, že faktura nemohla být uzavřena na 2. smlouvu, protože ta byla uzavřena až 9 let poté.

Tento příklad také poukazuje na to, že pravidla, která jsem definoval, nemusí platit vždy. Datum splatnosti faktury je 5. 1. 2010, ale datum zaplacení je až téměř měsíc poté dne 1. 2. 2010.

Dále i po vyřazení jedné ze dvou položek si stále nemohu být jistý tím, že faktura ke smlouvě patří. Například proto, že faktura může být na smlouvu, která hodnotou nepřevyšuje částku 50 000 Kč, a záznam o smlouvě tak nemusí existovat.

2.1.4 Předmět smlouvy a faktury

Předmět smlouvy je často položka, kterou člověk využije při rozhodování, zda záznamy patří k sobě. Často tento atribut obsahuje společné prvky, jako například smluvenou věc nebo službu. Někdy mohou být dokonce úplně stejné. Pro tento účel by bylo možné použít funkci, která by dokázala určit, že dva texty pojednávají o stejné věci. Ale aby tato funkce mohla fungovat, muselo by se jednat o funkci, která využívá algoritmy pro zpracování přirozeného jazyka.

Abych mohl použít předmět smlouvy a faktury, identifikuji jednotlivé prvky a vztahy, které mi mohou pomoci při rozhodování, bez použití komplikovanějších algoritmů.

2.1.4.1 Stejné předměty

Pokud je předmět smlouvy naprosto stejný jako předmět faktury, je větší pravděpodobnost, že budou tyto položky patřit k sobě, než dvě položky, které předmět stejný nemají.

Příkladem může být faktura Ministerstva obrany, kde dodavatelem je EUROGASTRO CATERING, S. R. O. Předmět smlouvy i faktury je „Stravování u VZ 551210 Štěpánov (doba plnění 4.1.2014- 3.1.2018)“.

Důležitou roli při rozhodování také hraje délka textu. Pokud se jedná o celou větu, které obsahuje několik slov, údajů a čísel, je tento údaje použitelnější, než předmět, který obsahuje pouze jedno nebo dvě slova.

2.1.4.2 Částečná shoda

Přesná shoda je silné pravidlo, ale existují případy, kdy se jedná pouze o částečnou shodu. Tato shoda může nastat například tehdy, když předmět jedné položky je součástí předmětu druhé položky.

Příkladem může být smlouva a faktura mezi Ministerstvem obrany a společností TONERSYSTEM, S. R. O., kde předmětem smlouvy je „18/7/1/102/2017-6624 Stavební a truhlářské řezivo“ a předmětem faktury „NS 662400, 18/7/1/102/2017-6624 Stavební a truhlářské řezivo, DUD 01701166“

Částečná shoda může také nastat, když některá slova v předmětu jedné položky, se objevují u položky druhé. Může se jednat o jedno nebo více slov. Pokud je jedna faktura a dvě smlouvy, které se mohou k účetnímu dokladu navázat a je potřeba rozhodnout, která smlouva je ta správná nebo vhodnější, je možné tento test provést, a podle výsledků se rozhodnout. Při rozhodování je ale potřeba sledovat délku společných slov, a jak velkou část tyto společná slova představují.

Například pokud předmět smlouvy je „Údržba modulů PIS, ORACLE a Stravování“ a předmět faktury „Podpora PIS, Oracle, Stravování, Docházka“, je možné si všimnout, že oba předměty mají společná slova „PIS“, „ORACLE“ a „Stravování“. Tato slova představují u smlouvy 3/6 slov (pokud je spojka „a“ počítána jako slovo). U faktury společná slova představují 3/5 slov. Na druhou stranu, pokud je smlouva s předmětem „551240 Řezivo a překližka“ a faktura s předmětem „Nákup rour a kolen kouřovodu“, tak při otestování je zjištěno, že mají společné pouze slovo „a“. Toto slovo představuje u smlouvy 1/4 všech slov a u faktury 1/5. Přestože mají společný pouze jeden znak, tak jsou tímto porovnáním získany celkem velká čísla. Je možné spočítat, jak velkou část společná slova představují z celkové délky slova. U smlouvy se podíl rovná 1/22 a u faktury 1/24. Tato čísla více odpovídají části, kterou v textu společná slova představují.

Kontrolu komplikuje fakt, že v českém jazyce jsou slova skloňována. Proto testy, které by kontrolovaly, zda se slovo (bez úprav) vyskytuje v předmětu druhé položky, nebudou mít tak velkou validitu.

Také v této části by mohlo být použity metody NLP (Natural Language Processing), ale to je nad rámec této práce.

2.1.4.3 Jiné atributy

Předmět faktury může obsahovat, krom informací o předmětu a službě, kterou se zavázal dodavatel dodat nebo vykonat, i některé další údaje, které mohou mít spojitost se smlouvou, na kterou jsou vykázány.

Jedním z těchto atributů je číslo smlouvy. Pokud se tento údaj vyskytuje v předmětu smlouvy, tak se zvyšuje šance, že je toto spojení správné.

Příkladem takového předmětu faktury může být následující předmět:
„135000 /606900, PZ 601740100959/12/17, sml.č. 155110134 - pož. list č. 40/2017

AUT.OBRN. IVECO 50B5“. Tento předmět obsahuje číslo smlouvy, díky kterému je možné dohledat smlouvu, ke které patří.

Některé předměty obsahují také časové údaje smlouvy. Jako je například rok začátku a ukončení platnosti nebo datum uzavření smlouvy. Často se ale nemusí jednat o tyto časové údaje a nemusí mít s danou smlouvou nic společného. Proto by se testům, které tuto skutečnost chtějí kontrolovat, neměla přikládat velká váha.

Některé atributy je potřeba před porovnáním upravit, protože jsou ve faktuře obsaženy, ale v neupravené formě. Úpravou je myšleno například odstranění nečíselných hodnot nebo převedení data do jiného formátu. Stejná úprava by měla proběhnout na straně předmětu, protože v něm může také nastat situace, že obsahuje číslo smlouvy, ale s přidanými znaky.

Další z vlastností, která nesmí být přehlédnuta, a která hraje roli při provádění těchto testů, je délka atributu. Různou váhu má atribut o jednom znaku a atribut o 7 nebo 10 znacích. A proto v případě, že se číslo smlouvy skládá z menšího počtu znaků, tento test nebrat jako validní. Hranice, při které je již délka atributu dostačující se pohybuje okolo 4–5 znaků.

2.1.5 Částka

Dalším atributem, který je možné použít při rozhodování, je částka, která je uvedena v účetním dokladu. Faktura může obsahovat částku bez DPH, částku s DPH nebo částku v jiné měně než v českých korunách. Tyto údaje mají jistý vztah k částkám, které mohou být uvedeny ve smlouvě. Zde je možné rozdělit smluvy do tří skupin.

Do první skupiny patří smlouvy, u kterých není možné odhadnout výslednou částku. U těchto smluv chybí údaje o částce a zde není žádný vztah mezi částkami uvedenými ve faktuře a částkami uvedenými ve smlouvě.

Do druhé skupiny patří smlouvy, kde je předmětem provedení jedné služby nebo dodání jednoho produktu. U těchto smluv se dá očekávat, že dodavatel po dokončení nebo dodání zašle ministerstvu jednu fakturu, která bude obsahovat částku, která je blízko té ve smlouvě.

Jako příklad uvedu smlouvu a fakturu mezi Ministerstvem obrany a společností PREMO, S. R. O., kde je smlouva na nákup tonerů uzavřena na částku 349 619.82 Kč (včetně DPH) a faktura, která byla vystavena 22. 3. 2017, obsahuje také částku 349 619.82 Kč. V takovém případě bychom měli spojení mezi touto smlouvou a fakturou upřednostnit před ostatními spojeními, kde je rozdíl v částkách větší.

Do třetí skupiny smluv patří ty, které jsou uzavřeny na delší časové období, skládají se z několika částí nebo jejich plnění probíhá postupně, ale je možné u nich odhadnout výslednou cenu. V takovém případě může dodavatel poslat několik faktur a součet částek v nich uvedených by měl být blízko částce uvedené ve smlouvě.

Tento vztah je vidět u smlouvy o poskytnutí vysílacího času mezi Ministerstvem obrany a společností CET 21 spol. s. r. o., kde celková částka je 4598000 Kč včetně DPH. K této smlouvě je možné vytvořit spojení na tři faktury, které jsou na částky 2 390 960 Kč, 1 547 992.93 Kč a 659 047.07 Kč. Součet těchto částek je 4 598 000 Kč, což je přesně smluvená částka.

2.1.6 Číslo smlouvy

Tento identifikátor představuje označení smlouvy v systémech ministerstva, které smlouvu zveřejnilo. Tento údaj může být často klíčový při testování spojení. Kontrola, zda se vyskytuje v předmětu faktury, může velice pomoci. Často se jedná o jedné vodítko, které ale má velkou váhu. Někdy je možné z tohoto identifikátoru odvodit některé další informace.

Například ministerstvo financí využívá čísla smluv, která jsou ve formátu „SML:YY/DDD/DDDD“, kde SML je označení smlouvy, „YY“ je označení roku, kdy byla smlouva uzavřena, poté dvě čísla, která jsou smlouvě přiřazeny podle neznámého pravidla. Pokud je k této smlouvě vytvořen dodatek, je označen stejným číslem smlouvy a na konec je přidáno „/“ a číslo dodatku. Jako příkladu uvedu smlouvu s číslem „SML:94/013/0002“ a pokud by existoval dodatek k této smlouvě, byl by označen číslem „SML:94/013/0002/01“.

2.1.7 Variabilní symbol

Variabilní symbol představuje sadu číslic, které jsou uvedeny ve faktuře a takový řetězec se může jevit jako ideální identifikátor smlouvy, ke které patří. Bohužel jsem při bližším zkoumání zjistil, že je variabilní symbol shodný s číslem smlouvy ojediněle nebo se zřídka vyskytuje v předmětu smlouvy, ale ve většině takových případů se jedná o náhodu. Takže tento údaj použít není možné.

2.2 Metoda mapování

V této části se zaměřím na metodu, která bude použita pro vytvoření spojení mezi fakturou a smlouvou. Popíši jednotlivé fáze a jaké jsou výstupy.

2.2.1 Vytvoření potenciálních spojení

Na začátku jsou data rozdělena na dvě skupiny – Faktury a smlouvy. Kdyby bylo žádoucí vytvořit všechna potenciální spojení, která mohou existovat, vytvořili bychom kartézský součin těchto množin. To by bylo neefektivní a zbytečné, ale měl bych jistotu, že jsem žádnou z možností nevynechal. Abych nemusel pracovat s tak velkými daty, omezil jsem, kdy bude spojení vytvořeno.

Pro tento účel mi poslouží identifikační údaje ministerstva a dodavatele. Tyto údaje jasně určují, kdo s kým uzavřel smlouvu a následně vystavené faktury budou obsahovat údaje stejné.

Proto na začátku vytvořím kartézský součin, ale musí platit, že IČO ministerstva a dodavatele je stejné na straně smlouvy a také na straně faktury.

Při vytváření spojení se často stává, že po aplikaci zmíněného pravidla zaniknou všechna potenciální spojení na některý záznam. To může být způsobeno tím, že data, která reálně patří k dané položce, nejsou k dispozici.

2.2.2 Zpracování a vyhodnocení spojení

Když mám vytvořenou množinu spojení, je nutné je nějakým způsobem porovnat s ostatními a poté rozhodnout, které spojení je to nejlepší.

Pro tento účel definuji dvě části:

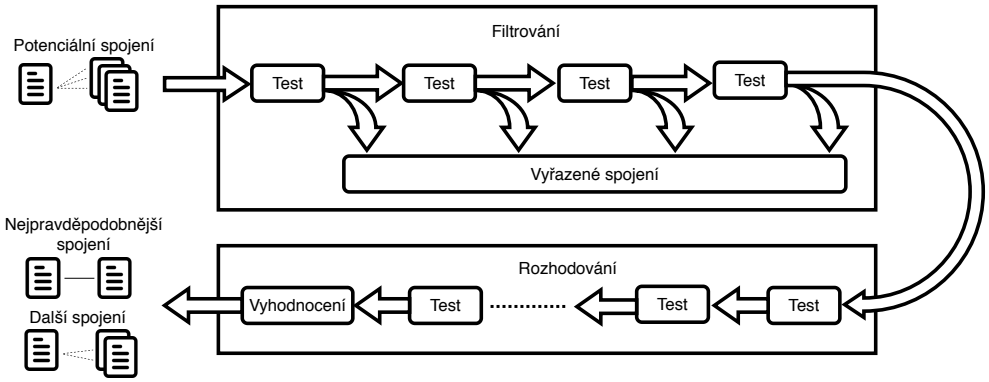
- Filtrování
- Rozhodování

Filtrování
Jak je vidět na obrázku 2.1, na začátku máme spojení, která se týkají jedné faktury. Tento set spojení je na začátku vložen do procesu filtrování. V tomto procesu jsou prováděny testy, které musí každé spojení splnit. Pokud některý z testů nesplní, je vyřazeno. Testy v tomto procesu využívají především časové údaje.

Rozhodování
Poté, co spojení projde filtrování, je předáno do procesu Rozhodování. V tomto procesu jsou prováděny testy, které mají za úkol ohodnotit spojení na základě atributů, ale také vztahu k ostatním spojením. Proto jsou testovány spojení na shodné části v předmětu, výskyt číselných údajů, testy na uvedené částky a další. Na konci tohoto procesu je vyhodnocení, které rozhodne, které spojení je to nejlepší.

Po dokončení procesu rozhodování jsou vráceny dvě položky. První je nejpravděpodobnější spojení, pokud bylo některé vybráno. Druhou položkou je množina spojení, která sice nebyla vybrána za nejpravděpodobnější, ale přesto jsou uchována.

Tento proces byl navrhnut na základě způsobu, jakým sám spojuji fakturu k smlouvě. Na začátku vyfiltruji ty, které nesplňují základní pravidla a následně na základě menších detailů a vztahů vyhodnocuji, které spojení je to nejlepší.



Obrázek 2.1: Diagram zpracování spojení

2.3 Databáze

Při práci s větším množstvím dat je potřeba data někam ukládat. Pro tento účel je použita relační databáze. V databázi budou vytvořeny tabulky pro uložení informací o fakturách a smlouvách, ale také pro uložení výsledků párování a testování.

Pro tento účel vzniknou následující tabulky:

- **contract**: pro uchování informací o smlouvách
- **invoice**: pro uchování informací o fakturách
- **possible_relation**: pro uchování informací o spojeních
- **test_result**: pro uchování informací o výsledcích testů
- **contract_warning**: pro označení smluv, které jsou vyhodnoceny jako podezřelé
- **ministry**: pro uchování informací o ministerstvech
- **statistics**: pro uložení statistik o datech v databázi a urychlení dotazování
- **blocked_supplier**: pro uložení výjimek z mapování

Jak je vidět na diagramu 2.2, tabulky pro smlouvu a fakturu jsou vytvořeny tak, aby mohly obsahovat většinu informací, které je možné dohledat. Zároveň, krom identifikátorů, nejsou žádné údaje povinné, protože se může stát, že každý údaj bude v některém případě chybět. Některé údaje mohou vypadat zbytečně, protože při analýze nebylo zjištěno, že by bylo možné je použít při

testování. Přesto je zde pro ně vytvořen sloupec, protože se v budoucnu může stát, že právě tento údaj bude nutné použít.

Protože databáze bude obsahovat velké množství faktur a smluv, vyhledávání v těchto datech může být pomalé. Při párování ale bude nutné získat smlouvy, které mohou patřit k faktuře, na základě IČO ministerstva a IČO dodavatele.

Pro urychlení je možné použít indexy. Indexy jsou databázové struktury, sloužící k urychlení vyhledávání v datech. Z tohoto důvodu jsou v databázi s daty vytvořeny indexy nad sloupci ministry_ico a supplier_ico a to jak u smlouvy, tak u faktury. Díky indexům nebude nutné prohledávat všechny záznamy v databázi při vyhledávání pomocí zmíněných atributů, ale použijí se indexy, které zapříčiní, že se prohledá pouze zlomek.

2.4 Modely

V této části jsou popsány základní třídy, které jsou použity při načítání a zpracování dat. Modely jsou vytvořeny tak, aby bylo možné je následně použít při práci s databází pomocí ORM knihovny SQLAlchemy. Jednotlivé třídy jsou také znázorněny pomocí diagramu tříd na obrázku 2.3.

2.4.1 Contract

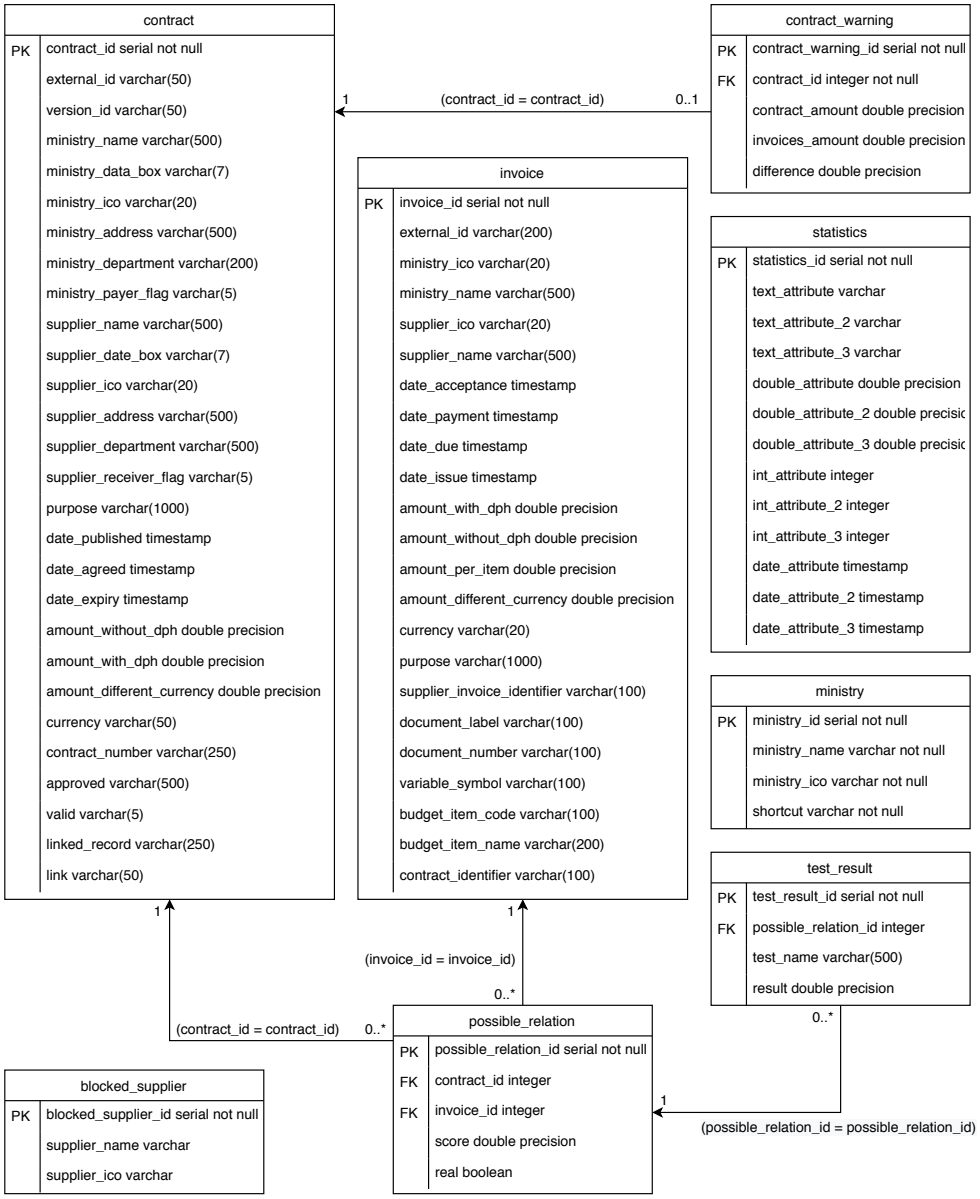
`Contract` představuje třídu pro reprezentaci smlouvy. `Contract` obsahuje všechny důležité atributy, které mohou být využity při párování. Každý atribut je nepovinný a to proto, že při analýze dat bylo zjištěno, že každý atribut může být prázdný. Buď kvůli tomu, že zdroj dat tyto údaje neobsahuje, nebo je chyba v datech, dále z důvodu ochrany soukromí a nebo pro zachování obchodního tajemství.

2.4.2 Invoice

`Invoice` přestavuje třídu pro reprezentaci faktury. Stejně jako třída `Contract` má všechny atributy nepovinné, protože i u faktury mohou být všechna pole prázdná.

2.4.3 PossibleRelation

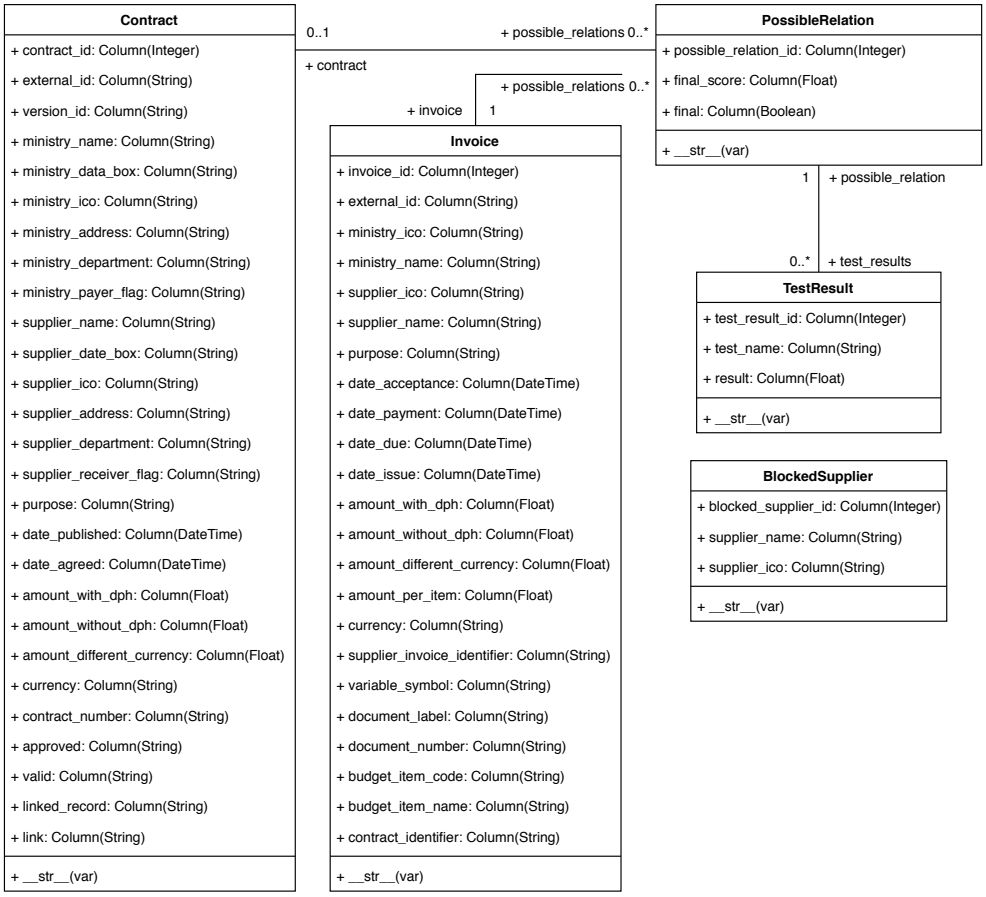
Třída `PossibleRelation` představuje možné spojení mezi fakturou a smlouvou. Objekt má atribut `score`, který představuje bodové hodnocení spojení. Čim více bodů spojení má, tím je spojení lepší. `Score` je přiřazeno na základě výsledků testů, kterým bylo spojení podrobeno. Po vytvoření a ohodnocení spojení je provedeno vyhodnocení a pokud je spojení označeno za nejpravděpodobnější, je hodnota atributu `real` nastavena na `True`. V opačném případě je hodnota nastavena na `False`.



Obrázek 2.2: Databázový diagram

2.4.4 TestResult

`TestResult` slouží jako záznam o výsledku testování. Každý test má své jméno, které popisuje, o jaký test se jedná. Výsledek testu poté definuje hodnota atributu `result`. Výsledek je ve formátu čísla s plovoucí desetinnou čárkou, protože není stanoven způsob, jak interpretovat výsledek. Interpre-



Obrázek 2.3: Diagram tříd - modely

tace je ponechána tomu, kdo vytváří výsledek testu a následné zpracování musí výsledek správně využít.

2.4.5 BlockedSupplier

Existují dodavatelé, kteří dělají především menší zakázky. U těchto dodavatelů vzniká mnoho faktur na smlouvy, které jsou pod hranicí 50 000 Kč. A po provedení párování se vytvoří mnoho spojení mezi fakturami a některými smlouvami, které v registr jsou, ale reálně faktury patří k těm, které zveřejněny nejsou. Ve výsledku vzniká velký rozdíl mezi hodnotou a smlouvy a vyfakturovanou částkou. Protože se většinou jedná o konkrétní dodavatele, je vytvořena tabulka `BlockedSupplier` a odpovídající model. Třída `BlockedSupplier` je později využita při filtrování.

2.5 Vzhled webového portálu

V této části představuji návrhy jednotlivých stránek webového portálu, na kterých budou prezentovány data a výsledky mapování faktur na smlouvy.

Webový portál by se měl skládat z následujících stránek:

- Přehled
- Přehled podle jednotlivých ministerstev
- Přehled faktur a smluv
- Detail faktury a smlouvy

2.5.1 Přehled

Přehled je první stránka, kterou uživatel uvidí. Tato stránka slouží k představení dat, která je možné najít na tomto webovém portálu a měla by obsahovat základní informace o datech, které prezentujeme. Společně s prezentací dat slouží také jako rozcestník k dalším stránkám.

Součástí přehledu by měl být počet faktur, smluv a počet nesrovnalostí. Společně s těmito čísly by zde měla být možnost přesměrování na seznam faktur, smluv a výčet nesrovnalostí, které byly na základě párování identifikovány.

Stránka může být doplněna o data, grafy a tabulky, které pomohou uživateli s orientací v datech. Zároveň by měla zůstat jednoduchá.

Návrh takové stránky je vidět na obrázku 2.4.

2.5.2 Přehled podle jednotlivých ministerstev

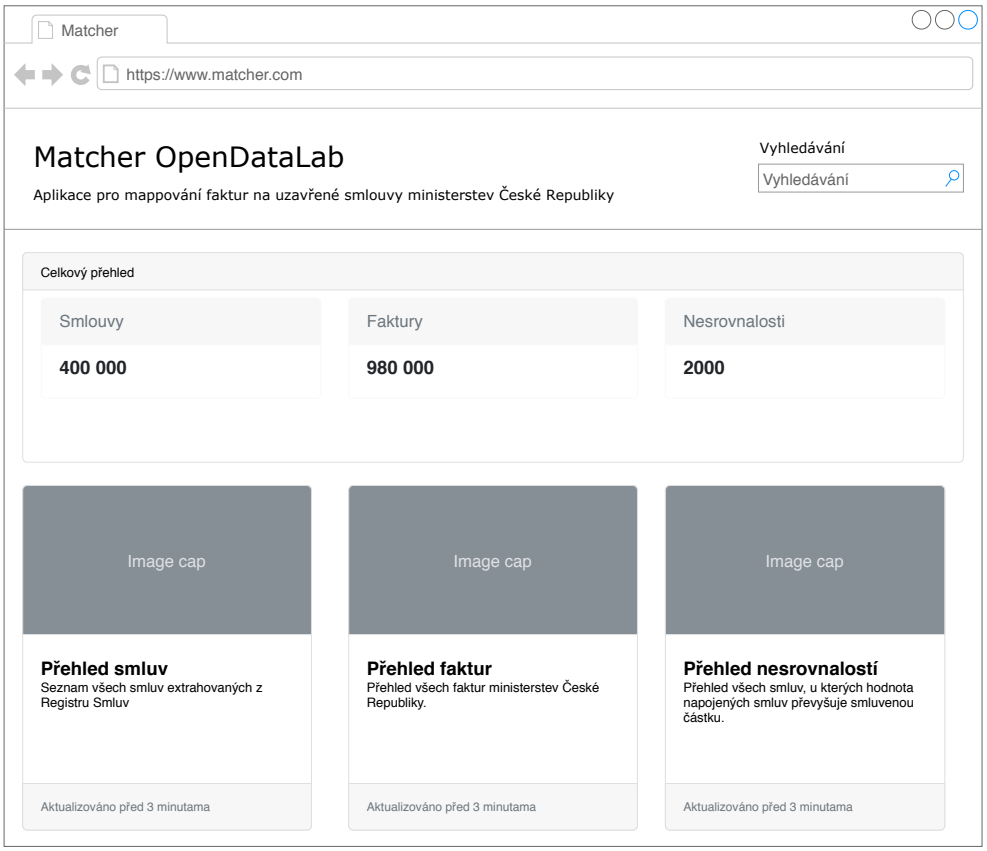
Přehled podle ministerstev má sloužit k představení dat rozdělených podle ministerstva, ke kterému patří. Každé ministerstvo by mělo mít v tomto přehledu sekci, kde budou informace uvedeny. Může se jednat o počty, grafy a tabulky.

Stránka by měla také sloužit k tomu, aby uživatel mohl porovnat výsledky napříč ministerstvy. Proto by jednotlivé sekce neměly být moc dlouhé. Zároveň každá informace, která je zde uvedena, by měla mít možnost přesměrování na stránku, kde je možné si konkrétní data prohlédnout.

Návrh takové stránky je vidět na obrázku 2.5.

2.5.3 Přehled faktur a smluv

Seznam faktur a smluv slouží jako velký přehled položek, které byly načteny ze zdrojů. Jedná se o dvě stránky, ale postavené na stejném principu. V tabulce jsou zde uvedeny jednotlivé záznamy, kde u každého je informace o ministerstvu, dodavateli, předmětu, datu vystavení nebo uzavření a částce. U každého



Obrázek 2.4: Návrh vzhledu stránky - Přehled

záznamu je navíc ještě možnost výběru operace, kterou je se záznamem možné provést. Například přesměrování na detail záznamu.

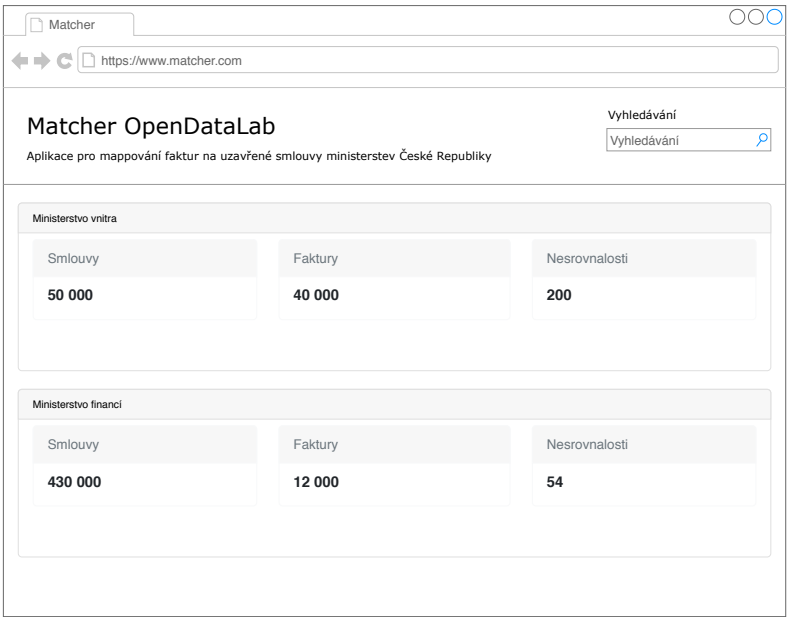
Stránka také obsahuje filtr, kde mohou uživatelé filtrovat podle základních údajů.

Návrh stránky se seznamem faktur a smluv je vidět na obrázku 2.6

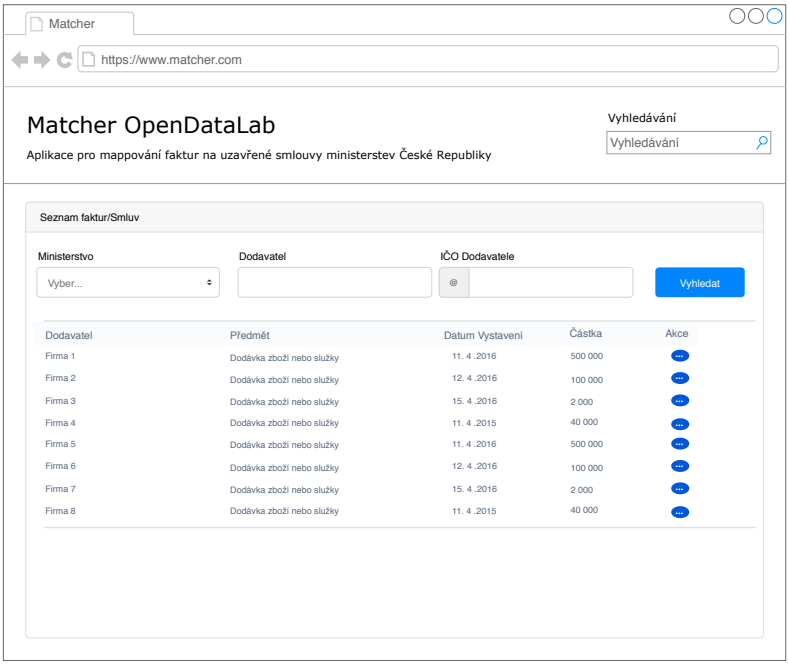
2.5.4 Detail faktury a smlouvy

Detail faktury a smlouvy slouží k tomu, aby si uživatel mohl zobrazit všechny detaily o konkrétním záznamu. Návrh je vidět na obrázku 2.7

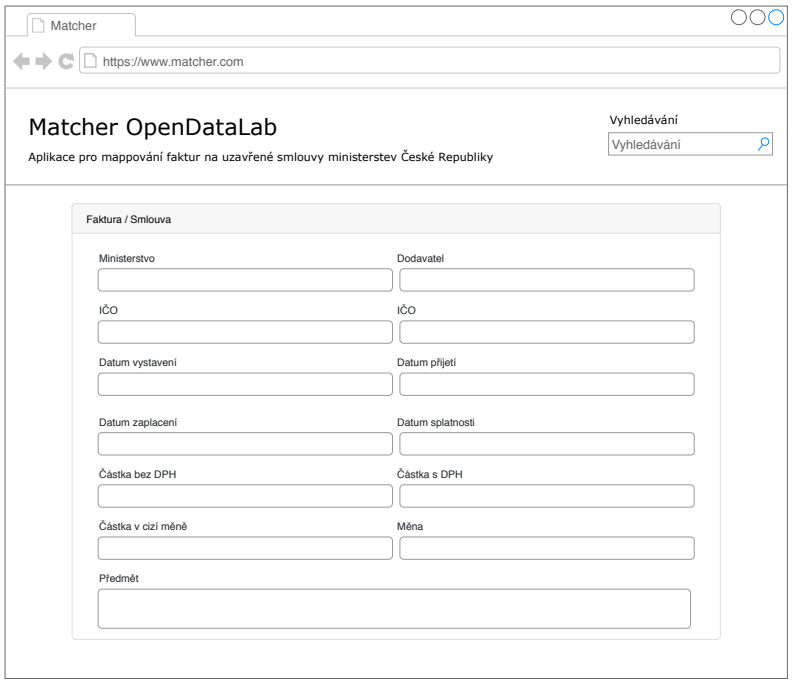
U smlouvy je také žádoucí, aby bylo uvedeno, zda jsou na smlouvu navázány faktury a pokud ano, tak zobrazit jejich počet, součet vyfakturovaných částek a porovnání s hodnotou smlouvy. Tyto faktury poté zobrazit v tabulce pod přehledem, aby uživatel mohl zkontrolovat, že je navázání správné. Pokud by potřeboval zobrazit detail, může se na něj přesměrovat pomocí tlačítka uvedeného u každého záznamu. Návrh přehledu je vidět na obrázku 2.8.



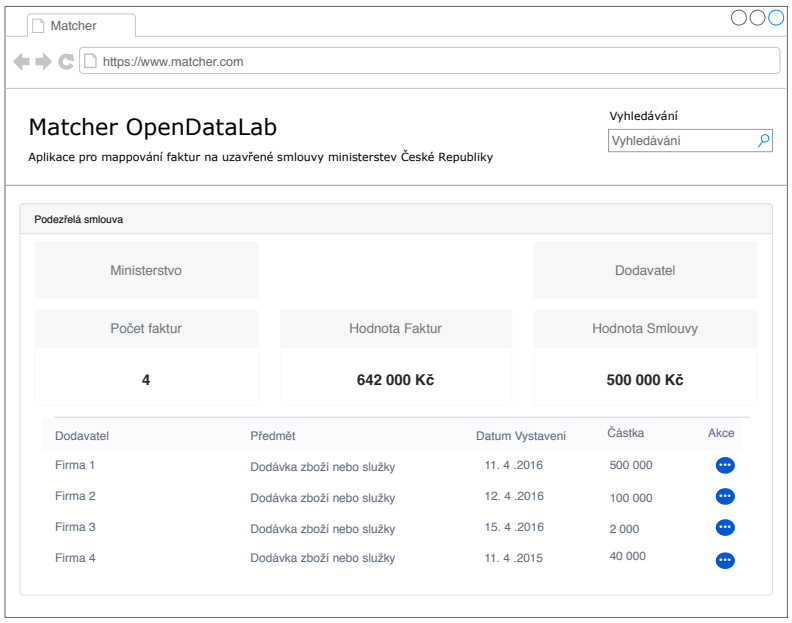
Obrázek 2.5: Návrh vzhledu stránky - Přehled podle jednotlivých ministerstev



Obrázek 2.6: Návrh vzhledu stránky - Seznam faktur a smluv



Obrázek 2.7: Návrh vzhledu stránky - Detail faktury a smlouvy



Obrázek 2.8: Návrh vzhledu stránky - Detail podezřelé smlouvy

Kapitola 3

Realizace

V této kapitole se budu věnovat popisu implementace aplikace, která vytváří spojení mezi fakturou a smlouvou, a také implementaci webové aplikace. Na začátku popíši technologie, které jsem si zvolil pro vývoj. Poté se zaměřím na konkrétní části aplikace, které jsou použity při párování faktur ke smlouvám. A na závěr popíši webovou aplikaci a jednotlivé stránky. Celý projekt je dostupný v GITHUB repositáři na adrese `https://github.com/opendatalabcz/invoice-contract-matcher`.

3.1 Použité technologie

V této části jsou popsány technologie, které jsou použity při tvorbě. Jsou zde popsány jak technologie pro tvorbu aplikace, která stahuje potřebná data, vytváří spojení, ale i pro tvorbu webového klienta a vystavení REST API.

3.1.1 Python

Python [16] je skriptovací a programovací jazyk, který se v dnešní době těší veliké popularitě. Jazyk navrhl v roce 1991 Guido van Rossum a od té doby se jazyk hodně vyvinul a stal se jedním z nejpoužívanějších na světě. Aktuální verze Python 3 přináší jednoduchou a přehlednou syntax, která dovoluje vývojářům psát přehledný kód. Od jazyků jako Java a C# se liší v tom, že využívá dynamickou kontrolu datových typů a také díky tomu se tento jazyk často používá při práci s daty, strojovém učení a při práci s umělou inteligencí.

Při zvažování, který programovací jazyk využiji při vývoji, jsem se rozhodl pro Python, jednak kvůli zkušenostem, které jsem s tímto jazykem měl a také kvůli tomu, že již od začátku jsem měl představu, že při párování faktur a smluv bude možné využít například umělou inteligenci pro rozpoznávání předmětu textu a další funkce a algoritmy, které jsou v tomto jazyce často vyvíjeny.

```python
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'
```

Výpis kódu 3.1: Minimální Flask aplikace

3.1.2 PostgreSQL

PostgreSQL [17] je objektově-relační databáze, která není vlastněna žádnou firmou a je vyvíjena jako open source. Tato databáze může být použita pro menší i větší projekty.

Příkladem dalších databází jsou například Oracle databáze nebo MySQL. Pro PostgreSQL jsem se rozhodl, protože se jedná o open source projekt, existuje dobrá dokumentace, je dobrá rozšiřitelnost a také proto, že ji využívá OpendataLab ve své aplikaci pro extrakci faktur a smluv ministerstev ČR.

3.1.3 Flask

Flask [18] je webový framework napsaný v jazyce Python. Jedná se mikroframework, protože Flask sám o sobě nepotřebuje další knihovny a nástroje pro spuštění.

Z kódu 3.1 je vidět, že vytvoření webové aplikace, která po navštívení zobrazí text „Hello, World!“ je velmi jednoduché.

Flask v základu neobsahuje ORM nebo další funkcionality, které ostatní webové frameworky, jako například Django, obsahují. To ale vývojářům nebrání v přidání rozšíření, které jsou navrženy přímo pro Flask a díky těmto rozšířením můžu doplnit funkcionality, které potřebuji. V této práci je Flask využit pro vystavení REST API, která bude poskytovat data webovým klientům.

Další podobné frameworky jsou například Django nebo Web2Py. Pro Flask jsem se ale rozhodl díky tomu, že je oproti zmíněným frameworkům extrémně jednoduchý, kdežto Django i Web2Py se řadí mezi full-stack frameworky.

3.1.4 SQLAlchemy

Příkladem knihovny, kterou je možné přidat k frameworku Flask, je SQLAlchemy. SQLAlchemy dává vývojáři možnost dotazovat se na relační databázi bez potřeby použití SQL. Tato funkcionalita je užitečná ve chvíli, kde se může používaná databáze změnit a bylo by nutné upravit velké množství kódu. Další výhodou používání ORM je automatická ochrana proti SQL Injection útokům, pokud nejsou použity příkazy s čistým SQL příkazem. Knihovnu SQLAlchemy je možné využít také bez použití v kombinaci s frameworkem Flask.

Mezi další python ORM se řadí například Django ORM nebo také SQLObject. Rozhodujícím faktorem v tomto případě byla možnost propojení s frameworkem Flask.

3.1.5 REST

REST (REpresentational State Transfer) je architektonický styl, který byl definován, aby pomohl s tvorbou a organizováním distribuovaných systémů.[20]

REST představuje čtyři principy:

- Všechny zdroje jsou identifikovány pomocí URI.
- Zdroj může mít více reprezentací.
- Zdroj může být vytvořen, získán, upraven nebo smazán pomocí standardních HTTP metod.
- Server si neukládá informaci o stavu.

HTTP metody, které slouží k práci se zdroji:

- GET: Slouží ke čtení zdrojů.
- POST: Většinou slouží k vytvoření nového zdroje.
- PUT: Většinou slouží k upravení nebo vytvoření nového zdroje.
- DELETE: Slouží ke smazání zdroje.

V této práci je princip REST využit především k jednoduchému přístupu k datům, která jsou v databázi. Ostatní operace mohou být přidány, pokud by byl portál rozšířen o prvky, které by vyžadovaly záznamy v databázi vytvářet, upravovat nebo mazat.

3.1.6 React

Pro tvorbu webového klienta jsem si zvolil framework React. React [19] je javascriptový webový framework pro vytváření webového uživatelského prostředí. Tento framework je vyvíjen společností Facebook a společně s frameworky Vue a Angular patří k nejpoužívanějším javascriptovým knihovnám pro tvorbu UI. Při vývoji jsou jsou důležité dvě části. První částí jsou komponenty. React je postaven na komponentách, které je možné skládat a vytvářet tak větší komponenty. Tento způsob umožňuje vytvářet části, které je možné testovat odděleně a v případě, že je potřeba některou část upravit, stačí upravit danou komponentu, a zbytek zůstane nedotčen. Druhou částí, která je při vývoji ve frameworku React důležitá, je stav. Každá komponenta si může udržovat svůj stav, který určuje, jak bude komponenta vypadat, která data bude obsahovat a jak se bude chovat. Pokud vezmu dvakrát jednu komponentu, kde obě budou

mít stejný stav, tak se zobrazí stejně. React zároveň dovoluje při změně stavu aktualizovat pouze ty komponenty, kterým je změněn stav. Díky tomu není potřeba znovu vykreslit celou stránku a UI je plynulejší.

I s využitím frameworku React může být tvorba uživatelského rozhraní obtížná. Cílem je vytvořit stránku, která bude přehledná a zároveň se nebude jednat o pouhý text. HTML obsahuje tlačítka, textová pole a další elementy, které je možné využít a aby vypadaly dobře, je nunté je doplnit o CSS a javascript. Abych nevytvářel něco, co již bylo vytvořeno, využil jsem jako základ šablonu od tvůrce jménem CreativeTim. Šablona se jmenuje Material Dashboard [21] a obsahuje kombinaci frameworku React a Bootstrap 4. Vzhled je inspirován principy Materal Design, které definovala společnost Google. K šabloně je dobrá dokumentace, která mi pomůže při upravení stránky tak, aby splňovala všechny naše potřeby.

Mezi podobné frameworky patří již zmíněný Angular a Vue, ale pro React jsem se rozhodl díky zkušenostem, množství dostupných zdrojů a způsobu práce se stavem a komponentami, který mně je sympatický.

3.2 Proces získání dat, párování a prezentace

Výsledkem této práce jsou čtyři programy, kde každý má za cíl jednu konkrétní funkcionalitu.

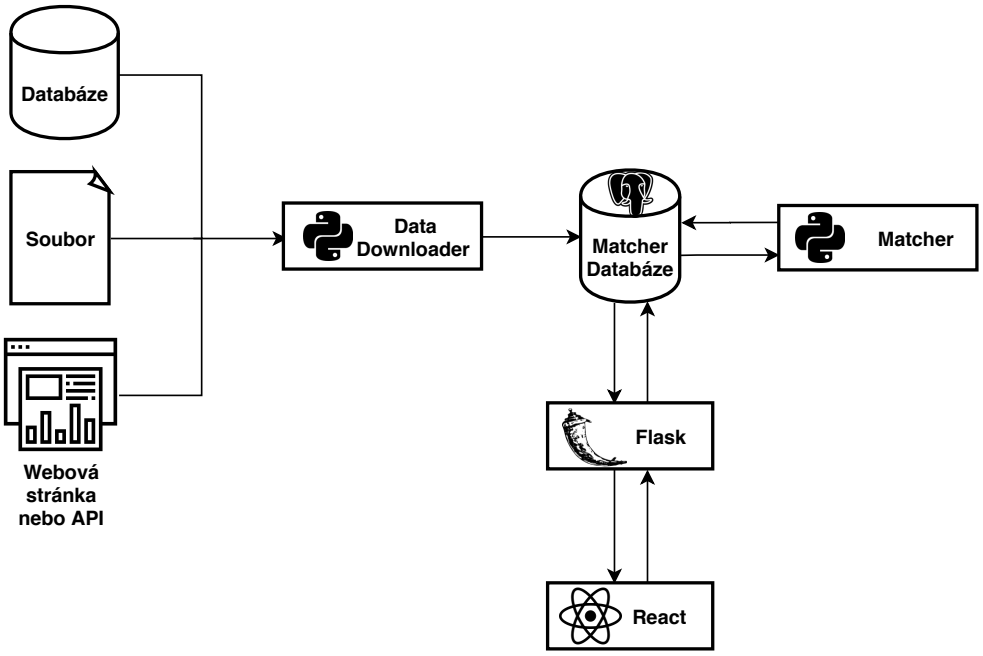
- DataDownloader - stažení informací o fakturách a uzavřených smlouvách
- Matcher - vytvoření spojení mezi fakturami a smlouvami
- Flask - vystavení REST API
- Webový portál - prezentace výsledků

Jak je vidět na obrázku 3.1, o získání dat z dostupných zdrojů se stará DataDownloader. Data jsou následně uložena v databázi, odkud jsou následně načteny aplikací Matcher, která vytvoří spojení mezi daty a uloží je zpět do databáze. Tato data jsou dostupná pomocí REST API, kterou poskytuje Flask aplikace. A poslední částí je Webový klient, který prezentuje data a výsledky párování.

V následujících kapitolách jsou popsány jednotlivé části a jejich funkcionality.

3.3 DataDownloader

DataDownloader slouží k získání všech potřebných dat, se kterými následně budu pracovat. Primárně tedy data o smlouvách z Registru smluv a informace o fakturách z databáze Opendata.



Obrázek 3.1: Tok dat při zpracování

3.3.1 Provider

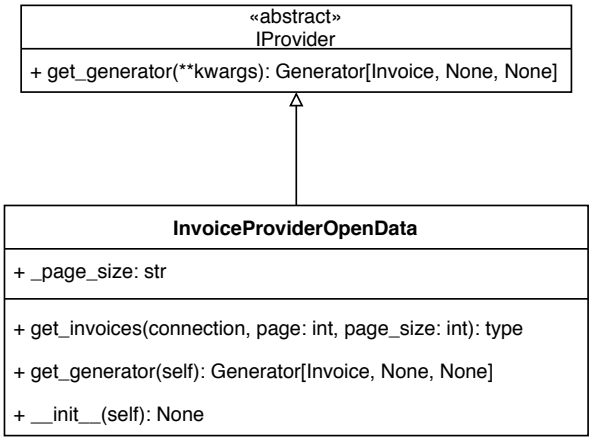
Na začátku procesu je nutné data získat a uložit na místo, kde s nimi budu pracovat. K tomuto účelu slouží providery. Providery mají za úkol získání dat ze zdrojů, které si zvolím. Zdrojem dat může být nějaké API rozhraní, existující databáze nebo webová stránka, ze které je možné data stáhnout pomocí webscrapingu. Výstupem takového provideru je generátor, který data může postupně poskytovat. Generátor, neboli generátorová funkce, je typ iterátoru, která nemusí obsahovat všechna data, přes které iteruje. Jedná se totiž o funkci, která postupně data generuje nebo získává z databáze. Zároveň paměť není zatěžována tím, že jsou stažena všechna data do paměti, a až poté jsou zpracovány.

Definuji dva druhy providerů, podle typu dat:

- InvoiceProvider
- ContractProvider

Jak již jméno napovídá, `IProvider` slouží k získání faktur. Jako zdroj faktur v této práci využívám databázi naplněnou programem opendata, který je navrhnut Opendata laboratoří.

Pro tento účel je vytvořena třída `InvoiceProviderOpenData`. Třída dědí z třídy `IProvider`, jak je vidět na obrázku 3.2, a proto musí implementovat



Obrázek 3.2: Diagram tříd - IProvider

funkci `get_generator`. Tato funkce se pomocí připojení, které je specifikováno v parametru, připojí k databázi a postupně vrací jednotlivé faktury. Faktury nejsou před zpracováním nijak upravené. Poté, co z databáze nejsou vráceny žádné výsledky, generátor se ukončí.

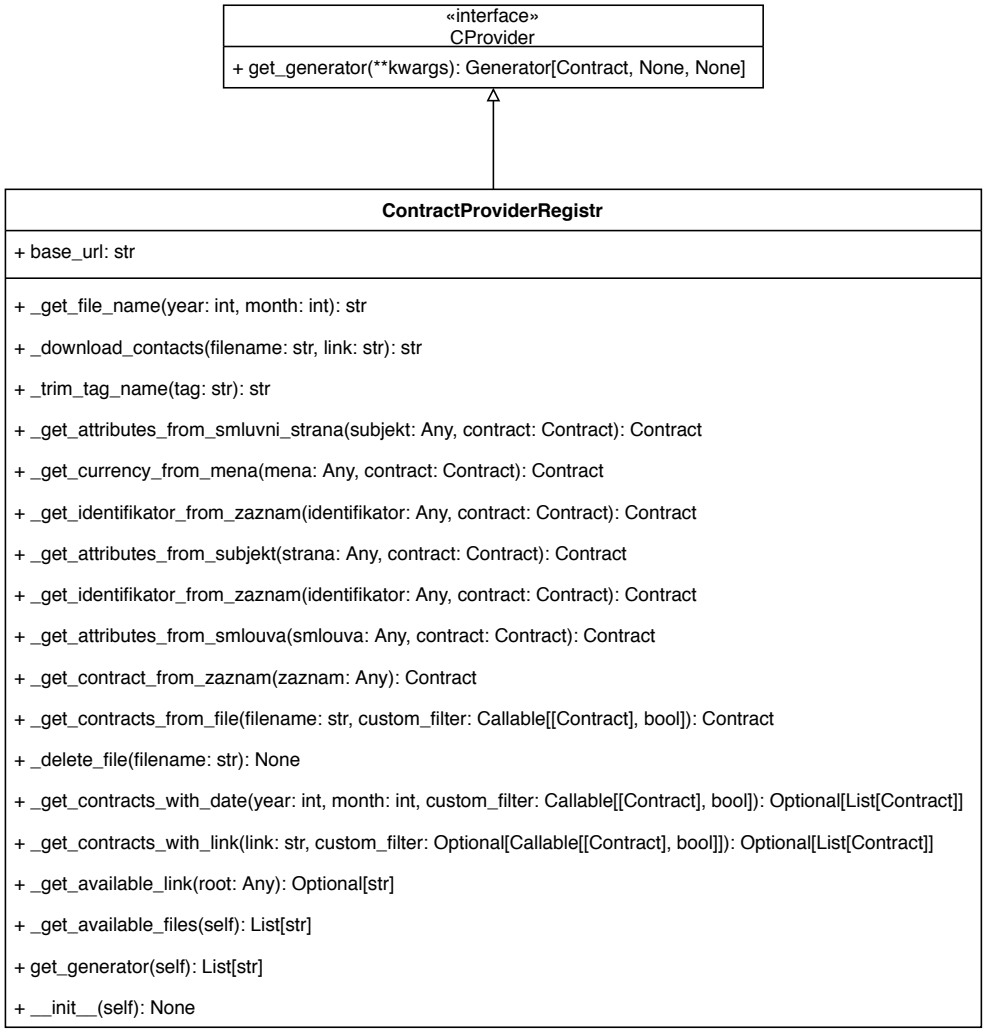
Použití generátoru zde ulehčuje mnoho práce, a to tím, že program nebo skript, který generátor využívá, se nemusí starat o to, zda jsou faktury načítány po stránkách nebo po souborech. To řeší provider sám.

`CProvider` slouží k získání smluv ze zvoleného zdroje. V této práci je využit Registr smluv jako primární zdroj informací o smlouvách, a proto využívám třídu `ContractProviderRegistr`, viz obrázek 3.3. Registr smluv poskytuje data ve formě xml souborů, které jsou zveřejňována po měsících. Seznam všech dostupných dump souborů je možné získat na adrese `https://data.smlouvy.gov.cz`. Ve výpisu se nacházejí odkazy na jednotlivé soubory, které mají formát názvu dump_yyyy_mm.xml, kde „yyyy“ představuje rok a „mm“ číslo měsíce, které je doplněno úvodní nulou, pokud je potřeba.

Generátor třídy `ContractProviderRegistr` jednotlivé odkazy extrahuje a začne je postupně stahovat. Po dokončení stahování dat začne extrahovat jednotlivé smlouvy. Smlouvy během parsování ukládá do pole, ze kterého jsou postupně vraceny generátorem. Po zpracování všech dat je soubor smazán a začne stahování dalšího souboru.

Na části kódu 3.2 je vidět, že je na začátku vytvořena instance třídy `InvoiceProviderOpenData`, která slouží ke stažení faktur. Následně je pomocí funkce `get_generator` získán generátor, přes který je iterováno a jednotlivé faktury jsou pomocí `matcher_conn` uloženy do databáze. V proměnná `matcher_conn` je uložena instance třídy `SQLAlchemyController`, která slouží pro komunikaci s databází.

Stejným způsobem jsou získány informace o uzavřených smlouvách z Registru smluv pomocí třídy `ContractProviderRegistr`.



Obrázek 3.3: Diagram tříd - CProvider

3.3.2 Database Controller

`DBController` je abstraktní třída a má za cíl provádění operací, které náš program potřebuje provést nad databází. Může se jednat o jednoduché CRUD operace, ale také o složitější operace, jako je například vytvoření statistik z dat, které databáze obsahuje. Tato třída definuje jednotlivé operace a následně Controller, který bude dědit z této třídy, musí jednotlivé operace implementovat, viz obrázek 3.4.

DataDownloader používá Controllery pro uložení dat do databáze, ze které následně budou použity pro párování. Stejný Controller je následně použit v aplikaci Matcher.

```
log.info("Starting to download invoices from opendata database")
iprovider = InvoiceProviderOpenData()
for i in iprovider.get_generator():
    log.debug(f"Inserting: {i}")
    matcher_conn.insert_invoice(i)
matcher_conn.commit()
log.info("Invoices downloaded.")

log.info("Starting to download contracts from registr")
cprovider = ContractProviderRegistr()
for c in cprovider.get_generator():
    log.debug(f"Inserting: {c}")
    matcher_conn.insert_contract(c)
matcher_conn.commit()
log.info("Contracts downloaded.")
```

Výpis kódu 3.2: Získání a uložení dat pomocí Data Downloaderu

3.3.3 SQLAlchemyController

`SQLAlchemyController` je třída, která dědí z třídy `DBController` a jejím cílem je umožnit provedení definovaných operací primárně nad databází PostgreSQL. Tento Controller využívá ORM knihovnu SQLAlchemy pro přístup k databázi. Díky využití ORM, je možné změnit zdrojovou databázi a implementace by zůstala v ideálním případě beze změny. Používané modely již obsahují informaci o tabulce v databázi a datový typ atributu.

`SQLAlchemyController` při připojení k databázi musí vytvořit `engine`, který obsahuje informace o přihlašovacích údajích, adrese databáze, portu a také druhu databáze. Poté, co je vytvořený `engine`, je vytvořena `session`, která je nadále používána k dotazování a provádění operací. Například v části kódu 3.3 je vidět, jak pomocí funkce `query()` získám smlouvy, které omezím kritériem. Poslední funkce `first()` zajistí, že pokud je dostupný výsledek, bude vrácen pouze první záznam.

```
def get_contract(self, contract_id: int) -> Optional[Contract]:
    assert self.session is not None
    contract = self.session.query(Contract)
               .filter(Contract.contract_id == contract_id)
               .first()
    return contract
```

Výpis kódu 3.3: Funkce pro získání smlouvy podle id

3.4 Matcher

Matcher je aplikace, která slouží k vytvoření spojení mezi smlouvami a fakturami, které jsou načteny do databáze pomocí programu DataDownloader. V této části popíši třídy, které jsou použity při mapování a způsob, kterým je mapování provedeno.

3.4.1 Pipeline

`Pipeline` je třída, která slouží ke zpracování možných spojení. Jak je vidět na diagramu 3.5 třída definuje dvě funkce, které musí každá dědící třída implementovat. První funkcí je `prepare`, která má za cíl připravit potřebná data, konfiguraci a připojení, které budou při zpracování potřeba. Může se jednat například o připojení k databázi, ze které budou získávána data. Nebo také stáhnout soubor s daty. Neboli každá funkce, kterou je možné provést jen jednou a mohla by být časově náročná.

Druhou funkcí je `process_invoice_relations`, která jako parametr požaduje List obsahující `PossibleRelation`, které mají být zpracovány. Výsledkem této funkce je `Tuple`, kde prvním prvkem je `List` obsahující spojení, které zpracováním prošly úspěšně, a druhým prvkem je List spojení, které byly během zpracování vyřazeny.

Jak bude rozdělení spojení definováno je čistě na konkrétní implementaci. Může se jednat o sadu testů a pokud jeden nebo více testů dané spojení nesplní, může být vyřazeno.

- **FilterPipeline**

  `FilterPipeline` je první `Pipeline`, která má za cíl vyřadit spojení, která nesplňují základní podmínky, a tím zmenšit množství dat, které bude při dalším vyhodnocení nutné zpracovat. Tato část je velice důležitá, protože na začátku vznikají možná spojení tím, že je vytvořen kartézský součin smluv a faktur, které jsou mezi stejným ministerstvem a dodavatelem. Testy, které jsou zde prováděny, by měly být rychlé a nenáročné, protože budou vykonávány pro každé spojení.

  Při zpracování pomocí této `Pipeline` je každé spojení otestováno testy, které se týkají časových údajů. Časové údaje smlouvy a faktury jsou údaje, které jsou často k dispozici a jejich využití dokáže množství snížit.

- **DecidingPipeline**

  Cílem této `Pipeline` je rozhodnutí, zda faktura ke smlouvě patří, nebo ne. Toto je provedeno pomocí otestování spojení pomocí testů a na základě výsledků přiřadit `score`, které hodnotí spojení.

  Každému testu je přiřazena váha, která bude reprezentovat v porovnání s ostatními, jak hodně tento poznatek napovídá, že faktura patří právě

k této smlouvě. Například, pokud faktura obsahuje číslo smlouvy (identifikátor, který používá ministerstvo ve svých systémech), mělo by toto spojení upřednostněno před spojením, kde jediné co je známe, je, že vyfakturovaná částka je menší, než smluvená cena.

`Pipeline` může zpracovávat data jednotlivě nezávisle na sobě, ale je možné využít i toho, že vstupním parametrem je `List` spojení. Pokud spojení budou tvořit skupinu, která tvoří nějaký celek, může nad nimi být proveden test, který této vlastnosti využije. Například, pokud ke zpracování budou předána všechna spojení, která se týkají jedné faktury, je možné po provedení testů, porovnat `score` a zvolit to s nejlepším hodnocením.

Při rozhodování, které spojení bude považováno za dostatečně důvěryhodné, je důležitá bodová hranice. Bodová hranice slouží k tomu, aby nebyla považována za nejpravděpodobnější spojení ta, u kterých výsledky testů nic nezjistily. Bez této hranice by mohli být považovány za nejpravděpodobnější spojení ty, které mají 0 bodů. Tato hranice je načtena z konfiguračního souboru a pokud spojení tuto hranici nepřekročí, záznam o testování bude uložen, ale nemůže být označen jako finální. Tím se může stát až spojení, které má stejný nebo větší počet bodů, než je bodová hranice a zároveň má nejvíce bodů ze všech spojení, které souvisejí s jednou fakturou.

3.4.2 Testy spojení

Když je spojení vytvořeno, je nutné ho otestovat a ohodnotit. A pro tento účel slouží testy spojení. Testy mohou být jednoduché, které pouze porovnají data, a nebo složitější, které využijí komplikovanější algoritmy a vztahy. Testy samy o sobě nerozhodují o tom, zda bude spojení mezi fakturou a smlouvou vyřazeno. Testy pouze hodnotí a interpretace výsledku je na Pipeline, která je volá.

Rozlišuji dva druhy testů:

- Contract Invoice Test
- Multiple Constracts Test

Contract-Invoice testy porovnávají jednotlivé atributy smlouvy a faktury a poté z nich odvodí výsledek, viz obrázek 3.6. Na vstupu těchto testů je, jak název napovídá, `Contract` a `Invoice`. Výstupem je `TestResult`, který obsahuje výsledek. Výsledkem těchto testů může být číselná hodnota typu `float`, která bude později interpretována a využíta k porovnání s ostatními spojeními.

Multiple Constracs Testy slouží k testování souboru smluv, které mohou patřit k jedné faktuře. Tyto testy hodnotí soubor jako celek. Tento způsob testování především pomůže při využití počtu smluv.

- **Test časových údajů**:

  Test časových údajů, které smlouva a faktura obsahují, je využit k vyřazení spojení. Smlouva obsahuje datum uzavření, datum ukončení smlouvy (datum zveřejnění při tomto testu není možné nijak využít). Faktura může obsahovat datum vystavení, datum přijetí, datum zaplacení a datum splatnosti.

  Protože je potřeba zjistit, zda aspoň jeden časový atribut faktury je menší, než datum vytvoření smlouvy, tak stačí vybrat nejmenší datum z časových údajů faktury a porovnat ho s datem vytvoření smlouvy. Pokud je větší nebo rovno datu uzavření smlouvy, je spojení možné a výsledek testu nastaven na hodnotu 1. V opačném případě je výsledek nastaven na hodnotu 0. Pokud nejsou dostupné potřebné údaje (alespoň jeden z údajů faktury a datum uzavření smlouvy) je výsledkem testu hodnota `None`.

- **Počet dní vystavení faktury po uzavření smlouvy**:

  Tento test porovná nejmenší z časových údajů faktury s datem uzavření smlouvy a výsledkem je počet dní mezi nimi. Výsledek tohoto testu může být využit, pokud je více smluv k jedné faktuře se stejným hodnocením. V takovém případě je možné vybrat smlouvu, u které je časový rozdíl nejmenší.

- **Vyfakturovaná částka menší nebo rovna hodnotě smlouvy**:

  Tento test cílí na smlouvy, na které je vytvořeno více faktur v průběhu plnění. Není běžné, že vyfakturovaná částka je větší než smlouva, na kterou je vytvořena. A proto pokud je hodnota faktury menší, než hodnota smlouvy, měla by být zvýhodněna při finálním vyhodnocení.

  Proto je na začátku nalezena maximální hodnota smlouvy a faktury (částka s DPH nebo částka bez DPH, podle toho, která hodnota je dostupná) a tyto dvě hodnoty jsou porovnány. Pokud je hodnota faktury menší nebo rovna, než je hodnota smlouvy, je jako výsledek testu uložena číselná hodnota 1. V opačném případě je výsledná hodnota testu rovna číslu 0.

- **Shodná částka**:

  Tento test porovnává částky, které jsou uvedeny ve smlouvě a faktuře. Cílí na smlouvy, na které je vytvořena pouze jedna faktura, která čerpá celou částku. Oba objekty mohou obsahovat částku s DPH, částku bez DPH a částku v jiné měně.

  Výsledkem testu bude hodnota mezi 0 a 1 včetně krajních hodnot, kde 1 je nejlepší možný výsledek. Proto jsou vytvořeny dvojice, podle toho o jaký typ částky se jedná a spočítá se podíl částek, který ve výsledku

řekne, jak hodně jsou částky stejné. Aby nebyly vráceny hodnoty větší než 1, je vždy určena větší a menší hodnota z dvojice a výsledek je roven menší částce vydělené větší.

Pokud jedna z hodnot ve dvojici chybí, výsledek se nezapočítává. Po vyhodnocené všech tří dvojic, je spočítán průměr a toto číslo je uloženo jako výsledek testu.

- **Číslo smlouvy v předmětu faktury**:

  V tomto testu chci zjistit, zda je číslo smlouvy obsaženo v předmětu faktury. Číslo smlouvy může obsahovat číslice a nečíselné znaky (znaky abecedy, dvojtečky, lomítka a další). Testy jsou podmíněny minimální délkou čísla smlouvy, která je nastavena na 4 znaky. V některých případech se stává, že číslo smlouvy obsahuje nečíselné znaky, ale faktura obsahuje pouze číslice. Proto jsou vytvořeny dva testy.

  První test zjišťuje, zda je číslo smlouvy v předmětu faktury bez větších úprav (oba údaje jsou před porovnáním převedeny na malá písmena, aby porovnání nebylo citlivé na velikost písmen). Pokud předmět faktury nebo číslo smlouvy není k dispozici, je výsledek testu nastaven na hodnotu 0. Pokud údaje jsou k dispozici, je použita funkce pro zjištění, zda je číslo smlouvy substring předmětu faktury. Pokud číslo smlouvy je obsaženo v předmětu faktury, je výsledek testu nastaven na hodnotu 1. V opačném případě je nastaven na hodnotu 0.

  Druhý test před porovnáním odstraní z čísla smlouvy a předmětu faktury všechny nečíselné znaky. Znaky jsou odstraněny z předmětu faktury také, protože by v něm mohly být použity oddělovače a další znaky, které by způsobily, že test selže. Při odstraňování znaků je nutná opatrnost, protože může nastat situace, že údaje se skládají pouze z nečíselných znaků a po odstranění by vznikly prázdné řetězce. Proto je nutné zkontrolovat, že po odstranění znaků jsou oba řetězce neprázdné a až poté zkontrolovat, zda je číslo smlouvy substring předmětu faktury po odstranění nečíselných znaků.

- **Počet smluv:**

  Počet smluv je dobrý ukazatel v především případech, že je dostupná pouze jedna smlouva, na kterou se faktura může navázat. Tento test je velice jednoduchý, protože je vrácen počet vstupních smluv.

3.4.3 Vyhodnocení podezřelých zakázek

Vyhodnocení podezřelých zakázek probíhá na základě vytvořených spojení. Podezřelou smlouvou je každá, na kterou je vyfakturovaná větší částka, než je uvedeno ve smlouvě. Pokud by byly započítány všechny takové smlouvy, seznam by obsahoval také smlouvy, u kterých součet vyfakturovaných částek

převyšuje částku smlouvy o korunu a více. Proto je při vytváření možnost využít spodní a horní limit, který umožní vyfiltrování smluv, u kterých je přírůstek příliš nízký.

Pro vytvoření záznamů o podezřelých zakázkách je využit příkaz SQL, který vybere smlouvy, které jsou spojeny s fakturami a podle součtu hodnot faktur spočítá přírůstek oproti hodnotě smlouvy. Pokud je hodnota faktur vyšší než hodnota smlouvy a procentuální přírůstek je vyšší než spodní limit, je informace o smlouvě, hodnotě smlouvy, hodnotě faktur a procentuálnímu přírůstku uložena do databáze.

3.5 Flask

Flask aplikace v mém případě slouží pouze jako REST API pro komunikaci mezi webovým klientem a databází. Pro komunikaci s databází je využívána knihovna SQLAlchemy.

3.5.1 SQLAlchemy

Při spuštění je nutné vytvořit spojení s databází, ze které budou získávána data. Proto je zajištěno, že atribut `SQLALCHEMY_DATABASE_URI` obsahuje připojovací řetězec ke zdrojové databázi. Pro vytvoření modelů, se kterými následně pracuji, využiji funkci `reflekt`, která je vytvoří podle existujících tabulek v databázi.

3.5.2 REST API

REST API se skládá z šesti částí.

- statistics
- invoices
- contracts
- relations
- warnings
- ministry

Spolu se zmíněnými částmi je na kořenové adrese dostupná dokumentace, která je vytvořena pomocí knihovny flask-restx. Flask-restx je rozšíření frameworku Flask a za pomocí dekorátorů v kódu vygeneruje dokumentaci, kde je možné vidět dostupné adresy, ale také API přímo otestovat, viz 3.8. Ukázka dokumentace je vidět na obrázku 3.7.

Pomocí API je také možné získat seznam smluv, faktur a podezřelých zakázek. Funkce využívají jeden povinný parametr určující číslo stránky, která bude vrácena. Velikost stránky je určena konstantou, která je nastavena při spuštění. Aby bylo možné v datech vyhledávat, je zde možnost využít nepovinné argumenty. Dostupné filtry jsou na IČO ministerstva, jméno dodavatele, IČO dodavatele. Dále je možné filtrovat také pomocí časových údajů a to u smlouvy podle data uzavření a u faktur podle data vystavení. Krom informací o smlouvě nebo faktuře samotné, je navíc připojena informace o potenciálních spojeních, která jsou se záznamem svázána.

Dále jsou také dostupné funkce pro získání informací o smlouvě nebo faktuře na základě identifikátoru v databázi. Na základě identifikátoru smlouvy nebo faktury je také možné vyhledat informace o potenciálních spojeních.

Také jsou dostupné funkce, které slouží pro získání statistik o datech, která databáze obsahuje. Tyto informace nejsou počítány při každém dotazu, ale po každém dokončení párování a jsou uloženy v tabulce `statistics`. Tento způsob zvyšuje rychlost odeslání odpovědi, protože není nutné nic počítat a pouze vybrat data z tabulky na základě textového identifikátoru.

Je také možné odeslat dotaz pro získání seznamu ministerstev a odpovídající IČO.

3.6 Webový portál

Jedná se o webový portál vytvořený pomocí frameworku React, kde je možné vidět základní statistiky o datech, které jsou dostupné a také výsledky párování. V této části popíši jednotlivé části, ze kterých se klient skládá.

3.6.1 Šablona

Šablona představuje okolí zobrazovaných dat. Skládá se z menu, ze kterého je možné dostat se na jednotlivé položky. Součástí šablony je také zápatí stránky, kde jsou uvedeny odkazy na klíčové stránky.

Všechny další stránky tvoří komponenty, které se zobrazí uvnitř této šablony.

3.6.2 Dashboard

Stránka Dashboard, zobrazená na obrázku 3.9, ukazuje celkový přehled počtu smluv, faktur, navázaných faktur a podezřelých zakázek. Po kliknutí na ikonu, u každého údaje, je uživatel přesměrován na stránku s přehledem. Dále obsahuje graf, který zobrazuje vývoj počtu zveřejněným smluv a faktur v čase a druhý graf, který zobrazuje porovnání počtu zveřejněných faktur a smluv podle ministerstva.

3.6.3 Ministerstva

Tato stránka, viz obrázek 3.10, zobrazuje data podle jednotlivých ministerstev. Ke každému ministerstvu jsou uvedeny počty zveřejněných smluv, faktur, počet vytvořených spojení mezi fakturou a smlouvou a počet podezřelých zakázek. Pomocí ikony u každého údaje je možné se přesměrovat na stránku s odpovídajícími daty. Takže pokud uživatel klikne například na ikonu s fakturou u ministerstva financí České republiky, bude přesměrován na stránku se seznamem faktur, kde bude automaticky filtr nastaven tak, aby byly zobrazeny pouze faktury zveřejněné daným ministerstvem.

Komponenta pro každé ministerstvo je generována na základě dat, které jsou načteny z databáze. Pokud by bylo nutné v budoucnu rozšířit o další instituci, stačí, aby byla instituce uložena v databázi a existovaly k ní data.

3.6.4 Faktury

Na této stránce, viz obrázek 3.11 jsou jednotlivé faktury zobrazeny v tabulce, která obsahuje ve sloupcích údaj o ministerstvu, dodavateli, datum vystavení a částku. Na konci každého řádku je tlačítko, které uživatele přesměruje na stránku s popisem faktury. Každé tlačítko navíc obsahuje ikonu připojení, která zobrazuje, zda je faktura propojena se smlouvou nebo ne.

Stránka také obsahuje vysouvací filtry, kde je možné vybrat ministerstvo, zadat začátek jména nebo IČO dodavatele, časově omezit datum vystavení faktury nebo vyhledat pouze navázané položky. Po kliknutí na tlačítko vyhledat jsou data v pozadí aktualizována a obsah tabulky je změněn podle výsledků vyhledávání.

Spolu s filtry je zde také možnost přepínat mezi stránkami. Při načítání dat se vždy zobrazí ukazatel průběhu, aby uživatel věděl, že se data načítají z databáze.

V URL stránky je možné uvést argumenty podle kterých budou nastaveny filtry. Je možné použít následující argumenty:

- **page**: Stránka, která se má zobrazit.
- **m_ico**: IČO ministerstva.
- **s_ico**: IČO dodavatele.
- **s_name**: Jméno dodavatele.
- **from**: Počáteční datum, pro omezení data vystavení faktury nebo data uzavření smlouvy.
- **to**: Koncové datum, pro omezení data vystavení faktury nebo data uzavření smlouvy.
- **linked**: Hodnota pro zobrazení pouze navázaných záznamů.

URL pro vyhledání faktur ministerstva obrany České republiky, u kterých je dodavatel firma, která začíná na "Marius Pedersen"a jsou navázány k některé ze smluv, vypadá následovně:

`/invoices?m_ico=60162694&s_name=Marius%20Pedersen&linked=true`
Zmíněná URL se skládá z následujících částí:

- **invoices**: je označení pro stránku s fakturami. Tato stránka může být nahrazena stránkou pro smlouvy (contracts) nebo podezřelé zakázky (warnings) a chování bude stejné. Po označení stránky následuje znak „?“, který značí, že začínají argumenty.
- **m_ico**: Zde je specifikováno, že má být nastaven filtr, aby se zobrazily pouze faktury spojené s ministerstvem dopravy České republiky, které má IČO 60162694.
- **s_name**: Zde je uveden začátek jména dodavatele, pro kterého se mají zobrazit faktury. Mezery jsou nahrazeny textem „%20“.
- **linked**: Zde je uvedena hodnota „true“, která značí, že se mají zobrazit pouze navázané faktury.

Mezi každým argumentem je znak „&“, který značí, kde končí hodnota jednoho argumentu a kde začíná argument druhý.

Tento způsob umožní přesměrování s již nastavenými parametry.

3.6.5 Smlouvy

Stránka se smlouvami je velmi podobná stránce s výčtem faktur. Také obsahuje tabulku se záznamy, filtry, možnost nastavení filtrů pomocí argumentů v URL, možnost přepínání stránek. Malý rozdíl je ten, že u tlačítka pro přesměrování na stránku s popisem smlouvy, není pouze ikona naznačující, zda je smlouva propojena s některou z faktur. V rohu tlačítka se zobrazí ikona s číslem, které reprezentuje počet navázaných faktur. Pokud je počet navázaných faktur větší než 99, zobrazí se pouze 99+. Viz obrázek 3.12.

3.6.6 Detail faktury a smlouvy

Stránka s detailem faktury zobrazuje jednotlivé údaje faktury spolu se smlouvou, se kterou byla spojena (pokud byla spojena), a také další smlouvy, které byly brány v úvahu při párování. U každé smlouvy je tlačítko, které zobrazí pop up stránku s informacemi o smlouvě. Tento způsob umožní uživateli prohlížet navázané položky bez nutnosti opuštění stránky.

Stejným způsobem je vytvořena stránka s detailem smlouvy, viz obrázek 3.14, s rozdílem, že u smlouvy jsou navíc zobrazeny informace o počtu navázaných faktur a součet vyfakturovaných částek, viz obrázek 3.13. Pokud je součet

vyšší než hodnota smlouvy, je barva ikony změněna ze zelené na žlutou, aby bylo lépe vidět, že se jedná o podezřelou zakázku. Na stránce s detaily faktury nebo smlouvy je dobře vidět, jak je jednoduché použití komponent. Každá kolonka s jednou informací se liší pouze v nadpisu, id, hodnotě a velikosti. A protože by bylo nutné ji na stránce opakovat pro každý atribut znova a kód by se tak mohl stát nepřehledným, vytvořil jsem komponentu, které stačí předat potřebné informace.

Jak je vidět v části kódu 3.4, jedná se o funkci, která na základě informací, uložených v parametru `props`, vrátí komponentu `GridItem`, která obsahuje nadpis, hodnotu a má definovanou velikost.

```
export default function InfoColumn(props) {
    return (
        <GridItem xs={props.xs} sm={props.sm} md={props.md}>
          <CustomInput
            labelText={props.label}
            id={props.id}
            formControlProps={{
              fullWidth: true
            }}
            inputProps={{
              disabled: true
            }}
          />
          <InputLabel>{props.value}</InputLabel>
        </GridItem>
    )
}
```

Výpis kódu 3.4: Komponenta InfoColumn

Následně jsou informace o faktuře převedeny do pole. Na toto pole je následně zavolána funkce `map` a pro každý prvek v poli je vytvořena nová komponenta, které jsou předány potřebné informace a velikost. Část kódu s generováním jednotlivých komponent z pole je vidět v kódu 3.5.

3.6.7 Podezřelé zakázky

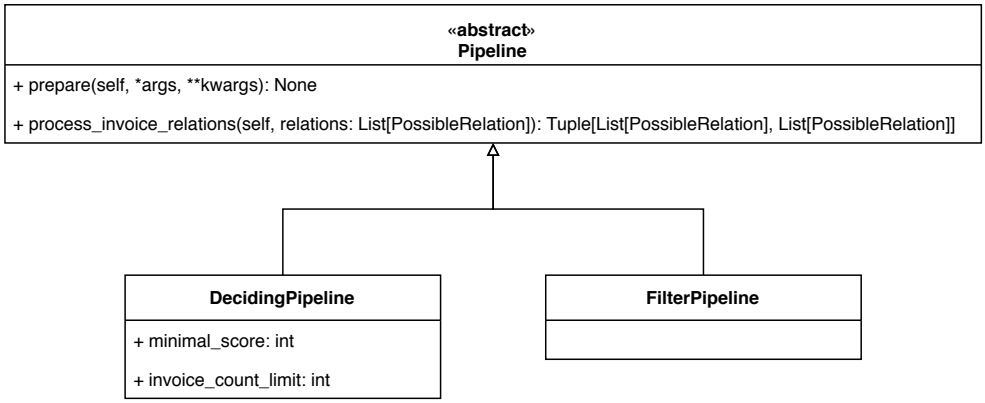
Podezřelé zakázky jsou zobrazeny v tabulce, kde jsou uvedeny základní údaje o smlouvě a procentuální přírůstek oproti ceně smlouvy. Každou smlouvu je možné si otevřít a podívat se na detailní popis a faktury, které jsou k smlouvě navázány.

Tato stránka implementuje možnosti filtrování, jak za pomoci komponenty přímo ve stránce, ale také přes argumenty v URL.

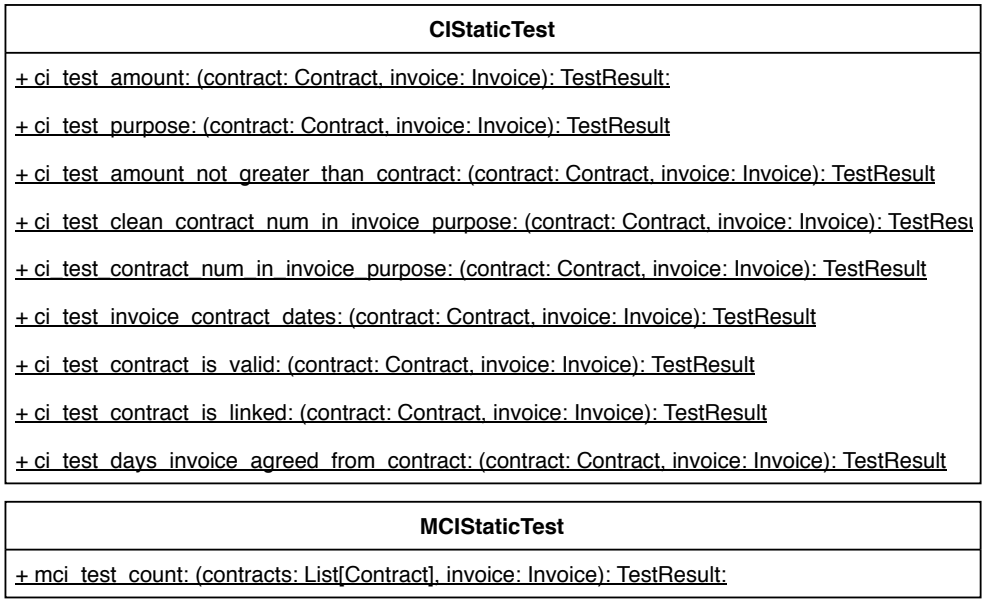
```
<GridContainer>
    {getAttributeArray(invoice).map(attribute => (
        <InfoColumn
            label={attribute.label}
            id={attribute.id}
            value={attribute.value}
            xs={attribute.xs ?? 12}
            sm={attribute.sm ?? 12}
            md={attribute.md ?? 6} />
    ))}
</GridContainer>
```

Výpis kódu 3.5: Použití komponenty InfoColumn

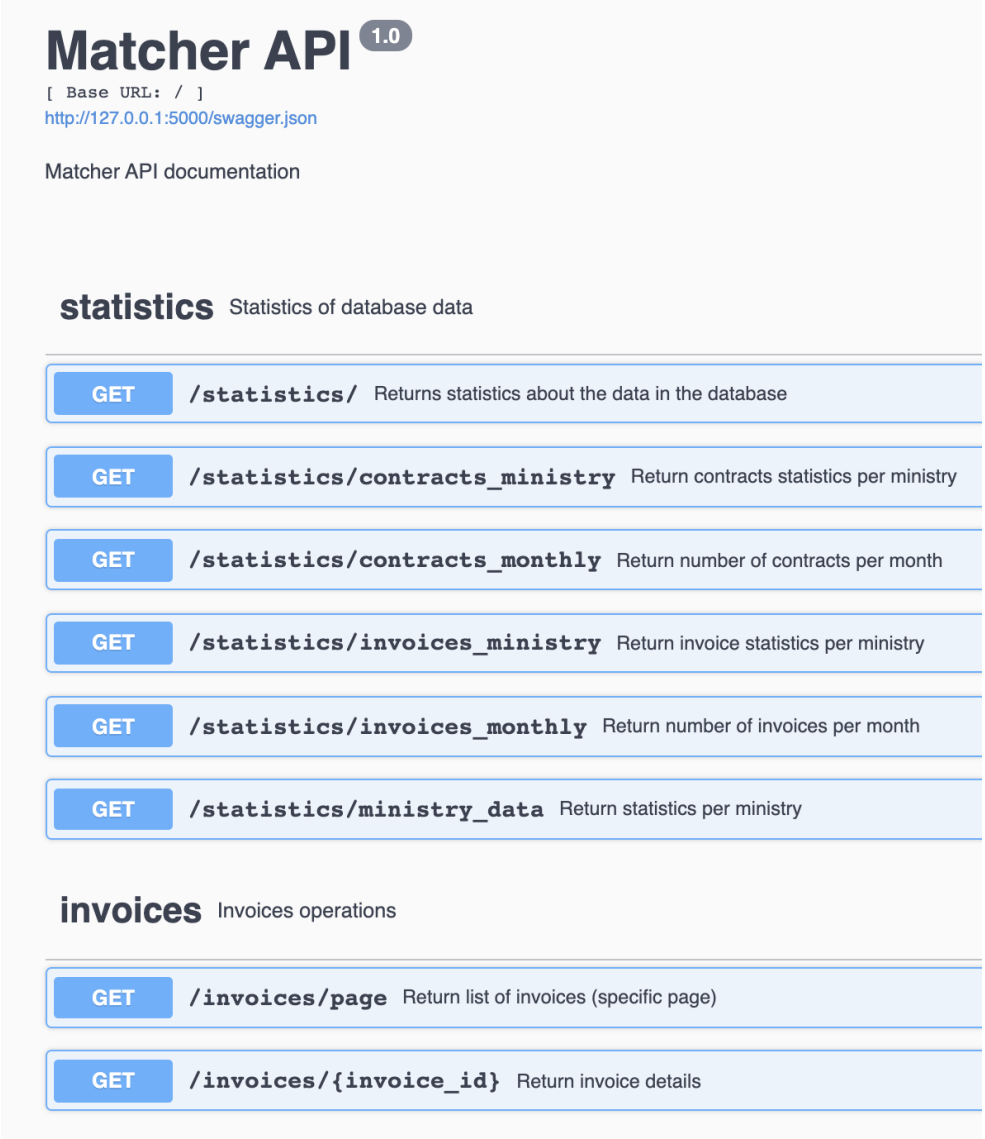
Obrázek 3.4: Diagram tříd - DBController



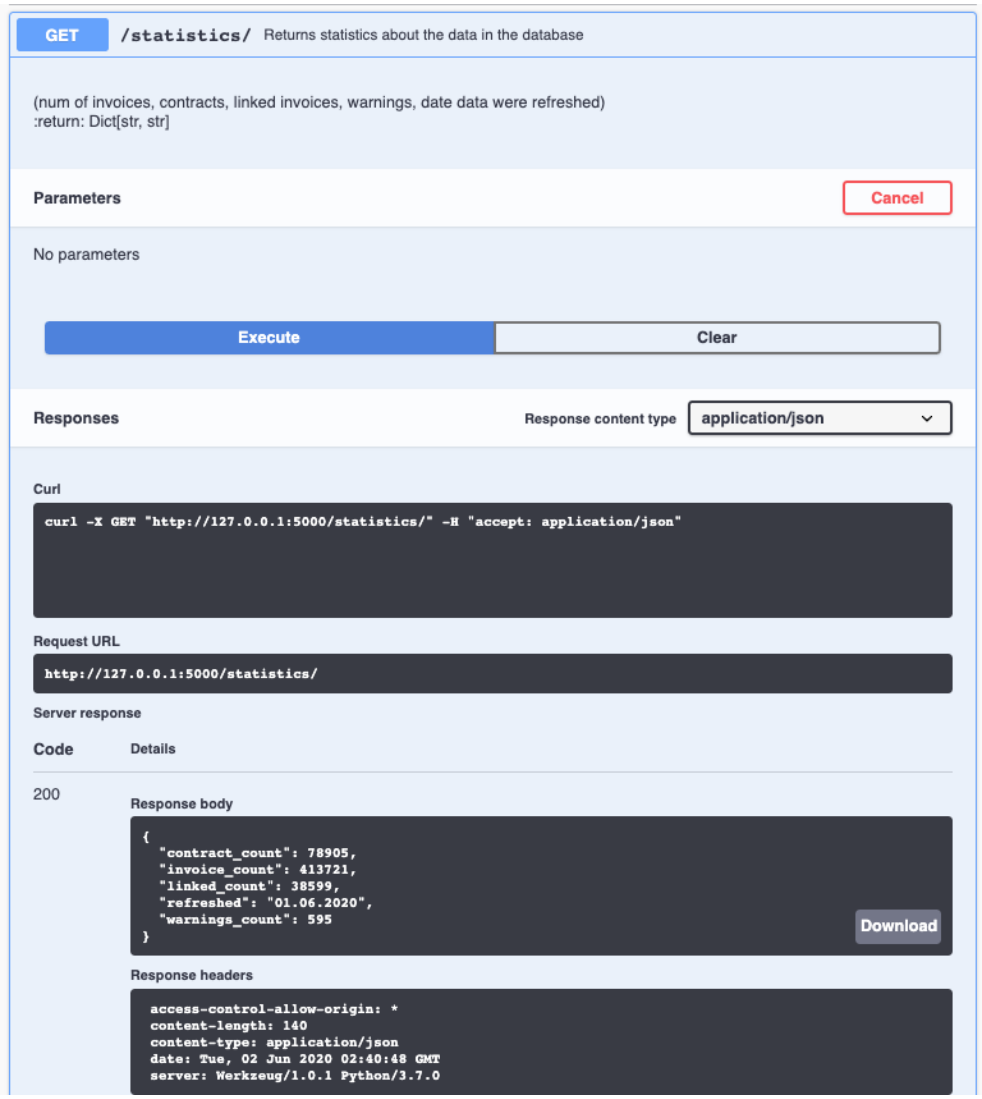
Obrázek 3.5: Diagram tříd - Pipelines



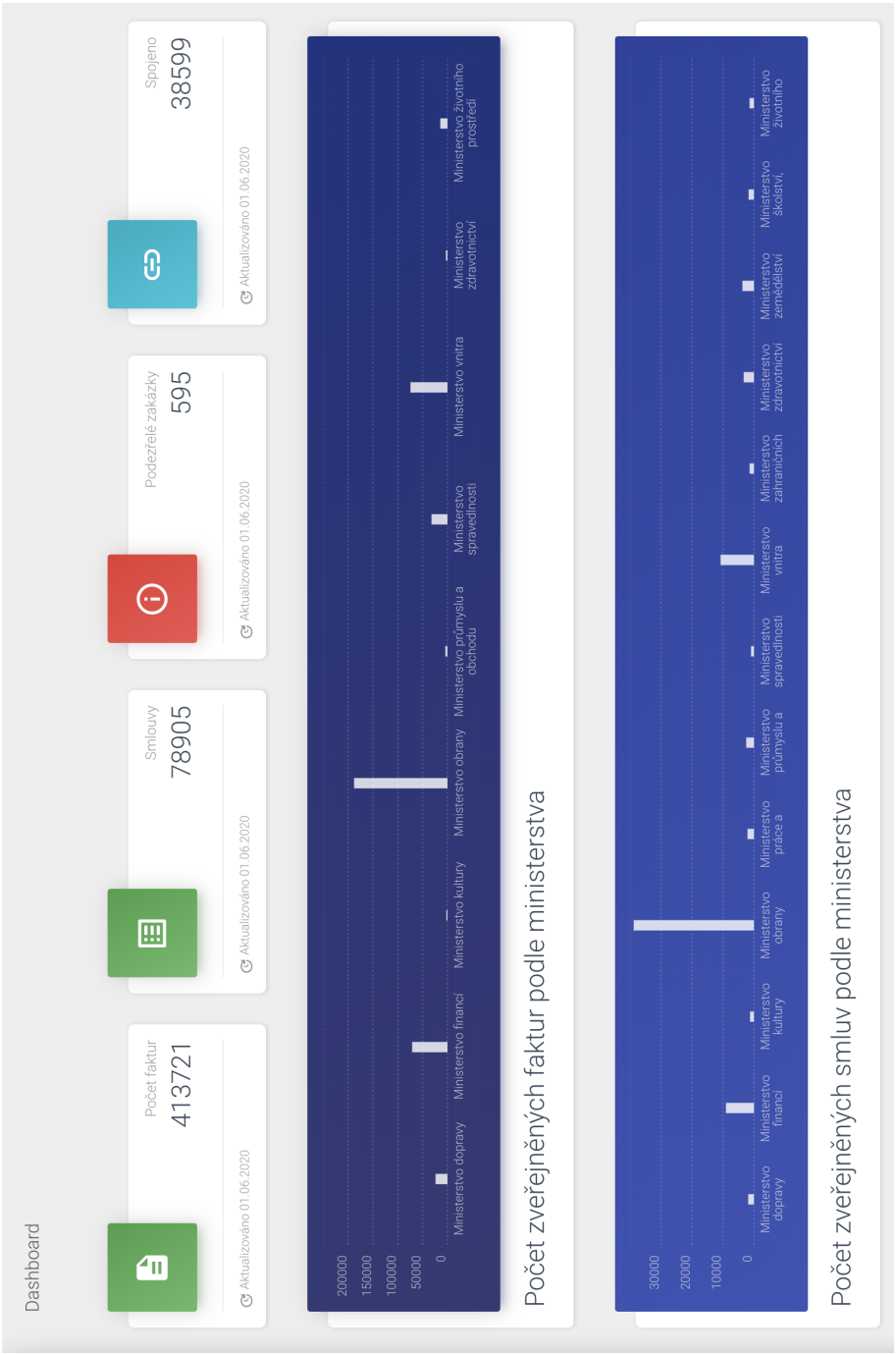
Obrázek 3.6: Diagram tříd - Testy



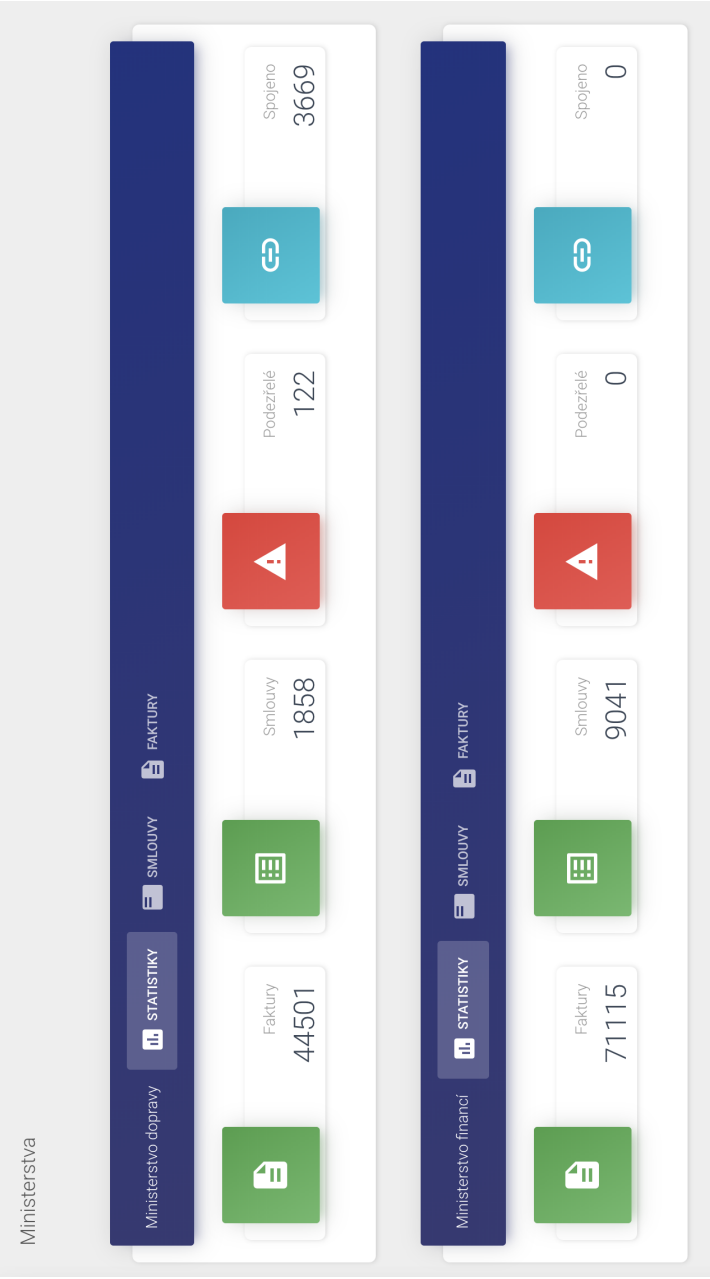
Obrázek 3.7: Ukázka API dokumentace - seznam adres



Obrázek 3.8: Ukázka API dokumentace - testování odpovědi



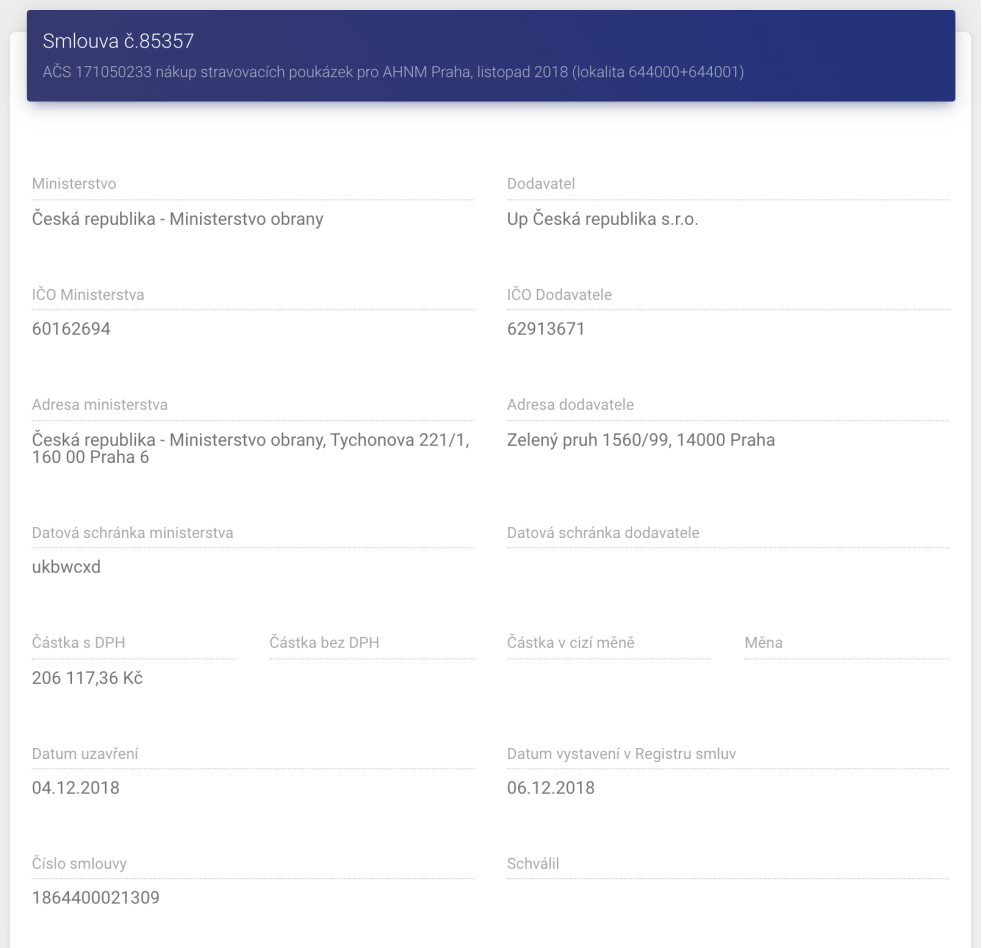
Obrázek 3.9: Stránka Dashboard



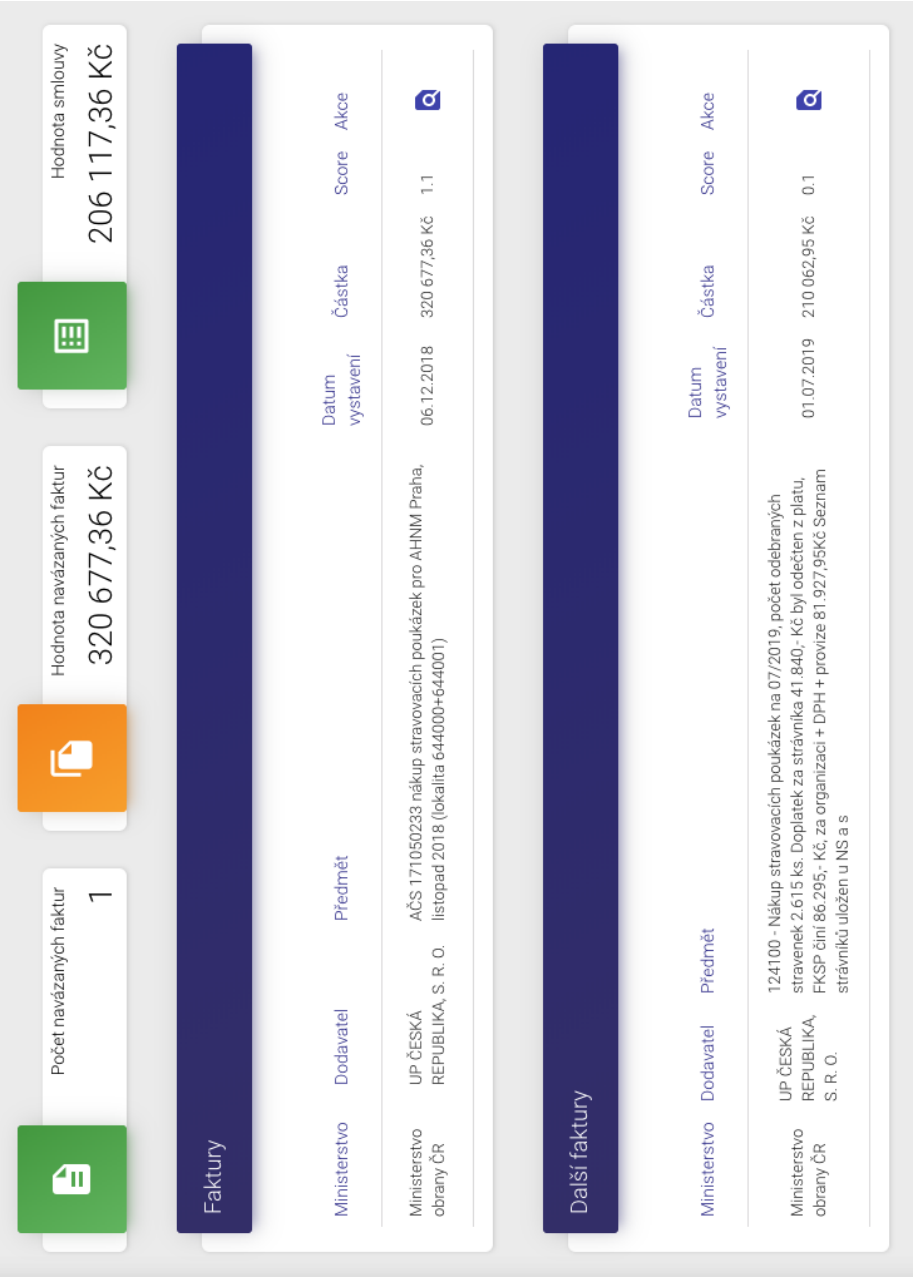
Obrázek 3.10: Stránka Ministerstva

Obrázek 3.11: Stránka Faktury

Obrázek 3.12: Stránka Smlouvy



Obrázek 3.13: Stránka Detail Smlouvy - Atributy



Obrázek 3.14: Stránka Detail Smlouvy - Namapované faktury

Závěr

Cílem této práce bylo analyzování dat, která ministerstva České republiky zveřejňují o uzavřených smlouvách a vyplacených fakturách. Dále v závislosti na této analýze navrhnout metody pro mapování faktur na smlouvy a následně je implementovat. A na závěr vytvořit webový portál, kde budou výsledky mapování prezentovány.

Při analýze dat jsem zjistil, že pro spojení faktur se smlouvami jsou nejvíce užitečné údaje smluvních stran a časové údaje, které je možné použít k zúžení možných spojení, předmět smlouvy nebo faktury, který obsahuje informace o zboží nebo službě, číslo smlouvy nebo také částka. Na základě těchto atributů a vztahů mezi nimi, jsem navrhl metodu, která pomocí sady testů identifikuje nejspolehlivější spojení, které je následně považováno za nejpravděpodobnější. V některých případech navržená metoda spojení nedokáže vytvořit, a to z důvodu chybějících dat nebo nedostatku klíčových prvků.

Pro stažení potřebných dat je vytvořena aplikace, která extrahuje informace o uzavřených smlouvách z registru smluv. Zároveň extrahuje informace o fakturách ministerstev České republiky získaných pomocí nástroje Laboratoře otevřených dat. Pro mapování faktur na smlouvy je vytvořen program, který implementuje navržené metody a výsledky uloží do databáze, odkud je s nimi možné dále pracovat.

Na závěr je vytvořeno REST API pro přístup k datům a webový portál, který zobrazuje informace o fakturách, smlouvách, vytvořených spojeních a upozorňuje na zjištěné nesrovnalosti, které byly identifikovány na základě vytvořených spojení.

Při spuštění program pro mapování faktur na smlouvy vyhodnotil, že z přibližně 80 000 smluv existuje 595 smluv, u kterých jsou zjištěny některé nesrovnalosti ve vyfakturované částce. Průměrně, z těchto 595 smluv, bylo vyfakturováno o 42 % více, než bylo uvedeno ve smlouvě. Výsledky je ale nutné ještě manuálně projít a zjistit, zda jsou položky navázány správně.

Tato práce může být nadále rozšiřována, například přidáním testů, které by využívaly metod zpracování přirozeného jazyka (NLP - Natural Langu-

age Processing). Také by bylo možné zdokonalení procesu získávání dat tak, aby byly informace o smlouvách a fakturách získávány inkrementálně. Nebo přidání funkcionalit do webového portálu, které by umožňovaly vytváření a upravování testů, nastavení konfigurace procesu mapování a možnost manuální úpravy spojení mezi fakturami a smlouvami.

Projekt je uveřejněn v repositáři na adrese `https://github.com/opendatalabcz/invoice-contract-matcher`, kde bude dál rozšiřován. Projekt je takto veřejně přístupný a můžou ho tak získat vývojáři, kteří chtějí využít výsledky této práce. Vývojáři, kteří tuto práci budou chtít využít, musí dodržet jeho GPL 3.0 licenci.

Literatura

[1] Registr smluv. *Registr smluv* [online]. Praha: Ministerstvo vnitra České republiky, 2016 [cit. 15. 4. 2020]. Dostupné z: https://smlouvy.gov.cz/

[2] Česko. § 3 odst. 7 zákona č. 106/1999 Sb., o svobodném přístupu k informacím. *In: Zákony pro lidi.cz* [online]. © AION CS 2010-2020 [cit. 15. 4. 2020]. Dostupné z: https://www.zakonyprolidi.cz/cs/1999-106#p3-7

[3] Národní katalog otevřených dat na Portálu veřejné správy In: *Ministerstvo vnitra České republiky* [online] Praha: Ministerstvo vnitra České republiky, c2020 [cit. 16. 5. 2020]. Dostupné z: https://www.mvcr.cz/clanek/zpravodajstvi-narodni-katalog-otevrenych-dat-na-portalu-verejne-spravy.aspx

[4] About. *CKAN* [online]. [cit. 16. 5. 2020]. Dostupné z: https://ckan.org/

[5] Česko. Zákon č. 340/2015 Sb., Zákon o zvláštních podmínkách účinnosti některých smluv, uveřejňování těchto smluv a o registru smluv (zákon o registru smluv) In: *Sbírka zákonů České republiky*. 2015, Dostupné z: http://aplikace.mvcr.cz/sbirka-zakonu/ViewFile.aspx?type=z&id=37369

[6] Česko. *Úplné znění Ústavního zákona České národní rady č. 1/1993 Sb., Ústava České republiky: Úplné znění Usnesení České národní rady č. 2/1993 Sb., o vyhlášení Listiny základních práv a svobod jako součásti ústavního pořádku České republiky ; některé další související právní předpisy. In: .* Vydání: čtrnácté. Praha: Armex Publishing, 2019. Edice kapesních zákonů. ISBN 978-80-87451-66-3.

[7] FABÍKOVÁ, Jana. *Veřejná správa a svobodný přístup k informacím.* Brno, 2006. Bakalářská práce. Masarykova univerzita, Fakulta právnická, Katedra správní vědy, správního práva a finančního práva. Vedoucí práce JUDr. Petr Kolman.

[8] Česko. Zákon č. 563/1991 Sb., o účetnictví, s vyznačením změn podle zákonného opatření Senátu č. 344/2013 Sb. In: *Sbírka zákonů České republiky*. 1991, § 11, s. 11. Dostupné z: https://www.mfcr.cz/assets/cs/media/Zak_1991-563_UZ-Zakon-c563-1991-s-vyznacenim-zmen-k-112014.pdf

[9] VESELÁ, Radomíra. *Základy občanského práva hmotného.* Kunovice: Evropský polytechnický institut, 2011. ISBN 978-80-7314-258-2.

[10] Metodický návod k aplikaci zákona o registru smluv. *Ministerstvo vnitra České republiky* [online]. 2019, 16. 10. 2019 [cit. 2020-04-15]. Dostupné z: https://www.mvcr.cz/soubor/metodicky-navod-k-aplikaci-zakona-o-registru-smluv-jez-slouzi-k-zakladni-orientaci-v-problematice-a-prinasi-zakladni-odpovedi-na-casto-kladene-dotazy.aspx

[11] Vize. *Hlídač státu* [online]. [cit. 26. 4. 2020]. Dostupné z: https://www.hlidacstatu.cz/texty/o-serveru/

[12] O programu. *Supervizor Ministerstva financí* [online]. [cit. 15. 4. 2020]. Dostupné z: https://supervizor.mfcr.cz

[13] *Microsoft Dinamics 365: Financial Management Software* [online]. Washington: Microsoft, 2020 [cit. 2020-04-15]. Dostupné z: https://dynamics.microsoft.com/en-in/finance/overview/

[14] Kdo jsme. *OpenDataLab* [online]. [cit. 26. 4. 2020]. Dostupné z: https://opendatalab.cz/#kdo-jsme

[15] GITHUB. *Opendata*. GitHub [online]. [San Francisco]: GitHub, c2020 [cit. 3. 5. 2020]. Dostupné z: https://github.com/opendatalabcz/opendata

[16] PYTHON SOFTWARE FOUNDATION. *Python*. 2020. Verze 3.8.2. Dostupné z: https://www.python.org/

[17] POSTGRESQL GLOBAL DEVELOPMENT GROUP. *PostgreSQL*. 2020. Verze 10.12. Dostupné z: https://www.postgresql.org

[18] ARMIN RONACHER. *Flask*. 2020. Verze 1.1.2 Dostupné z: https://flask.palletsprojects.com/en/1.1.x/

[19] FACEBOOK INC. *React*. 2020. Verze 16.13.1 Dostupné z: https://reactjs.org

[20] MEHTA, Bhakti. *RESTful Java Patterns and Best Practices: Learn best practices to efficiently build scalable, reliable, and maintainable high performance RESTful services*. 1. Livery Place 35 Livery Street Birmingham B3 2PB, UK.: Packt Publishing, 2014. ISBN 978-1-78328-796-3.

[21] GITHUB. *Material Dashboard*. GitHub [online]. [San Francisco]: GitHub, c2020 [cit. 3. 5. 2020]. Dostupné z: `https://github.com/creativetimofficial/material-dashboard`

Příloha A

Instalační příručka

Zde uvádím instalační příručku, která má pomoci těm, kteří moji práci budou spouštět. Tuto příručku dělím do dvou částí.

A.1 Backend

Tato část obsahuje návod, jak nainstalovat potřebné programy pro spuštění načítání dat, mapování faktur na smlouvy a vystavení REST API.

- Pro spuštění je nutné nainstalovat Python 3. Instalační soubor je možné stáhnout na stránce: `https://www.python.org/downloads/`

  Postup instalace je popsán na této stránce: `https://realpython.com/installing-python/`

- Po nainstalování je nutná instalace knihoven, které jsou využívány. Pro instalaci knihoven spusťte následující příkaz

  ```
  pip install -r backend/requirements.txt
  ```

  (zde uvádím příkazy pro UNIX systém. Na platformě windows se příkazy mohou trochu lišit)

  Pokud pracujete na systému, kde máte více verzí, specifikujte, že chcete použít python 3

  ```
  pip3 install -r backend/requirements.txt
  ```

- V této práci používám databázi PostgreSQL.

  Stažení je možné z oficiálních stránek:

  `https://www.postgresql.org/download/`

  Návod, jak nainstalovat databázi na vaše zařízení, naleznete například zde:

  `https://www.postgresqltutorial.com/install-postgresql/`

Pro práci s databází je možné využít terminál:

`https://www.postgresql.org/docs/12/app-psql.html`

Nebo je zde možnost využití nástroje pgAdmin:

`https://www.pgadmin.org/`

Nebo využít nástrojů ve vývojovém prostředí, například PyCharm:

`https://www.jetbrains.com/help/pycharm/relational-databases.html`

- Konfigurace

  Jako zdroj nastavení slouží soubor configuration.ini

  Soubor obsahuje 5 částí:

  Flask Obsahuje nastavení frameworku Flask
  - Parametry host a port určují, na jaké adrese bude REST API dostupné
  - Pokud chcete, aby bylo možné se k API připojit ze sítě, nastavte host na adresu 0.0.0.0

  matcherdb obsahuje údaje potřebné k připojení k databázi, která bude využita pro uložení dat, které budou následně použity při párování

  opendatadb obsahuje údaje potřebné k připojení k databázi, která slouží jako zdroj faktur

  contract_provider slouží ke specifikování parametrů pro třídu ContractProviderRegistr

  deciding_pipeline slouží k specifikování parametrů pro třídu DecidingPipeline

  V souboru configuration.ini je potřeba nastavit sekce matcherdb a opendatadb.

  Matcherdb jsou údaje pro připojení k databázi, kam budou data uložena.

  Při využití opendata databáze:

  - Opendatadb jsou údaje pro připojení k databázi, ze které se stáhnou faktury
  - Opendata databázi je možné naplnit daty pomocí aplikace OpendataLabu dostupné zde:

    `https://github.com/opendatalabcz/opendata`

  Při využití smluv z Registru smluv:

  - Pro stažení smluv z Registru smluv není potřeba upravovat nic.
  - Pouze zdrojovou adresu v sekci contract_provider, pokud se změnila.

A.1.1 Spuštění

Před spuštěním nahrávání dat je nutné vytvořit tabulky, do kterých jsou nahrány smlouvy a faktury.

Pro vytvoření tabulek v databázi a nahrání základních dat spusťte create script, který naleznete:

`backend/Database/scripts/drop_create_tables.sql`

A.1.2 Stažení dat

Po nastavení potřebných údajů, je třeba spustit soubor data_downloader.py

```
python data_downloader.py
```

Tento program stáhne nejdříve faktury a poté smlouvy.

A.1.3 Spuštění párování

Pro spuštění párování je spusťe matcher.py

```
python matcher.py
```

Tento program páruje faktury na smlouvy. Po dokončení jsou vytvořeny záznamy o podezřelých smlouvách a statistiky.

A.1.4 REST API

Pokud jsou faktury namapovány na smlouvy, je možné spustit Flask aplikaci, která vystaví REST API.

- Nejdříve nastavíme proměnou FLASK_APP, aby ukazovala na soubor flask_runner.py:

  ```
  export FLASK_APP=flask_runner.py
  ```

- Flask poté spustíme příkazem:

  ```
  python flask_runner.py
  ```

  Po spuštění jsou vidět requesty, které aplikace zpracovala.

  Nebo je možné použít příkaz:

  ```
  flask run
  ```

  Ale tento příkaz bude ignorovat nastavení, které je uložené v souboru configuration.ini

Host a port je možné nastavit pomocí parametrů v parametrech

```
flask run --host 0.0.0.0
```

A.2 Frontend

Pro spuštění webového klienta je nutné nainstalovat Node.js. Instalační soubor je dostupný zde:

`https://nodejs.org/en/download/`

Návod, jak Node.js nainstlovat neleznete zde:
`https://nodejs.org/en/download/package-manager/`

- Pro nainstalování potřebných balíčků spusťte:

  ```
  npm install
  ```

- Po nainstalování potřebných balíčků je možné spustit server pomocí příkazu:

  ```
  npm start
  ```

  Tento příkaz spustí server na adrese `http://127.0.0.1:3000`

  Na této adrese je také dostupná dokumentace k REST API, kde je možné spojení otestovat.

- Pro vytvoření produkční verze spusťte:

  ```
  npm run-script build
  ```

  Build je následně uložen v adresáři `frontend/build`

  Úprava adresy odkazující na REST API, ze kterého jsou získávány data, je možná v souboru:

  `frontend/src/variables/general.js`

  Nastavení npm příkazů je možné v souboru package.json

  V tomto souboru jsou uloženy také dependencies.

Příloha B

Seznam použitých zkratek

NKOD Národní katalog otevřených dat

CKAN Comprehensive Knowledge Archive Network

JSON Comprehensive Knowledge Archive Network

NLP Natural language processing

REST Representational state transfer

API Application program interface

CSV Comma separated values

XML Extensible markup language

IČO Identifikační číslo osoby

DPH Daň z přidané hodnoty

URL Uniform resource locator

ORM Object relational mapping

SQL Structured query language

HTTP Hypertext transfer protocol

UI User interface

HTML Hypertext markup language

CRUD Create, read, update, delete

Příloha C

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `src`
  - `backend` ..... zdrojové kódy implementace backend části
  - `frontend` ..... zdrojové kódy implementace frontend části
  - `README.md` ..... instalační příručka
- `text` ..... text práce
  - `thesis.pdf` ..... text práce ve formátu PDF