ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název: Asistovaná aktualizace informací o klientech firmy na základě služby ARES
Student: Jakub Šedý
Vedoucí: Ing. Miroslav Balík, Ph.D.
Studijní program: Informatika
Studijní obor: Softwarové inženýrství
Katedra: Katedra softwarového inženýrství
Platnost zadání: Do konce zimního semestru 2018/19

Pokyny pro vypracování

Registr ekonomických subjektů ARES (http://wwwinfo.mfcr.cz/ares/) nabízí rozhraní, které na základě IČO vrátí o subjektu další informace. Firmy si o svých obchodních partnerech či klientech vedou vlastní evidenci. Cílem práce je navrhnout a implementovat řešení, které umožní aktualizovat lokální data na základě registru ARES.
Pokyny pro vypracování:
1. Navrhněte a implementujte databázi, do které se budou ukládat data získaná z ARES.
2. Analyzujte a navrhněte vhodné způsoby aktualizace konkrétních položek získaných z ARES v případě, že nesouhlasí s lokálně uloženými daty.
3. Navrhněte a implementujte aplikaci v Javě, která na základě nastavení stáhne data o subjektech z ARES a porovná je s daty v lokálním úložišti. V případě nového záznamu data uloží, v případě plné shody ponechá stávající, v případě neshody navrhne adekvátní řešení aktualizace dat založené na analýze z bodu 2.
4. Aplikaci řádně zdokumentujte a vhodným způsobem otestujte.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 27. března 2017

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra softwarového inženýrství

Bakalářská práce

Asistovaná aktualizace informací o klientech firmy na základě služby ARES

Jakub Šedý

Vedoucí práce: Ing. Miroslav Balík, Ph.D.

16. května 2017

Poděkování

Chtěl bych poděkovat svojí rodině za podporu při studiu a při tvorbě bakalářské práce. Dále bych chtěl poděkovat vedoucímu práce Miroslavu Balíkovi za pevné nervy, cenné rady a trpělivost.

Prohlášení

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

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

V Praze dne 16. května 2017 .....

České vysoké učení technické v Praze
Fakulta informačních technologií
© 2017 Jakub Šedý. Všechna práva vyhrazena.
Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Šedý, Jakub. *Asistovaná aktualizace informací o klientech firmy na základě služby ARES*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Bakalářská práce se zabývá vytvořením aplikace pro získávání dat pomocí služeb ARES Ministerstva financí České republiky . Cíle dosahuje pomocí analýzy těchto služeb, výběrem technologií, návrhem databáze a výsledné aplikace společně s implementací a testováním. Dále se práce zabývá párovaním získaných dat, které zajišťuje webová, uživatelsky obsluhovaná část aplikace. Výsledná aplikace je již nasazena v testovacím režimu a úspěšně získává důležitá data pro firmu EOS KSI Česká republika, která umožňují snadněji identifikovat klienty.

Klíčová slova ARES, XML Služby, Java, XPath, JPA, JSF

Abstract

Bachelor thesis focuses on creating application for data acquirement using services ARES from Ministry of Finance of the Czech Republic. The goal is achieved by analyzing ARES services, selecting technologies, designing the database; designing, implementing and testing the application. The next part of this work is about pairing the acquired data. This part is provided by user-operated web application. Application is already deployed in test environment

and is successfully getting important data for company EOS KSI Czech Republic, which facilitate identifying clients.

Keywords ARES, XML Services, Java, XPath, JPA, JSF

Obsah

Úvod **1**

1 Teoretická část **5**
1.1 XML 5
1.2 Porovnávání řetězců 7
1.3 Přibližné porovnávání řetězců 7
1.4 Java Bean Standard 9
1.5 Persistence 9
1.6 Objektově relační mapování (ORM) 10
1.7 Java Persistance API (JPA) 10
1.8 JavaServer Faces 15

2 Analýza **19**
2.1 Funkční požadavky 19
2.2 Nefunkční požadavky 20
2.3 Popis služby Ares 20
2.4 Aktualizace položek získaných z ARES 27

3 Návrh **31**
3.1 Databáze 31
3.2 Popis běhu aplikace pro stahování dat 35
3.3 Popis stavů 36
3.4 Use Case aplikace pro stahování 37
3.5 Use Case webové služby 37

4 Implementace **39**
4.1 AresUpdate 39
4.2 AresWorker 39
4.3 StandardQuery 39
4.4 RZPQuery 39

4.5 EntitySeeker 40
4.6 EntityUploader 40
4.7 Entitní třídy 40
4.8 Controller 40

5 Testování **41**
5.1 Zpětná vazba uživatelů 41
5.2 Nalezené chyby a úpravy 41
5.3 Získaná data 42

Závěr **43**

Literatura **45**

A Seznam použitých zkratek **47**

B Obsah přiloženého CD **49**

C Instalační příručka **51**
C.1 Aplikace pro stahování dat z ARES 51
C.2 Webová aplikace pro párování 52

D Uživatelská příručka webové aplikace **53**

Seznam obrázků

1.1 Struktura JPA architektury, převzato z [1] 15
1.2 Příklad JSF stránky 16

2.1 Struktura XML získaného po dotazu GET s vyznačenými elementy, s kterými aplikace pracuje 25
2.2 Struktura elementu dtt:Adresy s vyznačenými elementy, s kterými aplikace pracuje 26
2.3 Struktura elementu dtt:Osoby s vyznačenými elementy, s kterými aplikace pracuje 27
2.4 Porovnání dat ze služby ARES a klientské tabulky v databázi . . . 29

3.1 Logický model, popisující strukturu tabulek pro uloženi dat z ARES 31
3.2 Diagram aktivit popisující běh aplikace pro stahování dat 35
3.3 Stavový diagram popisující stav stažených dat 36
3.4 Use Case aplikace pro stahování 37
3.5 Use Case webové aplikace 38

D.1 Ukázka přihlašovaní webové aplikace 54
D.2 Příklad obrazovky porovnání klientů 55
D.3 Příklad obrazovky po spárování klienta 56

Úvod

Informace o klientech jsou pro většinu firem to nejdůležitější pro jejich podnikání. Často ovšem nejsou vyplňovány s takovou pečlivostí a někdy údaje chybí, nebo jsou nesprávné. Mnohé z chybějících dat lze na Internetu najít pomocí různých služeb.

Podobný problém řeší firma EOS KSI Česká republika, ve které pracuji a pro kterou je bakalářská práce určena. Pokusím se získat co nejvíce informací pomocí XML služeb ARES Ministerstva financí České republiky o jejich klientech.

Téma jsem si zvolil především z důvodu, že budu mít možnost sledovat využití aplikace a jejich přínosů v praxi.

V bakalářské práci se zabývám popisem použitých technologií, analýzou, návrhem, implementací a testováním. Aplikaci jsem vytvořil pomocí programovacího jazyku Java, který má mnoho možností, jak problematiku řešit. Data z XML souborů získávám pomocí dotazovacího jazyku XPath, které propojuji s databází pomocí objektového relačního mapování.

První část práce je převážně teoretická a popisuje použité technologie, které jsou nezbytné pro pochopení zbylých částí práce. Po ní následuje analýza problematiky a seznámení se se službami ARES. Dále se zabývám návrhem databáze a výsledného programu. Ve čtvrté části se věnuji implementaci aplikace a popisu nejdůležitějších částí a v poslední části, která se věnuje testování, se zabývám zkoušením výsledné aplikace a popisem zjištěných chyb.

Cíl práce

Cílem práce je nalezení chybějících informací ke klientům v databázi. Tento cíl se dále rozděluje na teoretickou a praktickou část.

Smyslem teoretické části práce je seznámení s nezbytnými technologiemi pro pochopení praktické části.

Cílem praktické části je získání dat z webové služby ARES Ministerstva financí České republiky a následné ukládání zpět do tabulky klientů. Práce naplňuje cíle pomocí analýzy služby ARES, návrhem databázové struktury odpovídající XML struktuře služby ARES, návrhem, implementací a testováním aplikace.

Literární rešerše

Získáváním dat pomocí služby ARES se jistě zabývá velké množství aplikací, protože k tomu byla služba určena. Žádná taková aplikace ovšem není veřejně přístupná a každá organizace s daty pracuje jinak. I tato práce je určená pouze firmě EOS KSI Česká republika.

Kapitola 1

Teoretická část

Následující kapitola se věnuje popisem technologií použitých v praktické části.

1.1 XML

XML znamená rozšiřitelný značkovací jazyk. XML je textově založený jazyk vyvinutý ze standarního obecného značkovacího jazyka (SGML). Je popisovaný jako třída datových objektů zvaných XML dokumenty. Tyto dokumenty skladují strukturu XML elementů, které jsou označeny pomocí XML tagů. Tyto XML elementy identifikují data a slouží převážně pro uložení a organizování dat než na jejich zobrazení. [2] Další známý značkovací jazyk je HTML, který byl vytvořen s cílem zobrazování dat, nikoliv pro skladování jako XML. Další výhoda XML je, že nemá předdefinované XML tagy.[3]

1.1.1 XML Parser

XML Parser slouží k transformaci XML dokumentu do formy, s kterou se programátorovi jednodušeji pracuje. Kontroluje správnost formátu XML dokumentu a umožnuje i jeho validaci. Java nabízí veliké množství XML parserů, nejznámnější jsou:

- Dom Parser,
- SAX Parser,
- JSoup Parser,
- JDOM Parser,
- StAX Parser,
- XPath Parser,
- DOM4J Parser. [4]

Pro vývoj této aplikace byl zvolen JSoup z těchto důvodů:

1. Zkušenost autora s prácí s touto knihovnou,
2. možnost stažení XML z webové stránky,
3. podpora XPath jazyka.[1]

1.1.2 XML Path Language (XPath)

Dotazovací jazyk pro nalezení elementů v XML dokumentu. Výrazy vypadají velice podobně jako v tradičním souborovém systému. Obsahuje přes 200 vestavěných funkcí a je velkou součástí XSLT standardu.[3]

1.1.2.1 XPath Syntaxe

Pro nalezení správných elementů existuje několik klíčových znaků:

- Název elementu – najde všechny elementy s tímto názvem,
- / - absolutní cesta (hledá od kořenového elementu),
- // - najde všechny elementy nehledě na to, kde jsou,
- . – najde samo sebe,
- .. – najde svého rodiče,
- @ - najde atributy,
- * - jakýkoliv element,
- @* - jakýkoliv atribut.[3]

Znaků a funkcí existuje mnohem více, ale v aplikaci jsou použity pouze některé z těchto znaků. Kompletní seznam je dostupný v XPath dokumentaci.[5]

1.1.2.2 XPath příklady

Pro pochopení jazyku je uvedeno několik praktických příkladů z aplikace:

- /odpovedi/Odpoved/VH/K – nalezne všechny elementy K, které jsou přesně pod touto cestou,
- PSC – nalezne všechny elementy s názvem PSC, které jsou potomky elementu, z kterého se XPath spouští,
- //PSC – nalezne všechny elementy s názvem PSC a je jedno, kde ve stromu se nachází,

[1] S pomocí knihovny XSoup

- odpovedi//PSC – nalezne všechny elementy s názvem PSC, které jsou kdekoliv pod elementem odpovedi,
- PSC[1] – nalezne první element s názvem D:PSC, který je potomek elementu, z kterého se XPath spouští,
- A/* - nalezne všechny potomky elementu A, který je potomek elementu, z kterého se XPath pouští.

1.2 Porovnávání řetězců

Při získávání nových informací je potřeba zjistit, ke kterým datům patří. Dost často se stává, že není žádný jednoznačný identifikátor, který spojuje nově získané informace s těmi starými. Poté nezbývá nic jiného, než porovnávat části, které mají společné. Tyto části nemusí být vždy 100% stejné a člověk je schopný rozhodnout o shodě. Lidské porovnávání je ale časově náročné. Ovšem automatický, lehko implementovatelný algoritmus neexistuje. Při vytváření takových algoritmů narazíme na konflikt mezi přesností a úplností. Je tedy potřeba se rozhodnout, zda chceme pouze správně spárovaná data, ale ne všechna, nebo všechna správná data za cenu toho, že se spárují i některé nesprávné.[6]

1.3 Přibližné porovnávání řetězců

Přibližné porovnávání řetězců se nezabývá otázkou „je řetězec A a B stejný?“, ale otázkou „jak moc si jsou řetězce A a B podobné?“. Na tuto otázku se dá odpovědět přesněji než pouze „ano“ nebo „ne“. Stupeň podobnosti pro textové porovnávání je nazýván editační vzdálenost, podle kterého se určuje, jak moc podobné dva řetězce jsou.[6]

1.3.1 Editační vzdálenost

Editační vzdálenost mezi dvěma řetězci definovaná jako nejmenší počet úprav jednoho řetězce, aby se shodoval s druhým řetězcem. Vypočítáním této vzdálenosti se zabývá mnoho lidí a existuje mnoho metod, lišících se v povolených operacích a cenách každé operace. Hodnota vzdálenosti se počítá takto:

1. Základní stav: Slovo, které se bude měnit.
2. Operace: Nejznámnější jsou vkládání, mazání, nahrazování, transpozice.
3. Cílový stav: Výsledné slovo, s kterým má být řetězec identický.
4. Vzdálenost: Výsledná cena.[7]

1.3.1.1 Hammingova vzdálenost

Umožnuje pouze nahrazování jako jedinou operaci úprav. Porovnávané řetězce musí být stejně dlouhé[7]

Příklad Řetězec „Jiri Pfof“ chceme porovnat s řetězcem „Jiří Poff“. Je potřeba nahradit všechna vyznačená písmena, vzdálenost je tedy 4.

1.3.1.2 Nejdelší společná podposloupnost

Hledá nejdelší společnou podposloupnost dvou řetězců. Má stejný efekt, jako kdyby bylo možné pouze mazat a přidávat znaky s cenou 1. Měří se délka společné podposloupnosti.[7]

Příklad Řetězec „Jaromír MarekTodth - ADDOSS“ chceme porovnat s řetězcem „Jaromír Toth“. Jak je vidět, první řetězec obsahuje celý druhý řetězec. Délka podposloupnost je pro tento příklad největší možná - 12 .

1.3.1.3 Levenshteinova vzdálenost

Patří k nejpoužívanějším a nejběžnější a jednotka vzdálenosti je často počítána právě s touto vzdáleností. Povolené úpravy jsou vkládání, mazání nebo nahrazování jednoho znaku. Každá operace má cenu 1.[7]

Příklad Řetězec „aronír Doth - ADDOSS“ chceme porovnat s řetězcem „Jaromír Toth“. Je potřeba nahradit zelená písmena, smazat všechna červená (včetně mezer), a přidat chybějící první písmeno J. Všechny operace mají cenu 1, celková vzdálenost je tedy 12.

1.3.1.4 Damerau–Levenshteinova vzdálenost

Velice podobná Levenshteinově vzdálenosti, ale dovoluje 4. operaci, kterou je transpozice dvou sousedících znaků. Tyto dva znaky spolu musí sousedit jak před úpravami, tak po úpravách.[7]

Příklad Řetězec „Jiri Pfof“ chceme porovnat s řetězcem „Jiří Poff“. Červeně vyznačená písmena je potřeba nahradit, ovšem zeleně vyznačená lze využít transpozici - prohození písmen. Celková vzdálenost je tedy 3.

1.3.1.5 Vážená/generalizovaná vzdálenost

Opět velice podobná Levenshteinově vzdálenosti, ale zde se ohodnocují operace cenou.[7]

Příklad Řetězec „aronír Doth - ADDOSS“ chceme porovnat s řetězcem „Jaromír Toth“. Cena operací je 1 pro nahrazení, 2 pro mazání a 3 pro vkládání. Je potřeba nahradit zelená písmena, smazat všechna červená (včetně mezer) a přidat chybějící první písmeno J. Celkově se maže devět znaků, dva se nahrazují a jeden se vkládá, celková vzdálenost je tedy 23.

1.4 Java Bean Standard

Následující kapitoly (Java Persistance API a JavaServer Faces) využívají třídy, které jsou brány jako Java Bean, proto musí dodržovat konvence:

1. Každá třída bean musí mít konstruktor bez parametrů.

2. Instanční proměnné se dají rozdělit mezi Boolean a non-Boolean.

3. Non-Boolean mají getter a setter metody.

4. Boolean mají setter a is metody.

5. Každá getter metoda by měla začínat slovem get následovaná názvem proměnné s velkým prvním písmenem. Instanční proměnná city by tedy měla getter getCity().

6. Každá setter metoda by měla začínat slovem set a, stejně jako v getter metodě, následovaná jménem proměnné s velkým prvním písmenem. Pro proměnnou city tedy setCity(String city).

7. Každá is metoda by měla začínat slovem is, následovaná názvem proměnné s velkým písmenem. Pro proměnnou rightPerson by tedy byla isRightPerson().[8]

1.5 Persistence

Persistence v informatice odkazuje na objekty a procesy charakteristické tím, že existují i po ukončení procesu, který je ukončil, nebo byl vypnut stroj, na kterém běžely. Pokud je potřeba mít objekt persistentní, je nutno ho uložit na stálé a bezpečné místo (například harddisk). Persistentní objekt by měl být vymazán pouze tehdy, když je určen k smazání. Pokud tedy například spadne program nebo se vypne počítač, persistentní objekty by měly zůstat nedotčené. Persistentní vlákna a procesy jsou takové procesy, které nemohou být vypnuté nebo mrtvé, což také většinou platí pro procesy klíčové k běhu systému.[9]

1.6 Objektově relační mapování (ORM)

„Mechanismus, který umožnuje adresovat, přistupovat a manipulovat s objekty bez potřeby zvážení, jak se tyto objekty vážou k jejich zdrojům. ORM umožnuje programátorům udržovat konzistentní pohled nad objekty v čase, i přesto, že se zdroje, přijímací schránky nebo aplikace změní. Založené na abstrakci, ORM se stará o mapování dat mezi množinami objektů a relačními databázemi, XML repozitáři nebo jinými zdroji dat, zatímco zakrývá často měnící se detaily v relačních rozhraních vývojářů a kódu, který vytváří. “[10, přeloženo autorem]

Mezi hlavní výhody ORM patří:

- Je eliminováno všechno ruční mapování.
- Lze generovat kód pomocí relačního nebo doménového modelu.
- Existuje mnoho knihoven, které propojení usnadňují.
- Podporuje objektově orientované programování a modelování.
- Je nezávislé na databázi a schématu.[11]

1.7 Java Persistance API (JPA)

Pro uchování persistentích dat byla zvolena specifikace Java Persistance API. Tato specifikace velice usnadňuje synchronizaci persistentních dat s databází pomocí objektově relačního mapování. Jednoduše nám stačí udělat entitní třídy (lze je i vygenerovat) odpovídající tabulkám a správně použít anotace pro mapování.[12] Důvody, proč bylo zvoleno právě JPA:

- Standard Javy
- Zjednodušení a přehlednost kódu

1.7.1 JPA Provider

JPA je specifikace, existuje tedy mnoho poskytovatelů, které tuto specifikaci naimplementovali (JPA Provider). Nejznámější jsou:

- Hibernate,
- EclipseLink,
- Toplink,
- Spring Data JPA,
- DataNucleus,

- ObjectDB,
- Apache OpenJPA.

1.7.2 Mapování

Mapovat třídy na tabulky v databázi lze dvěma způsoby:

1. Pomocí anotací.
2. Pomocí atributů v xml souboru.

Používání anotací je Standard a proto jsou v práci tabulky mapovány právě pomocí anotací.

1.7.2.1 Anotace

Anotace je v Javě zápis, jak přiřadit nějakému elementu (třída, metoda, proměnná) jakýsi příznak, metadata, informaci mimo běžný kód. Anotaci poznáme v Java kódu tak, že má před názvem znak "@", který symbolizuje kompilátoru, že se jedná o anotaci. [13]

Nejjednodušší anotace je tohoto typu:

@Entity

Anotace mohou mít elementy, kterým můžeme zadávat hodnoty. To se dělá takto:

@Entity (tableName = “CZ_PERSON“)

tableName je jméno elementu a CZ_PERSON je hodnota, pokud chceme zadat více parametrů, tak je potřeba je oddělit čárkou:

@Entity (tableName = "CZ_PERSON", primaryKey = "PERSON_ID")

Pokud mají anotace pouze jeden element, je konvence ho pojmenovat „value“ a pokud ho chceme nastavit, stačí zadat do závorek hodnotu:

@InsertNew(“yes“)

Použité anotace s elementy Použité anotace, které byly použity v této práci jsou popsány níže. Kompletní seznam dostupný v Oracle dokumentaci[14].

Entity deklaruje třídu jako entitu nebo tabulku.

Table slouží k nastavení elementů tabulky:

- name: název tabulky,
- schema: název schématu.

GeneratedValue specifikuje, jak nastavit identitu atributu. Použity byly tyto elementy:

- strategy: strategie generování identity (např. automatická),
- generator: název generátoru.

SequenceGenerator specifikuje hodnotu z GeneratedValue. Použity byly tyto elementy:

- name: název generátoru (zadaný v GeneratedValue),
- sequenceName: jméno sekvence, z které se berou hodnoty,
- allocationSize: velikost alokace dané sekvence.

Embedded specifikuje instanční proměnné, které jsou vestavěné dohromady (např. COMPANY_ID a PERSON_ID v PERSON_IN_COMPANY tabulce, pro unikátnost).

EmbeddedId ukazuje, že primární klíč je tvořený z více proměnných.

Id označí instanční proměnnou jako primární klíč.

Column přiřadí sloupec z tabulky instanční proměnné. Použit byl element name, který nastavuje název sloupce.

Transient označuje instanční proměnou, která není persistentní (nebude se ukládat).

ManyToOne označuje asociaci na jinou tabulku ve formě N:1.

JoinColumn určuje, jakým sloupcem se relačně váže v ManyToOne vztahu pomocí elementu name, který nastavuje název sloupce, přes který se spojují.

OneToMany definuje asociaci na jinou tabulku ve formě 1:N, používá tyto elementy:

- mappedBy: Používá se v OneToMany, aby ukázal, že klíč je na druhé straně.
- cascade – vkláda se CascadeType, který určuje, jestli máme provádět akce i s relačně spojenými entitami.

1.7.3 Připojovací řetězec

Řetězec inicializačních proměnných pro připojení k úložišti dat, který jednoduše umožnuje skladovat informace v aplikaci nebo ho mezi aplikacemi posílat. Bez Připojovacího řetězce by se musela poslat struktura několika proměnných pro připojení. V aplikaci je použit JDBC ovladač pro připojení k databázi. Každá databáze má trochu jinou formu připojovacího řetězce[15]. Nás zajímají pouze připojovací řetězce pro oracle, které vypadají takto:

1. driverName:driverType:username/password@DBadress:Port:SID,
2. driverName:driverType:@DBadress:Port:SID,
3. driverName:driverType:@DBadress:Port/Service.

Kde platí:

- driverName: název driveru,
- driverType: typ driveru,
- username: jméno uživatele,
- password: heslo uživatele,
- DBadress: adresa databázového serveru,
- Port: port databázového serveru,
- SID: SID databázového serveru,
- Service: název Oracle služby.

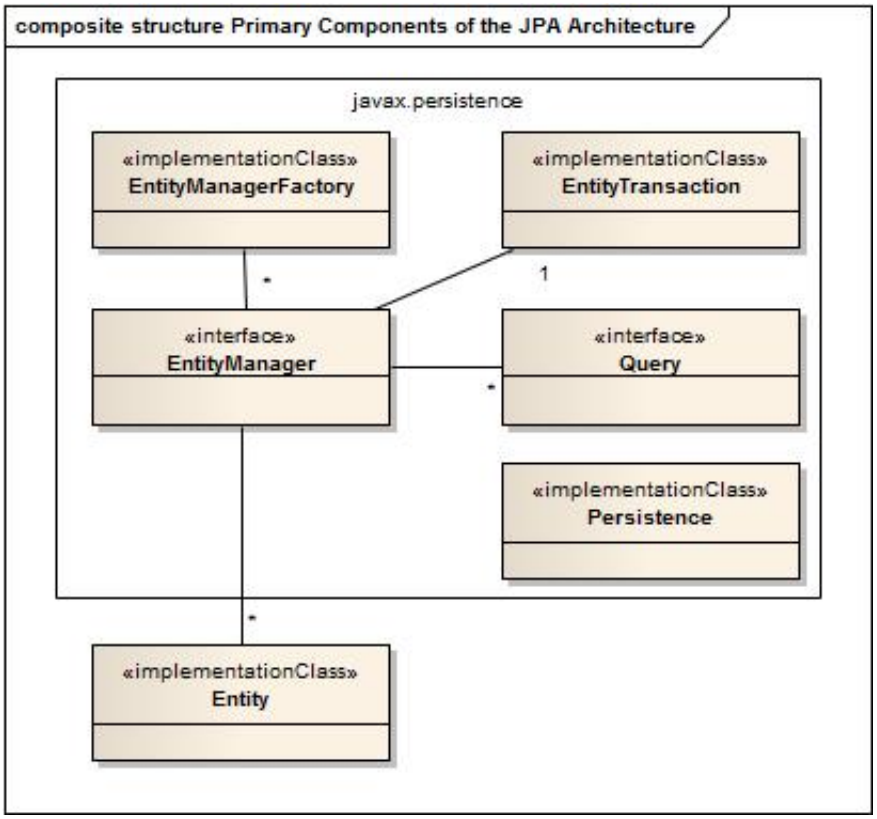
1.7.3.1 Persistence.xml

Tento soubor hraje velice důležitou roli v konceptu JPA. Definují se zde připojení k databázi a musí zde být uvedeny všechny třídy, které jsou anotovány pomocí @Entity. V souboru nás zajímá element persistence-unit, kterých může být v xml několik a každá má vlastní entity a properties. V této aplikaci stačí jedna. Do elementů class je potřeba dát všechny entity a v properties se nastavuje právě dané připojení pomocí proměnných:[12]

- javax.persistence.jdbc.driver – ovladače pro danou databázi,
- javax.persistence.jdbc.url – Connection String pro připojení k databázi,
- javax.persistence.jdbc.user – Uživatel, pod kterým chceme dělat operace v aplikaci,
- javax.persistence.jdbc.password – Heslo k uživateli.

1.7.4 Připojení k databázi

O připojení k databázi se stará rozhraní EntityManager, které spravuje persistentní operace a vytváří databázové dotazy. Je kontrolovano třídou EntityManagerFactory, která je schopná vytvářet a starat se o několik EntityManageru najednou. Každé rozhraní EntityManager má právě jednu EntityTransaction, která udržuje všechny operace. Při vytváření EntityManagerFactory třídy Persistance je potřeba zadat jméno persistence-unit, které je uvedené v persistence.xml.[12]



Obrázek 1.1: Struktura JPA architektury, převzato z [1]

1.7.5 Java Persistence Query Language (JPQL)

JPQL jazyk je jeden ze základních součástí Java Persistence, který umožnuje psát přenosné dotazy, které fungují nezávisle na úložišti dat. Využívá syntaxi podobnou SQL pro vybrání objektů nebo hodnot entit. Tyto dotazy mohou být:

- staticky definované v anotaci (@NamedQuery(name="CzAddress.findAll", query="SELECT c FROM CzAddress c")
- dynamicky vytvořené v kódu (SELECT p FROM CzPerson p WHERE p.firstName=?1 AND p.lastName=?2 AND p.birthDate=?3) [12]

1.8 JavaServer Faces

JavaServer Faces(JSF) je specifikace, zjednodušující vývoj sofistikovanější webové aplikace uživatelského rozhraní. Toho dosahuje pomocí modelu dobře svázaného s procesováním životního cyklu žádostí. Umožňuje:

- Java programátorům vyvíjet aplikaci bez větší znalosti HTTP,
- autorům stránek pracovat s uživatelským rozhraním bez znalosti programování.

Přestože je HTML značkovací jazyk pro většinu webových aplikací, JSF není limitované pouze na HTML nebo jiný značkovací jazyk. JSF model je velice podobný tomu, který byl využíván pro GUI samostatných aplikací.[16]

JSF stránky jsou vytvořené v XML formátu. Obsahující tagy z definovaných jmenných prostorů. Komunikují s kódem pomocí JSF výrazového jazyka. Pomocí těchto stránek se generují výsledné stránky.[12]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core">

<ui:composition template="/WEB-INF/templates/BasicTemplate.xhtml">
    <ui:define name="content">
        <h:form>
            <h:panelGrid columns="2">
                <h:outputText value="Name"></h:outputText>
                <h:inputText value="#{loginBean.name}"></h:inputText>
                <h:outputText value="Password"></h:outputText>
                <h:inputSecret value="#{loginBean.password}"></h:inputSecret>
            </h:panelGrid>
            <h:commandButton value="Login" action="#{loadNextBean.loadNext}"></h:commandButton>
        </h:form>
    </ui:define>
</ui:composition>
</html>
```

Obrázek 1.2: Příklad JSF stránky

1.8.1 Jmenné prostory

V elementu html jsou definovány jmenné prostory pomocí atributu xmlns. Tento atribut má hodnotu webové stránky s definovanými tagy. Protože může více jmenných prostorů obsahovat stejný tag, je potřeba je rozlišit. Rozlišují se pomocí hodnoty za dvojtečkou elementu xmlns. K elementům jmenného prostoru xmlns:ui se potom přistupuje pomocí ui:define, kde define je jméno elementu a ui je název jmenného prostoru. [12]

1.8.2 JSF výrazový jazyk

JSF výrazový jazyk umožnuje autorům stránek jednoduše použít výrazy pro dynamický přístup k datům z JavaBean komponent. Na obrázku 1.2 se takto přistupuje ke komponentám loginBean a loadNextBean. Konkrétně se zadávají

hodnoty do formuláře, které se uloží při kliknutí na tlačítko Login. Výrazový jazyk je používán k:

- Okamžitému vyhodnocení výrazu,
- získání nebo nastavení dat,
- zavolání metod.[17]

Kapitola 2

Analýza

2.1 Funkční požadavky

F_1 **Vytvoření databázového modelu**

Data je potřeba ukládat a v databázi firmy EOS KSI Česká republika není žádná struktura pro data ze služby ARES. Je potřeba vytvořit takový model, který odpovídá XML struktuře v ARESu.

F_2 **Evidence získaných dat**

Získaná data je potřeba evidovat v databázi pro další použití.

F_3 **Možnost lehké změny konfigurace**

Pokud se změní některé vstupní proměnné (proxy,limity pro ARES nebo připojení do databáze), je možné je jednoduše změnit bez zkompilování programu.

F_4 **Hlídání počtu dotazů**

Počet dotazů je omezen na straně ARESu a je důležité toto omezení hlídat.

F_5 **Automatické dotazování na web**

Aplikaci pro dotazování a stahování dat bude možno spouštět bez zásahu uživatele jako denní automatickou službu.

F_6 **Vyhodnocování výsledků a ukládání pouze fyzických osob**

Aplikace bude umět rozeznat, jestli se jedná o osobu, která firmu zajímá a poté jí uloží do databáze. V opačném případě nikoliv.

F_7 **Párování dat pouze se shodnými uživateli**

Nesmí se stát, aby aplikace doplnila ke klientovi nesprávné údaje.

F_8 **Aplikace bude zapisovat výstupy do souborů**

Aplikace bude při běhu jako služba zapisovat svojí aktivitu do souboru, aby ulehčila hledání chyb a jejich opravování.

F_9 **Určování správnosti výsledku pomocí webové služby**

Pro párování dat mezi získanými daty a tabulkou klientů bude sloužit webová služba obsluhovaná osobou.

2.2 Nefunkční požadavky

N_1 **Počet dotazů omezen službou ARES**

Nelze položit více dotazů, než ARES dovoluje a počet získaných dat je omezen právě tímto pravidlem. K datu 1.5. 2017 je tento limit 1000 ve špičce (8.-18. hodina) a 5000 mimo špičku.

N_2 **Jednoduchá rozšiřitelnost pro získávání více dat z ARES**

Jednoduše lze přidělat další funkcionality Například přidat získávání dat z dalších služeb, které ARES nabízí.

N_3 **Webová služba poběží na všech běžných internetových prohlížečích**

Všechny běžně používané internetové prohlížeče[1] zobrazí webovou službu správně

2.3 Popis služby Ares

„ARES je IS, který zpřístupňuje veřejné údaje o ekonomických subjektech z informačních systémů (zdrojů) veřejné správy. Obsahuje údaje ze základních (majoritních) zdrojů, které jsou formou odkazů doplněny údaji z dalších zdrojů. Při zpracování se používají též kontrolní zdroje.“[18]

2.3.1 Popis zdrojů a aplikací ARES

"Majoritní zdroje:

- *Veřejný rejstřík (VR), vedený rejstříkovými soudy,*
- *Živnostenský rejstřík (RŽP), vedený Ministerstvem průmyslu a obchodu ČR,*
- *Registr ekonomických subjektů (RES), vedený Českým statistickým úřadem,*

[1] Google Chrome, Internet Explorer, Mozila Firefox, Safari, Opera

- *Registr církví a náboženských společností (RCNS), vedený Ministerstvem kultury ČR,*
- *Registr zdravotnických zařízení (RZZ), vedený Ústavem zdravotnických informací a statistiky ČR,*
- *Seznam občanských sdružení a spolků (OSS), vedený Ministerstvem vnitra ČR,*
- *Evidence zemědělského podnikatele (EZP), která je vedena Ministerstvem zemědělství ČR,*
- *Seznam politických stran a hnutí (PSH), vedený Ministerstvem vnitra ČR,*
- *Rejstřík škol a školských zařízení (RŠ), vedený Ministerstvem školství a tělovýchovy ČR.*

Další zdroje:

- *Registr plátců daně z přidané hodnoty (DPH, SkDPH, údaje o nespolehlivém plátci DPH), vedený Finanční správou ČR,*
- *Registr plátců spotřební daně (SD), vedený Celní správou ČR,*
- *Účelový registr organizací systému ARIS (RARIS), vedený Ministerstvem financí ČR,*
- *Centrální evidence dotací z rozpočtu (CEDR), která je vedena Ministerstvem financí ČR,*
- *Centrální evidence úpadců (CEU), která je vedena Ministerstvem spravedlnosti ČR,*
- *Insolvenční rejstřík (IR), který je veden Ministerstvem spravedlnosti ČR."*[18]

2.3.2 Podmínky provozu

„Účelem www aplikace ARES Ministerstva financí je souhrnně zpřístupnit údaje z informačních systémů pro vedení registrů a evidencí veřejné správy o ekonomických subjektech. ARES přehledně zpřístupňuje údaje, přebírané ze zdrojových registrů do databáze ARES a současně umožňuje přímé přepnutí do www aplikací orgánů veřejné správy, které příslušné informační systémy provozují, pokud tyto aplikace již existují.

Všechny údaje, které ARES zpřístupňuje, mají pouze informativní charakter a nemají charakter úřední listiny. Jejich výpis (prostřednictvím ARES) proto nemůže být použit jako průkazní materiál pro úřední jednání.

Vzhledem k charakteru aplikace ARES nenese Ministerstvo financí odpovědnost za aktuálnost a úplnost údajů z databáze ARES ani za nepřetržitý provoz www aplikací pro zpřístupnění údajů z informačních systémů zdrojových registrů ani za korektnost a aktuálnost jimi zpřístupňovaných informací. Ministerstvo financí neodpovídá za případné škody způsobené použitím údajů získaných prostřednictvím aplikace ARES.

Účelem provozování aplikace je poskytnout rychlé a obecně dostupné informace o jednotlivých subjektech. K tomuto účelu není pro běžného uživatele přístup k aplikaci ARES omezen. S ohledem na charakter provozu ARES a jeho zabezpečení si Ministerstvo financí vyhrazuje právo omezit nebo znemožnit přístup k www aplikaci ARES uživatelům, kteří:

- *odešlou k vyřízení více než 1000 dotazů v době od 8:00 hod. do 18:00 hod.,*
- *odešlou k vyřízení více než 5000 dotazů v době od 18:00 hod. do 8:00 hod. rána následujícího dne,*
- *se snaží o porušení bezpečnostní ochrany www serveru Ministerstva financí,*
- *opakovaně posílají nesprávně vyplněné dotazy,*
- *opakovaně posílají stejné dotazy,*
- *mají větší počet současně zadaných dotazů (pro automatizované XML dotazy),*
- *obcházejí povolené limity využíváním dotazování z většího množství IP adres,*
- *automatizovaně propátrávají databázi náhodnými údaji nebo generují většinu nesprávných dotazů.*

Zejména u automatizovaného dotazování přes XML služby doporučujeme dbát na správné vyplňování parametrů platnými a smysluplnými hodnotami. V opačném případě může dojít k zamezení přístupu do IS ARES. Obnovení přístupu k údajům aplikace ARES po jeho případném zákazu bude řešeno zásadně na základě dostatečných písemných záruk, které uživatel dohodne s Ministerstvem financí.“[18]

2.3.3 XML služby

„Součástí informačního systému ARES je i XML rozhraní pro vyhledání subjektu a zpřístupnění jeho veřejných údajů ze zdrojových registrů. Služby XML může využívat každý, pokud bude respektovat podmínky provozu. Jsou nastaveny určité limity pro počty dotazů, které může jeden uživatel během dne do

ARES poslat. V době od 8:00 do 18:00 se předpokládá větší množství dotazujících uživatelů, proto je pro tuto dobu omezení počtu dotazů přísnější. Hodnoty limitů pro denní a noční provoz jsou uvedeny v podmínkách provozu.

V případě odůvodněné mimořádné potřeby využívat XML služby ARES v nadlimitním režimu je možné:

1. *požádat ministerstvo financí na adrese ares@mfcr.cz o možnost zvýšení limitu a sdělit předpokládaný počet dotazů,*
2. *dodat seznam IP adres, ze kterých se bude na XML služby ARES přistupovat,*
3. *dodat kontaktní e-mailové adresy pro možnost zasílání upozornění o případných výpadcích v provozu IS ARES.*

Kompletní technická dokumentace je na vyžádání na adrese ares@mfcr.cz."[18]

2.3.3.1 Popis metod GET

„Následující seznam nabízí výběr aplikací, které poskytují data z ARES v XML pro dotazy metodou GET bez nutnosti digitálního podpisu. Název služby je současně příkladem dotazu do příslušné služby.

- *Standard - Základní dotaz na výpis identifikačních údajů,*
- *Seznam_Reg – dotaz na seznam registrací subjektu ve zdrojích VR,RES a RŽP,*
- *Vypis_OR – výpis dat z Veřejného rejstříku,*
- *Vypis_RZP – výpis dat z Registru živnostenského podnikání,*
- *Vypis_RES – výpis dat ze statického registru RES,*
- *Vypis_CNS – výpis z Registru církví a náboženských společností,*
- *Vypis_CEU – výpis dat z Centrální evidence úpadců,*
- *Vypis_OSS – výpis dat ze Seznamu občanských sdružení a spolků,*
- *Vypis_RZZ – výpis dat z Registru zdravotnických zařízení,*
- *Vypis_SZR – výpis dat ze zemědělského registru,*
- *Vypis_PSH – výpis dat z Registru politických stran a hnutí,*
- *Vypis_SKO – výpis dat z Rejstříku škol,*
- *přehled ekonomických subjektů,*
- *dotaz na standardizovanou adresu,*
- *přehled změn ekonomických subjektů.“*[18]

2.3.3.2 Standard

Pro vstup a správné vyhodnocení dotazu Standard je potřeba zadat parametr ico nebo obchodni_firma. Seznam všech parametrů dostupný na stránkách ARESu.

příklady volání

1. http://wwwinfo.mfcr.cz/cgi-bin/ares/darv_std.cgi?ico=27074358,
2. http://wwwinfo.mfcr.cz/cgi-bin/ares/darv_std.cgi?obchodni_firma =ASSECO a.s.,
3. http://wwwinfo.mfcr.cz/cgi-bin/ares/darv_std.cgi?obchodni_firma =ASSECO+a.s.,
4. http://wwwinfo.mfcr.cz/cgi-bin/ares/darv_std.cgi?ico=00216151&jazyk =cz&xml=1&aktivni=false.

Výstup dotazu V dotazu standard nás zajímají elementy:

- Priznaky_subjektu – informace o subjektu popsané níže,
- Pocet_zaznamu – návratová hodnota kterou je třeba kontrolovat, jestli byl nalezený subjekt pro konkrétní IČO.

Priznaky_subjektu se skládá z třiceti znakového řetězce obsahující hodnoty:

- A – platná registrace,
- Z – zaniklá registrace,
- S – v pozici č. 6 označuje skupinovou registrace DPH,
- P – v pozici č. 15 označuje pozastavenou činnost,
- E – v pozici č. 22 označuje, že existuje záznam v Insolvenčním rejstříku. Nutno prověřit stav řízení,
- N (nebo jiný znak) – není v evidenci.

Pozice znaku určuje typ příznaku, je dostupných na stránkách služeb ARES. [18]

2.3.3.3 Vypis_rzp

Dotaz má povinný parametr ico Příklady volání:

1. http://wwwinfo.mfcr.cz/cgi-bin/ares/darv_rzp.cgi?ico=27074358& xml=0&ver=1.0.4
2. http://wwwinfo.mfcr.cz/cgi-bin/ares/darv_rzp.cgi?ico=27074358& xml=1&rozsah=2

Výstup dotazu Ve výsledném XML výsledku jsou důležité hodnoty v elementech vyznačených na obrázku:


```
− <are:Ares_odpovedi odpoved_datum_cas="2017-05-01T17:10:44" odpoved_pocet="1"
  odpoved_typ="Vypis_RZP" vystup_format="XML" xslt="klient"
  validation_XSLT="http://wwwinfo.mfcr.cz/ares/xml_doc/schemas/ares/ares_odpovedi.xsl"
  xsi:schemaLocation="http://wwwinfo.mfcr.cz/ares/xml_doc/schemas/ares/ares_answer_rzp
  /v_1.0.4 http://wwwinfo.mfcr.cz/ares/xml_doc/schemas/ares/ares_answer_rzp/v_1.0.4
  /ares_answer_rzp_v_1.0.4.xsd" Id="ares">
  − <are:Odpoved>
      <dtt:Pomocne_ID>1</dtt:Pomocne_ID>
    − <dtt:Vysledek_hledani>
        <dtt:Kod>1</dtt:Kod>
      </dtt:Vysledek_hledani>
      <dtt:Pocet_zaznamu>1</dtt:Pocet_zaznamu>
    + <dtt:Uvod></dtt:Uvod>
    − <dtt:Vypis_RZP>
      − <dtt:Zakladni_udaje>
          <dtt:Stav>A</dtt:Stav>
          <dtt:Datum_zmeny>2015-06-17</dtt:Datum_zmeny>
          <dtt:ICO>27074358</dtt:ICO>
          <dtt:Obchodni_firma>Asseco Central Europe, a.s.</dtt:Obchodni_firma>
        − <dtt:Pravni_forma>
            <dtt:Kod_PF>121</dtt:Kod_PF>
            <dtt:Nazev_PF>Akciová společnost</dtt:Nazev_PF>
            <dtt:PF_osoba>P</dtt:PF_osoba>
            <dtt:Text>právnická osoba tuzemská</dtt:Text>
          </dtt:Pravni_forma>
        + <dtt:Zivnostensky_urad></dtt:Zivnostensky_urad>
          <dtt:Prvni_zivnost>2004-03-22</dtt:Prvni_zivnost>
          <dtt:Vsech_zivnosti>26</dtt:Vsech_zivnosti>
          <dtt:Aktivnich_zivnosti>5</dtt:Aktivnich_zivnosti>
          <dtt:Aktivnich_provozoven>ano</dtt:Aktivnich_provozoven>
        </dtt:Zakladni_udaje>
      + <dtt:Adresy></dtt:Adresy>
      + <dtt:Osoby></dtt:Osoby>
      + <dtt:Zivnosti></dtt:Zivnosti>
      </dtt:Vypis_RZP>
    </are:Odpoved>
  </are:Ares_odpovedi>
```


Obrázek 2.1: Struktura XML získaného po dotazu GET s vyznačenými elementy, s kterými aplikace pracuje

- dtt:Obchodni_firma: název společnosti,
- dtt:Kod_PF: kód právní formy,
- dtt:Adresy: adresy společnosti,
- dtt:Osoby: osoby spojené se společností.

Adresy Pod elementem dtt:Adresy jsou adresy, kde jsou důležité tyto elementy(žádný z elementů není povinný):

```
– <dtt:Adresy>
  – <dtt:Adresa kvadres="1">
      <dtt:ID_adresy>29512295</dtt:ID_adresy>
      <dtt:Nazev_obce>Praha</dtt:Nazev_obce>
      <dtt:Nazev_casti_obce>Michle</dtt:Nazev_casti_obce>
      <dtt:Nazev_ulice>Budějovická</dtt:Nazev_ulice>
      <dtt:Cislo_domovni>778</dtt:Cislo_domovni>
      <dtt:Typ_cislo_domovni>1</dtt:Typ_cislo_domovni>
      <dtt:Cislo_orientacni>3</dtt:Cislo_orientacni>
      <dtt:Cislo_do_adresy>3a</dtt:Cislo_do_adresy>
      <dtt:PSC>14000</dtt:PSC>
    </dtt:Adresa>
  </dtt:Adresy>
```

Obrázek 2.2: Struktura elementu dtt:Adresy s vyznačenými elementy, s kterými aplikace pracuje

- dtt:Nazev_obce: název obce,
- dtt:Nazev_casti_obce: název části obce,
- dtt:Nazev_ulice: název ulice,
- dtt:Cislo_domovni: číslo domovní,
- dtt:Cislo_orientacni: orientační číslo,
- dtt:PSC: poštovní směrovací číslo.

Osoby Pod elementem dtt:Osoby jsou osoby, kde jsou důležité tyto elementy:

```
− <dtt:Osoby>
  − <dtt:Osoba>
      <dtt:Role>S</dtt:Role>
      <dtt:Jmeno>David</dtt:Jmeno>
      <dtt:Prijmeni>Stoppani</dtt:Prijmeni>
      <dtt:Datum_narozeni>1981-08-16</dtt:Datum_narozeni>
      <dtt:Titul_pred>Ing.</dtt:Titul_pred>
    + <dtt:Bydliste>
        <dtt:ID_adresy>29512296</dtt:ID_adresy>
        <dtt:Nazev_obce>Zlín</dtt:Nazev_obce>
        <dtt:Nazev_casti_obce>Zlín</dtt:Nazev_casti_obce>
        <dtt:Nazev_ulice>Podlesí IV</dtt:Nazev_ulice>
        <dtt:Cislo_domovni>4960</dtt:Cislo_domovni>
        <dtt:Typ_cislo_domovni>1</dtt:Typ_cislo_domovni>
        <dtt:PSC>76005</dtt:PSC>
      </dtt:Bydliste>
    </dtt:Osoba>
  + <dtt:Osoba></dtt:Osoba>
  + <dtt:Osoba></dtt:Osoba>
  </dtt:Osoby>
```


Obrázek 2.3: Struktura elementu dtt:Osoby s vyznačenými elementy, s kterými aplikace pracuje

- dtt:Role: role vůči firmě,
- dtt:Jmeno: křestní jméno,
- dtt:Prijmeni: příjmení,
- dtt:Datum_narozeni: datum narození ve formátu RRRR-MM-DD,
- dtt:Bydliste: trvalé bydliště se stejnou strukturou jako adresa.

2.4 Aktualizace položek získaných z ARES

Tabulka klientů, s kterou se párují data ze služby ARES má momentálně pouze čtyři sloupce shodné se službou: IČO, křestní jméno, příjmení a datum narození. Aplikace spouští dohledávání pouze pro klienty, které nemají vyplněné datum narození a rodné číslo. Na základě je IČO je volán dotaz. Může nastat několik situací:

1. IČO v tabulce klienta je správně:
   a) Křestní jméno a příjmení se shodují.
   b) Křestní jméno chybí.
   c) Příjmení chybí.
   d) Křestní jméno nebo příjmení se neshodují.

2. IČO je v tabulce klienta špatné:

   a) Dotaz není nalezen.

   b) Najde se IČO jiné společnosti.

Pouze v situaci, kdy jsou všechny společné informace(1a), je jisté, že by odpovědná osoba určitě zvolila spárování, a proto to může dělat aplikace sama.

V ostatních případech je velice riskantní nechat aplikaci párovat data automaticky. Tento problém vyřeší webová aplikace obsluhovaná osobou, která bude rozhodovat, jestli jsou klienti stejní nebo ne. Aplikace bude na webu řadit osoby podle editační vzdálenosti. Bude se tedy zobrazovat nejpodobnější nezpracované křestní jméno a příjmení.

2.4.1 Porovnání dat v tabulkách

Pro párování je potřeba nalést vhodný algoritmus pro výpočet editační vzdálenosti mezi daty ze služby ARES a tabulkou klientů, proto je potřeba udělat porovnání získaných dat.

Data v databázi EOS-KSI

CLIENT_FIRST_NAME	CLIENT_LAST_NAME	CLIENT_COMPANY_NAME	CLIENT_REG_...
Aleš	Maxilon	(null)	10048227
(null)	PENZION - RESTAURANT U RYTÍŘE	(null)	10048227
Lukáš	Zeman	(null)	10048227
(null)	VÁCLAV PABLICH - PASO	(null)	10049410
Radoslav	Babuška	(null)	10049410
Václav	Pablich	(null)	10049410
Zdeněk	ZDENĚK VAŘEČKA - AOV	(null)	16459636
(null)	RASTISLAV FEFKO - RM STAV	(null)	25266004
Robert	Demeter	(null)	25266004
ALITEX I.E., s.r.o.	ALITEX I.E. S.R.O.	(null)	25942701
Antonín	Juráň - juma	(null)	42075424
JUMA - služby elektro	(null)	(null)	42075424
Petr	Polka - rydaz	(null)	42820570
PF INTERIER	Mgr Jiří Pfof	(null)	44901305
Ladislav Didi	AUTOBAZAR DIDI-KARS	(null)	46738550
Jaromír Toth - ADOSS	Toth- adoss	(null)	47851902
Vlastislav	Hezoučký, vh tours	(null)	63798638
Diviš	J. -fa+spol.	(null)	66728061
Jan	Horák ročník 60	(null)	68495901

Data získaná z ARES

COMPANY_REG_NO	COMPANY_NAME	PERSON_FIRST_NAME	PERSON_LAST_NAME
10048227	Aleš Maxilon	Aleš	Maxilon
10048227	Aleš Maxilon	Aleš	Maxilon
10048227	Aleš Maxilon	Aleš	Maxilon
10049410	Václav Pablich	Václav	Pablich
10049410	Václav Pablich	Václav	Pablich
10049410	Václav Pablich	Václav	Pablich
16459636	Zdeněk VAŘEČKA	Zdeněk	VAŘEČKA
25266004	Rastislav Fefko	Rastislav	Fefko
25266004	Rastislav Fefko	Rastislav	Fefko
25942701	ALITEX I.E. s.r.o. v likvidaci	Ali	Shayto
42075424	Antonín Juráň	Antonín	Juráň
42075424	Antonín Juráň	Antonín	Juráň
42820570	Václav Pohořalý	Václav	Pohořalý
44901305	Mgr. Jiří Pfof	Jiří	Pfof
46738550	Ladislav Didi	Ladislav	Didi
47851902	Jaromír Toth	Jaromír	Toth
63798638	Vlastislav Hezoučký,FALKO TRAVEL	Vlastislav	Hezoučký
66728061	Jaroslav Diviš	Jaroslav	Diviš
68495901	Jan Horák	Jan	Horák

Obrázek 2.4: Porovnání dat ze služby ARES a klientské tabulky v databázi

Jak je vidět v tabulce, klientská data mohou mít ve sloupci LAST_NAME a FIRST_NAME různá data. Z tohoto důvodu se bude vzdálenost měřit pro složené řetězce těchto dvou sloupců s řetězci FIRST_NAME a LAST_NAME ze služby ARES.

Pro výpočet editační vzdálenosti byl z tohoto důvodu zvolen algoritmus pro nalezení nejdelší podposloupnost, který bude spuštěn pro složené řetězce klientského křestního jména a příjmení odděleného mezerou. Tento řetězec bude porovnáván s křestním jménenem a příjmením, složeným oběma kombinacemi ("příjmení jméno"a "jméno příjmení"). Získané dvě hodnoty se porovnají a větší hodnota se uloží jako jejich editační vzdálenost.

2.4.1.1 Porovnání vzdáleností

Při porovnávání dat nelze použít Hammingovu vzdálenost, protože řetězce nemusí být stejně dlouhé. Damerau-Levensheinova vzdálenost je výhodnější nez Levenshteinova, protože je možné, že v databázi jsou překlepy.

Porovnání pro spočítání vzdálenosti využijeme řádek „Zdeněk ZDENĚK VAŘEČKA - AOV“, který se porovnává s „Zdeněk VAŘEČKA“.

Damerau-Levenshteinova vzdálenost Pro získání vzdálenosti je potřeba se zbavit rětězce „ ZDENĚK“ a řetězce „ - AOV“. Vzdálenost je tedy 13.

Nejdelší společná podposloupnost První řetězec obsahuje celou sekvenci druhého, jeho délka je tedy nejdelší možná - 14.

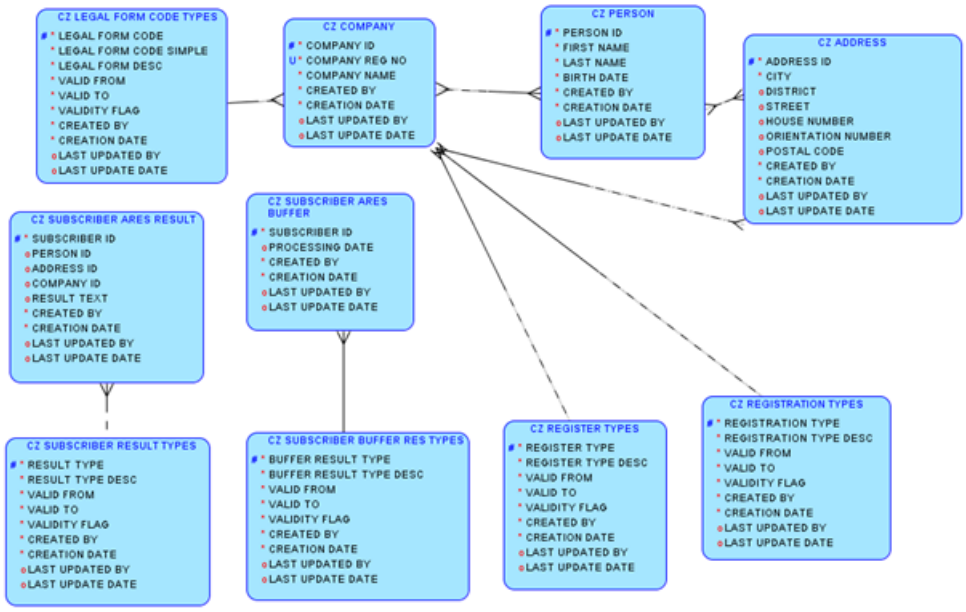
Závěr porovnání dat Při použití Damerau-Levenshteinovi vzdálenosti vyšla veliká vzdálenost - 13. Při porovnání prázdného řetězce s většinou dat v databázi by vyšla menší vzdálenost a tato metoda počítání podobnosti je tedy nevhodná. U nejdelší společné podposloupnosti se podařilo úspěšně najít celou podsekvenci a je tedy velice pravděpodobné, že se křestní jméno a příjmení shodují. Pro porovnávání řetězců bude pro zvolena právě metoda pro nalezení nejdelší společné podposloupnosti, řazená podle největší vzdálenosti. Jediná nevýhoda této metody je, že záleží na délce jména a příjmení. Nevýhoda bude eliminována tak, že pokud délka podposloupnosti bude délka jednoho z porovnávaných řetězců, přičte se k vzdálenosti konstanta, která určí tomuto vstupu přednost.

Kapitola 3

Návrh

3.1 Databáze

Tato část se věnuje databázi, která je navrhnutá podle XML struktury služby ARES a slouží k uchovávání dat pouze ze služby ARES. Podčásti této kapitoly detailně popisují nejdůležitější sloupce těchto tabulek.



Obrázek 3.1: Logický model, popisující strukturu tabulek pro uloženi dat z ARES

3.1.0.1 CZ_SUBSCRIBER_ARES_BUFFER

Tabulka plněná pomocí databázové procedury. Plní se na základě dat z interní tabulky SUBSCRIBERS, v které chybí data, která se dají získat . Důležité jsou tyto sloupce:

- SUBSCRIBER_ID – Primární klíč tabulky SUBSCRIBERS.
- PROCESSING_DATE – Datum, kdy byl řádek zpracován.
- BUFFER_RESULT_TYPE – Jednoznakový typ výsledku, všechny validní znaky jsou popsány v tabulce CZ_SUBSCRIBER_ARES_RESULT.

3.1.0.2 CZ_SUBSCRIBER_ARES_RESULT

Rejstřík, obsahuje všechny možné validní znaky s jejich popisem a příznakem, jestli jsou stále validní. Důležité jsou tyto sloupce:

- BUFFER_RESULT_TYPE – Jednoznakový typ výsledku, je více popsán v dalším sloupci.
- BUFFER_RESULT_TYPE_DESC – Podrobné popsání výsledku.

3.1.0.3 CZ_COMPANY

Hlavní tabulka, při každém dotázání se vytvoří maximálně jedna, s vazbami na N Adres a osob. Důležité jsou tyto sloupce:

- COMPANY_ID – primární klíč.
- COMPANY_REG_NO – IČO firmy (Unikátní).
- COMPANY_NAME – jméno firmy.
- LEGAL_FORM_CODE – kód právní formy.

3.1.0.4 CZ_ADDRESS

Tabulka reprezentující bydliště osob a sídla firem. Unikátnost tabulky je dána těmito sloupci:

- CITY - město.
- DISTRICT – městská část.
- STREET – ulice.
- HOUSE_NUMBER – domovní číslo.
- ORIENTATION_NUMBER – číslo orientační.

- POSTAL_CODE – poštovní směrovací číslo.

Jakýkoliv ze sloupců může být prázdný, pokud jsou všechny prázdné, adresa se neukládá.

3.1.0.5 CZ_PERSON

Tabulka reprezentující osobu, je vztažená k firmě s nějakou rolí pomocí relační tabulky CZ_PERSON_IN_COMPANY a k adrese bydliště. Unikátnost tabulky je dána těmito sloupci:

- FIRST_NAME – křestní jméno.
- LAST_NAME – příjmení.
- BIRTH_DATE – datum narození.

Pro osobu není více údajů v ARESu, a tak je možné mít dvě osoby reprezentované jednou řádkou (Dva Jani Nováci se stejným datem narození).

3.1.0.6 LEGAL_FORM_CODE_TYPES

Rejstřík, popisující všechny právní formy. Měl by být identický s tabulkou na http://wwwinfo.mfcr.cz/ares/aresPrFor.html.cz. Důležité jsou tyto sloupce:

- LEGAL_FORM_CODE – primární klíč, kód právní formy.
- LEGAL_FORM_CODE_SIMPLE – jednoznakový kód určující, jestli je firma právní (P) nebo fyzická (F).
- LEGAL_FORM_DESC – popis právní formy.

3.1.0.7 CZ_REGISTER_TYPES

Rejstřík, popisující příznaky subjektu popsané výše. Důležité jsou tyto sloupce:

- REGISTER_TYPE – primární klíč, pozice znaku.
- REGISTER_TYPE_DESC – popis, co pozice znamená.

3.1.0.8 CZ_REGISTRATION_TYPES

Rejstřík, popisující hodnoty příznaků subjektů. Důležité jsou tyto sloupce:

- REGISTRATION_TYPE – primární klíč, jeden znak.
- REGISTRATION_TYPE_DESC – Popis, co daný znak znamená.

3.1.0.9 CZ_SUBSCRIBER_ARES_RESULT

Tabulka popisující finální výsledek po kontrole, jestli je osoba z ARESu stejná jako v databázi. Důležité jsou tyto sloupce:

- RESULT_TYPE – finální výsledek po kontrole
- RESULT_TEXT – komentář k výsledku.

3.1.0.10 CZ_COMPANY_ADDRESS

Relační tabulka mezi CZ_ADDRESS a CZ_COMPANY v relaci N:N.

3.1.0.11 CZ_PERSON_IN_COMPANY

Relační tabulka mezi CZ_PERSON a CZ_COMPANY v relaci N:N obsahující navíc řádek „ROLE“ symbolizující roli osoby ve firmě.

3.1.0.12 CZ_PERSON_ADDRESS

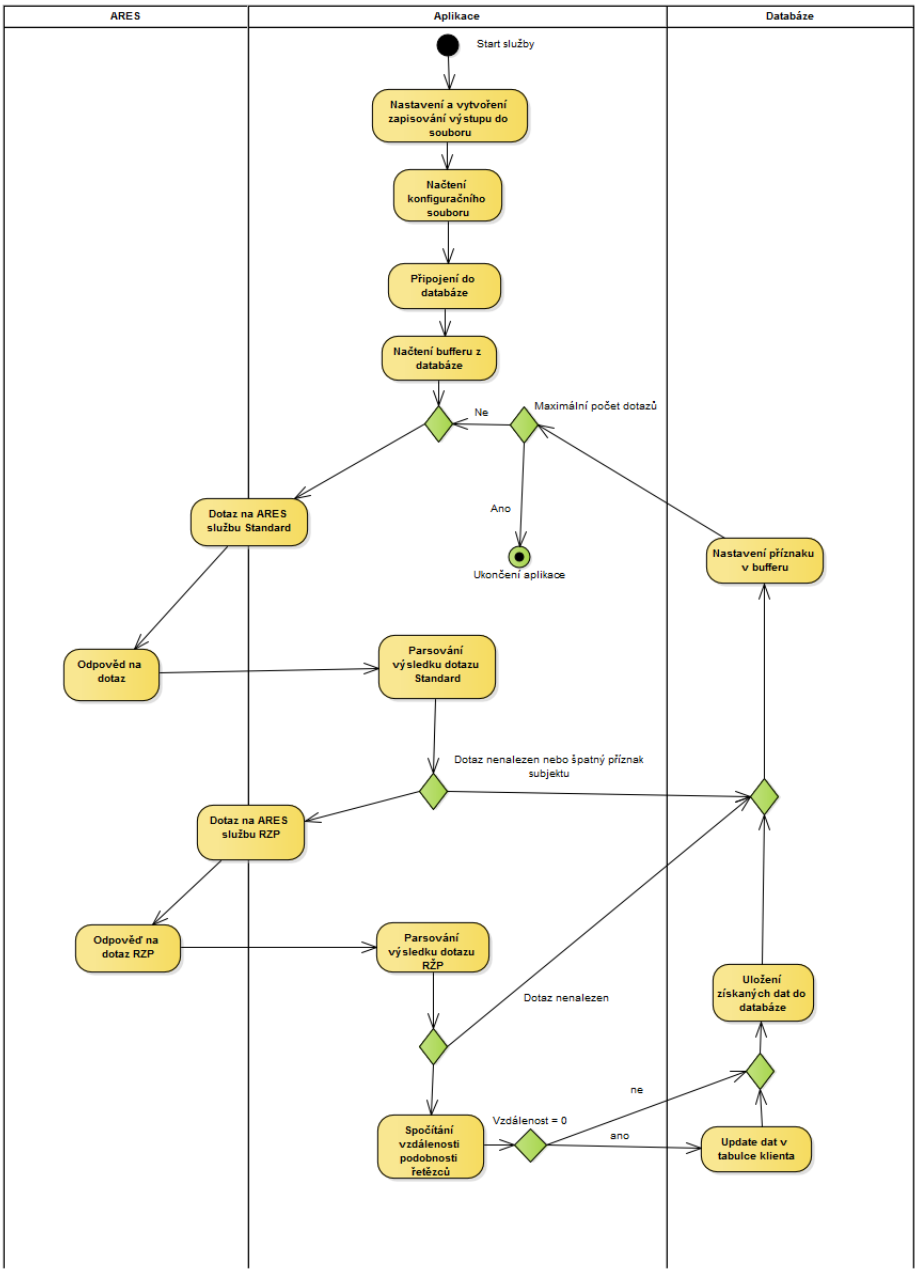
Relační tabulka mezi CZ_PERSON a CZ_ADDRESS v relaci N:N symbolizující bydliště osoby.

3.1.0.13 CZ_PERSON_ADDRESS_COMP

Relační tabulka mezi CZ_PERSON, CZ_ADDRESS a CZ_COMPANY, slouží k získání bydliště osoby vázané k firmě.

3.2 Popis běhu aplikace pro stahování dat

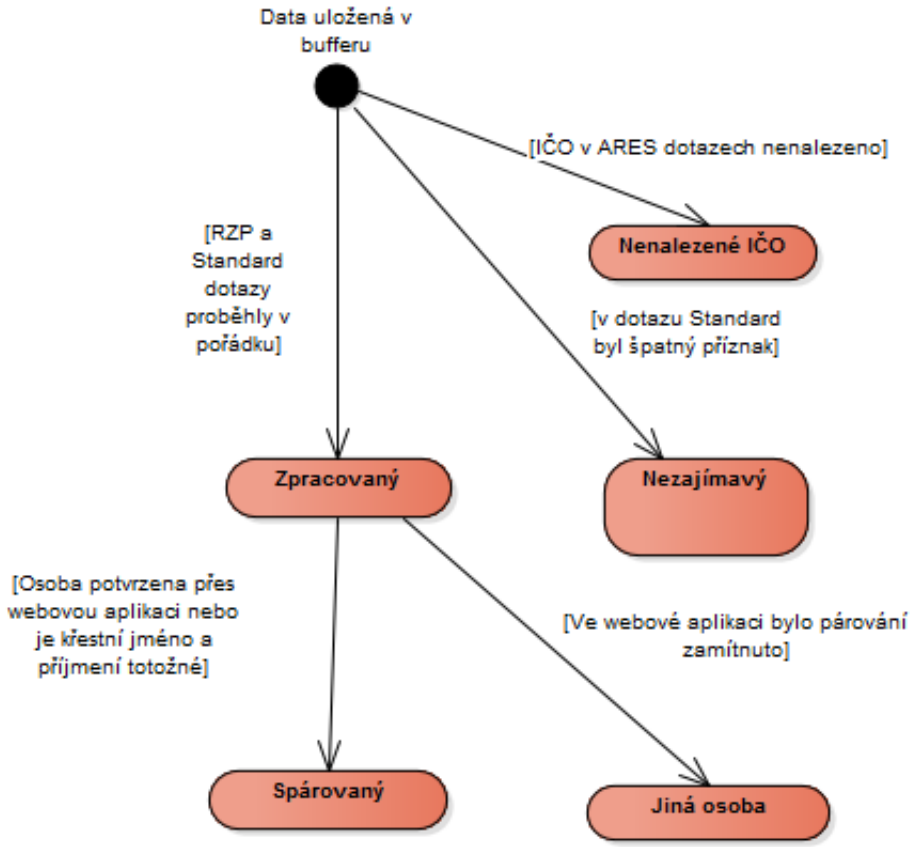
Pro popis běhu byl zvolen diagram aktivit, který zobrazuje všechny důležité aktivity, které program dělá, a všechny možné scénáře.



Obrázek 3.2: Diagram aktivit popisující běh aplikace pro stahování dat

3.3 Popis stavů

Stažená data mají stav, podle kterého se určuje, co s nimi dále. Tyto stavy jsou popsány v diagramu stavů.



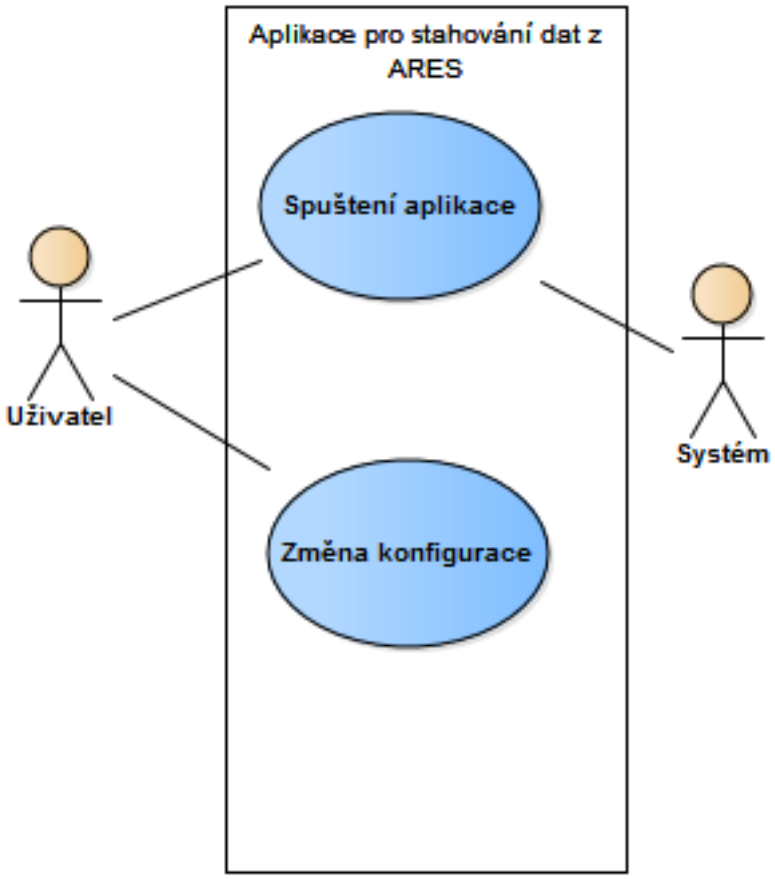
Obrázek 3.3: Stavový diagram popisující stav stažených dat

- Bez stavu: Data čekají na zpracování aplikací.
- Zpracovaný: Data se úspěšně stáhla a jsou připravena ke spárování.
- Spárovaný: Data se uložila do tabulky klientů.
- Jiná osoba: Data v ARESu se neshodují s daty v tabulce klientů.
- Nenalezené IČO: IČO nebylo v databázi služby ARES nalezeno.
- Nezajímavý: Stahovací aplikace vyhodnotila příznaky negativně a klient tedy pro firmu není zajímavý.

3.4 Use Case aplikace pro stahování

Aplikace pro stahování má dva aktéry:

- Systém, který může pouze spouštět aplikaci.
- Uživatele, který může spouštět aplikace a měnit konfiguraci aplikace.



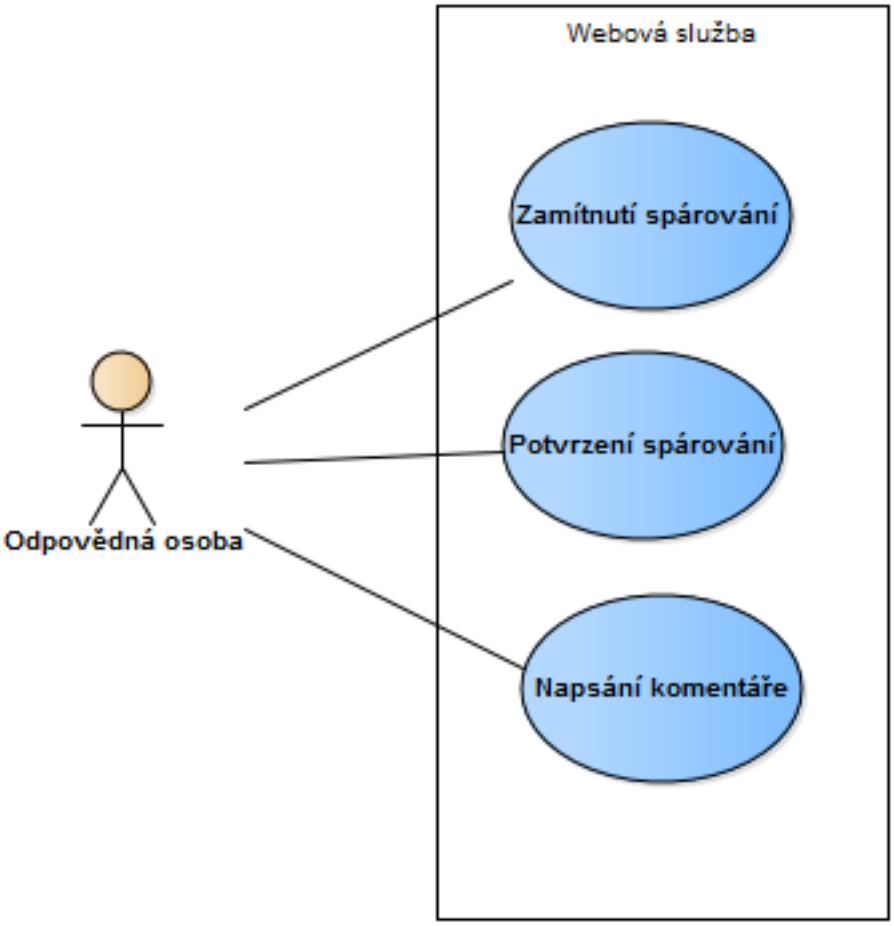
Obrázek 3.4: Use Case aplikace pro stahování

3.5 Use Case webové služby

Webová služba má pouze jednoho aktéra - odpovědnou osobu, která může:

- Zamítnout spárování: Určí, že se data na ARESu neshodují s daty v klientské tabulce.

- Potvrdit spárování: Určí, že se data shodují a přidá do tabulky klientů zjištěné údaje.
- Napsání komentáře: Napíše poznámku k údajům, týkající se párování.



Obrázek 3.5: Use Case webové aplikace

Kapitola 4

Implementace

Tato kapitola se věnuje popisem nejdůležitějších tříd aplikace a jejich nejdůležitějších metod. Kompletní seznam tříd a okomentovaných metod je v javadocu na přiloženém CD.

4.1 AresUpdate

Třída AresUpdate obsahuje metodu main, přes kterou se spouští aplikace pro stahování dat. V této metodě se nastavuje vypisování výstupu do souboru a vytváří se instance třídy AresWorker, na kterou se volá metoda doRoutine.

4.2 AresWorker

Třída AresWorker se stará pomocí metody doRoutine o celý běh programu, hlídá počet dotazů, získává data ze služby ARES pomocí tříd RZPQuery a StandardQuery, vyhodnocuje je a pomocí tříd EntitySeeker a EntityUploader je ukládá a hledá.

4.3 StandardQuery

Třída StandardQuery se stará o dotazování do ARES přes službu Standard pomocí metody getNext. Tato metoda stáhne XML soubor, zpracuje ho a vrátí třídu Registers.

4.4 RZPQuery

Třída RZPQuery se stará o dotazování do ARES přes službu RZP pomocí metody getNext. Tato metoda stáhne XML soubor, vytvoří všechny entity korespondující s XML strukturou a vrátí entitní třídu CzCompany svázanou se všemi získanými entitami.

4.5 EntitySeeker

Třída EntitySeeker slouží pro dotazování do databáze pomocí jazyku JPQL. Pomocí metody getDataFromBuffer získává klientů s chybějícími daty narození a pomocí metod findAddress,findPerson, FindCompany kontroluje, jestli v databázi entita se stejnými daty existuje.

4.6 EntityUploader

Třída EntityUploader slouží pro ukládání dat do databáze, převážně pomocí metody insertCompanyAndItsRelations.

4.7 Entitní třídy

Entitní třídy korespondující s databázovým návrhem jsou v balíčku model.

4.8 Controller

Třída Controller se stará o všechny akce webové aplikace.

Kapitola 5

Testování

5.1 Zpětná vazba uživatelů

Zde uvádím doslovný přepis zpětné vazby uživatelů.

„Aplikace byla původně plánována pouze pro doplnění data narození v tabulce klientů, ovšem při vývoji se narazilo na několik nedostatků v databázi, které nám aplikace pomůže vyřešit.“

„Užívání webové aplikace je jednoduché a přehledné. Co se týče jejího vzhledu, tak se mi moc líbí, že je ve firemních barvách.“

„I když je zatím aplikace pouze v testovacím režimu, tak nám již pomohla získat důležitá data a usnadnila práci. V budoucnu bychom rádi rozšířili aplikaci o další funkčnosti.“

5.2 Nalezené chyby a úpravy

Při prvních dnech testování aplikace bylo nalezeno několik chyb a nedostatků. Všechny jsou sepsány níže.

P_1 **Adresa ve službe ARES nemá žádné povinné atributy**

V tabulce Adresy byla všechna pole povinná z důvodu nalezení povinných atributů adresy. Zjistilo se, že žádný atribut není povinný, a proto jsou i v tabulce nepovinné.

P_2 **Nekontrolovala se návratová hodnota výsledků služby ARES**

ARES při neúspěšném vyhledání (špatné IČO) vrátil jiný formát XML, který bylo nutné zpracovat a vyhodnotit jinak.

P_3 **Zapisování komentáře při spárování klienta**

Pokud se ve webové aplikaci zvolilo spárování klienta, komentář se nezapsal do databáze.

P_4 **Komentář se nezobrazoval**

Komentář se ve webové aplikaci zobrazoval prázdný, i když v databázi prázdný nebyl.

5.3 Získaná data

Ke dni 9. 5. 2017 bylo aplikací zpracováno 4121 klientů, z toho po stažení byly data ve stavu:

- 1370 úspěšně zpracovaných,
- 2645 nezajímavých,
- 106 nenalezených.

Z úspěšně zpracovaných bylo ihned spárovaných pouze 60 klientů. Webovou aplikací bylo s asistencí osoby zpracováno 338 klientů, z toho bylo 268 úspěšně spárováno a u 70 byla zjištěna chyba.

Závěr

Cílem bakalářské práce bylo vytvořit aplikaci pro stahování dat o klientech na základě chybějících údajů v databázi pomocí XML služby ministerstva financí ARES a párování získaných dat s klienty.

K vývoji práce bylo využito Eclipse Java EE IDE for Web Developers. Zvolený JPA provider pro tuto práce byl EclipseLink 2.5.2. Jako XML parser byl využit JSoup. Pro webovou část byla použita implementace JavaServer Faces Mojarra 2.2.0 a server Apache Tomcat 6.0.53. K porovnávání řetězců byla nejvhodnější nejdelší společná podposloupnost.

Vytvořena aplikace je nasazena a testována jako každodenní služba a úspěšně získává data. Počet dotazů se dá jednoduše nastavit v konfiguračním souboru. Aplikace momentálně ukládá údaje o firmě, její adresy a všechny zainteresované osoby a jejich bydliště společně s rolí vůči firmě. Používá k tomu služby RZP a Standard. S klientskými tabulkami ovšem momentálně páruje pouze datum narození. V budoucnosti je možné aplikaci rozšířit o získávání dalších dat a používání dalších XML služeb, které ARES nabízí.

Literatura

[1] Thaker, B.: JPA EclipseLink architecture. [online], [cit. 2017-13-5]. Dostupné z: http://bhaveshthaker.com/images/jpaeclipselinkdemoimages/JPAEclipseLinkDemo_Project_Img001.JPG

[2] Bray, T.; Paoli, J.; Sperberg-McQueen, C. M.; aj.: Extensible markup language (XML) 1.0. 2008.

[3] W3Schools: XML Tutorial. [online], [cit. 2017-13-5]. Dostupné z: https://www.w3schools.com/xml/

[4] Oracle: XML Parsing for Java. [online], [cit. 2017-26-4]. Dostupné z: https://docs.oracle.com/cd/B28359_01/appdev.111/b28394/adx_j_parser.htm

[5] W3C: XML Path Language (XPath). [online], [cit. 2017-13-5]. Dostupné z: https://www.w3.org/TR/xpath/

[6] R-bloggers: Fuzzy String Matching - a survival skill to tackle unstructured information. Dostupné z: https://www.r-bloggers.com/fuzzy-string-matching-a-survival-skill-to-tackle-unstructured-information/

[7] Hyyrö, H.: *Practical methods for approximate string matching*. Tampereen yliopisto, 2003.

[8] Oracle: JavaBeans(TM) Specification 1.01 Final Release. [online], [cit. 2017-13-5]. Dostupné z: http://download.oracle.com/otndocs/jcp/7224-javabeans-1.01-fr-spec-oth-JSpec/

[9] Inc, T.: What is Persistance? - Definition from Techopedia. [online], [cit. 2017-26-4]. Dostupné z: https://www.techopedia.com/definition/8842/persistence-computing

[10] TechTarget: What is an object-relational mapping(ORM)? - Definition from WhatIs.com. [online], [cit. 2017-26-4]. Dostupné z: http://searchwindevelopment.techtarget.com/definition/object-relational-mapping

[11] Stoltmann, T.: Object-relational Mapping. Dostupné z: https://www.informatik.hu-berlin.de/de/forschung/gebiete/rok/wbi/teaching/archive/ws1314/sp_semtext/07_ORM.pdf

[12] Daněček, J.: Přednášky [BI-EJA Enterprise java]. [online], [cit. 2017-13-5]. Dostupné z: https://edux.fit.cvut.cz/courses/BI-EJA/lectures/start

[13] ApS, J.: Java Annotations. [online], cit. 2017-1-5. Dostupné z: http://tutorials.jenkov.com/java/annotations.html

[14] Oracle: javax.persistence (Java(TM) EE7 Specification APIs. [online], [cit. 2017-13-5]. Dostupné z: https://docs.oracle.com/javaee/7/api/javax/persistence/package-summary.html

[15] ConnectionStrings.com: Connection Strings Explained - ConnectionStrings.com. [online], [cit. 2017-13-5]. Dostupné z: https://www.connectionstrings.com/connection-strings-explained/

[16] Bergsten, H.: *JavaServer Faces: building Web-based user interfaces.* "O'Reilly Media, Inc.", 2004.

[17] Oracle: Overview of the EL - The Java EE 6 Tutorial. [online], [cit. 2017-13-5]. Dostupné z: http://docs.oracle.com/javaee/6/tutorial/doc/bnahq.html

[18] MFČR: ARES-Administrativni registr ekonomický subjektů. [online], [cit. 2017-1-5]. Dostupné z: http://wwwinfo.mfcr.cz/ares/

Příloha **A**

Seznam použitých zkratek

ARES Administrativní registr ekonomických subjektů

HTML HyperText Markup Language

IČO Identifikační číslo

JAR Java archive

JPA Java Persistence API

JSF JavaServer Faces

MFČR Ministerstvo financí České republiky

ORM Objektově relační mapování

RŽP Registr živnostenského podnikání

WAR Web application archive

XML Extensible markup language

Příloha **B**

Obsah přiloženého CD

```
readme.txt ..................................... stručný popis obsahu CD
exe ....................... adresář se spustitelnou formou implementace
src
  AresService ..... projekt aplikace se zdrojovými kódy a dokumentací
  thesis ...................... zdrojová forma práce ve formátu LaTeX
text ..................................................... text práce
  thesis.pdf ........................... text práce ve formátu PDF
```

Příloha C

Instalační příručka

Tato kapitola popisuje, jak vytvořit prostředí pro spuštění aplikace.

C.1 Aplikace pro stahování dat z ARES

Pro spuštení aplikace je potřeba dodržet adresářovou strukturu, v kořenovém adresáři musí být:

- Vygenerovaný jar soubor,
- složka config se souborem config.cfg,
- složka logs.

Pro spuštění aplikace v adresáři se správnou strukturou je potřeba nastavit připojení k databázi. Připojení se nastavuje v souboru persistence.xml, uloženém uvnitř JAR souboru v adresáři META-INF. Poté stačí bez parametrů spustit JAR soubor. Výstup aplikace je poté v souboru složky logs, pojmenovaném podle časové značky yyyy-MM-dd_HH-mm-ss s příponou log.

C.1.1 Popis konfiguračního souboru

Hodnoty proměnných se nastavují ve formátu názevProměnné=hodnota. Proměnné, s kterými se pracuje:

- proxyHost: nastavuje adresu proxy serveru,
- proxyPort: nastavuje port proxy server,
- proxyUser: nastavuje uživatele proxy serveru,
- proxyPassword: nastavuje heslo uživatele proxy serveru,
- dayStart: nastavuje začátek dne aplikace,

- dayEnd: nastavuje konec dne aplikace,
- dayLimit: nastavuje maximální denní limit pro počet dotazů na ARES,
- nightLimit: nastavuje maximální noční limit pro počet dotazů na ARES.

C.2 Webová aplikace pro párování

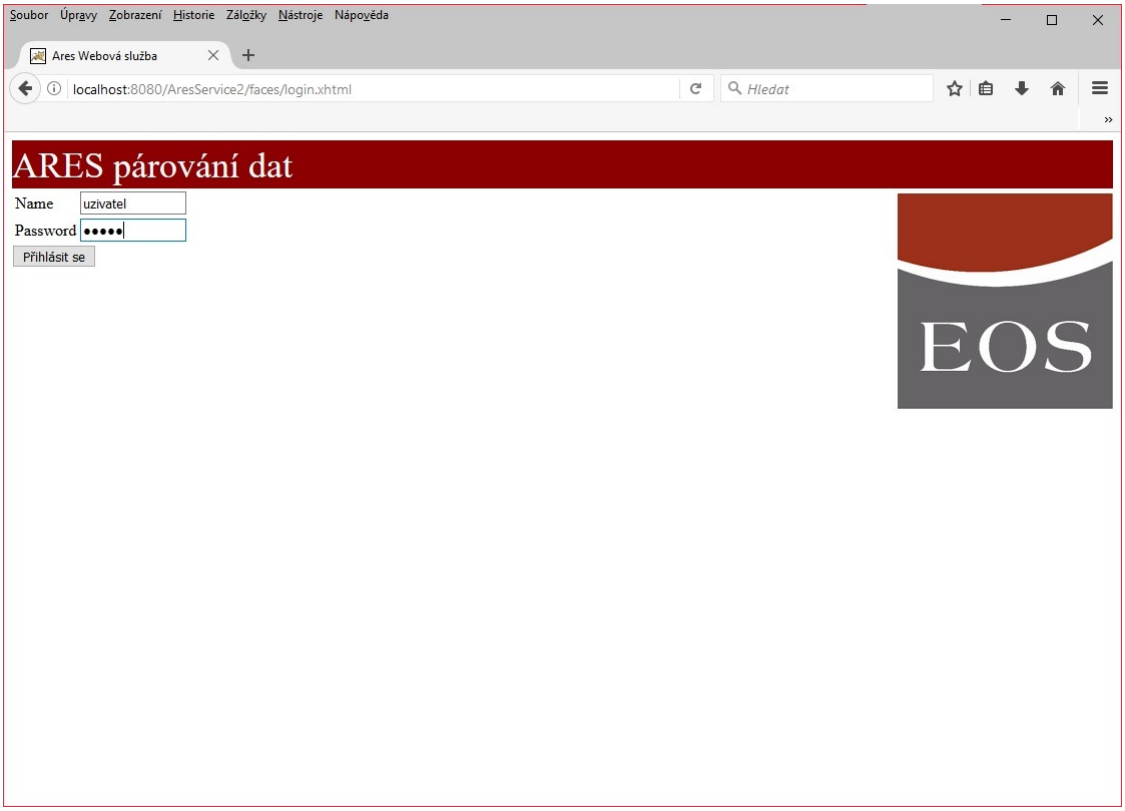
Aplikace se nasazuje na server. Na serveru je nezbytné mít knihovny eclipselink.jar, javax.persistence.jar, cdi-api-1.2.jar a ojdbc.jar. Je potřeba ve WAR souboru nastavit připojení k databázi. Připojení se nastavuje v souboru persistence.xml, uloženém uvnitř WAR souboru v adresáři WEB-INF/classes/META-INF. Poté stačí pouze vygenerovaný WAR soubor vložit na server [1] a restartovat server.

[1] U Tomcat serveru do složky webapps

Příloha D

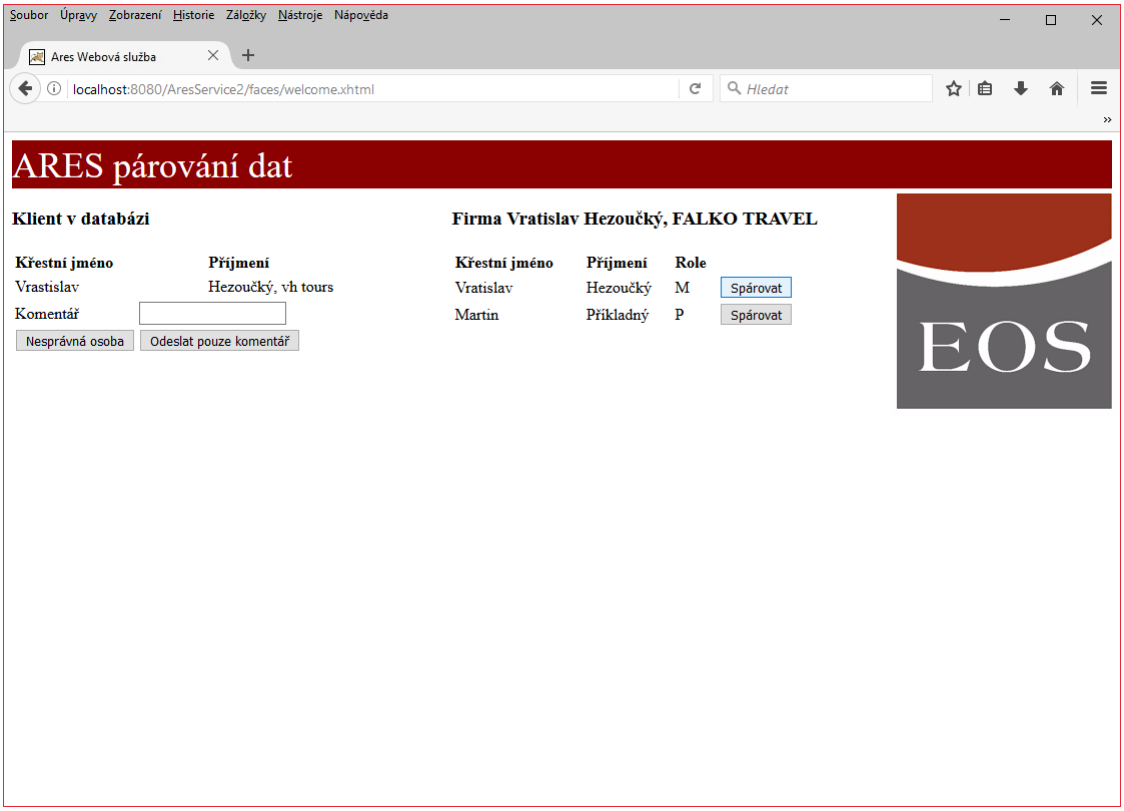
Uživatelská příručka webové aplikace

Tato kapitola popisuje uživatelské rozhraní webové aplikace pro párování dat. Pro vstup do aplikace je nutné se přihlásit.



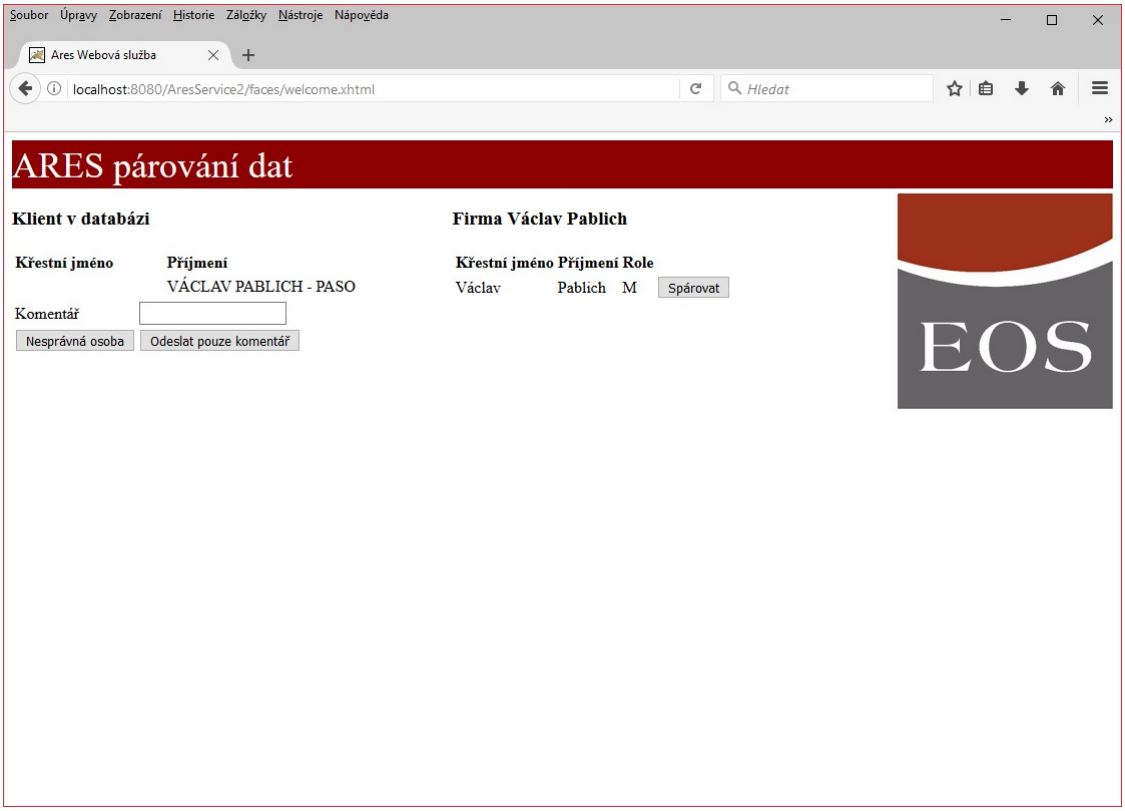
Obrázek D.1: Ukázka přihlašovaní webové aplikace

Po přihlášení může uživatel rozhodovat o správnosti pomocí tlačítek nebo napsat komentář, který se ukládá i při zamítnutí a spárování. Firma může mít více osob a uživatel vybírá správnou osobu pomocí tlačítka „spárovat“ vedle ní.



Obrázek D.2: Příklad obrazovky porovnání klientů

Při jakékoliv akci se načte další klient ve stejném formátu.



Obrázek D.3: Příklad obrazovky po spárování klienta