

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Automatizované získávání dat z veřejných rejstříků a generování dokumentů

Bakalářská práce

Vedoucí práce:
Ing. Petra Čáčková, Ph.D.

Jonáš Petrovský

Brno 2014

Na tomto místě bych chtěl poděkovat vedoucí své bakalářské práce Ing. Petře Čáčkové, Ph.D. za cenné rady a připomínky k práci, stejně jako za trpělivost při odpovídání na mé dotazy. Dále bych rád poděkoval své rodině, že mě během celého studia i tvorby této práce podporovala.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Automatizované získávání dat z veřejných rejstříků a generování dokumentů**

vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 15. května 2014

.....

Abstract

PETROVSKÝ, J. Automated acquiring of data from public registers and document generation. Bachelor thesis. Brno, 2014.

The thesis deals with public registers, options of acquiring data from them and document generation. Firstly the system of basic registers is described and the overview of registers in Czech Republic is created. Then the module for data acquiring from chosen registers is created (in PHP). Finally the web application for generating standard types of documents (document model in XML, generating to PDF) is created. It is focused on agreements, but it can be extended on other document types. It consists of two modules: 1. administrative for user and document type management, 2. user for document management. Used technologies: FPDF, Nette, curl, PHP, MySQL, Apache.

Keywords

open data, basic registers, public registers, acquiring data from web, document generation, web application, HTML parsing, XML, PDF, PHP, Nette, FPDF, curl

Abstrakt

PETROVSKÝ, J. Automatizované získávání dat z veřejných rejstříků a generování dokumentů. Bakalářská práce. Brno, 2014.

Práce se zabývá veřejnými rejstříky, možnostmi získávání dat z nich a generováním dokumentů. Nejdříve je popsán systém základních registrů a vytvořen přehled rejstříků v ČR. Poté je vytvořen modul pro získávání dat z vybraných rejstříků (v PHP). Nakonec je vytvořena webová aplikace pro generování standardních typů dokumentů (model dokumentu v XML, generování do PDF). Je zaměřená na smlouvy, ale lze ji rozšířit i na další typy dokumentů. Tvoří ji dva moduly: 1. administrační pro správu uživatelů a typů dokumentů, 2. uživatelský pro správu dokumentů. Použité technologie: FPDF, Nette, curl, PHP, MySQL, Apache.

Klíčová slova

otevřená data, základní registry, veřejné rejstříky, získávání dat z webu, generování dokumentů, webová aplikace, parsování HTML, XML, PDF, PHP, Nette, FPDF, curl

Obsah

1	Úvod a cíl práce	13
1.1	Úvod	13
1.2	Cíl práce	15
2	Veřejné rejstříky	16
2.1	Rejstřík a registr	16
2.2	Systém základních registrů	16
2.2.1	Zákon o základních registrech	16
2.2.2	Informační systém ORG	17
2.2.3	Informační systém základních registrů	17
2.3	Přehled rejstříků	19
2.3.1	Rejstříkové soudy	20
2.3.2	Podnikatelské registry	20
2.3.3	Některé důležité registry	22
2.3.4	Další užitečné registry	23
2.3.5	Další zajímavé registry	24
2.3.6	Seznamy organizací	25
2.3.7	Evidence osob	26
2.3.8	Registry dlužníků	26
3	Modul pro získávání dat	27
3.1	Vybrané rejstříky	27
3.1.1	Testovací skripty	27
3.2	ARES	28
3.2.1	Popis	28
3.2.2	Nevýhody	29
3.3	Požadavky	30
3.3.1	Funkční požadavky	30
3.3.2	Nefunkční požadavky	30
3.4	Návrh modulu	31
3.4.1	Adresářová struktura	31
3.4.2	Diagram tříd	31
3.5	Implementace modulu	32
3.5.1	Soubor core.php	32
3.5.2	Princip získání dat z registru	33
3.5.3	Obecný popis třídy registru	35
3.5.4	Popis jednotlivých tříd registrů	37
3.5.5	Použití modulu	39

4	Generování standardních typů dokumentů	40
4.1	Metodika	40
4.2	Princip aplikace	40
4.2.1	Model dokumentu	40
4.2.2	Základní princip	41
4.2.3	Bod 2 – vytvoření formuláře	41
4.2.4	Bod 4 – generování PDF	42
4.3	Umístění aplikace v globální architektuře	42
4.4	Požadavky	42
4.4.1	Administrátor	43
4.4.2	Uživatel	43
4.4.3	Nefunkční požadavky	43
4.5	Návrh aplikace	43
4.5.1	Datový model	43
4.5.2	Uživatelské rozhraní	45
4.5.3	Další návrhy	45
4.6	Implementace aplikace	46
4.6.1	Použité technologie	46
4.6.2	Architektura MVC	46
4.6.3	Struktura aplikace	47
4.6.4	Modelové třídy	47
4.6.5	Routování a presentery	48
4.6.6	Generování PDF	48
4.6.7	Vybrané funkce aplikace	49
5	Diskuze	52
5.1	Přehled veřejných rejstříků	52
5.2	Modul pro získávání dat	52
5.3	Webová aplikace pro generování dokumentů	52
6	Závěr	54
7	Literatura	55
	Seznam zkratk	59
	Přílohy	60
A	Elektronická příloha	61
B	E-mailová korespondence s Mgr. Hejdukem	62
C	Soubor core.php	63

OBSAH	11
D Odeslání souboru přes cURL	64
E XML dotazy na RŽP	65
F Webová aplikace – obrazovky	66

1 Úvod a cíl práce

1.1 Úvod

Každý stát potřebuje uchovávat základní údaje o fyzických a právnických osobách nebo dalších entitách (např. budovy), které se v daném státě vyskytují. Tyto údaje se nacházejí v tzv. rejstřících (registrech). Před vynálezem výpočetní techniky byly tyto rejstříky vedeny v listinné podobě. Pokud chtěl občan do nějakého rejstříku nahlédnout, musel přijít na určité úřední místo a požádat o výpis či náhled. Toto nebylo příliš pohodlné. A to ani pro státní úředníky, kteří v případě potřeby museli složitě vyhledávat informace o daném subjektu na různých místech.

Pokud se zaměříme na Českou republiku, budeme uvažovat dobu od vzniku samostatné ČR (1. 1. 1993), jelikož před tímto datem byla situace poněkud komplikovanější. V 90. letech 20. století započal obrovský rozvoj výpočetní techniky. Tudiž bylo možné vybavit pracoviště orgánů veřejné moci osobními počítači a pořídit výkonné servery. To spolu se vznikem Internetu dalo impuls pro elektronizaci státní správy (tzv. *E-government*), která trvá dodnes.

Jako příklad můžeme vzít obchodní rejstřík. Cesarová (2008, s. 31) potvrzuje rychlé vybavování rejstříkových soudů výpočetní technikou od 90. let 20. století. Také ovšem zdůrazňuje, že „propojení pro vzájemnou komunikaci mezi soudy a přijetí potřebné právní úpravy mělo značné zpoždění“. A například trestní rejstřík se dočkal plné elektronizace až v roce 2004 (ADVICE.CZ, 2008).

Dalším krokem E-governmentu je dálkové zpřístupnění údajů občanům státu a to prostřednictvím internetu. V tomto ohledu je česká vláda poměrně aktivní, velké množství rejstříků je takto přístupných. Nicméně k ideálnímu stavu je ještě hodně daleko. Bylo by užitečné inspirovat se v zahraničí, kde mají s podobnými aktivitami více zkušeností. Progresivní je v tomto ohledu také náš soused – Slovenská republika. Ta již zavedla několik opatření, o kterých se u nás stále pouze mluví. Jak píše Kočí (2012), „už druhým rokem zveřejňují slovenské úřady veškeré smlouvy v Centrálním registru smluv“ a také existuje centrální Katalog vládních dat.

V širších souvislostech se v poslední době mluví o tzv. *open data* – otevřených datech, která zveřejňují vládní instituce (či soukromé subjekty) ve strojově čitelném formátu, který umožňuje automatizované zpracování (Fond Otakara Motejla, 2014). Problematikou *open government data* se zabývá celá řada iniciativ. Česká republika přistoupila v roce 2011 k mezinárodní iniciativě Open Government Partnership a 2012 byl vytvořen *Akční plán České republiky – Partnerství pro otevřené vládnutí* (Vláda ČR, 2012). V tomto plánu se ČR zavázala k určitým krokům:

1. přijetí Zákona o úřednicích veřejné správy – k 1. 1. 2014
2. zefektivnění systému svobodného přístupu k informacím – novela Zákona o svobodném přístupu k informacím – k 1. 1. 2014
3. zpřístupnění dat a informací

Podle hledání na www.zakonyprolidi.cz nebyly body jedna a dva k datu 6. 4. 2014 naplněny. Bod tři obsahuje 4 části, které zřejmě byly splněny všechny, nicméně je otázka, do jaké hloubky. Každopádně vznikl Katalog dat ČR na adrese <http://cz.ckan.net>, který byl vyvinut v rámci iniciativy opendata.cz. Nejedná se zatím o pravou otevřenou datovou infrastrukturu, je to spíše příprava pro její budování (Opendata.cz, 2012).

Můžeme se také podívat na roli státu v prezentování informací, získaných ze zdrojových dat. V článku od Robinson (2009) zaznívá názor, že stát by měl snížit svou roli při prezentování důležitých informací občanům. Tedy místo vytváření webových stránek, které uspokojí všechny možné potřeby uživatelů, by měl spíše vytvářet kvalitní technickou infrastrukturu pro poskytování dat. Tato data potom mohou použít soukromé subjekty, které lépe znají potřeby občanů a mohou neustále aktualizovat webové služby dle jejich požadavků. Tento názor je logický a dá se s ním souhlasit. Nicméně je asi potřeba mít několik státních webových služeb pro základní veřejné rejstříky, aby člověk měl nějakou garanci, že dostává informace z oficiálního zdroje.

V souvislosti s výše uvedenými fakty se dá usoudit, že plně otevřených vládních dat se v ČR v nejbližší době nedočkáme. Je proto nutné využít již existující zdroje dat a pokusit se data získat v co největší kvalitě a kvantitě. Jak je zřejmé, takových dat je spousta. V dalším textu bude pozornost zaměřena na veřejné rejstříky, poskytující základní informace o entitách, vyskytujících se v ČR.

Generování dokumentů

Lidstvo se od nepaměti snaží uchovávat informace. Ty mohou být strukturované nebo nestrukturované. V každém případě je potřeba tyto informace nějak prezentovat. V dávných dobách lidské civilizace se k tomu používaly hliněné tabulky, později inkoust a pergamen (či papír) a konečně knihtisk. Tyto metody umožňovaly, pokud pomineme umělecká díla, pouze malé možnosti formátování textu. To ovšem nebylo na škodu, jelikož tím pádem výsledné dokumenty vypadaly podobně a byly přehledné.

Pokud se přesuneme na konec 20. století, situace se radikálně změnila. Pomocí počítače a vhodného softwaru bylo možné vytvořit téměř jakýkoliv grafický návrh dokumentu a tento potom na tiskárně vytisknout. Možnosti nastavit barvu textu nebo pozadí, podtrhnout text čarou nebo změnit velikost písma uchvátily nejdříve uživatele natolik, že byl schopen vyprodukovat dokument, který porušoval snad všechna typografická pravidla.

V současné době je situace podobná. Nicméně existuje množství metod, jak zajistit produkci dokumentů, které mají kvalitní technické a také estetické provedení. První možností je generování dokumentu z dat uložených v databázi. To je vhodné například pro tisk faktury, nabídky zboží nebo výpisu z veřejného rejstříku. Druhou možností jsou systémy pro generování dokumentů standardních typů, kde uživatel pouze upraví nebo zadá jiné údaje. Tyto se následně zpracují a výsledkem je

originální dokument, který ovšem odpovídá určenému typu dokumentu (šabloně). V relevantní kapitole práce je diskutován a implementován systém tohoto typu. Systém je zaměřen na generování smluv a to na základě podnětu z praxe, který si vyžádal ověřit schůdnost a funkčnost takového řešení.

1.2 Cíl práce

Na základně zadání práce je stanoven následující cíl práce: Vytvořit přehled veřejných rejstříků v ČR, vytvořit modul pro automatizované získávání dat z vybraných veřejných rejstříků a následně vytvořit webovou aplikaci pro generování standardních typů dokumentů.

Pro splnění cíle je potřeba provést tyto kroky:

- Informovat se o tom, jaké veřejné rejstříky v České republice vlastně existují.
- Seznámit se s možnostmi automatizovaného získávání dat z nich.
- Navrhnout, jak data získávat a implementovat tento návrh do podoby programového modulu.
- Navrhnout a implementovat webovou aplikaci, která generuje dokumenty standardních typů. Aplikace je zaměřená na smlouvy, nicméně je možné ji rozšířit i o další typy dokumentů.

Nakonec bude celá práce zhodnocena.

2 Veřejné rejstříky

V následující kapitole je uveden přehled veřejných rejstříků a jsou popsány možnosti automatického získávání dat z těchto rejstříků.

2.1 Rejstřík a registr

Rejstřík lze definovat jako úředně vedený seznam (Ústav pro jazyk český, 2011). Ve stejném významu se také používá slovo registr. Tento seznam může být:

- veřejný – Kdokoliv do něj může volně nahlížet.
- neveřejný – Pro nahlížení je potřeba podat žádost a dostat povolení k nahlédnutí.
- selektivní – Nahlízet mohou pouze subjekty určitého typu, ostatní ne.

Dále se zaměříme především na veřejné rejstříky spravované státem.

Nový občanský zákoník operuje v oblasti věcných práv s jejich evidencí ve veřejných seznamech. Veřejný seznam má dva hlavní účely: centrální evidence zapisovaných údajů a upevnění právní jistoty společnosti. Druhý bod lze chápat tak, že osoby mají dobrou víru v to, že co je v seznamu psáno, je také dáno – je to pravdivé. A naopak – co v seznamu chybí, tak ve skutečnosti také neplatí (MS ČR, 2014b).

2.2 Systém základních registrů

Potřeba vytvořit jednotné registry pro státní zprávu byla diskutována už na začátku 90. let 20. stol. Ovšem dlouhou dobu se jednalo o izolované systémy, které neumožňovaly sdílení údajů, obsahovaly nejednotná data a neexistovala právní úprava pro sdílení a efektivní využití těchto údajů. Cílem nového právní úpravy je přeměna současného systému sběru dat v systém nový, který umožní stanovené informace sbírat a se zárukou spolehlivosti využívat v celé státní správě (Mates, 2012, s. 90).

Ostrý provoz systému základních registrů byl zahájen v souladu se Zákonem č. 111/2009 Sb., o základních registrech dne 1. 7. 2012 (SZR, 2014a).

2.2.1 Zákon o základních registrech

Zákon č. 111/2009 Sb., o základních registrech (mj.) specifikuje základní registry – konkrétní informační systémy veřejné správy (ISVS), které na jeho základě vznikly:

- základní registr obyvatel (registr obyvatel),
- základní registr právnických osob, podnikajících fyzických osob a orgánů veřejné moci (registr osob),
- základní registr územní identifikace, adres a nemovitostí (registr územní identifikace),
- základní registr agend orgánů veřejné moci a některých práv a povinností (registr práv a povinností).

Zákon dále zavádí tyto pojmy:

- referenční údaj – údaj vedený v registru, který je označen jako referenční,
- orgán veřejné moci (OVM) – státní orgán, územní samosprávný celek a fyzická nebo právnická osoba, byla-li jí svěřena působnost v oblasti veřejné správy,
- editor – OVM, který je oprávněn zapisovat referenční údaje do základního registru a provádět změny zapsaných referenčních údajů,
- agenda – souhrn činností spočívajících ve výkonu vymezeného okruhu vzájemně souvisejících činností v rámci působnosti orgánu veřejné moci,
- agendový informační systém – ISVS sloužící k výkonu agendy.

2.2.2 Informační systém ORG

Důležitou součástí zákona je také IS ORG – tzv. *převodník identifikátorů*. Tento systém spravuje Úřad pro ochranu osobních údajů. Chrání osobní údaje fyzických osob v základních registrech.

Systém operuje se dvěma typy identifikátorů (Mates, 2012, s. 93):

- zdrojový identifikátor fyzické osoby (ZIFO) – fyzické osobě jej přidělí Úřad
- agendový identifikátor fyzické osoby (AIFO) – odvozen ze ZIFO a kódu agendy; je užíván výlučně v agendě, pro kterou byl přidělen

Oba dva identifikátory jsou neveřejné a musí být unikátní. Systém ORG převádí AIFO jedné agendy na AIFO druhé agendy (toto dokáže jako jediný v systému základních registrů). Mimo ZIFO a AIFO neobsahuje žádné jiné osobní údaje. Komunikuje pouze s Informačním systémem základních registrů (SZR, 2014c).

2.2.3 Informační systém základních registrů

Zákon o základních registrech také zřizuje Informační systém základních registrů (ISZR). Tento ISVS poskytuje prostředí, v jehož rámci 4 výše zmíněné registry fungují. Provozuje jej Správa základních registrů.

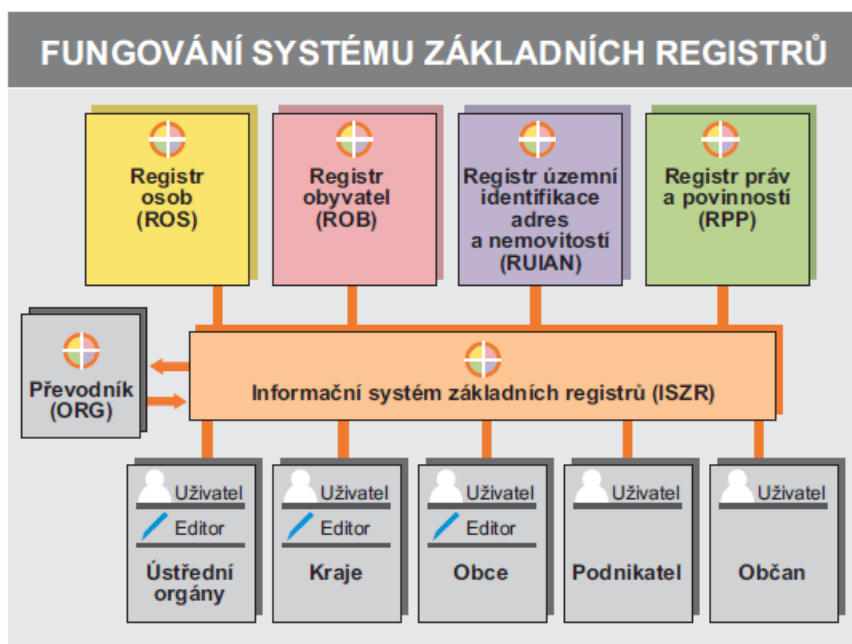
Autor práce vznesl na Správu základních registrů e-mailem dotaz, kdo všechno vlastně může přistupovat k ISZR. Hlavně ho zajímalo, jestli existuje veřejný přímý přístup do základních registrů přes webovou službu. Velmi rychle přišla odpověď ze Service desku v tomto znění: „Připojovat se k ISZR mohou pouze OVM. Firmy či jednotlivci mohou získávat některé referenční údaje ZR přes PVS gov.cz odkaz Základní registry s použitím datové schránky přes tam uvedené formuláře nebo prostřednictvím kontaktních míst veř. správy CzechPOINT.“ Z toho se dá usoudit, že takový přístup neexistuje, a k získání dat je potřeba využít konkrétní agendové systémy (ty veřejné).

O několik dnů později přišla odpověď od Mgr. Marka Hejduka (pracovníka pro vztah s veřejností), kterou lze nalézt v příloze B. Zpráva je obsáhlá a obsahuje užitečné shrnující informace.

V kombinaci se zdrojem (SZR, 2014a) lze ISZR charakterizovat takto:

- Základní registry byly vybudovány především pro potřebu orgánů veřejné moci. Představují pro ně jediný relevantní zdroj údajů.
- Pro osoby to znamená zefektivnění komunikace se státem. To se týká rychlejšího vyřízení žádostí nebo například změny údajů – tu je potřeba provést jen na jednom místě, do ostatních registrů se změna automaticky promítne.
- Poskytuje komplexní služby definované v katalogu eGON služeb. Tyto služby nad základními registry poskytuje všem subjektům s ohledem na jejich aktuální oprávnění v registru práv a povinností.
- Fyzické a právnické osoby mohou k referenčním údajům přistupovat pouze prostřednictvím k tomu speciálně určených agendových informačních systémů:
 - Czech POINT – na základě žádosti u přepážky
 - Datové schránky – přístup k údajům uskutečňuje vlastník datové schránky s využitím formulářů uveřejněných na Portálu veřejné správy

Princip fungování systému základních registrů znázorňuje obrázek 1.



Obrázek 1: Schéma fungování systému základních registrů (SZR, 2014b)

2.3 Přehled rejstříků

Tato část kapitoly obsahuje seznam vybraných rejstříků a informace o nich – jaká data obsahují, jestli jsou volně přístupné přes internet a zda poskytují API¹.

Oficiální seznam všech státních rejstříků (resp. ISVS) představuje Informační systém o informačních systémech veřejné správy (přístupný na <https://www.sluzby-isvs.cz/ISoISVS/Applets/DefaultSSL.aspx>). Těch s veřejnou částí bylo k 9. 5. 2014 přesně 1443. Toto množství je příliš velké – vhodné rejstříky je tedy nutné vybrat jiným způsobem.

Postup hledání rejstříků spočíval v podstatě ve dvou krocích:

1. zaměření se na zdroje dat, které se používají v základních registrech
2. zjišťování informací o dalších rejstřících

Pro krok dva byly využity následující zdroje – kromě prvního webové stránky:

- Zákon č. 304/2013 Sb., o veřejných rejstřících právnických a fyzických osob
- Portál veřejné správy – Rejstříky:
<https://portal.gov.cz/portal/rejstriky/data/index.html>
- ARES – Administrativní registr ekonomických subjektů:
<http://wwwinfo.mfcr.cz/ares/ares.html.cz>
- BusinessInfo.cz – Registry, databáze:
<http://www.businessinfo.cz/cs/online-nastroje/registry-database.html>
- statnisprava.cz – Rejstříky a registry:
<http://www.statnisprava.cz/rstsp/redakce.nsf/i/rejstriky>
- Rejstříky.cz – seznam rejstříků všech druhů: <http://www.rejstriky.cz/>
- Wikipedia.org
 - Veřejný rejstřík: http://cs.wikipedia.org/wiki/Ve%C5%99ejn%C3%BD_rejst%C5%99%C3%ADk
 - Kategorie: Veřejné registry v Česku: http://cs.wikipedia.org/wiki/Kategorie:Ve%C5%99ejn%C3%A9_registry_v_%C4%8Cesku
 - Registr dlužníků: http://cs.wikipedia.org/wiki/Registr_dlu%C5%BEn%C3%ADk%C5%AF

V dalším textu předpokládáme, že pokud není uvedeno jinak, je rejstřík veřejně přístupný a neobsahuje API.

Neveřejné jsou např. Registr obyvatel, Centrální registr vozidel a Rejstřík trestů.

¹Application Programming Interface – rozhraní pro programový přístup k dané aplikaci

2.3.1 Rejstříkové soudy

Dne 1. 1. 2014 nabyl účinnosti Zákon č. 304/2013 Sb., o veřejných rejstřících právnických a fyzických osob. Ten stanovuje, že:

- Veřejnými rejstříky (VR) podle tohoto zákona se rozumí: spolkový rejstřík, nadační rejstřík, rejstřík ústavů, rejstřík společenství vlastníků jednotek, obchodní rejstřík a rejstřík obecně prospěšných společností.
- Do VR se zapisují zákonem stanovené údaje o právnických a fyzických osobách.
- VR je ISVS, je veden v elektronické podobě a vede ho rejstříkový soud.

Pro dálkový přístup k rejstříkům lze použít web Ministerstva spravedlnosti:

<https://or.justice.cz/ias/ui/rejstrik>

Ten umožňuje vyhledávání, ale bohužel nenabízí API.

2.3.2 Podnikatelské registry

Zde jsou představeny registry, týkající se podnikatelů (podnikající právnické a fyzické osoby).

Obchodní rejstřík – OR

- správce (OVM): Ministerstvo spravedlnosti – rejstříkové soudy
- eviduje: obchodní společnosti a družstva (obchodní korporace); fyzické osoby, které jsou podnikateli a požádají o zápis.
- URL: <https://or.justice.cz/ias/ui/rejstrik>

Živnostenský rejstřík – RŽP

- správce: Ministerstvo průmyslu a obchodu
- eviduje: fyzické a právnické osoby, které provozují živnost (jak je definována v Zákonu č. 455/1991 Sb., o živnostenském podnikání).
- URL: <http://www.rzp.cz>
- API: ano – popis: http://www.rzp.cz/docs/RZP02_XML_25.pdf

Registr ekonomických subjektů – RES

- správce: Český statistický úřad
- eviduje: Ekonomickým subjektem je každá právnická osoba, fyzická osoba s postavením podnikatele a organizační složka státu, která je účetní jednotkou (ČSÚ, 2012).
- URL: <http://registry.czso.cz/irsw/>
- API: ne – Dle e-mailu od Ondřeje Košaty lze pouze koupit databázi RES (nebo výběry z ní) jako produkt.

Evidence zemědělského podnikatele – EZP

- správce: Ministerstvo zemědělství
- eviduje: fyzické nebo právnické osoby, které hodlají provozovat zemědělskou výrobu za účelem zisku (podle Zákona č. 252/1997 Sb., o zemědělství).
- URL: <http://eagri.cz/public/app/SZR/EZP>

Registr plátců daně z přidané hodnoty – DPH

- správce: Finanční správa
- eviduje: fyzické nebo právnické osoby, které jsou plátcí DPH. Je tam uvedené, jestli je to spolehlivý nebo nespolehlivý plátec.
- URL: http://adisreg.mfcr.cz/cgi-bin/adis/idph/int_dp_prij.cgi

Registr plátců spotřebních a ekologických daní – SD

- správce: Celní správa
- eviduje: fyzické nebo právnické osoby, které jsou plátcí spotřebních nebo ekologických daní. Je tam uvedeno, za jaké produkty nebo činnosti tuto daň platí.
- URL: <http://www.celnisprava.cz/cz/aplikace/Stranky/SpdInternet.aspx?act=findspd>

Centrální registr dotací – IS CEDR III

- správce: Ministerstvo financí
- eviduje: informace o poskytnutých účelových dotacích ze státního rozpočtu.
- URL: <http://cedr.mfcr.cz/cedr3internetv416/CommonPages/ConditionPage.aspx?queryType=0>

Centrální evidence úpadců – CEU

- správce: Ministerstvo spravedlnosti
- eviduje: dlužníky, proti kterým bylo zahájeno konkurzní či vyrovnací řízení před 1. 1. 2008 (MS ČR, 2014a).
- URL: http://www.justice.cz/cgi-bin/sqw1250.cgi/upkuk/s_i8.sqw
- API: ano, pouze dávkový výstup o změnách v evidenci:
http://www.justice.cz/IS02/upkuk/ceu_davka6.html

Insolvenční rejstřík – IR

- správce: Ministerstvo spravedlnosti
- eviduje: dlužníky, proti kterým bylo zahájeno insolvenční řízení po 1. 1. 2008 a nebyli z rejstříku vyškrtnuti dle § 425 insolvenčního zákona (MS ČR, 2014a).
- URL: <https://isir.justice.cz/isir/common/index.do>
- API: ano, pouze hromadné stahování dat – popis:
https://isir.justice.cz/isir/help/Popis_WS.pdf

2.3.3 Některé důležité registry

Registr územní identifikace, adres a nemovitostí – RÚIAN

- správce: Český úřad zeměměřický a katastrální
- eviduje: územní prvky, územně evidenční jednotky, adresy, územní identifikace.
- URL – Veřejný dálkový přístup: <http://vdp.cuzk.cz/>
- API: ano – XML soubory obsahující adresy a územní členění:
<http://vdp.cuzk.cz/vdp/ruian/vymennyformat/vyhledej>

Katastr nemovitostí – KN

- správce: Český úřad zeměměřický a katastrální
- eviduje: údaje o nemovitostech (pozemky a stavby) – soupis a popis, geometrické a polohové určení. Také eviduje práva k těmto nemovitostem (ČÚZK, 2013).
- URL – Nahlížení do KN: <http://nahliznidokn.cuzk.cz/>
- API: ano – Dálkový přístup do KN: <https://katastr.cuzk.cz/uvod/>. Jedná se o placenou službu. Ing. Tomáš Chotěnovský přes e-mail sdělil (mimo jiné) následující: „Zřízení Dálkového přístupu, ani vedení zákaznického účtu není zpoplatněno. Účtována je pouze úplata za odebrané údaje, tedy ty které uživatel zobrazí na svém počítači/zařízení nebo je uloží do svého počítače/zařízení.“

Ověřování DIČ systémem VIES

- správce: Evropská komise
- funkce: VIES je elektronický systém, který umožňuje přenos informací týkajících se registrace k DPH v rámci Evropské unie (Evropská komise, 2014).
- URL: http://ec.europa.eu/taxation_customs/vies/vatRequest.html
- API: ano – SOAP, definice webové služby:
http://ec.europa.eu/taxation_customs/vies/checkVatService.wsdl

Obchodní věstník – OV

- správce: Ministerstvo vnitra
- eviduje: změny v obchodním rejstříku.
- URL: <http://ov.gov.cz/>

Věstník veřejných zakázek – VVZ

- správce: Ministerstvo pro místní rozvoj
- eviduje: základní informace o veřejných zakázkách, které jsou zadávány v souladu se Zákonem č. 137/2006 Sb., o veřejných zakázkách.
- URL: <http://www.vestnikverejnychzakazek.cz/cs/Searching>
- API: ano – je nutná registrace, info: <http://www.vestnikverejnychzakazek.cz/cs/PublishAForm/XMLInterfaceForISVZUS>

Seznam kvalifikovaných dodavatelů – SKD

- správce: Ministerstvo pro místní rozvoj
- eviduje: dodavatele, kteří splnili kvalifikaci podle § 53 (základní kvalifikační kritéria) a § 54 (profesní kvalifikační kritéria) zákona o veřejných zakázkách a splnění kvalifikace doložili ministerstvu příslušnými doklady a současně zaplatili správní poplatek (MMR ČR, 2014).
- URL: <http://www.isvz.cz/ISVZ/SKD/Filter.aspx?type=1>

2.3.4 Další užitečné registry

Seznamy regulovaných a registrovaných subjektů finančního trhu

- správce: Česká národní banka
- funkce: Poskytuje veřejnosti možnost ověřit si, zda jsou subjekty, se kterými se mají možnost setkat na českém finančním trhu, oprávněny k nabízení a poskytování finančních služeb.
- URL: https://apl.cnb.cz/apljerrsdad/JERRS.WEB09.DIRECT_FIND

Registr odcizených vozidel

- správce: Policie ČR
- eviduje: údaje o vozidlech a registračních značkách, jejichž odcizení bylo oznámeno Policii České republiky.
- URL hledání: <http://aplikace.policie.cz/patrani-vozidla/>
- Ministerstvo dopravy – odcizené registrační doklady a tabulky RZ: http://www.mdcz.cz/cs/Silnicni_doprava/rv_rzd.htm
- Občanská databáze odcizených vozidel: <http://www.odcizena-vozidla.cz>

Letecký rejstřík

- správce: Úřad pro civilní letectví
- eviduje: letadla (zahrnuje všechny létající stroje), jejichž provozovatelem je fyzická osoba s trvalým pobytem nebo právnická osoba se sídlem v České republice.
- URL hledání: <http://portal.caa.cz/letecky-rejstrik>
- Úřad spravuje ještě další rejstříky, viz <http://www.caa.cz/rejstriky>.

Registr smluv

- správce: Ministerstvo financí
- eviduje: záznamy (smlouvy, objednávky, finanční plnění), které byly zveřejněny orgány veřejné moci (dobrovolně).
- URL: výpis podle měsíců a let: <https://portal.gov.cz/portal/rejstriky/data/10013/>
- API: <https://portal.gov.cz/portal/rejstriky/data/10013/popis.html>

Monitor – informační portál

- správce: Ministerstvo financí
- funkce: Jak bylo sděleno v e-mailu od Komunikace SP: „Informační systém Monitor umožňuje volný přístup k rozpočtovým a účetním informacím ze všech úrovní státní správy a samosprávy. Uváděné informace pocházejí ze systému Státní pokladny (IISSP – Integrovaný informační systém státní pokladny) a Centrálního systému účetních informací (CSÚIS) a jsou aktualizovány čtvrtletně.“
- URL: <http://monitor.statnipokladna.cz/>
- API: ano; webové služby – XML: <http://monitor.statnipokladna.cz/2013/webove-sluzby/>, zdrojová data – CSV: <http://monitor.statnipokladna.cz/2013/data/>

Veřejný registr zpracování osobních údajů

- správce: Úřad pro ochranu osobních údajů
- eviduje: osoby, které zpracovávají osobní údaje. Umožňuje přesvědčit se, zda osoba splnila svou zákonnou povinnost takové zpracování oznámit Úřadu.
- URL: bohužel nefunguje (zobrazuje se prázdná stránka): <https://www.uoou.cz/verejny-registr-zpracovani-osobnich-udaju/ds-1513/p1=1513>

Databáze ochranných známek, patentů, užitných a průmyslových vzorů

- správce: Úřad průmyslového vlastnictví
- eviduje: ochranné známky, patenty, užitné vzory, průmyslové vzory – umožňuje provádět informativní rešerše.
- URL: <http://www.upv.cz/cs/sluzby-uradu/databaze-on-line.html>

Centrální registr administrativních budov – CRAB

- správce: Úřad pro zastupování státu ve věcech majetkových
- eviduje: administrativní objekty v majetku státu, jejich obsazenost a dislokaci státních zaměstnanců. Primárně jej využívají státní instituce. Veřejnost může pouze zobrazit nemovitosti nabízené k prodeji nebo pronájmu.
- URL: <http://crab.uzsvm.cz/majetek/vse/filtr/0-0-1-0-0/>

2.3.5 Další zajímavé registry

- Centrální evidence sbírek (CES) – Ministerstvo kultury:
<http://ces.mkcr.cz/cz/search.php>
- Integrovaný registr znečišťování (IRZ) – Ministerstvo životního prostředí:
<http://portal.cenia.cz/irz/>
- Databáze léků (RLL) – Státní ústav pro kontrolu léčiv:
<http://www.sukl.cz/modules/medication/search.php>

- Registr sčítacích obvodů a budov (RSO) – Český statistický úřad:
<http://registry.czso.cz/irso/home.jsp>
- Ústřední seznam ochrany přírody (ÚSOP) – Agentura ochrany přírody a krajiny ČR: <http://drusop.nature.cz/>
- Informační systém výzkumu, vývoje a inovací – Rada pro výzkum, vývoj a inovace – součástí je známý systém RIV: <http://www.isvav.cz/>
- Národní soustava kvalifikací (NSK) – MŠMT: <http://www.narodnikvalifikace.cz/vyber-kvalifikace/profesni-kvalifikace>
- Registr komunálních symbolů (RKS) – Parlament České republiky:
<http://rekos.psp.cz/vyhledani-symbolu>
- Státní odrůdová kniha České republiky (SOK) – Ústřední kontrolní a zkušební ústav zemědělský : <http://nou.ukzuz.cz/ido/index.html>
- Ústřední seznam kulturních památek ČR (USKP) – Ministerstvo kultury:
<http://monumnet.npu.cz/monumnet.php>
- Báňské registry – Státní báňská správa ČR:
<http://www.cbusts.cz/sekce-registry.aspx>
- Telekomunikační registry – Český telekomunikační úřad : <http://www.ctu.cz/ctu-online/vyhledavaci-databaze/prehled-vyhledavacich-databazi.html>
- Registrované subjekty SVS – Státní veterinární správa:
<http://eagri.cz/public/web/svs/portal/registrovane-subjekty/>
- Seznam provozovatelů vysílání – Rada pro rozhlasové a televizní vysílání: <http://www.rrtv.cz/cz/static/prehledy/seznamy-provozovatelu/index.htm>
- Adresář exportérů – vládní agentura CzechTrade:
<http://exporters.czechtrade.cz/cs/katalog-firem/>

2.3.6 Seznamy organizací

- Rejstřík církví a náboženských společností (RCNS) – Ministerstvo kultury:
http://www3.mkcr.cz/cns_internet/
- Registr zdravotnických zařízení (RZZ) – Ústav zdravotnických informací a statistiky: <https://snzr.uzis.cz/viewzz/rzz.htm>
- Seznam politických stran a hnutí (PSH) – Ministerstvo vnitra:
<http://aplikace.mvcr.cz/seznam-politickych-stran/>
- Rejstřík škol a školských zařízení (RŠ) – Ministerstvo školství, mládeže a tělovýchovy: <http://rejskol.msmt.cz/>

- Rejstřík veřejných výzkumných institucí (RVVI) – Ministerstvo školství, mládeže a tělovýchovy: <http://rvvi.msmt.cz/>

2.3.7 Evidence osob

Následující část obsahuje seznamy osob, které působí podle zvláštního právního předpisu nebo jsou sdružené v profesní komoře.

- Evidence znalců a tlumočnicků – Ministerstvo spravedlnosti
- Seznam notářů – Notářská komora České republiky
- Přehled soudců – Ministerstvo spravedlnosti
- Přehled státních zástupců – Ministerstvo spravedlnosti
- Seznam daňových poradců – Komora daňových poradců České republiky
- Přehled energetických specialistů – Ministerstvo průmyslu a obchodu
- Přehled insolvenčních správců – Ministerstvo spravedlnosti
- Seznam advokátů a koncipientů – Česká advokátní komora
- Seznam architektů – Česká komora architektů
- Seznam autorizovaných inženýrů a techniků – Česká komora autorizovaných inženýrů a techniků činných ve výstavbě
- Seznam soudních exekutorů – Exekutorská komora České republiky
- Seznam auditorů – Komora auditorů České republiky
- Seznam patentových zástupců – Komora patentových zástupců České republiky
- Seznam veterinářů – Komora veterinárních lékařů České republiky
- Seznam registrovaných lékařů – Česká lékařská komora

2.3.8 Registry dlužníků

Informace pro tuto část byly čerpány z (Wikipedia, 2014) a z webových stránek jednotlivých subjektů. Jedná se o selektivní rejstříky, ke kterým mají přístup pouze zákonem dané nebo registrované subjekty.

- Centrální registr úvěrů (CRÚ) – přístup mají banky
- Czech Banking Credit Bureau (CBCB) – banky a spořitelny
- Czech Non-Banking Credit Bureau (CNCB) – registrované finanční instituce
- SOLUS – každý může požádat o výpis, procházet údaje mohou pouze členové
- Centrální registr dlužníků (CERD) – pochybný subjekt²

²Registr není registrován Úřadem pro ochranu osobních údajů pro svou činnost. Dne 3. 6. 2010 byla Českou televizí zveřejněna reportáž zpochybňující jeho činnost. <http://www.ceskatelevize.cz/porady/1097429889-kerne-ovce/210452801080603/3772-valassko/?sekce=13&clanek=1930>

3 Modul pro získávání dat

V této kapitole je popsána tvorba modulu pro získávání dat. Nejprve jsou vybrány rejstříky, z kterých je žádoucí získávat data. Potom jsou vytvořeny testovací skripty, které mají ověřit, že lze z daných rejstříků data získávat. Následně jsou stanoveny požadavky na funkce modulu. V dalším kroku je modul navržen a implementován.

3.1 Vybrané rejstříky

Jak je možné vidět v kapitole 2, veřejných rejstříků existuje opravdu velké množství. Navrhnout řešení pro získávání dat ze všech těchto rejstříků nemá velkého smyslu. Prvním důvodem je vysoká časová náročnost na zpracování tolika zdrojů dat. Druhým a důležitějším důvodem je, že pouze malé množství rejstříků má praktické využití – poskytují užitečná, potřebná a v praxi využitelná data.

Pro výběr rejstříků, z kterých získávat data, byla použita tato kritéria:

- obsah (užitečnost) rejstříku
- dostupnost API, složitost přístupu bez API
- poplatek za přístup k rejstříku

Nakonec bylo rozhodnuto, že se pokusíme vytvořit řešení pro následující rejstříky:

- Obchodní rejstřík (+ další na rejstříkových soudech)
- Živnostenský rejstřík
- Registr ekonomických subjektů
- Registr plátců daně z přidané hodnoty
- Registr plátců spotřební daně
- Centrální registr dotací (IS CEDR III)
- Centrální evidence úpadců
- Insolvenční rejstřík
- Registr územní identifikace, adres a nemovitostí
- Ověřování DIČ systémem VIES

3.1.1 Testovací skripty

V dalším kroku bylo nutné zjistit, zda lze z daných rejstříků získávat data automaticky. Pro tvorbu skriptů byl vybrán jazyk PHP. Důvodem je, že autor práce je s tímto jazykem dobře seznámen a jazyk poskytuje vše potřebné pro požadovanou činnost. Velkou výhodou je, že PHP je dostupný na drtivé většině webhostingů, takže modul (který bude taky v PHP) bude moci využít široký okruh uživatelů.

Postup byl takový, že pro každý rejstřík byl vytvořen testovací skript, který měl za úkol provést stažení dat z rejstříku. Podrobnější informace o způsobu získávání dat jsou uvedeny v části 3.5.2. Níže jsou uvedeny výsledky testování – tj. jestli je možné data z daného rejstříku získat nebo ne.

- Obchodní rejstřík (OR) – ano
- Živnostenský rejstřík – ano
- Registr ekonomických subjektů – ano, ale občas nastane výrazné zpomalení
- Registr plátců daně z přidané hodnoty – ano
- Registr plátců spotřební daně – NE
- Centrální registr dotací (IS CEDR III) – NE
- Centrální evidence úpadců – ano
- Insolvenční rejstřík (IR) – ano
- Registr územní identifikace, adres a nemovitostí – ano
- Ověřování DIČ systémem VIES – ano

Jak je vidět, až na dva registry bylo testování úspěšné. RES občas funguje nespolehlivě, takže jeho použití je diskutabilní, nicméně do modulu integrován bude.

Aplikace registrů mají stanovené provozní podmínky. Je tam obvykle uvedeno, kolik požadavků může uživatel odeslat za určitý čas. Např. v případě IR a OR se jedná o 3000 požadavků za 1 den a 50 požadavků za 1 minutu. Pokud uživatel tyto limity překročí, může mu být k aplikaci zakázán přístup.

3.2 ARES

Zde je nutné zastavit se u webové aplikace ARES na <http://wwwinfo.mfcr.cz/ares/ares.html.cz>. Tento „Administrativní registr ekonomických subjektů“ není veřejným rejstříkem v pravém slova smyslu. „Je to IS, který zpřístupňuje veřejné údaje o ekonomických subjektech z informačních systémů (zdrojů) veřejné správy“ (MF ČR, 2013). Jedná se tedy o aplikaci, která získává data z vybraných zdrojů (rejstříků), a umožňuje v nich vyhledávat.

3.2.1 Popis

Aplikace má dvě hlavní funkce:

1. umožňuje interaktivní práci přes webový formulář
2. poskytuje veřejné API, které je možné využít k získání odpovědi v XML

Příklad volání API metodou GET: standardní dotaz – vyhledání podle IČO: http://wwwinfo.mfcr.cz/cgi-bin/ares/darv_std.cgi?ico=27074358. Server pošle textovou odpověď ve formátu XML, ve které lze najít základní informace o nalezeném subjektu. Jsou to: typ prioritního registru, datum vzniku, právní forma, obchodní firma, IČO, adresa, kód FÚ, příznaky registrů.

Volání na registr má tedy tento tvar:

`http://wwwinfo.mfcr.cz/cgi-bin/ares/darv_<typ>.cgi?parametry`

Do parametrů je možné zadat: IČO – ico nebo obchodní firmu – obchodni_firma

Jsou dostupná také další volání. Jejich popis je možné vyhledat na (MF ČR, 2013) v sekci XML služby. Typy dotazů:

- Basic – výpis z více registrů: bas
- Seznam registrací subjektu v OR, RES a RŽP: reg
- Úplný dotaz na registr: or, rzp, res, szr, ceu, cns, oss, rzz, psh, sko
- Dotaz na standardizovanou adresu:
`darv_adr.cgi?obec=Svitavy&ulice=Nerudova&xml=0&typ=A`
- Přehled změn ekonomických subjektů v ARES:
`darv_zm.cgi?cislo_zdroje=2&cislo_davky_od=2&cislo_davky_do=3`
- Přehled fyzických osob – nelze vyhledávat podle IČO:
`ares_fo.cgi?jmeno=Kalivoda&obec=Praha&cis_or=14&maxpoc=200&xml=0`
- Přehled ekonomických subjektů – v jakých registrech je subjekt zapsán: es

Na doplňkové registry je k dispozici pouze odkaz, který vede na web registru s detailem subjektu. Jedná se o tyto registry: DPH, SD, CEDR, CEU, IR.

3.2.2 Nevýhody

Mohlo by se tedy zdát, že řešení pro získávání dat z veřejných rejstříků již existuje, a do jisté míry to je pravda. Nicméně po bližším prozkoumání situace je možné konstatovat, že ARES neřeší všechny požadavky a má několik nevýhod.

Hlavním problémem je vysoká odezva aplikace. Ta je znát jak v interaktivní části, tak ve veřejném API. Pro ARES byl vytvořen testovací skript a byla provedena řada měření. Pro dotaz Standard se odezva pohybovala v průměru okolo 6 sekund. Nejnižší čas 2–3 sekundy byl dosažen jen občas. Výsledná doba odezvy záleží na momentálním vytížení serveru. Někdy je možné dostat odpověď za dvě sekundy, jindy může přijít za 10 (nebo i více) sekund. Odezva je ještě vyšší například u výpisu z obchodního rejstříku, kde se pohybovala okolo 30 sekund.

Při psaní práce se odezva zdá být rychlejší než při testování, nicméně stále se jedná o vysoké hodnoty. Tento názor sdílí i další lidé – např. Dvořák (2013) zvolil kvůli rychlosti možnost parsování výstupu z webu `http://or.justice.cz`. Ve shodě s vedoucí práce bylo konstatováno, že pro optimální a interaktivní práci uživatele je odezva okolo 5–10 sekund příliš vysoká a bude tedy lepší vytvořit vlastní řešení.

Další nevýhodou je, že výše uvedené doplňkové registry nejsou přístupné v rámci výsledného XML souboru, takže pro získání údajů z nich je stejně potřeba data nějak získat a zpracovat.

3.3 Požadavky

Zde je uvedeno, jaké funkce má výsledný modul mít a jaké požadavky musí splňovat.

3.3.1 Funkční požadavky

Úkolem modulu je poskytnout volající aplikaci data z veřejného rejstříku. A to ve formátu, se kterým může ihned pracovat. Za nejlepší řešení se dá považovat asociativní pole (klíč => hodnota).

Modul má tyto funkce:

1. Získat základní informace o subjektu podle zadaného IČO – název, adresa, datum vzniku, právní forma.
2. Vyhledat subjekt(y) podle zadaného názvu a získat o nich základní informace (viz výše + IČO) – bude nalezeno jeden nebo více subjektů.
3. Podle zadaného IČO zjistit přítomnost a hodnotu záznamu ve speciálních registrech (DPH, CEU, IR).
4. Zkontrolovat platnost adresy – přes RUIAN.
5. Ověřit, zda je subjekt s daným DIČ plátcem DPH ve své zemi – přes VIES, funguje v celé Evropské unii.
6. Přistoupit ke speciálním údajům konkrétního registru.

Pro funkce 1 a 2 jsou využity: Živnostenský rejstřík (RŽP), Obchodní rejstřík (OR), ARES. Funkce probíhá tak, že se nejdříve podívá do RŽP. Pokud tam subjekt najde, vrátí data. Pokud ho nenajde, podívá se do OR. Pokud ho nenajde ani tam, podívá se do ARES.

Modul neukládá ani nezaznamenává žádná data (např. logy) – v případě potřeby si toto musí zařídit klientská aplikace.

3.3.2 Nefunkční požadavky

- Modul je vytvořen v programovacím jazyce PHP. Nepoužívá žádná speciální rozšíření – musí fungovat na co největším počtu základních konfigurací.
- Modul je vytvořen pomocí objektově orientovaného přístupu. Součástí je jedna hlavní třída pro společné funkce a pro každý registr existuje zvláštní třída, kterou lze také samostatně použít.

Modul lze použít pouze v jiném skriptu PHP. Neposkytuje žádné GUI.

3.4 Návrh modulu

Na základě požadavků je nyní potřeba navrhnout strukturu a formu modulu.

3.4.1 Adresářová struktura

Modul je umístěn ve složce, dostupné aplikaci, která chce modul využít.

```
CzechRegisters/ ..... kořenová složka modulu
├── helpers/ ..... pomocné třídy a funkce
├── libs/ ..... programové knihovny, které modul využívá
├── registers/ ..... třídy jednotlivých registrů
├── testing/ ..... skripty pro vyzkoušení modulu
├── core.php ..... skript pro autoloading tříd
└── CzechRegisters.php ..... hlavní rozhraní pro interakci s modulem
```

Pro použití v jiné aplikaci je nutné načíst všechny potřebné třídy – tj. ty ze složek helpers, libs a registers + hlavní třídu CzechRegisters. To si zajistí aplikace sama svým autoloadingem nebo je možné využít soubor core.php. Ten je použit v testovacím skriptu regs-test.php.

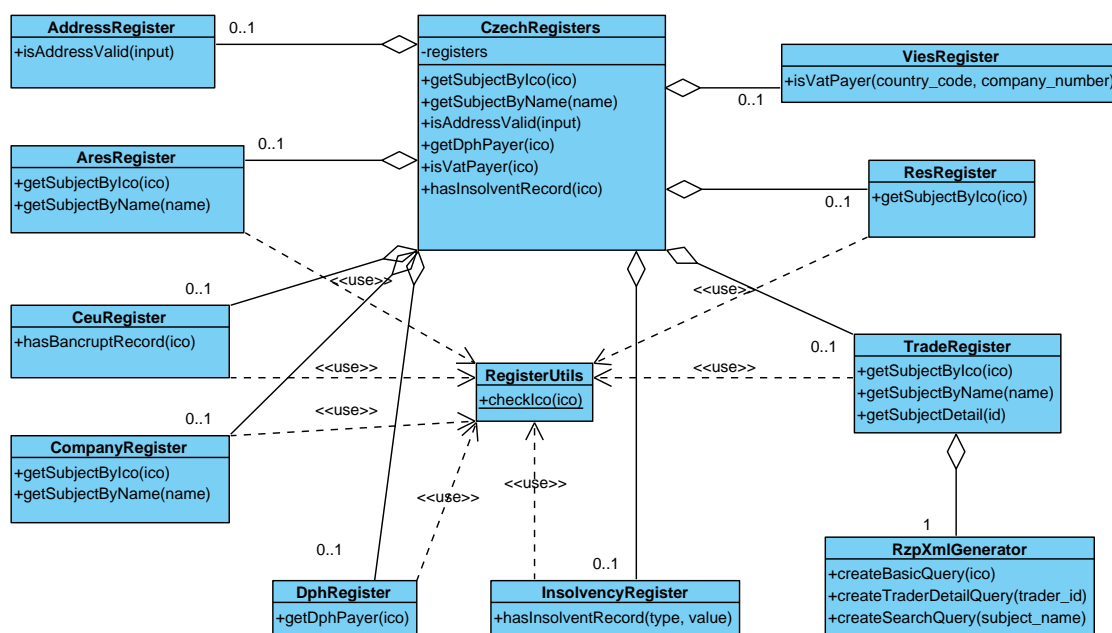
Následně aplikace vytvoří objekt třídy CzechRegisters, na kterém může volat základní metody. Pokud bude potřeba využít specifické funkce nějakého registru, aplikace vytvoří objekt dané třídy a může s ním potom pracovat.

3.4.2 Diagram tříd

Modul nemá příliš komplikovanou strukturu – obsahuje hlavní třídu, pomocné třídy a třídy registrů. Třídy ve složce libs zde nejsou uvedeny.

Třída je koncept z OOP – abstrakce objektů, které mají společné chování a vlastnosti. Má deklaraci, atributy, operace. Objekt (abstrakce předmětu/problému z reálného světa) je instance třídy – obsahuje její atributy a operace. Konkrétní hodnoty atributů určují tzv. *stav objektu* a objekt má pouze metody, definované ve třídě.

Strukturu znázorníme pomocí jednoduchého diagramu tříd (Class diagram). Tento je zapsán pomocí notace UML – je to standardní jazyk pro vizuální modelování prvků projektu, které se zabývají vývojem softwaru. Standard spravuje organizační skupina OMG. UML obsahuje 3 základní stavební kameny: artefakty (prvky), vztahy a diagramy (Rábová, 2008, s. 59).



Obrázek 2: Diagram tříd (modul pro získávání dat)

3.5 Implementace modulu

Na základě návrhu bylo přikročeno k implementaci. Ta se skládala z několika fází:

1. Vytvoření souboru core.php – autoloading tříd
2. Vytvoření souboru regs-test.php – testovací skript pro vyzkoušení tříd
3. Tvorba tříd pro jednotlivé registry. Základem pro ně byly testovací skripty, vytvořené ve fázi výběru rejstříků.
4. Vytvoření třídy CzechRegisters – hlavní rozhraní pro komunikaci s modulem.
5. Úprava a testování všech tříd až do finální podoby.

3.5.1 Soubor core.php

Pokud chceme ve skriptu využít třídu, která není interní třídou PHP, musíme její definici do skriptu nějak „dostat“. K tomu lze využít funkci `require`, která jako parametr přijímá cestu k souboru, který chceme vložit. Pro třídy ovšem musíme využít variantu `require_once`, protože ta nám zajistí, že třída nebude deklarována vícekrát (to je zakázáno).

Tímto způsobem lze načtení souborů zajistit – nad každou třídu musíme napsat `require` pro všechny knihovny, a do našeho skriptu `require` pro všechny použité třídy. Je ale možné, že se vyskytnou nějaké problémy a hlavně to je velmi pracné a zdlou-

havé. Lepším řešením je využít vestavěnou *magickou*³ funkci PHP `autoload`, která se spustí v případě, že chceme použít ještě nedefinovanou třídu. Jméno této třídy je vloženo do parametru funkce. Funkci musíme definovat a v jejím těle zajistit, aby byl pomocí `require` načten soubor se stejným jménem jako daná třída.

Před definicí funkce `autoload` musíme pomocí funkce `set_include_path` nastavit adresáře, ve kterých bude `require` hledat dané třídy.

Pro celé toto řešení byl inspirací článek (NABITO.NET, 2008). Obsah souboru `core.php` je umístěn v příloze C.

3.5.2 Princip získání dat z registru

Nyní můžeme uvést, jakým způsobem vlastně získáme a následně zpracujeme data z registrů. Postup se liší v závislosti na tom, jaké rozhraní webová aplikace daného registru nabízí. Možnosti jsou tyto:

1. API – přístupné přes GET nebo POST HTTP požadavek
2. formulář odesílající data metodou GET
3. formulář odesílající data metodou POST

API rozhraní

Pokud je možné použít API, tak máme situaci ulehčenou. Po zaslání požadavku dostaneme odpověď ve standardním formátu (obvykle XML), kterou je možné lehce zpracovat. Pro zpracování XML v PHP lze využít standardní knihovny jazyka: SimpleXML, SAX, DOM a XMLReader. Výběr vhodného rozhraní záleží na povaze dat, která chceme číst, a na tom, jak je potřebujeme zpracovat (Kosek, 2009, s. 16).

Pro vytvoření XML souboru (kvůli zaslání na server) bylo použito rozhraní DOM, které se pro tuto činnost nejvíce hodí. Stejně rozhraní bylo využito také pro čtení došlé odpovědi, protože umí dobře pracovat s jmennými prostory elementů, které se v odpovědi většinou vyskytují. Jelikož mají výsledné soubory malou velikost, nebylo nutné použít pro jejich čtení „streamovací“ rozhraní SAX nebo XMLReader.

HTML stránka

Pokud daný registr API nenabízí, znamená to pro nás práci navíc. Dostaneme totiž odpověď ve formě běžné HTML stránky, jejíž obsah musíme prohledat a získat data, která nás zajímají. HTML stránka obvykle není tzv. *well-formed XML document*⁴. Proto není možné pro její zpracování použít standardní knihovny pro práci s XML.

³Metody tříd začínající na `__` jsou považovány za magické. Ty výchozí obsahuje každá třída, jejich seznam lze nalézt na (The PHP Group, 2014b). Spustí se automaticky při určitých událostech.

⁴Tj. takový, který má právě jeden kořenový element (tag), všechny otevírací tagy mají odpovídající zavírací tagy, a všechny tagy jsou korektně vnořené. Atributy elementů musí mít unikátní jména, všechny deklarace jmenných prostorů musí mít jiné předpony a všechny použité předpony jmenných prostorů musí mít odpovídající deklaraci (Skonnard, 2001, s. 13).

Resp. lze použít DOM a funkci `DOMDocument::loadHTML`, ale následná práce s objektem dokumentu je potom v případě HTML vstupu velmi těžkopádná.

Z tohoto důvodu byla použita pro práci s HTML externí knihovna PHP Simple HTML DOM Parser (verze 1.5). Jak se píše na jejím webu (Chen, 2013), je to „HTML DOM parser napsaný v PHP5+, který vám umožní manipulovat s HTML velmi jednoduchým způsobem“. Podporuje nevalidní HTML a umožňuje výběr elementů na HTML stránce pomocí selektorů (jako javascriptová knihovna jQuery). Tato knihovna poskytuje konstruktor (načte a zpracuje dokument – z URL, souboru nebo řetězce) a několik základních metod. Tou hlavní je `find`, která v načteném dokumentu najde elementy, splňující podmínku zadanou v parametru. Výsledek je uložen do proměnné – pole nalezených elementů. Následně lze přistoupit ke konkrétnímu elementu a získat jeho obsah nebo atributy.

Příklad práce s knihovnou:

```
$html = file_get_html('http://www.seznam.cz/');  
// najít na stránce všechny obrázky a vypsát jejich adresu  
foreach($html->find('img') as $element)  
    echo $element->src . '<br>';  
// vytvoření objektu html přes OOP  
$html = new simple_html_dom();  
$html->load_file('http://www.seznam.cz/');  
$first = $html->find('img',0); // najít první obrázek
```

Požadavek přes GET

Pokud aplikace registru poskytuje veřejné API, ke kterému lze přistoupit skrz požadavek GET, je naše práce snadná. Stačí sestavit správné URL s požadovanými parametry, na to přistoupit a zpracovat odpověď serveru. Využijeme funkci `file_get_contents` a jako parametr zadáme požadované URL. Funkce pošle GET požadavek na tuto adresu a vrátí nám odpověď serveru, kterou potom můžeme snadno zpracovat. Funkce též přijímá jako parametr cestu k lokálnímu souboru.

Pokud web registru nenabízí API, ale pouze HTML formulář, musíme vytvořit požadavek GET, který bude emulovat odeslání formuláře. Nejdříve je ovšem nutné zjistit, na jakou URL se formulář odesílá a s jakými parametry. K tomu lze využít postup popsáný v části „Požadavek přes POST“ nebo se jednoduše podívat na výslednou URL, která vznikne po odeslání formuláře.

Po zjištění tvaru URL využijeme opět funkci `file_get_contents` a získáme tak obsah výsledné HTML stránky. Potom je nutné použít již zmíněnou knihovnu PHP Simple HTML DOM Parser a najít požadovaná data. Pro přesnost uvedme, že vlastně stačí použít funkci této knihovny pro načtení HTML, která interně využívá právě výše zmíněnou funkci.

Požadavek přes POST

Pokud web nabízí API, ale je nutné poslat požadavek přes POST, musíme využít knihovnu cURL – Client URL Library. Tato je postavená na knihovně libcurl, která nám „umožňuje připojit se k mnoha typům serverů pomocí mnoha typů protokolů“. Podporuje např. protokoly HTTP, HTTPS, FTP, GOPHER a TELNET. Podporuje také SSL certifikáty, HTTP POST, HTTP PUT, FTP uploading, HTTP form based upload, proxies, cookies, user+password autentizaci aj. (The PHP Group, 2014a).

Práce s knihovnou je poměrně jednoduchá. Obsahuje tyto základní kroky:

1. příprava dat – URL a parametry
2. inicializace cURL session
3. nastavení možností pro transfer
4. provedení požadavku (session)
5. zavření session
6. zpracování odpovědi serveru

Příklad jak pracovat s knihovnou se nachází v příloze D. Pokud je potřeba odeslat soubor, je nutné před cestu k němu uvést znak @. Toto je zároveň způsob, jak pracovat s rejstříkem RŽP. Pokud chceme vytvářet soubory dynamicky, je to možné – použijeme funkci `tempnam` pro vytvoření dočasného souboru s unikátním jménem, zapíšeme do něj požadovaný text, odešleme soubor a nakonec ho smažeme. Tento postup využívá třída modulu TradeRegister.

Pokud web nenabízí API, ale pouze formulář odesílající požadavek přes POST, použijeme také knihovnu cURL. Je potřeba zjistit, na jakou URL se formulář odesílá a s jakými parametry. K tomu lze využít dva způsoby:

1. Zobrazit zdrojový kód stránky a lokalizovat v něm formulář. URL se nachází v atributu `action` elementu `form`. Parametry představují položky (tagy `input`) – jejich názvy obsahuje atribut `name`.
2. Použít rozšíření webové prohlížeče (např. Firebug pro Firefox), kde v záložce Sít můžeme sledovat, na jakou URL a s jakými parametry byl formulář odeslán.

Je vhodné odeslat požadavek se všemi nalezenými parametry. Předejde se tak podezření, že formulář neodeslal uživatel, ale počítačový program.

3.5.3 Obecný popis třídy registru

Jak vyplývá z části Návrh modulu, pro každý registr byla vytvořena třída, která s ním pracuje. Všechny mají podobnou strukturu: Obsahují jednu (nebo více) veřejných metod, které může volat klientský program. Veřejná metoda obvykle používá několik privátních metod, které dohromady získají požadovaná data. Tato jsou nakonec vrácena volajícímu programu jako asociativní pole.

Pokud nebylo (podle zadaných podmínek – parametrů) nic nalezeno, vrátí metoda hodnotu `false`. Pokud jsou zadány špatné parametry nebo se vyskytne neo-

čekávaná chyba (např. nelze se připojit k serveru rejstříku), vyhodí objekt výjimku, kterou by měl klientský program zpracovat. Pro špatně zadané parametry byla vytvořena výjimka *RegInvalidArgumentException*. Lze v ní např. logovat neplatná volání metod.

Veřejné metody jsou okomentované podle standardu PHPDoc (popis metody, parametry, návratová hodnota, event. výjimka).

Třída *RegisterUtils* obsahuje několik užitečných statických metod.

Třída CzechRegisters

Tato hlavní třída pro práci s modulem obsahuje několik veřejných metod. Tyto v podstatě odpovídají funkčním požadavkům uvedeným v části 3.3.1. Níže jsou vypsané názvy metod včetně parametrů. Všechny vracejí pole nebo **false**.

- `getSubjectByIco(ico)`
- `getSubjectByName(name)`
- `isAddressValid(input)`
- `getDphPayer(ico)`
- `isVatPayer(country code, company number)`
- `hasInsolventRecord(ico)`

Ve třídě je pro přístup k registrům použit tzv. *lazy loading*. Ten zajišťuje, že se zbytečně nevytváří objekty, které nejsou potřeba. Toto chování umožňuje speciální privátní metoda třídy, přes kterou se v jejím těle k registrům přistupuje. Tento přístup se nazývá *lazy initialization* (Fowler, 2003, s. 200).

```
// atribut s odkazy na instance tříd registrů
private $registers = array();
// funkce pro lazy loading
private function getRegister($name)
{
    $name = ucfirst($name);
    if(array_key_exists($name, $this->registers)) {
        return $this->registers[$name];
    }
    $className = 'CzechRegisters\\'. $name. 'Register';
    $this->registers[$name] = new $className();
    return $this->registers[$name];
}
// následné použití
$subj = $this->getRegister('Trade')->getSubjectByIco($ico);
```

3.5.4 Popis jednotlivých tříd registrů

V této části jsou v krátkosti představeny všechny třídy registrů.

AddressRegister

- zdroj dat: RUIAN = Registr územní identifikace, adres a nemovitostí
- způsob získání dat: GET požadavek a zpracování HTML
- základní URL: <http://vdp.cuzk.cz/vdp/ruian/overeniadresy/vyhledej>
- hlavní metoda: `isAddressValid(adresa)`
- parametry: `adresa` = ulice, č. popisné, č. orientační, město, městská část, PSČ
- funkce: Kontroluje, jestli je zadaná adresa správná (tj. existuje a je unikátní).

AresRegister

- zdroj dat: ARES = Administrativní registr ekonomických subjektů
- způsob získání dat: GET požadavek a zpracování XML
- URL: http://wwwinfo.mfcr.cz/cgi-bin/ares/darv_std.cgi
- hlavní metody: `getSubjectByIco(ico)`, `getSubjectByName(name)`
- funkce: Získá základní informace o subjektu.

CeuRegister

- zdroj dat: CEU = Centrální Evidence Úpadců
- způsob získání dat: GET požadavek a zpracování HTML
- URL: http://www.justice.cz/cgi-bin/sqw1250.cgi/upkuk/s_i6.sqw
- hlavní metoda: `hasBancruptRecord(ico)`
- funkce: Zjistí, jestli má zadaný subjekt záznam o insolvenčním řízení (do 1. 1. 2008). Pokud ano, vrátí podrobnosti.

CompanyRegister

- zdroj dat: Obchodní rejstřík + další rejstříky na soudech
- způsob získání dat: GET požadavek a zpracování HTML
- URL: <https://or.justice.cz/ias/ui/rejstrik-dotaz>
- hlavní metody: `getSubjectByIco(ico)`, `getSubjectByName(name)`
- funkce: Získá základní informace o subjektu.

DphRegister

- zdroj dat: DPH – Registr plátců daně z přidané hodnoty
- způsob získání dat: POST požadavek a zpracování HTML
- URL: `http://adisreg.mfcr.cz/cgi-bin/adis/idph/int_dp_prij.cgi`
- hlavní metoda: `getDphPayer(ico)`
- funkce: Zjistí, jestli je subjekt plátce DPH v ČR. Pokud je, vrátí jestli je spolehlivý nebo nespolehlivý. Kvůli omezení serveru trvá metoda min. 2 sekundy.

InsolvencyRegister

- zdroj dat: IR = Insolvenční rejstřík
- způsob získání dat: GET požadavek a zpracování HTML
- URL: `https://isir.justice.cz/isir/ueu/vysledek_lustrace.do`
- hlavní metoda: `hasInsolventRecord(type, value)`
- parametry: `type` = IČO nebo rodné číslo, `value` = hodnota daného typu
- funkce: Zjistí, jestli má zadaný subjekt záznam o insolvenčním řízení (po 1. 1. 2008). Pokud ano, vrátí podrobnosti.

ResRegister

- zdroj dat: RES = Registr ekonomických subjektů
- způsob získání dat: POST požadavek a zpracování HTML
- URL: `http://registry.czso.cz/irsw/hledat.jsp`
- hlavní metoda: `getSubjectByIco(ico)`
- funkce: Získá statistické informace o subjektu – např. kategorie počtu zaměstnanců. Občas je zdrojový web velmi zpomalený a dotaz trvá až 1 minutu.

TradeRegister

- zdroj dat: RZP = Registr živnostenského podnikání
- způsob získání dat: POST požadavek a zpracování XML
- URL: `http://www.rzp.cz/cgi-bin/aps_cacheWEB.sh`
- metody: `getSubjectByIco(ico)`, `getSubjectByName(name)`, `getSubjectDetail(id)`
- funkce: Získá základní resp. detailní informace o subjektu.

U tohoto registru bylo nutné vytvořit podle XSD schématu, uvedeného v dokumentaci, platný XML dotaz (nebyl dostupný žádný příklad). Naštěstí dotaz nemá příliš složitou strukturu. Základem je element `Kriteria`, který obsahuje elementy určující podmínky, podle kterých bude probíhat hledání. Dotazy se nacházejí v příloze E.

Při spouštění na lokálním PC probíhalo vše v pořádku. Na hostingovém serveru byla funkce nějakou dobu velmi pomalá (doba průběhu asi 60 sekund). Zřejmě to bylo kvůli vytváření dočasného XML souboru. Nyní je vše v pořádku, nicméně pro jistotu je na veřejný hosting lepší použít jiné registry. Pořadí registrů, použitých ve třídě `CzechRegisters`, určuje její proměnná `callOrder` – obsahuje pole názvů registrů.

ViesRegister

- zdroj dat: VIES – Ověřování DIČ pro účely DPH
- způsob získání dat: SOAP
- URL: `http://ec.europa.eu/taxation_customs/vies/checkVatService.wsdl`
- hlavní metoda: `isVatPayer(country code, company number)`
- funkce: Zjistí, jestli je subjekt ve své zemi registrován jako plátce DPH. Funguje pro státy EU. Metoda může vracet i další informace jako název nebo adresu firmy, ale to se liší podle země původu (např. německé firmy detaily neobsahují).

Pro komunikaci se serverem byla využita třída `SoapClient`. Na lokálním PC stačilo do jejího konstruktoru zadat URL webové služby a poté zavolat metodu `checkVat`. Na hostingu se ovšem připojení nezdařilo. Důvodem zřejmě bylo, že používá IPv6. Pro zprovoznění bylo nutné přidat do konstruktoru pole `options` s klíčem `user_agent` a libovolnou hodnotou.

3.5.5 Použití modulu

Použití je popsáno už v části 3.4.1. Pro použití v jiné aplikaci je nutné načíst všechny potřebné třídy. To si zajistí aplikace sama svým autoloadingem nebo je možné využít soubor `core.php`. Ten je použit v testovacím skriptu `regs-test.php`. Následně aplikace vytvoří instanci třídy `CzechRegisters`, na které může volat hlavní metody. Je také možné vytvořit instanci konkrétní třídy registru a pracovat s ní.

Doba získání dat se pohybuje okolo 1 sekundy (0,5–1,5) – záleží na konkrétním registru. To se dá považovat za přijatelnou hodnotu pro interaktivní práci.

Níže je uveden příklad použití:

```
require( 'core.php' );  
$ir = new InsolvencyRegister();  
var_dump( $ir->hasInsolventRecord( 'ico', '27074358' ) );  
$cr = new CzechRegisters();  
var_dump( $cr->getSubjectByIco( '27074358' ) );
```

4 Generování standardních typů dokumentů

Tato kapitola práce se zabývá systémem pro generování standardních typů dokumentů. Tento je vytvořen jako webová aplikace. Postupně je vysvětlen princip aplikace, vytvořen model dokumentu a stanoveny požadavky na aplikaci. Tato je následně navrhována a implementována.

4.1 Metodika

Životní cyklus vývoje softwaru obsahuje zhruba tyto etapy:

1. specifikace problému – definice požadavků
2. analýza a návrh
3. implementace
4. zavedení a testování
5. provoz a údržba

Tyto kroky v podstatě odpovídají Vodopádovému modelu životního cyklu, ve kterém je navíc během vývoje možné specifikovat nové požadavky a vrátit se tak do bodu 2. Tento tzv. *Waterfall model* je vlastně první průkopnický model SW inženýrství. I když je v dnešní době už překonaný, jeho základní struktura se stále využívá i v novějších metodikách (Rábová, 2008, s. 40).

Není použita žádná speciální metodika vývoje (např. RUP), jelikož cílová aplikace není příliš rozsáhlá. Základem pro použitou metodiku je výše uvedený vodopádový model. Krok dva obsahuje návrh datového modelu, uživatelského rozhraní a základních tříd.

4.2 Princip aplikace

Účelem aplikace je generovat standardní typy dokumentů. Konkrétní zaměření bylo stanoveno na smlouvy. Předpokládáme, že aplikace bude implementovaná jako webová a bude vytvořená v jazyce PHP.

Administrátor spravuje typy dokumentů a uživatele.

Uživatel vytváří a edituje dokumenty, generuje PDF.

4.2.1 Model dokumentu

Nejdříve je nutné stanovit, jakou formu bude mít onen standardní typ dokumentu (šablona). Resp. v jaké souborovém formátu bude uložen. Možnosti jsou široké: HTML, RTF, DOCX, XML, \LaTeX aj. Další možností je uložení přímo v DBMS.

Jako nejvhodnější se jeví obecný formát XML. Půjde tak snadno oddělit obsah od formy a je to obyčejný textový soubor. Možnou nevýhodou je, že vytváření šablon

dokumentů nebude probíhat v žádném WYSIWYG⁵ editoru a bude nutné dodržet striktní syntaxi XML. Nicméně základní (vzorová) šablona bude již vytvořená, takže administrátor pouze upraví text a některé hodnoty a rovnou může vytvořit nový typ dokumentu. Naopak výhodou takového modelu je, že k editaci a vytvoření šablony bude stačit jakýkoliv textový editor a šablona bude mít malou velikost.

Byla vytvořena výchozí šablona dokumentu – smlouva. Obsahuje hlavní oddíly jako název, smluvní strany, články a podpis. Tato a několik dalších šablon (pro konkrétní typy smluv) se nacházejí v elektronické příloze:
`docugen-app/sandbox/www/doc-uploads/doctypes`.

4.2.2 Základní princip

Z hlediska koncového uživatele funguje aplikace následovně:

1. Uživatel zvolí typ dokumentu – již vytvořená šablona v XML.
2. Šablona se přemění ve formulář, uživatel zadá nové hodnoty a odešle ho.
3. Hodnoty z formuláře se uloží do XML souboru.
4. Uživatel chce vygenerovat dokument – XML soubor se zpracuje a vygeneruje se PDF.

4.2.3 Bod 2 – vytvoření formuláře

Pro transformaci šablony na formulář bylo uvažováno nad dvěma možnostmi:

- zpracovat XML soubor pomocí XSLT a převést jej do HTML;
- zpracovat XML soubor pomocí PHP, získat hodnoty a vytvořit formulář.

Na první pohled se zdál jako lepší způsob ten přes XSLT. Dosáhlo by se tak nezávislosti na jazyku, v jakém je aplikace napsána. Po další úvaze byl ovšem objeven zásadní problém. V dnešní době se při vývoji webových aplikací téměř výlučně používá nějaký aplikační framework – tj. připravená množina zdrojového kódu a postupů, podle které lze vytvořit kompletní aplikaci (Lecky-Thompson, 2009, s. 355). Framework má obvykle svůj způsob, jak vytvářet formuláře, a vygenerované HTML je poměrně specifické. Takže při použití XSLT by bylo nutné dopředu zvolit pro jaký framework HTML generovat a zapsat tuto transformaci do stylu. V některých případech by to možná ani nebylo proveditelné.

Z tohoto důvodu byla zvolena druhá možnost, tj. zpracování přes PHP a vytvoření formuláře pomocí daného frameworku. Proces získání dat z XML je obecně použitelný, následné vygenerování formuláře je specifické pro daný framework. Tento způsob umožní co nejefektivnější zpracování a poskytne uživateli co nejlepší výsledek.

⁵What You See Is What You Get – vizuální editor

4.2.4 Bod 4 – generování PDF

Pro vygenerování PDF z XML lze zvažovat dvě možnosti:

- zpracovat XML, vložit data do TEX souboru a pomocí L^AT_EX vygenerovat PDF;
- zpracovat XML a pro vytvoření PDF použít nějakou PHP knihovnu.

Opět se zdálo, že lepší bude první možnost. L^AT_EX je rozšířená nadstavba nad systémem počítačové sazby T_EX a umožňuje tvorbu estetických a technicky kvalitních dokumentů. Jeho využití pro generování PDF je tedy přirozené. Nevýhodou ovšem je, že tento systém musí být nainstalovaný na serveru, na kterém poběží výsledná aplikace. Navíc L^AT_EX obsahuje obrovské množství balíčků a konečná velikost instalace může přesáhnout i 3 GB. Výsledná aplikace by měla být přístupná co největšímu počtu uživatelů a běžet na co největším počtu konfigurací serverů, z nichž drtivá většina (např. běžné webhostingové plány) nemá L^AT_EX nainstalovaný.

Z tohoto důvodu je nutné použít druhou možnost, tj. PHP knihovnu pro generování PDF. Příprava dokumentu bude poněkud složitější a výsledek zřejmě nebude tak dokonalý, ale na druhou stranu bude aplikace fungovat na libovolném serveru s podporou PHP.

4.3 Umístění aplikace v globální architektuře

Výsledná aplikace může být součástí informačního systému nějaké organizace. Jakou roli bude hrát v její globální architektuře IS/ICT? Můžeme připomenout, že to je „hrubý návrh IS/ICT, který zachycuje jednotlivé komponenty IS/ICT a jejich vazby“. Rozlišujeme v ní vertikální dimenzi (hierarchické členění managementu do tří úrovní) a horizontální dimenzi (podnikové útvary) (Rábová, 2008, s. 16).

Z hlediska horizontální dimenze lze uvažovat celou řadu oddělení (právní, účetní, personální apod). Záleží na konkrétním typu dokumentu.

Umístění ve vertikální dimenzi je také nejednoznačné. Teoreticky ji mohou využívat všechny úrovně managementu (operativní, střední, vrcholový), ale pokud vezmeme generování smluv, tak to budou nejspíše provádět pracovníci používající TPS – Transaction Processing System nebo MIS – Management information system.

Aplikaci je ale určitě součástí OIS – Office Information System. Tento blok se prolíná do všech vertikálních úrovní globální architektury. Podporuje kancelářské a administrativní práce a týmovou spolupráci. Jeho součástí může být např. Microsoft Office nebo systémy pro správu dokumentů (např. DMS – Document management system).

4.4 Požadavky

V krátkosti lze popsat požadavky na funkci aplikace takto (podle role uživatele): Administrátor spravuje typy dokumentů a uživatele. Uživatel vytváří a edituje dokumenty, generuje PDF.

4.4.1 Administrátor

má tyto možnosti:

- přihlásit se, změnit svoje údaje
- spravovat uživatele: přidat, upravit, smazat, zablokovat
- spravovat typy dokumentů (šablony): přidat, upravit, smazat
- spravovat skupiny uživatelů: přidat, upravit, smazat, nastavit práva pro šablony

4.4.2 Uživatel

má tyto možnosti:

- přihlásit se, resetovat heslo, změnit svoje údaje
- spravovat dokumenty: vytvořit, upravit, smazat, vygenerovat PDF, exportovat
- Při editaci dokumentu doplnit do údajů o osobách data získaná z veřejného rejstříku (podle zadaného IČO).

4.4.3 Nefunkční požadavky

Údaje o uživatelích, skupinách uživatelů, typech dokumentů a dokumentech jsou uloženy v DBMS. Samotné dokumenty a typy dokumentů (šablony) jsou uloženy jako soubory (ve formátu XML) v adresářích na pevném disku serveru.

Aplikace bude implementována jako webová, tj. poběží na webovém serveru a uživatelé k ní budou přistupovat přes webový prohlížeč. Bude vytvořena v PHP. Důvody jsou: Pro tento jazyk je dostupné mnoho kvalitních frameworků. Modul z kapitoly 3 je také napsán v PHP, takže s ním aplikace bude moct spolupracovat.

4.5 Návrh aplikace

V této části je vytvořen návrh datového modelu a uživatelského rozhraní.

4.5.1 Datový model

Pro návrh datového modelu lze využít tzv. *Entity Relationship modelování*, jehož výstupem je *Entity Relationship Diagram* (ERD) – diagram entit a jejich vztahů. Ten představuje konceptuální resp. logický model dat. Z toho je nakonec vytvořen fyzický model pro konkrétní typ DBMS.

Entita představuje skupinu objektů se stejnými vlastnostmi. Objekty mohou být reálné nebo abstraktní. Entity získáme analýzou požadavků a problémové oblasti. Entita má jméno a atributy – významné vlastnosti entity.

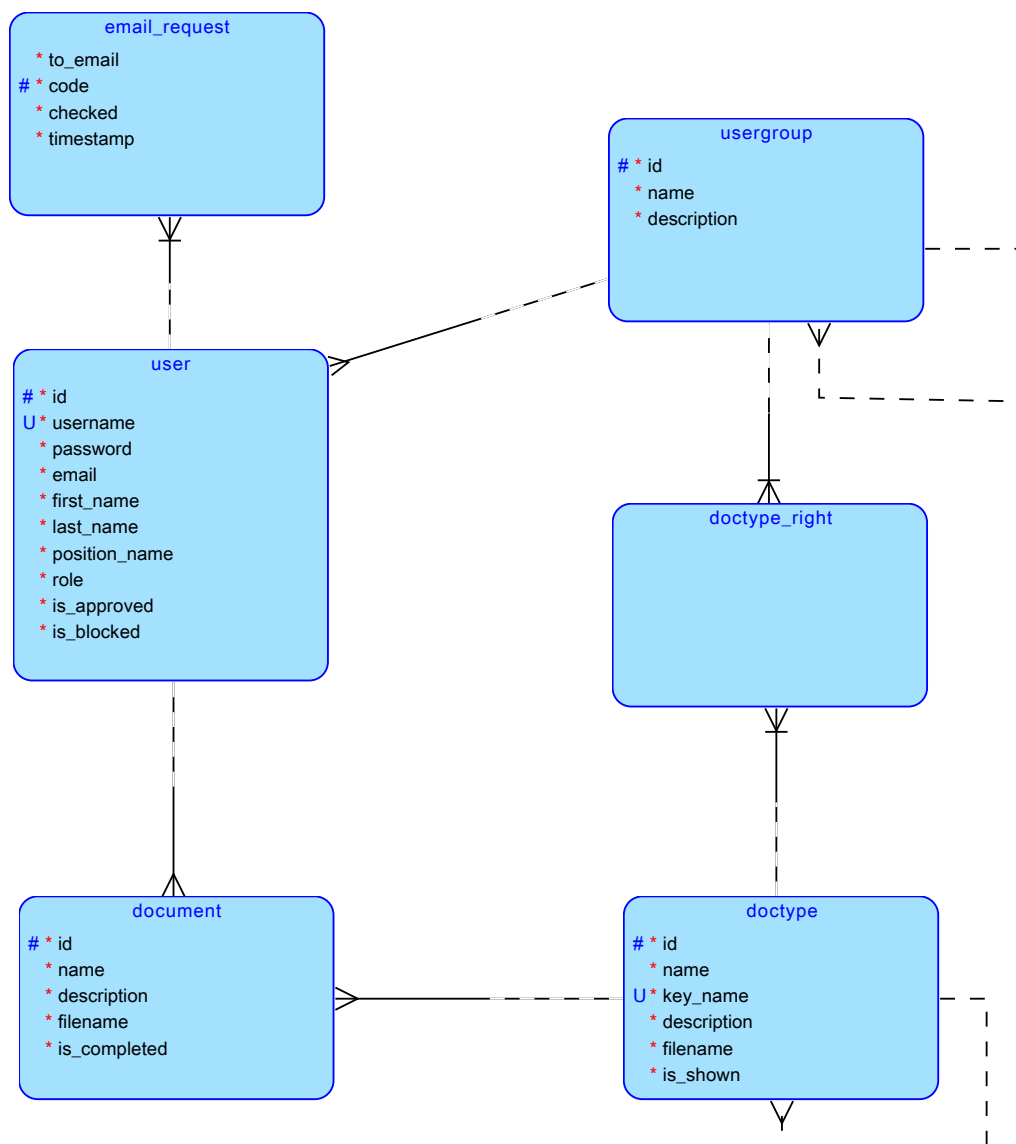
Vztah je množina asociací mezi jednou nebo více entitami. Vztah má jméno, které popisuje jeho funkci (Connolly, 2005, s. 343–346). Má kardinalitu – počet výskytů

objektů entit, které se vztahu účastní (1:1, 1:n, m:n) a parcialitu – povinnost nebo volitelnost jeho existence (Šimůnek, 1999, s. 46).

Byly identifikovány tyto entity: *user* (uživatel), *doctype* (typ dokumentu), *document* (dokument), *usergroup* (skupina uživatelů), *doctype right* (právo k typu dokumentu).

Atributy a vztahy byly také zvoleny, jsou uvedeny v diagramu níže.

Diagram využívá notaci Barker.



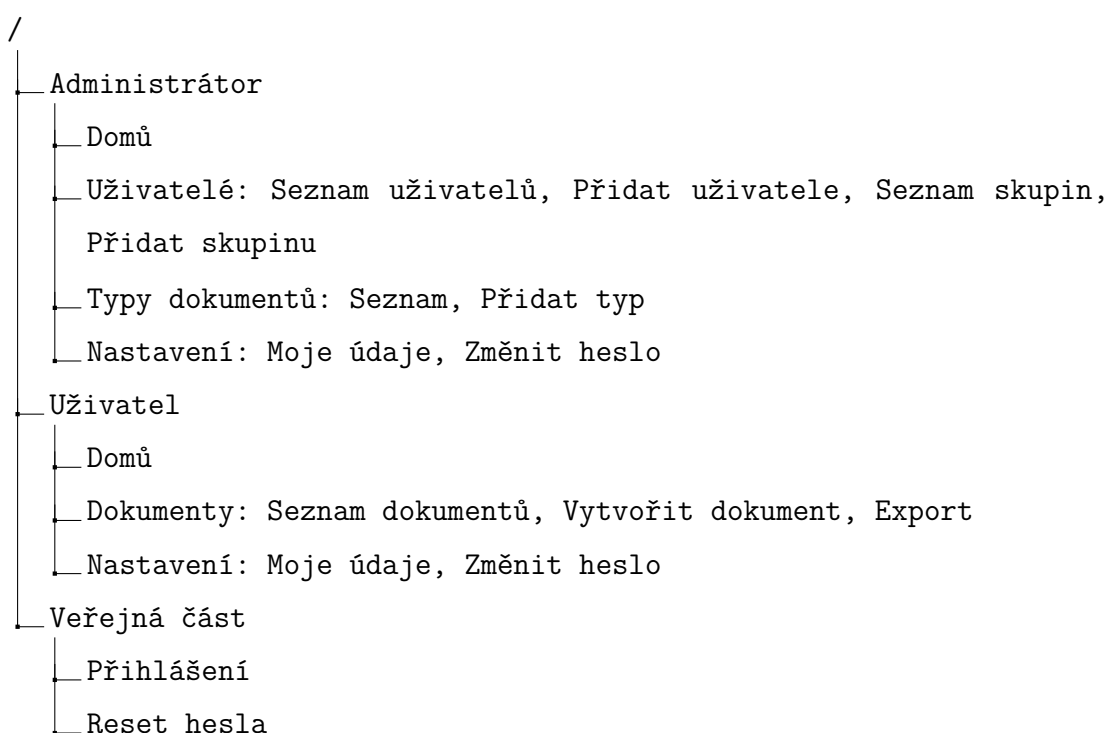
Obrázek 3: Logický datový model – ERD

Později byla přidána tabulka *email_request*, která obsahuje požadavky uživatelů na resetování hesla. Jako primární klíč byl zvolen umělý klíč v jednotném tvaru *id*, díky čemuž lze při implementaci bez problémů používat ORM.

4.5.2 Uživatelské rozhraní

Dalším krokem v návrhu aplikace je základní návrh uživatelského rozhraní. K němu lze využít tzv. *wireframe* – náčrt podoby aplikace, který znázorňuje rozvržení obrazovky – obsah, elementy pro interakci a navigační systémy. Obvykle postrádá styl, barvy nebo grafiku. Zaměřuje se na zobrazené informace, dostupné funkce a chování při práci uživatele (Brown, 2011, s. 166–169). Návrh byl proveden tužkou na papír a nemá velkou technickou kvalitu. Uveden zde tudíž nebude.

Základní struktura



4.5.3 Další návrhy

Na tomto místě by se mohly nacházet některé další diagramy. Například Use Case diagram – diagram případů užití. Nicméně ten by v zásadě nepřinášel nic nového, vše důležité je uvedeno v části o požadavcích.

Další možností je Class diagram – diagram tříd. Jelikož pro implementaci aplikace bude použit aplikační framework, tak toto nemá velkého významu. Jeho součástí je totiž tolik tříd, že by nebylo možné je všechny nebo i jen část do diagramu umístit. Resp. bylo by možné do něj umístit autorem navržené a vytvořené třídy – ale těch je několik desítek. Diagram by tudíž byl velmi nepřehledný a je otázka, zda by byl vůbec k něčemu dobrý.

4.6 Implementace aplikace

4.6.1 Použité technologie

Na základě požadavků a návrhu musíme rozhodnout, jaké technologie použít pro implementaci. Jako programovací jazyk zvolíme **PHP** (verze 5.4). Tento je velmi vhodný pro webové aplikace, existuje pro něj celá řada frameworků a také je v PHP napsán modul z kapitoly 3, který bude aplikace využívat.

Součástí aplikace je databáze, pro její správu bude využit databázový systém **MySQL** (verze 5.6). Tento DBMS je velmi používaný, autor s ním má zkušenosti a plně postačuje potřebám aplikace.

Pro tvorbu aplikace je vhodné využít nějaký framework, který nám usnadní všechny činnosti s tvorbou spojené. Poskytne databázovou vrstvu, šablonovací systém, vysoké zabezpečení nebo snadnou tvorbu a obsluhu formulářů. Frameworků pro PHP existuje velké množství. V zahraničí jsou populární např. Zend, Symfony 2, CakePHP nebo CodeIgniter. V České republice je situace jiná, protože zde dominuje framework **Nette** (verze 2.1). Ten vytvořil český vývojář David Grudl a framework se brzy rozšířil mezi českou webovou komunitu. V poslední době se do vývoje připojilo více lidí, což je rozhodně pozitivní. Připravovaná verze 2.2 rozdělí celý framework na části (komponenty), které lze používat samostatně (Grudl, 2014). Jelikož má autor s Nette již nějaké zkušenosti a framework poskytuje potřebné funkce na vysoké úrovni, byl zvolen jako základ pro tvorbu aplikace.

Mezi další použité „technologie“ je možné zařadit HTML pro tvorbu obsahu webové stránky a CSS pro definici vzhledu stránky.

Pro běh webové aplikace je potřeba webový server, který na požadavek klienta poskytne (resp. předtím vygeneruje) HTML stránku. Mezi nejznámější a nejpoužívanější software pro webový server patří **Apache HTTP Server** (verze 2.4). Pro naši aplikaci poskytuje všechny potřebné funkce a přirozeně se doplňuje s PHP a MySQL, takže byl použit právě tento.

4.6.2 Architektura MVC

Nette framework podporuje softwarovou architekturu Model-view-controller (MVC). Ta řeší problém, jak oddělit uživatelské rozhraní programu (view), aplikační logiku (model) a vnitřní zpracování a rozhodování (controller) tak, aby představovaly tři různé komponenty (Lecky-Thompson, 2009, s. 306). Podrobnější popis (Nette Foundation, 2014):

- Model – Je to datový a funkční základ celé aplikace. Model spravuje svůj vnitřní stav a ven nabízí rozhraní. Model o existenci view nebo controlleru neví.
- View – Zobrazuje výsledek požadavku uživatele, na základě pokynu od controlleru. Obvykle používá šablonovací systém.
- Controller – Zpracuje požadavek uživatele, podle něho zavolá vhodnou aplikační logiku (model), připraví data a požádá view o jejich vykreslení.

4.6.3 Struktura aplikace

Výchozí adresářovou strukturu aplikace lze najít na (Nette Foundation, 2014). Hlavní prvky:

```

/sandbox
├── app/ ..... adresář s aplikací
│   ├── model/ ..... třídy modelové vrstvy
│   ├── presenters/ ..... třídy presenterů (controllerů)
│   └── templates/ ..... šablony ve formátu latte
├── vendor/ ..... samotné Nette a další knihovny
└── www/ ..... veřejný adresář

```

V případě naší aplikace je velmi podobná. Rozdílem je použití modulů pro administrativní a uživatelskou část. Ve složce `app` byly vytvořeny složky `AdminModule` a `FrontModule`, které každá obsahují složky `presenters` a `templates`. Tím pádem je celá struktura přehlednější a nemíchají se funkce pro administrátora a uživatele.

4.6.4 Modelové třídy

Můžeme krátce popsat třídy ve složce `model`. Nachází se zde abstraktní třída *Repository*, která zajišťuje připojení k databázi a základní práci s daty. Pro každou tabulku je vytvořen potomek třídy *Repository* se jménem tabulky (např. *DocumentRepository*). *Repository* obsahuje metodu `getTable`, která podle názvu dané třídy vrátí správnou tabulku. S tou je potom možné snadno pracovat pomocí efektivního rozhraní. Dále obsahuje metody pro CRUD operace, získání všech záznamů a nalezení záznamů podle zadaných podmínek. Potomci *Repository* poté můžou definovat další specifické metody.

Třída *Passwords* zajišťuje práci s hesly. Metoda `hash` vytvoří ze zadaného řetězce tzv. *salted hash*. Tímto způsobem se ukládají hesla do databáze (jsou tudíž nečitelná). Metoda `verify` zkontroluje, zda zadané heslo odpovídá dodanému hashi.

Třída *UserManager* spravuje uživatele – při přihlášení zkontroluje zadaný login a heslo. Pokud je heslo správné, přihlásí uživatele do aplikace. Pokud uživatel neexistuje nebo je heslo špatné, vyhodí výjimku.

Další skupina tříd zabezpečuje práci s dokumenty (XML soubory). Jsou více popsány v části 4.6.6.

- `AgreementFormProcessor` – uložení dat z formuláře do XML souboru
- `AgreementXMLParser` – zpracování XML souboru do podoby pole dat
- `AgreementPDFGenerator` – vygenerování PDF souboru z dat, která jí poskytne `XMLParser`

4.6.5 Routování a presentery

Ve frameworku existuje tzv. *router*, který zpracovává požadavky uživatele a podle jejich tvaru zavolá příslušný presenter. Požadavek, tj. zvaná URL, má tento formát: `<presenter>/<action>/<id>`, přičemž na začátku se ještě může vyskytovat jméno modulu a část `<id>` je nepovinná. Takže zvaní URL `http://myweb.com/article/edit/10` router přesměruje na presenter `Article` a na jeho akci `edit`, a parametr `id` bude mít hodnotu 10.

Třída presenteru se vždy jmenuje `<Nazev>Presenter`. Obsahuje metody pro zpracování akcí, které mají také jednotné názvosloví – jsou to `action<NazevAkce>` nebo `render<NazevAkce>`. Metoda `action` se provádí dříve než metoda `render`. Používá se hlavně pokud daná akce provede určitý úkon (přihlášení uživatele, smazání dat v databázi) a potom je vhodné přesměrovat uživatele jinam. Metoda `render` obvykle získá data z modelu a vloží je do šablony. Soubor šablony je určen podle názvu presenteru a akce.

Pokud vezmeme výše uvedený příklad, tak po přístupu na danou URL se stane toto: soubor `index.php` spustí aplikaci, router zanalyzuje URL, vytvoří objekt `ArticlePresenter` a spustí metodu `renderEdit` s parametrem 10. Ta získá data o článku č. 10 a pošle je do šablony v souboru `templates/Article/edit.latte`. Ta obsahuje kromě běžného HTML speciální makra, která např. zobrazí obsah přijatý z presenteru. Vygenerovaná HTML stránka je nakonec poslána klientovi.

Součástí presenterů jsou také tzv. *továrničky* na komponenty (např. formuláře). Představují je metody s názvem `createComponent<NazevKomponenty>`. Používají se pro vkládání komponent do šablony. Fungují následovně: V šabloně je uvedeno makro např. `{control userForm}`. Při parsování šablony je zvaná metoda `createComponentUserForm`, která vytvoří formulář a vloží ho do šablony, kde je poté zobrazen. Komponenty je možné využít také například pro zobrazení navigace nebo výpis položek z databáze.

4.6.6 Generování PDF

Důležitou součástí aplikace je generování PDF souborů z dokumentů uložených v XML souborech. Postup je následující:

1. Získat cestu k XML souboru dokumentu a načíst soubor do objektu DOM.
2. Získat data z XML souboru (`AgreementXMLParser`).
3. Vytvořit PDF soubor s využitím získaných dat (`AgreementPDFGenerator`).

Pro vytvoření PDF souboru byla použita knihovna FPDF. Resp. doplňková třída `PDF_WriteTag`⁶, která umožňuje měnit formátování textu podle tagů.

Jedná se o prověřenou knihovnu s jednoduchým rozhraním a dostatkem funkcí. Na jejím webu <http://www.fpdf.org/> se nalézá mnoho tipů jak s ní pracovat.

⁶<http://www.fpdf.org/en/script/script23.php>

Má ovšem dvě nevýhody: nepodpora UTF-8 a nutnost speciálního formátu fontu. Existuje upravená verze knihovny s názvem tFPDF⁷, která podporuje UTF-8. Bohužel tuto se nepodařilo úspěšně zprovoznit. Dále existuje knihovna TCPDF (<http://www.tcpdf.org/>), která staví na FPDF a nabízí více funkcí a podporu UTF-8. Její rozhraní je ale o dost komplikovanější a knihovna zabírá hodně místa (869 KB vs 49 KB).

Každopádně ve výchozí instalaci se české znaky nezobrazují správně. Pro podporu češtiny je potřeba použít font s českými znaky a ve skriptu používat řetězec v kódování cp-XXX nebo iso-XXX.

Dokumenty (XML soubory) jsou uloženy v UTF-8. Pro převod údajů z XML-Parseru do jiného kódování (konkrétně windows-1250) byla využita funkce `iconv`.

České fonty lze nalézt například v operačním systému Windows 8 ve složce `C:\Windows\Fonts`. Pro dokumenty byl vybrán font Arial v řezech tučný, kurzívní a normální. Následně je nutné vygenerovat speciální formát fontu – lze využít nástroj na <http://fpdf.fruit-lab.de/>. Vybereme soubor fontu, zvolíme kódování cp-1250 a odešleme formulář. Následně stáhneme vygenerované soubory a umístíme je do složky font v instalaci FPDF. Soubor s příponou PHP můžeme přejmenovat např. na `czArialn`. Fonty jsou chráněny autorským zákonem – musíme získat licenci nebo splnit její podmínky (např. nekomerční použití). Kvalitní české fonty, které jsou zdarma dostupné, se hledají obtížně (Blažek, 2009). V práci použitý font Arial je tedy vhodný jenom pro testování. Pro veřejné resp. komerční použití aplikace by musel být zvolen jiný font.

V konstruktoru třídy FPDF použijeme metodu `AddFont(nazev,font.php)`, která do objektu přidá možnost použít tento font (přes `SetFont(nazev,'',velikost)`). Pro zprovoznění formátování podle tagů je ještě nutné zavolat pro každý tag metodu `SetStyle`, která nastaví vlastnosti pro daný tag (font, velikost, barvu). Základní metody pro tisk textu jsou `Cell`, `Multicell` a `Write`. Pro zpracování textu s tagem se používá metoda `WriteTag`. Dodejme, že tato očekává jako vstup řetězec obsahující kořenový element (např. `p` nebo uměle vytvořený `root`). Je vhodné vytvořit hlavní metodu `printAll`, která obsahuje privátní metody, zajišťující tisk různých částí dokumentu.

Nakonec je zavolána metoda `Output('soubor.pdf','par')`, která dle hodnoty druhého parametru: zobrazí dokument v prohlížeči (i), nabídne stažení souboru (d), uloží soubor na lokální disk (f) nebo vrátí dokument jako řetězec (s). V aplikaci je použita volba 'd'.

Komplexní příklad, jak použít tuto knihovnu, lze nalézt právě ve třídě `AgreementPDFGenerator`.

4.6.7 Vybrané funkce aplikace

V krátkosti popíšeme některé metody, které jsou použité pro práci s dokumenty a uživateli. Náhledy obrazovek se nacházejí v příloze F.

⁷<http://www.fpdf.org/en/script/script92.php>

Admin – vložit/upravit typ dokumentu

Formulář obsahuje pole název, popis, soubor šablony, nadřazený typ a jestli je přístupný uživatelům. Po odeslání formuláře:

1. Zjistit, jestli je to úprava (URL obsahuje ID) nebo vložení typu.
2. Zkontrolovat soubor šablony – Je to validní XML a odpovídá XML schématu pro obecný typ dokumentu. Pokud je typ smlouva, zkontrolovat podle speciálního schématu. Pokud není splněna nějaké podmínka, je vyhozena výjimka. Jinak je získán klíčový název z XML souboru.
3. Vložit údaje do databáze – insert nebo update. Klíčový název typu musí být unikátní – pokud není, je vyhozena výjimka.
4. Zpracovat soubor šablony – nahrát do složky `www/doc-uploads/doctypes` a aktualizovat název v databázi.
5. Zobrazit zprávu uživateli a přesměrovat ho – metody `flashMessage` a `redirect`.

Schémata pro krok 2 se nacházejí ve složce `www/doc-uploads/schemas`.

Uživatel – vytvořit/upravit dokument

Vytvoření nového dokumentu zahrnuje dva kroky:

1. Vybrat typ dokumentu – Zobrazí se všechny typy a jejich podtypy, ke kterým má uživatel přístup. Zaprvé musí být typ označen jako přístupný. Zadruhé musí být uživatel ve skupině, která má nastaveny práva pro přístup k tomuto typu.
2. Vložit základní údaje o dokumentu (název, popis). Potom je dokument vložen do databáze, soubor typu je zkopírován do složky `www/doc-uploads/documents` a uživatel je přesměrován na stránku pro úpravu.

Stránka pro úpravu dokumentu obsahuje dva formuláře – jeden pro základní údaje, druhý pro obsah. Také jsou zde odkazy na stáhnutí XML souboru a vygenerování PDF souboru. Pokud typ není smlouva, není formulář pro obsah (ani odkaz na PDF) zobrazen. Pro další typy dokumentů by bylo nutné vytvořit nové třídy podle vzoru `AgreementPDFParser`, `XMLParser`, `PDFGenerator`.

Formulář pro obsah je vytvořen pomocí třídy `DocumentContentForm` na základě dat získaných z XML souboru. Následně může uživatel upravit daná pole. Pro články smlouvy lze využít omezenou množinu formátovacích příkazů – číslovaný seznam bodů článku a řezy písma (tučný, kurzivní). Pro toto byla využita javascriptová knihovna TinyMCE (verze 4.0, <http://www.tinymce.com>). Smluvní strany jsou dvě a lze zvolit jestli to bude právnická nebo fyzická osoba. Podle toho se zobrazí vhodná pole. Pokud zadáme IČO, lze použít funkci pro získání dat z veřejného rejstříku – využívá modul `CzechRegisters` z kapitoly 3. Funkce je realizována přes AJAX.

Po odeslání formuláře je načten stávající soubor XML. Tento je na základě nových hodnot upraven pomocí třídy `AgreementFormProcessor` a znovu uložen.

Úprava hodnot je poměrně komplikovaný proces. Je použito rozhraní DOM a XPath. Bylo také nutné zajistit vložení HTML tagů do obsahu článku. Toho bylo dosaženo vymazáním všech potomků elementu a poté těmito příkazy:

```
$f = $this->dom->createDocumentFragment();  
$f->appendXML($newText); // textový obsah článku, může obsahovat tagy  
$clause->appendChild($f); // připojit fragment k uzlu článku
```

V případě, že článek obsahuje body, je situace komplikovanější – je potřeba vytvořit „umělý“ DOMDocument, získat v něm elementy li a zpracovat jejich obsah přes:

```
private function getInnerHtml($node)  
{  
    $innerHTML = '';  
    foreach ($node->childNodes as $child) {  
        $innerHTML .= $child->ownerDocument->saveXML($child);  
    }  
    return $innerHTML;  
}
```

Generování PDF je popsáno v části 4.6.6.

Admin – přidání uživatele

Admin může přidávat uživatele. Formulář obsahuje: login, e-mail, jméno, příjmení, název pozice, skupina, podskupina.

Volby v selectu podskupina se dynamicky načítají přes AJAX, kdy při změně hlavní skupiny je zavolána metoda presenteru, která vrátí data ve formátu JSON, obsahující ID a název podřízených skupin.

Po odeslání formuláře jsou data vložena do databáze. Pokud login není unikátní, je vrácena chyba. V metodě uložení je vygenerováno heslo. Pokud vše proběhne úspěšně, je uživateli na zadaný e-mail odeslána zpráva, obsahující přihlašovací údaje.

Pokud uživatel zapomene heslo, může na stránce pro přihlášení požádat o jeho resetování. Zadá svůj login a na registrovaný e-mail mu přijde zpráva s odkazem. Pokud na něj klikne, aplikace vygeneruje nové heslo, uloží je do databáze a pošle je na e-mail uživatele. Ten se potom může přihlásit a případně si heslo změnit.

Uživatel – export

Uživatel může exportovat své dokumenty – ve formátu XML nebo PDF. Výsledkem je ZIP soubor (archiv), obsahující soubory zvoleného typu. Tento soubor je poté nabídnut uživateli ke stažení. Archiv je vytvořen pomocí třídy ZipArchive.

V případě PDF je nutné vygenerovat PDF soubory ze všech dokumentů, uložit je do dočasného adresáře, vložit cesty k nim do archivu, uložit archiv a nakonec PDF soubory smazat.

5 Diskuze

Tato kapitola práce se zabývá zhodnocením dosažených výsledků. Ty byly třech typů: přehled veřejných rejstříků, modul pro získávání dat a webová aplikace pro generování dokumentů.

5.1 Přehled veřejných rejstříků

První část práce se věnovala veřejným rejstříkům. Bylo nalezeno a popsáno velké množství veřejných rejstříků. Seznam je poměrně vyčerpávající a hlavní rejstříky tam jsou uvedeny. Bylo uvažováno nad tím, uvést o některých rejstřících více informací. Ale jelikož to přímo nesouvisí s tématem práce, která je navíc už tak dost rozsáhlá, byl zachován stávající obsah. Určitě existuje celá řada dalších rejstříků a dat, ke kterým je možné se v České republice dostat. Nicméně kapitola představuje základ, na kterém by šlo dále stavět.

5.2 Modul pro získávání dat

Druhá část práce se zabývala tvorbou modulu pro získávání dat z vybraných rejstříků. Po analýze možností přístupu k datům bylo navrženo řešení, které dokáže data získávat. Faktem je, že ARES poskytuje ta nejdůležitější data přes kvalitní rozhraní, ale je velmi pomalý. Navíc k některým registrům nemá přímý přístup. Vytvořené řešení zpřístupňuje údaje z mnoha různých registrů a je poměrně rychlé. Nevýhodou je, že RŽP ani OR nemají adresy uložené v jednotném formátu, takže nelze z adresy získat jednotlivé položky. RES není občas přístupný resp. dotaz na něj trvá velmi dlouho. Důvodem může být nízký výkon serveru nebo nějaká ochrana proti automatickému použití formuláře.

Celkově se dá říct, že modul je dobře použitelný. Bylo by možné ho rozšířit o další rejstříky a přidat pomocné funkce (např. pro ukládání získaných dat nebo logování volání metod). Zdrojový kód modulu bude umístěn na internet, kde si ho případný zájemce může stáhnout a následně integrovat do své aplikace nebo si jej upravit k obrazu svému.

5.3 Webová aplikace pro generování dokumentů

Třetí část práce byla zaměřena na návrh a tvorbu webové aplikace pro generování standardních typů dokumentů. Byl popsán princip aplikace a byl vytvořen model dokumentu (šablona) ve formátu XML. Jedná se o smlouvu – obsahuje hlavní oddíly jako název, smluvní strany, články, podpis. Je použito několik atributů a jmenných prostorů. Šablona je funkční a v aplikaci se dá dobře použít. Bylo vytvořeno několik šablon pro konkrétní typy smluv (dohoda o provedení práce, kupní smlouva na movitou věc, licenční smlouva).

Dále byla navržena a implementována webová aplikace. Tato se skládá ze dvou modulů:

1. administrační – správa typů dokumentů (šablon) a uživatelů,
2. uživatelský – správa dokumentů a údajů uživatele.

První modul zajišťuje základní funkce, potřebné pro použití aplikace uživateli. Lze přidat další typy smluv. Také je možné systém doplnit o další (hlavní) typy dokumentů. To by si ovšem vyžádalo vytvoření nového modelu dokumentu a speciálních tříd pro práci s ním. Bylo by možné přidat další funkce – např. import uživatelů ze vzdálené databáze resp. souboru nebo interaktivní tvorba šablon.

Druhý modul umožňuje uživateli vytvářet, upravovat a generovat dokumenty. Uživatelské rozhraní pro editaci dokumentů je funkční a umožňuje všechny základní operace. Je ale strohé – bylo by možné udělat je přehlednější a komfortnější. Lze uvažovat o dalších možnostech pro uživatele (např. spolupráce na dokumentech), ale ty by vycházely hlavně z požadavků uživatelů, získaných během testování. Aplikace neřeší vícejazyčnost GUI, v tomto směru by tedy bylo možné ji vylepšit, ale vyžádalo by si to poměrně velký zásah do kódu programu.

Nápad na aplikaci byl inspirován požadavkem z praxe, že by se hodil nástroj na doplňování údajů do smluv. Nakonec ale bylo zvoleno řešení přímého generování smluv. Je pravděpodobné, že firmy obsluhující velký počet zákazníků mají své vlastní řešení, zakomponované do podnikového IS. Aplikace ovšem může být užitečná menším firmám, které tolik smluv (či jiných dokumentů) nevytvářejí, ale chtěly by podobné řešení využívat, aby jejich dokumenty měly stejnou strukturu a kvalitu.

V neposlední řadě může být aplikace zajímavá pro širokou veřejnost, které nabídne ověřený zdroj smluv (resp. jiných dokumentů) a možnost tvořit kvalitní dokumenty. V tomto případě by bylo nutné přidat veřejnou registraci a upravit možnosti administrátora.

6 Závěr

Práce se zabývala veřejnými rejstříky a generováním dokumentů.

Cílem práce bylo vytvořit přehled veřejných rejstříků v ČR, vytvořit modul pro automatizované získávání dat z vybraných veřejných rejstříků a následně vytvořit webovou aplikaci pro generování standardních typů dokumentů. Cíl práce resp. všechny dílčí cíle byly splněny.

První část práce se věnovala veřejným rejstříkům. Byl popsán systém základních registrů a proveden průzkum, jaké veřejné rejstříky v České republice existují. Byl vytvořen poměrně vyčerpávající seznam rejstříků a jejich popis. Jeho součástí byla analýza možností automatického získávání dat z nich.

Druhá část práce se zabývala tvorbou modulu pro získávání dat z veřejných rejstříků. Byly vybrány určité rejstříky a otestovány možnosti získávání dat. Následně bylo navrženo a vytvořeno řešení pro toto získávání.

Třetí část práce byla zaměřena na návrh a tvorbu webové aplikace pro generování standardních typů dokumentů. Byl popsán princip aplikace, byl vytvořen model dokumentu (smlouva) a byly stanoveny požadavky na aplikaci. Poté byla aplikace navržena a implementována.

Nakonec byla v diskuzi vytvořená řešení okomentována a zhodnocena a byly nastíněny možnosti pro jejich další rozvoj do budoucna.

V závěru byl celý obsah práce shrnut.

7 Literatura

- ADVICE.CZ. Rejstřík trestů vás eviduje po 100 let. In: *ISVS.CZ: zpravodajství o ISVS a eGovernmentu* [online]. 31. 10. 2008 [cit. 2014-04-06]. Dostupné z: <http://www.isvs.cz/rejstrik-trestu-vas-eviduje-po-100-let/>.
- BLAŽEK, F. Jak si pořídit legálně fonty. In: *Unie grafického designu: Návody* [online]. 26. 2. 2009 [cit. 2014-04-30]. Dostupné z: <http://unie-grafickeho-designu.cz/jak-si-poridit-legalne-fonty/#.U2Cnolf0j1F>.
- BROWN, D. M. *Communicating Design: Developing Web Site Documentation for Design and Planning*. 2nd ed. Berkeley, Calif: New Riders, 2011. ISBN 978-013-1385-399.
- CESAROVÁ, A. *Obchodní rejstřík po novele obchodního zákoníku*. Brno, 2008. Diplomová práce. Masarykova univerzita, Právnická fakulta, Katedra obchodního práva.
- CHEN, S. C. *PHP Simple HTML DOM Parser* [online]. Latest version of the library: 15. 4. 2013 [cit. 2014-04-09]. Dostupné z: simplehtmldom.sourceforge.net.
- CONNOLLY, T. M., BEGG, C. E. *Database Systems: A Practical Approach to Design, Implementation, and Management*. 4th ed. Harlow: Addison-Wesley, 2005. ISBN 03-212-1025-5.
- ČESKÝ STATISTICKÝ ÚŘAD. O registru – RES. ČSÚ [online]. Aktualizováno dne: 17.2. 2012 [cit. 2014-04-07]. Dostupné z: http://www.czso.cz/csu/redakce.nsf/i/o_registru_res.
- ČESKÝ ÚŘAD ZEMĚMĚŘICKÝ A KATASTRÁLNÍ. Účel katastru. ČÚZK [online]. © 2013 [cit. 2014-04-07]. Dostupné z: <http://www.cuzk.cz/Katastr-nemovitosti/0-katastru-nemovitosti/Ucel-katastru.aspx>.
- DVOŘÁK, T. Načítání dat z obchodního rejstříku justice.cz. In: *Blog Tomáše Dvořáka* [online]. 23. 5. 2013 [cit. 2014-04-08]. Dostupné z: <http://www.tomas-dvorak.cz/clanky/nacitani-dat-z-obchodniho-rejstriku-justicecz>.
- EVROPSKÁ KOMISE. VIES FAQ. *Evropská komise* [online]. [2014] [cit. 2014-04-07]. Dostupné z: http://ec.europa.eu/taxation_customs/vies/faq.html.
- FOND OTAKARA MOTEJLA. Co jsou otevřená data. *Otevřená data* [online]. © 2014 [cit. 2014-04-06]. Dostupné z: <http://www.otevrenadata.cz/otevrena-data/co-jsou-otevrena-data/>.
- FOWLER, M. *Patterns of enterprise application architecture*. Boston: Addison-Wesley, 2003. ISBN 0-321-12742-0.
- GRUDL, D. Nette Revolution 2.2. In: *PhpFashion* [online]. 20. 4. 2014 [cit. 2014-04-29]. Dostupné z: <http://phpfashion.com/nette-revolution-2-2>.

- KOČÍ, P. Podívejte se, jak nás Slovensko předbíhá v otevírání vládních dat. In: *Data, blog o datové žurnalistice* [online]. 8. 6. 2012 [cit. 2014-04-06]. Dostupné z: <http://data.blog.ihned.cz/c1-55963040-podivejte-se-jak-nas-slovensko-predbiha-v-otevirani-vladnich-dat>.
- KOSEK, J. *PHP a XML*. Praha: Grada, 2009. ISBN 978-80-247-1116-4.
- LECKY-THOMPSON, E., NOVICKY, S. D., MYER, T. *Professional PHP6*. Indianapolis: Wiley, 2009. ISBN 978-0-470-39509-7.
- MATES, P., SMEJKAL, V. *E-government v České republice: právní a technologické aspekty*. 2., podstatně přeprac. a rozš. vyd. Praha: Leges, 2012. ISBN 978-80-87576-36-6.
- MINISTERSTVO FINANCÍ ČR. *Administrativní registr ekonomických subjektů* [online]. © 2013 [cit. 2014-04-06]. Dostupné z: <http://www.info.mfcr.cz/ares/ares.html.cz>.
- MINISTERSTVO PRO MÍSTNÍ ROZVOJ ČR. Seznam kvalifikovaných dodavatelů. *ISVZ – Informační systém o veřejných zakázkách* [online]. [cit. 2014-04-07]. © 2005–2013. Dostupné z: http://www.isvz.cz/ISVZ/SKD/ISVZ_SKD_text.aspx.
- MINISTERSTVO SPRÁVEDLNOSTI ČR, 2014a. Formulář pro lustraci. *ISIR – Insolvenční rejstřík* [online]. [2014] [cit. 2014-04-07]. Dostupné z: <https://isir.justice.cz/isir/common/index.do>.
- MINISTERSTVO SPRÁVEDLNOSTI ČR, 2014b. Zdůraznění role veřejných seznamů. In: *Nový občanský zákoník* [online]. © 2013–2014 [cit. 2014-04-07]. Dostupné z: <http://obcanskyzakonik.justice.cz/vecna-prava/konkretni-zmeny/zdrazneni-role-verejnych-seznamu/>.
- NABITO.NET. Autoload v PHP. In: *NABITO.NET* [online]. 29. 10. 2008 [cit. 2014-04-09]. Dostupné z: <http://www.nabito.net/autoload-v-php/>.
- NETTE FOUNDATION. MVC aplikace & presentery. In: *Nette Framework: Dokumentace* [online]. © 2008–2014 [cit. 2014-04-10]. Dostupné z: <http://doc.nette.org/cs/2.1/presenters>.
- OPENDATA.CZ. Katalog dat. *Opendata.cz – Iniciativa za otevřenou datovou infrastrukturu* [online]. [2012] [cit. 2014-04-06]. Dostupné z: <http://opendata.cz/cs/node/25>.
- RÁBOVÁ, I. *Podnikové informační systémy a technologie jejich vývoje*. Brno: Tribun EU, 2008. ISBN 978-80-7399-599-7.
- ROBINSON, D. et al. Government data and the invisible hand. *Yale Journal of Law and Technology* [online]. 2009, Vol. 11: Iss. 1, Article 4 [cit. 2014-04-06]. Dostupné z: <http://digitalcommons.law.yale.edu/yjolt/vol11/iss1/4>.

- SKONNARD A., GUDGIN M. *Essential XML quick reference: a programmer's reference to XML, XPath, XSLT, XML Schema, SOAP, and more*. Boston: Addison-Wesley, 2001. ISBN 02-017-4095-8.
- SPRÁVA ZÁKLADNÍCH REGISTRŮ, 2014a. Co jsou to základní registry. *Správa základních registrů* [online]. © 2010–2014 [cit. 2014-04-06]. Dostupné z: <http://www.szrcr.cz/co-jsou-to-zakladni-registry>.
- SPRÁVA ZÁKLADNÍCH REGISTRŮ, 2014b. Informační systém základních registrů. *Správa základních registrů* [online]. © 2010–2014 [cit. 2014-04-06]. Dostupné z: <http://www.szrcr.cz/informacni-system-zakladnich-registru>.
- SPRÁVA ZÁKLADNÍCH REGISTRŮ, 2014c. ORG – převodník. *Správa základních registrů* [online]. © 2010–2014 [cit. 2014-04-06]. Dostupné z: <http://www.szrcr.cz/informacni-system-zakladnich-registru>.
- ŠIMŮNEK, M. *SQL: kompletní kapesní průvodce*. Praha: Grada, 1999. ISBN 80-716-9692-7.
- THE PHP GROUP, 2014a. cURL introduction. In: *PHP manual* [online]. © 2001–2014 [cit. 2014-04-09]. Dostupné z: <https://php.net/manual/en/intro.curl.php>.
- THE PHP GROUP, 2014b. Magic Methods. In: *PHP manual* [online]. © 2001–2014 [cit. 2014-04-09]. Dostupné z: <http://www.php.net/manual/en/language.oop5.magic.php>.
- ÚSTAV PRO JAZYK ČESKÝ AKADEMIE VĚD ČR. Heslo rejstřík. In: *Slovník spisovného jazyka českého* [online]. © 2011 [cit. 2014-05-09]. Dostupné z: <http://ssjc.ujc.cas.cz/search.php?sti=74846>.
- VLÁDA ČESKÉ REPUBLIKY. Akční plán České republiky „Partnerství pro otevřené vládnutí“ [online]. 4. 4. 2012 [cit. 2014-04-06]. Dostupné z: <http://www.vlada.cz/assets/clenove-vlady/pri-uradu-vlady/karolina-peake/tiskove-zpravy/Akcni-plan-OGP.pdf>.
- WIKIPEDIA, 2014. Registr dlužníků. In: *Wikipedia: the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, poslední aktualizace 2. 3. 2014 [cit. 2014-04-07]. Dostupné z: http://cs.wikipedia.org/wiki/Registr_dlu%C5%BEn%C3%ADk%C5%AF.

Seznam zkratek

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CRUD	Create Read Update Delete
CSS	Cascading Style Sheets
DBMS	Database management system
DOM	Document Object Model
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
ICT	Information and Communication Technologies
IS	Informační systém
ISVS	Informační systém veřejné správy
ISZR	Informační systém základních registrů
IČO	Identifikační číslo osoby
JSON	JavaScript Object Notation
OMG	Object Management Group
OOP	Object-oriented programming
ORM	Object-relational mapping
OVM	Orgán veřejné moci
RUP	Rational Unified Process
SOAP	Simple Object Access Protocol
UML	Unified Modelling Language
URL	Uniform Resource Locator
WYSIWYG	What You See Is What You Get
XML	Extensible Markup Language
XSLT	eXtensible Stylesheet Language Transformations

Přílohy

A Elektronická příloha

Přílohou této práce je CD s elektronickým obsahem.

/	
└ docugen-app/	Webová aplikace pro generování dokumentů
└└ sandbox/	Adresář s aplikací
└└ docugen_db.sql	Exportovaná MySQL databáze
└└ instalace.txt	Poznámky k instalaci
└└ online_verze.txt	Odkaz na online verzi
└└ prihlaseni.txt	Výchozí přihlašovací údaje
└└ priklad_smlouvy.pdf	Vygenerovaná smlouva v PDF
└└ priklad_smlouvy.xml	Zdrojový soubor smlouvy v XML
└ rejstriky-modul/	Modul pro získávání dat
└└ CzechRegisters/	Adresář s modulem
└└ phpdoc/	PHP dokumentace
└└ online_verze.txt	Odkaz na online verzi
└└ pouziti.txt	Poznámky k použití
└ bc-prace.pdf	Elektronická verze bakalářské práce

B E-mailová korespondence s Mgr. Hejdukem

Hejduk Marek (marek.hejduk@szrccr.cz)

RE: externí přístup k registrům

11. 2. 2014, 8:49:08

Komu: jond@post.cz

Dobrý den,

základní registry byly vybudovány především pro potřebu orgánů veřejné moci. Představují pro ně jediný relevantní zdroj údajů. Referenční údaje obsažené v základních registrech jsou považovány za správné, pokud není prokázán opak nebo pokud nevznikne oprávněná pochybnost o jejich správnosti.

Podle § 5 odst. 1 zákona o základních registrech (zákon č. 111/2009 Sb., ve znění pozdějších předpisů) využívají orgány veřejné moci údaje ze všech základních registrů, a to v rozsahu, v jakém jsou oprávněny tyto údaje využívat podle tohoto zákona nebo podle jiných právních předpisů, aniž by ověřovaly jejich správnost.

Každý úředník je povinen při výkonu své činnosti referenční údaje využívat a po občanovi je smí požadovat pouze v případech, kdy tyto nejsou v základních registrech obsaženy, jsou označeny za nesprávné nebo vznikla pochybnost o jejich správnosti.

Úřady přistupují k referenčním údajům prostřednictvím tzv. agendových informačních systémů, tedy v rámci agendy, kterou vykonávají.

Přístup k referenčním údajům v základních registrech mají umožněn i fyzické osoby (jednotlivci) nebo právnické osoby (firmy). Tyto subjekty neprovozují žádné agendové informační systémy a k referenčním údajům mohou přistupovat pouze prostřednictvím k tomu speciálně určených agendových informačních systémů. V praxi se jedná o dva možné přístupy:

- přístup prostřednictvím komunikačních kanálů Czech POINT; přístup k údajům na základě žádosti zprostředkuje pracovník u přepážky kontaktního místa veřejné správy,
- prostřednictvím informačního systému datových schránek (ISDS – seznam orgánů veřejné moci); přístup k údajům uskutečňuje vlastník datové schránky s využitím formulářů uveřejněných na Portálu veřejné správy.

V obou případech mohou jednotlivci nebo firma získat údaje vedené o nich v základních registrech i záznamy o využívání jejich údajů (tedy o tom, kdo, kdy a za jakým účelem jejich údaje využil). Mohou získat i veřejné údaje o jiných subjektech a jednotlivcích. Subjekt registru obyvatel může navíc poskytnout svůj souhlas k tomu, aby jeho údaje byly poskytovány jiným osobám.

Oficiální seznam všech registrů/rejstříků existuje. Je jím Informační systém o informačních systémech veřejné správy, který je dostupný na <https://www.sluzby-isvs.cz/ISoISVS/Applets/DefaultSSL.aspx>. U každého informačního systému je údaj o tom, zda je veřejný, částečně veřejný nebo zcela neveřejný.

S pozdravem

Mgr. Marek Hejduk – pracovník pro vztah s veřejností

C Soubor core.php

```
// DS - oddělovač pro adresáře
define ( 'DS', DIRECTORY_SEPARATOR );

// absolutní cesta k tomuto souboru
define ( 'ABSPATH', __DIR__.DS );

// nastavíme do include path potřebné cesty ke všem třídám
set_include_path (
    ABSPATH.'libs'.DS.PATH_SEPARATOR.           // knihovny
    ABSPATH.'registers'.DS.PATH_SEPARATOR.       // třídy registrů
    ABSPATH.'helpers'.DS.PATH_SEPARATOR.         // pomocné třídy
    get_include_path()
);

// autoload "magická" funkce
function __autoload($class_name)
{
    if (!class_exists($class_name, false) || !
        interface_exists($class_name, false))
    {
        // obsahuje název třídy namespace?
        $slashPos = strpos($class_name, '\\');
        if ($slashPos) {
            $class_name = substr($class_name, $slashPos+1);
        }
        require_once ($class_name.'.php');
    }
}
```

D Odeslání souboru přes cURL

```
// prepare data
$url = 'http://www.rzp.cz/cgi-bin/aps_cacheWEB.sh';
$fullpath = '@'.realpath('dotaz.ico.xml');
$data = array(
    'VSS_SERV' => 'ZVWSBJXML',
    'filename' => $fullpath
);

//open connection
$ch = curl_init();

// set parameters
curl_setopt($ch, CURLOPT_URL, $url);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

//execute post
$ok = curl_exec($ch);

//close connection
curl_close($ch);

// show response
echo $ok;
```


E XML dotazy na RŽP

Hledání podle IČO

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<VerejnyWebDotaz xmlns="
urn:cz:isvs:rzp:schemas:VerejnaCast:v1" version="2.5">
<Kriteria>
    <IdentifikacniCislo>27074358</IdentifikacniCislo>
    <PlatnostZaznamu>1</PlatnostZaznamu>
</Kriteria>
</VerejnyWebDotaz>
```

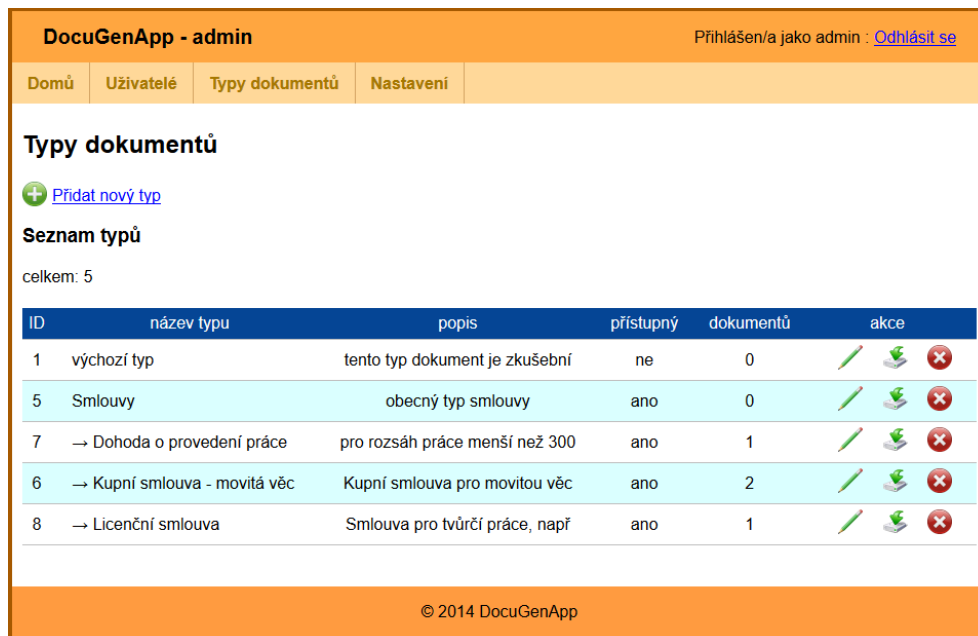
Hledání podle obchodního jména

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<VerejnyWebDotaz xmlns="
urn:cz:isvs:rzp:schemas:VerejnaCast:v1" version="2.5">
<Kriteria>
    <ObchodniJmeno>ing. petrovský</ObchodniJmeno>
    <CastecneDohledani>0</CastecneDohledani>
    <Adresa> <!-- nepovinný element, lze vynechat -->
        <Obec>brno</Obec>
    </Adresa>
    <PlatnostZaznamu>1</PlatnostZaznamu>
</Kriteria>
</VerejnyWebDotaz>
```

Detail podnikatele

```
<?xml version="1.0" encoding="ISO-8859-2"?>
<VerejnyWebDotaz xmlns="
urn:cz:isvs:rzp:schemas:VerejnaCast:v1" targetNamespace=
"urn:cz:isvs:rzp:schemas:VerejnaCast:v1"
elementFormDefault="qualified" version="2.5">
    <PodnikatelID>60e1e7c703d8e8f666</PodnikatelID>
    <Historie>0</Historie>
    <DruhVypisu>XML</DruhVypisu>
</VerejnyWebDotaz>
```

F Webová aplikace – obrazovky



Obrázek 4: Admin – Seznam typů dokumentů

Typy dokumentů

Upravit typ č. 7

Název:

Klíčový název: bude doplněn podle hodnoty v XML souboru

Popis:

Soubor šablony: Soubor nevybrán. Můžete jej nahradit. Současný lze [stáhnout](#)

Nadřazený typ:

☒ přístupný uživatelům

zpět na [seznam typů](#)

Obrázek 5: Admin – Upravit typ dokumentu

DocuGenApp - admin
Přihlášen/a jako admin : [Odhlásit se](#)

[Domů](#)
[Uživatelé](#)
[Typy dokumentů](#)
[Nastavení](#)

Uživatelé

[Seznam uživatelů](#)
[Přidat uživatele](#)
[Seznam skupin](#)
[Přidat skupinu](#)

Seznam uživatelů

Filtrovat podle skupiny:

celkem: 6

ID	login	jméno	skupina	blokovaný	dokumentů	akce
1	admin	Jonáš Petrovský	10 - admin	ne	0	
2	jontes	Johny Petrovsky	4 - IT oddělení	ne	4	
3	jontes2	Johnys Petrovskys	6 - vývoj a správa IS	ne	0	
4	franta45	František Veliký	2 - Oddělení skladu	ne	0	
7	petr.veliky	Petr Veliký	8 - networking	ne	0	
8	oskar.malen	Oskar Malen	9 - Public relations	ne	0	

© 2014 DocuGenApp

Obrázek 6: Admin – Seznam uživatelů

DocuGenApp - uživatel
Přihlášen/a jako jontes : [Odhlásit se](#)

[Domů](#)
[Dokumenty](#)
[Nastavení](#)

Dokumenty

[Moje dokumenty](#)
[Vytvořit dokument](#)
[Export](#)

Moje dokumenty

celkem: 4

ID	název	popis	typ	hotový	akce
6	Kupní smlouva - sekačka	smlouva pro nákup bazarové sek	Kupní smlouva - movitá věc	ne	
7	Dohoda o PP - pokládka dlažby	smlouva pro člověka, který na	Dohoda o provedení práce	ne	
8	Licenční smlouva - logo pro firmu	smlouva pro grafika, který nám	Licenční smlouva	ne	
10	Kupní smlouva - notebook Sony	Kupní smlouva pro bazarový not	Kupní smlouva - movitá věc	ne	

© 2014 DocuGenApp

Obrázek 7: Uživatel – Seznam dokumentů