

Informační systém překladače

Information System Translators

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Peter Rafaj**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Informační systém překladatele
Information System Translators**

Zásady pro vypracování:

Cílem bakalářské práce je zjistit požadavky na informační systém (IS) překladatele s ohledem na privátní a veřejnou část aplikace, vypracovat analýzu a návrh implementace. Pro implementaci zvolte moderní vývojové prostředí. Navržený IS implementujte a porovnejte s dostupnými IS.

1. Zjistěte a specifikujte požadavky na informační systém překladatele s ohledem na privátní a veřejnou činnost.
2. Vypracujte analýzu a návrh implementace pro vhodně zvolený moderní HW a SW.
3. Implementujte IS pomocí vhodné moderní technologie.
4. IS otestujte a proveďte zhodnocení dosažených výsledků.
5. Porovnejte vámi vypracovaný IS s podobnými IS.
6. Vypracujte uživatelskou a programátorskou dokumentaci.

Seznam doporučené odborné literatury:

- [1] ŠARMANOVÁ, Jana. *Teorie zpracování dat* [online]. druhé, přepracované. Ostrava : VŠB – Technická univerzita Ostrava, 2007 [cit. 2010-11-09]. Dostupné z www:
<<http://www.elearn.vsb.cz/archivcd/FEI/TZD/TZD.pdf>>, ISBN 978-80-248-1498-8
- [2] KOSEK, Jiří. *PHP Tvorba interaktivních internetových aplikací: Podrobný průvodce*. Vydání 1. Praha: Grada Publishing, 1999. 492 s., ISBN 80-7169-373-1

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Ing. Lačezar Ličev, CSc.**

Konzultant bakalářské práce: Ing. Radim Bača, Ph.D.

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne. Uviedol som všetky literárne
pramene a publikácie, z ktorých som čerpal.

V Ostrave 4. mája 2012


.....

Rád by som poďakoval Ing. Lačezaru Ličevovi, CSc., za odbornú pomoc a konzultáciu pri vytváraní tejto bakalárskej práce.

Abstrakt

Cieľom tejto bakalárskej práce je vytvorenie informačného systému pre podporu prekladateľskej činnosti prekladateľov a znalcov jazyka. Práca sa zaoberá kompletným vývojom informačného systému od analýzy, spracovania návrhu, implementácie až po testovanie a nasadenie informačného systému. Systém bude umožňovať manažment objednávky od jej vytvorenia zákazníkom, schválenie tejto objednávky administrátorom, vypracovania objednávky prekladateľom a dokončenia objednávky administrátorom. Systém bude určený pre zákazníkov, ktorí chcú využiť služby, ktoré ponúka prekladateľ a pre prekladateľov, ktorí si chcú evidovať informácie o svojej prekladateľskej činnosti. Práca sa skladá z objasnenia problematiky prekladateľov, analýzy funkčných požiadaviek na informačný systém a následnej implementácie . Posledná časť hodnotí dosiahnuté výsledky.

Kľúčová slova: prekladatelia, výkazy, preklady , objednávka, faktúra, pdf, spring framework, spring MVC, spring security, hibernate

Abstract

The main idea of this thesis is to create an information system in order to support the work of translators and language experts. The Thesis deals with complete development of the information system. This process starts with the analysis of requirements on the system, continues with the processing of draft, implementation and finally comes to the testing and deployment of the information system. The system will allow the management of orders from creating the order by customer, approval of the order by administrator, complete the order by translator and final check of the order by administrator. The system is designed for customers who want to use the services offered by translators and for translators, who want to keep records of their work. The Thesis consists of clarifying the issue of translation and analysis of functional requirements for the information system and subsequent implementation. The last section assesses the results of work.

Keywords: translator, expense, translation, order, invoice, pdf, spring framework, spring MVC, spring security, hibernate

Zoznam použitých symbolov a skratiek.

MVC	– Model View Controller
IS	– Informačný systém
J2EE	– Java 2 Platform, Enterprise Edition
POJO	– Plain Old Java Object
HQL	– Hibernate Query Language
SQL	– Structured Query Language
JDBC	– Java Database Connectivity
CSS	– Kaskádové štýly
HTML	– HyperText Markup Language
XML	– Extensible Markup Language
JSP	– JavaServer Pages
EJB	– Enterprise JavaBeans

Obsah

1	Úvod	6
2	Úvod do problematiky prekladateľov a prekladov	7
2.1	Prekladatelia	7
2.2	Užívateľ	7
2.3	Preklady	8
2.4	Objednávka	8
2.5	Administrátor	8
3	Analýza požiadaviek na informačný systém	9
3.1	Funkčné požiadavky	9
3.2	Kto bude so systémom pracovať	9
3.3	Vstupy do systému	10
3.4	Očakávané výstupy zo systému	13
3.5	Funkcie IS	13
3.6	Nefunkčné požiadavky	16
3.7	Bezpečnosť	16
3.8	Dátová analýza	16
3.8.1	E-R diagram	17
3.8.2	Lineárny zápis typu entít	17
3.8.3	Analýza životného cyklu objednávky	18
4	Implementácia IS	23
4.1	Architektúra	23
4.2	Spring Framework	23
4.2.1	Spring MVC framework	24
4.2.2	MVC Model – View – Controller	25
4.2.3	Architektúra Spring MVC	25
4.2.4	Implementácia kontrolérov	26
4.2.5	Prezentačná vrstva	27
4.3	Servisná vrstva	27
4.4	Databázová vrstva	28
4.5	Prevádzka systému	30
5	Záver	32
6	Literatúra	33
	Přílohy	33

A Příloha	34
A.1 Obsah CD	34
A.2 Schéma databáze	34
A.3 Datový slovník	34

Zoznam tabuliek

1	Tabuľka Language	36
2	Tabuľka Translator	36
3	Tabuľka Role	37
4	Tabuľka Reference	37
5	Tabuľka TranslatorUtils	37
6	Tabuľka Customer	37
7	Tabuľka UserRole	38
8	Tabuľka Invoice	38
9	Tabuľka Info	38
10	Tabuľka Expense	38
11	Tabuľka Area	39
12	Tabuľka Administrator	39
13	Tabuľka Address	39
14	Tabuľka Util	39
15	Tabuľka Translate	40
16	Tabuľka File	40
17	Tabuľka OrderForm	41
18	Tabuľka User	41
19	Tabuľka OrderFormFile	42
20	Tabuľka CustTransRelation	42

Zoznam obrázkov

1	Vztáh medzi aktérmi a systémom	10
2	Use case zobrazenia funkcionality	14
3	E-R diagram	17
4	Stavový diagram objednávky	19
5	Sekvenčný diagram schvaľovania objednávky administrátorom	21
6	Sekvenčný diagram generovanie výslednej faktúry	22
7	Architektúra frameworku spring	24
8	Architektúra Spring MVC	25
9	Ukážka prostredia pre administrátora	30
10	Ukážka zobrazenia informácií o objednávke	31
11	Schéma databáze	35

Zoznam výpisov zdrojového kódu

1	Implementácia kontroléru v spring MVC	26
---	---	----

1 Úvod

V dnešnej mul ti kultúrnej dobe sa denne stretávame s cudzími jazykmi. Prakticky každý z nás ovláda minimálne náš materinský jazyk. V dobe pred rokom 1989 v Čechách a na Slovensku dominoval hlavne Ruský jazyk. Väčšina našich rodičov sa stretla v škole práve s týmto jazykom. Zmena vnímania cudzích jazykov prišla až po roku 1989, kedy bolo umožnené mladým ľuďom študovať cudzie jazyky a vzdelávať sa v tejto oblasti. Dnešní mladí ľudia radi študujú cudzie jazyky a taktiež sa chcú vzdelávať v tejto oblasti. Mnoho ľudí už dnes ovláda niektorý zo svetových jazykov ako je angličtina, nemčina alebo španielčina na profesionálne úrovni. Dokážu sa slovne aj písomne dohovoriť cudzím jazykom. Problém však nastáva ak chceme takýto preložený dokument použiť ako právny dokument. Právna legislatíva vyžaduje aby bol preložený dokument overený úradným prekladateľom. Takúto možnosť majú len notársky overení prekladatelia. Prekladateľ je človek, ktorý ovláda niekoľko cudzích jazykov, rozumie im a dokáže v maximálnej miere preložiť text alebo hovorové slovo z jedného jazyka do druhého. Samozrejme ak to preklad vyžaduje, dokáže preklad notársky overiť a zaručiť tým rovnaké znenie dokumentu pred a po preložení.

V prvej časti mojej bakalárskej práce som sa venoval analyzovaniu požiadaviek na informačný systém prekladateľov s ohľadom na privátnu a verejnú časť. Na začiatku analýzy sa budem snažiť stručne popísať problematiku prekladov, prekladateľov a objednávky, ktorá vzniká medzi zákazníkom a prekladateľom. Čo znamená preklad z pohľadu prekladateľa a ako vyzerá preklad z pohľadu užívateľa. Preklad môžeme chápať ako službu, ktorú prekladateľ ponúka. Môže ísť o preloženie určitej entity ako text či hovorové slovo z jedného jazyka do druhého. Taktiež môže ísť o tlmočenie, korektúru textu ale dokonca aj vyučovacie hodiny cudzieho jazyka. Typ služby je preto veľmi dôležitý parameter pretože od neho sa odvíja množstvo ďalších parametrov. Medzi prekladateľom a zákazníkom vzniká vzťah, ktorý je reprezentovaný objednávkou. Objednávka je záväzná zmluva medzi osobou, právnickou osobou alebo organizáciou a prekladateľom. Náš informačný systém by mal ako hlavnú funkcionality ponúkať celkový manažment a životnosť objednávky či už z pohľadu užívateľa a taktiež prekladateľa. Pretože objednávka je záväzný dokument, musí objednávka obsahovať množstvo dôležitých parametrov ako napríklad rozsah prác, ktoré požaduje objednávateľ po prekladateľovi, dátum ukončenia prác, adresy, informácie pre platbu za objednávku.

V poslednej časti bakalárskej práce sa venujem implementácií informačného systému. Väčšina dnešných informačných systémov je vytvorených ako webové aplikácie. Preto aj ja som sa rozhodol tento informačný systém vytvoriť ako webovú aplikáciu. Prostriedkov pre vývoj webových technológií je veľké množstvo. Pre vývoj informačného systému som si vybral platformu Java. Z tejto platformy som si vybral veľmi populárny rámec (framework) spring. Je určený hlavne pre vývoj J2EE (Java 2 Enterprise Edition) aplikácií. V poslednej fáze tvorby informačného systému som funkčne otestoval informačný systém a zhodnotil výsledky.

2 Úvod do problematiky prekladateľov a prekladov

Nasleduje analýza informačného systému. V tomto kroku sa budem snažiť objasniť problematiku prekladov, prekladateľov a objednávok. Náš systém bude riadiť vzťah medzi zákazníkom a prekladateľom a taktiež by mal uľahčovať prekladateľovi evidenciu o jeho práci.

2.1 Prekladatelia

Prekladatelia sú ľudia, ktorí sa venujú cudzím jazykom na profesionálnej úrovni. Ovládajú všetky zákonitosti spojené s prekladmi. Preto ich využívajú ľudia na preloženie textu v čo najpresnejšej podobe. Pri preklade je veľmi dôležité aby prekladateľ rozumel problematike prekladaného textu, pretože znalosť problematiky pomôže vernejšie interpretovať daný text. Pri preklade hrá taktiež veľkú úlohu či ide o rodinného hovorca alebo nie. Znalosť kultúry daného jazyka napomáha k vernejšiemu prekladu. Je veľmi jednoduché preložiť určitý text pomocou známych internetových prekladačov ale preložiť text so zapojením kultúrnych aspektov, prípadne dialektov daného jazyka je už pre internetové prekladače nemožné. Každý z nás predsa ovláda materský jazyk a vie o nejakých zaujímavých slovách a slovných spojeniach, ktoré by mohli byť pre cudzinca zle pochopiteľné.

Prekladateľ ponúka portfólio svojich služieb. S väčšinou ponúkaných služieb sú určitým spôsobom spojené referencie na tieto služby. Prekladateľ by v našom systéme mal možnosť vytvárať si svoje vlastné referencie a odkazovať sa tak na svoje znalosti a zručnosti. Ďalšie doplnkové informácie, ktoré sa týkajú prekladateľa, sú nástroje, ktoré pri svojej práci využíva. Bežnou praxou je, že prekladateľ pri preklade dokáže spracovať rôzne formáty pritom dokáže využiť nástroje, ktoré zefektívnia jeho prácu. Informácie o nástrojoch, ktoré prekladateľ využíva pri preklade informujú zákazníka o tom či je prekladateľ schopný prijať súbor na preloženie v danom formáte.

S prekladateľskou činnosťou je taktiež spojená evidencia výkazov. Výkaz si môžeme predstaviť ako jednoduchý zápis o tom kto prekladateľovi platil, koľko mu zaplatil, koľko daná osoba prekladateľovi dlží a aký je celkový zisk zo služby. Spolu s týmito údajmi ešte prekladatelia pri výkazoch evidujú číslo faktúry a číslo v prekladateľskom slovníku. Výsledkom takejto evidencie by mohol byť prehľadný rozpis všetkých týchto parametrov spolu s výsledným ziskom či stratou.

2.2 Užívateľ

Užívateľ je osoba, ktorá chce využiť služby, ktoré ponúkajú prekladatelia a potrebuje si vybrať z určitej množiny prekladateľov a služieb podľa rôznych kritérií. Pre tieto potreby teda navštíví náš informačný systém. Z právneho hľadiska môže systém používať ľubovoľná skupina užívateľov. Môže ísť o právnický subjekt súd, fyzickú osobu alebo políciu.

2.3 Preklady

Na svete existuje množstvo cudzích jazykov a nárečí, preto nie je možné rozumieť všetkým jazykom. Aj keby sme sa naučili daný jazyk, nikdy ho nebudeme ovládať ako rodinný hovorca, pretože nepoznáme kultúru a slangy daného jazyka. Preklady sú určitým spôsobom služby, ktoré daný prekladateľ ponúka. Nemusí ísť úplne vždy o preklady textov. S doménou jazykov sa viaže množstvo služieb akými sú napríklad tlmočenie, korektúra textov. Taktiež môže ísť o výučbu cudzích jazykov. Hlavným atribútom prekladov by malo byť z akého zdrojového jazyka do akého cieľového jazyka je daný preklad uskutočnený. Medzi ďalšie atribúty patrí o akú službu ide. Ďalšou kľúčovou vlastnosťou je obor prekladu. Prekladateľ by mal poznať obor prekladu aby správne rozumel kontextu daných slov. Veľmi dobrým príkladom sú napríklad obory informatika a medicína. Tieto dva obory sú diametrálne odlišné a obsahujú diametrálne odlišnú skupinu slov s odlišným významom. Ako príklad uvediem slovíčko „socket“, ktoré v angličtine v informatickom kontext znamená „spojenie“ , „zásuvka“ ale v medicínskom kontexte môže znamenať „klbová jamka“. Poslednú vlastnosť, ktorá sa týka prekladov je určite cena, za ktorú prekladateľ daný preklad ponúka. Cena môže byť samozrejme v rozmedzí od do a je na prekladateľovi, ktorý sa rozhodne podľa náročnosti, akú výslednú cenu stanoví.

2.4 Objednávka

Objednávka je ponuka k uzavretiu kúpnej zmluvy či zmluvy o poskytnutí služby. Je vytvorená možným odberateľom a je určená pre dodávateľa. Písomná objednávka je účtovný doklad, ktorý slúži občanom i podnikateľom ako prvotný doklad pre dodávku tovaru alebo služieb. Ak sa teda užívateľ rozhodne využiť niektorú zo služieb, ktoré prekladateľ ponúka. Vytvorí v systéme objednávku na prekladateľove služby.

2.5 Administrátor

Medzi hlavné úlohy administrátora informačného systému je riadenie, správa a vytváranie užívateľov a doplnkové informácie ako správa slovníkov a podobne. Prekladatelia, ktorí budú chcieť využívať služby informačného systému sa budú musieť registrovať. Táto registrácia by však nemala byť hneď aktívna. Iba administrátor sa rozhodne či povolí registráciu konkrétneho prekladateľa alebo nie.

Aby sme zaručili výnosnosť naše ho informačného systému administrátor by mal určitým spôsobom zasahovať to životného cyklu objednávky. Administrátor by mal vystupovať v systéme pomocou dvoch profilov. Prvým profilom by mal byť profil prekladateľa, kedy všetky objednávky budú smerovať na neho. On sa rozhodne komu z možných prekladateľov objednávku pridelí. Druhým profilom by bol profil zákazníka. Práve týmto profilom bude žiadať prekladateľov o vyhotovenie objednávky, ktorú ale nevytvoril on ale zákazník. Po dokončení prekladu administrátor navýši faktúru prekladu o administratívnu čiastku.

3 Analýza požiadaviek na informačný systém

3.1 Funkčné požiadavky

Z doterajšej analýzy vieme povedať niekoľko základných funkčných požiadaviek, ktoré by náš systém mal spĺňať.

- Umožňovať užívateľom a prekladateľom sa registrovať a využívať funkcie informačného systému.
- Vyhľadanie prekladu užívateľom, vytvorenie objednávky na prekladateľove služby a následné sledovanie tejto objednávky užívateľom až do jej ukončenia.
- Z pohľadu administrátora by mu systém mal umožňovať správu užívateľov a prekladateľov, správa slovníkov informačného systému. Hlavnou funkcionalitou by však malo byť riadenie životného cyklu objednávky od jej zadania až po ukončenie.
- Z pohľadu prekladateľa mu musí systém umožňovať prehľadnú správu prekladov, objednávok a výkazov.

3.2 Kto bude so systémom pracovať

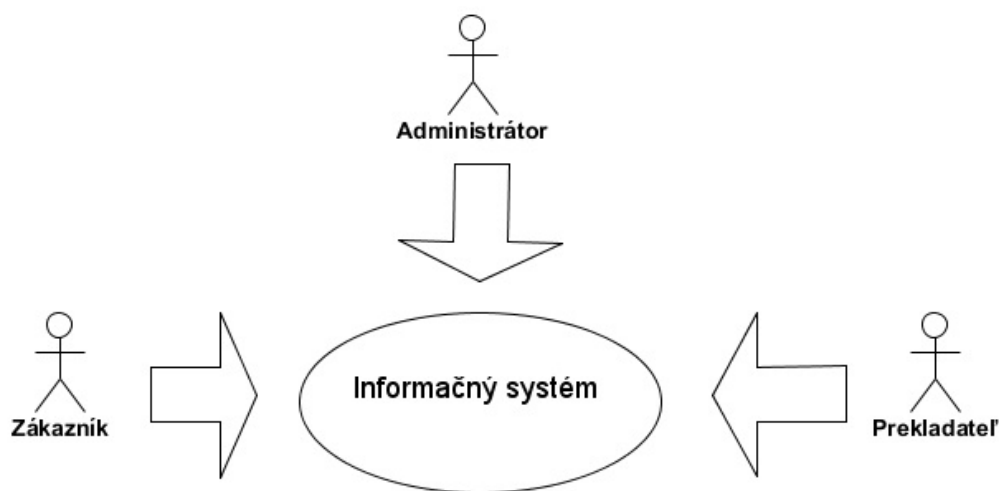
Z doterajšej analýzy vieme povedať, že systém budú môcť využívať štyri skupiny užívateľov. Prvou skupinou bude obyčajný návštevník systému. Tento užívateľ nebude zaregistrovaný a bude mať v maximálnej miere obmedzené možnosti využívania informačného systému. Týmto užívateľom sa stáva prakticky každý návštevník informačného systému. Jedinú funkcionalitu, ktorá takémuto užívateľovi bude umožnená bude zistenie či je v systéme zaregistrovaná požadovaná dvojica jazykov spolu s ostatnými parametrami. Oveľa dôležitejšími skupinami sú registrovaný zákazník, administrátor a prekladateľ.

- **Zákazník** by mal mať možnosť vybrať si z portfólia služieb, ktoré ponúkajú prekladateľa. Objednať si tieto služby a vytvoriť objednávku. Ako náhle bude objednávka v systéme vygenerovaná. Mal by mať možnosť sledovať stav tejto objednávky až do jej uzavretia. Po dokončení objednávky zo strany informačného systému by malo byť užívateľovi umožnené zobrazit' si faktúru a ak objednávka obsahovala nahraté súbory na preloženie mal by mať možnosť si ich stiahnuť. Zákazník si bude v systéme môcť taktiež upravovať svoj profil so základnými informáciami.
- **Administrátor** by mal mať možnosť zobrazovať všetkých užívateľov v systéme. Či už ide o zákazníkov alebo prekladateľov. U každého užívateľa by mal mať možnosť upravovať ich profily. Administrátor bude informovať užívateľov o zmenách v systéme. Tieto informácie sa zobrazia zákazníkovi alebo prekladateľovi po prihlásení. Ďalšie funkcie, ktoré musia byť prekladateľovi umožnené sú správa jazykov, ktoré môžu byť do systému registrované a oblasti prekladu, ktoré si bude môcť prekladateľ vybrať pri vytváraní svojho portfólia. Hlavnou funkcionalitou však bude riadenie objednávky od jej vytvorenia užívateľom, vygenerovania faktúry pre

prekladateľa a zákazníka až po uzavretie objednávky a presunutie objednávky do archívu. S generovaním faktúr je spojené taktiež vedenie databázy výkazov.

- **Prekladateľ** musí mať možnosť pridávať a upravovať portfólio svojich služieb. Upravovať svoj profil a taktiež uchovávať informácie o svojich referenciách. Zobrazovať objednávky, ktoré mu boli pridelené. Meniť stav objednávok od stavu kedy bude objednávka zadaná až do stavu kedy bude objednávka archivovaná. Spolu so správou objednávok musí mať prekladateľ možnosť evidovať svoje výkazy o prekladateľskej činnosti a taktiež nástroje, ktoré používa.

Vzťahy medzi aktérmi a informačným systémom môžeme vyjadriť prehľadným obrázkom.



Obr. 1: Vzťah medzi aktérmi a systémom

3.3 Vstupy do systému

Po analýze a definícií rolí sa dostávame do bodu dátovej analýzy informačného systému. Tento bod je veľmi dôležitý pretože v následnej implementácii nám správne navrhnutý dátový model uľahčí manipuláciu s dátami. Nasleduje zoznam jednotlivých entít, ktoré som sa rozhodol evidovať.

Zákazník (Customer) – (id zákazníka, meno, priezvisko, email, telefón, dátum registrácie, typ užívateľa, názov organizácie, bližšie informácie o organizácii, ičo, dič). U zákazníka musíme evidovať základné informácie a taktiež kontaktné údaje ako telefón. Spolu s kontaktnými informáciami musíme evidovať taktiež adresu zákazníka. Táto adresa však bude v inej tabuľke. Z právneho hľadiska môžu systém využívať rôzni užívatelia. Môže

ísť o súdne orgány, policajné oddelenia, právnické osoby ale taktiež bežný užívateľia.

Prekladateľ (Translator) –(id prekladateľa, titul, meno, priezvisko, dátum vytvorenia, ičo, dič, podpis ako súbor, telefón, email, názov banky, adresa banky, číslo banky, názov účtu, číslo účtu, iban, bic). Každý prekladateľ musí uviesť svoje základné informácie ako meno priezvisko, kontaktné informácie ako telefón a email. Pre uľahčenie tejto činnosti bude môcť prekladateľ do systému nahrať obrázok s podpisom alebo pečiatkou a pri generovaní faktúry sa tento podpis automaticky nastaví ako podpis pri faktúre. Prekladateľovi musí byť taktiež umožnené evidovať jeho referencie a výkazy, ktoré vznikajú pri vystavovaní faktúry.

Referencia (Reference) – (id referencie, stručný popis referencie, podrobný text referencie). Spolu so základnými informáciami o prekladateľovi musíme evidovať jeho referencie. U referencií budeme evidovať popis referencie.

Nástroje (Util) -(id nástroja, popis nástroja, poznámka k nástroju). Tabuľka nástrojov bude obsahovať stručný popis nástroja spolu s poznámkou k danému nástroju. Pre správu nástrojov, ktoré prekladateľ využíva som sa rozhodol vytvoriť separátnu tabuľku nástroje. Nástrojov, ktoré môžu prekladateľa využívať nie je veľa a preto ich bude v systéme vytvárať administrátor a prekladateľ si už bude vyberať z ponuky týchto nástrojov.

Preklad (Translate) – (id prekladu, preklad z jazyka, preklad do jazyka, je prvý jazyk materský, je druhý jazyk materský, oblasť prekladu, typ služby, notársky overený, typ ponúkanej služby, cena za, cena od, cena do). Prekladateľ si pomocou prekladov vytvorí portfólio svojich služieb. Prekladateľ môže ponúkať rôzne druhy prekladov. Preklad bude obsahovať dvojicu jazykov, ktoré prekladateľ ponúka na preklad. Aby nám v systéme nevznikalo veľa podobných jazykov, rozhodol som sa tieto jazyky vyčleniť do samostatnej tabuľky. Prekladateľ si pri tvorbe svojho portfólia služieb už vyberie z množiny jazykov. Spolu s jazykmi bude preklad obsahovať všetky potrebné informácie ako oblasť prekladu, či ide o materinský jazyk, rozsah ceny, za ktorú danú službu prekladateľ ponúka ale aj či tento preklad dokáže prekladateľ notársky overiť.

Jazyk (Language) - (id jazyka, názov jazyka, anglický názov jazyka). Tabuľka jazykov, bude predstavovať slovník možných jazykov, ktoré bude možné vyhľadať v systéme a taktiež si z týchto jazykov bude môcť prekladateľ vyberať pri vytváraní svojich služieb.

Oblasť prekladu (Area) – (id oblasti, oblasť). Oblasť prekladu si bude vyberať prekladateľ pri zostavovaní svojich prekladov. Bude ho vytvárať administrátor. Pôjde o jednoduchý slovník bez žiadnej väzby na inú tabuľku databázy.

Informácie (Info) –(id oznamu, nadpis, správa, dátum vytvorenia, určená pre). Administrátor bude môcť informovať užívateľov systému o zmenách v systéme alebo o výpadkoch. Tieto informácie sa budú zobrazovať užívateľom po prihlásení. Oznam bude obsahovať nadpis, správu a pre koho je správa určená.

Objednávka (OrderForm) – (id objednávky, dátum vytvorenia, dátum želaného dokončenia, dátum potvrdenia, dátum ukončenia, dátum splatnosti faktúry, status objednávky, skutočná cena, meno, priezvisko, email, telefón, pozícia kontaktnej osoby, nadpis faktúry,

popis faktúry, informácie o živnosti, číslo faktúry, špecifické číslo, adresa doručenia). Hlavná entita, ktorá reprezentuje vzťah, ktorý vzniká medzi zákazníkom a prekladateľom. Objednávka by mala obsahovať dôležité a zrozumiteľné informácie pre obe strany. Medzi základné informácie, ktoré musíme u objednávky evidovať patria číslo objednávky, dátum potvrdenia objednávky, dátum dokončenia objednávky a v taktiež dátum splatnosti faktúry. Objednávka by mala obsahovať stručný popis vykonanej práce. V niektorých typoch faktúr prekladateľ uvádza aj určité znenie zákona alebo právne popisuje rozsah svojej práce. Posledným zo základných atribútov by mal byť dátum vytvorenia faktúry. Hlavným atribútom, ktorý musíme evidovať je výsledná cena, ktorú požaduje prekladateľ uhradiť za služby.

Položka faktúry (Invoice) – (id položky, položka, špecifické číslo v prekladateľskom slovníku, cena). Každá objednávka bude obsahovať zoznam položiek faktúry. Z jednotlivých položiek faktúry, bude vo výsledku uzavretia faktúry spočítaná výsledná cena. U položky som sa rozhodol evidovať popis položky. Pôjde o textový popis položky a tento popis sa tiež zobrazí v tabuľke vygenerovanej faktúry. S prekladmi a celkovou správou prekladov sa viaže povinnosť prekladateľa evidovať svoje preklady v takzvanom prekladateľskom denníku, kde každý preklad má svoje evidenčné číslo. Toto evidenčné číslo som sa rozhodol uložiť práve v položke faktúry. Posledným atribútom, ktorý je potrebné evidovať je cena danej položky.

Adresa (Address) – (id adresy, ulica, mesto, smerové číslo, krajina). Tabuľka adresa bude obsiahnutá v rôznych entitách a obsahuje bežné atribúty zachytenia adresy.

Výkaz (Expense) – (id výkazu, typ výkazu, číslo v prekladateľskom slovníku, číslo objednávky, dátum, zaplatil, dlh, cena, zisk). Prekladateľ si pri svojej činnosti zvykne evidovať svoje výdaje. Výkaz informuje prekladateľa o zisku alebo strate a je spojený s jednotlivou objednávkou pomocou čísla objednávky.

Súbor (File) – (id súboru, názov súboru, prefix, informácie, vlastník, sôbor, typ odoslania súboru, typ prijatia súboru). Užívateľ pri vytvorení objednávky bude môcť do systému nahrať súbor, ktorý sa týka konkrétnej objednávky. Túto možnosť taktiež bude mať prekladateľ. Obsahuje základné informácie ako názov a prefix. Atribút vlastník súboru určuje či súbor nahral prekladateľ alebo zákazník. Spôsob odoslania a prijatia definuje či užívateľ odošle poštou, nahrá tento súbor na server alebo ho prinesie osobne.

Objednávka prekladateľa a zákazníka (CustTransRelation) - (id relácie, špecifické číslo, číslo objednávky pre zákazníka, číslo objednávky pre prekladateľa, dátum vytvorenia dvojice). Táto tabuľka reprezentuje dvojicu objednávok a prepojenie objednávky od zákazníka až po objednávku od administrátora, ktorá smeruje k prekladateľovi. Tieto dve objednávky majú spoločné špecifické číslo, ktoré bude taktiež zobrazené vo výslednej faktúre oboch objednávok.

Administrátor (admin) – (id administrátora, id užívateľa) Ide o bežnú tabuľku administrátora, ktorá neobsahuje špecifické informácie.

Pre účely autorizácie a autentifikácie musíme ešte vytvoriť tabuľky používateľa a tabuľku rolí, do ktorej konkrétny užívateľ patrí. Tabuľka užívateľ bude obsahovať login,

ktorý použije užívateľ na prihlásenie, zašifrované heslo a hodnotu či je užívateľ schválený alebo nie. Tabuľka rolí bude obsahovať názov role. Tento názov sa bude neskôr využívať pri overení užívateľa, či môže prístupovať k danej metóde servisnej vrstvy.

3.4 Očakávané výstupy zo systému

Prehľad všetkých prekladov, ktoré sú v systéme zaregistrované. Užívateľ bude môcť vyhľadať konkrétny preklad pomocou filtra a pokročilého vyhľadávania so všetkými atribútmi, ktoré sú obsiahnuté v preklade.

Zobrazenie súhrnných a dôležitých informácií o objednávke. Súhrnné zobrazovanie bude obsahovať, čo si zákazník objednal, dátum vytvorenia, želaný dátum dokončenia, údaje pre kontaktnú osobu. Spolu s objednávkou bude môcť vidieť zoznam súborov, ktoré sa týkajú objednávky. Tieto informácie bude môcť vidieť užívateľ, prekladateľ ale aj administrátor počas celého trvania objednávky.

Zobrazenie výslednej faktúry. Túto možnosť bude mať administrátor, prekladateľ, užívateľ. Po dokončení objednávky si užívateľ bude môcť vygenerovať výslednú faktúru vo formáte PDF.

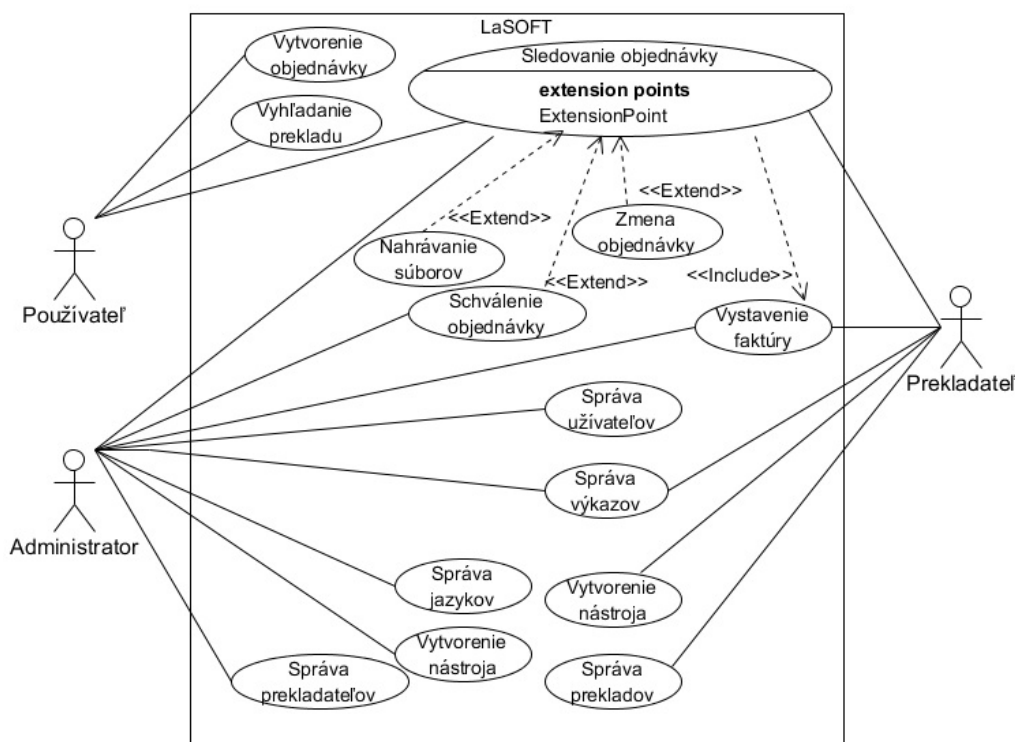
Zobrazenie všetkých zákazníkov a prekladateľov. Administrátor bude mať možnosť zobrazit' všetkých užívateľov v systéme a bude môcť meniť ich profily.

Zobrazenie jazykov a oblastí prekladov. Administrátor bude mať možnosť meniť použité jazyky a oblasti prekladov.

Zobrazenie výkazov. Prekladateľ a administrátor bude mať možnosť zobrazovať evidované výkazy za určité časové obdobie v prehľadnej tabuľke vo forme PDF.

3.5 Funkcie IS

Analýza požiadaviek v systéme popisuje chovanie systému a definuje úlohy, ktoré vstupujú do rozhodovania o potrebách a podmienkach, ktoré sú kladené na systém. Pre modelovanie funkcionality naše ho informačného systému môžeme využiť prípady užívania, ktoré zachytávajú vonkajší pohľad na systém a tým pomáhajú odhaliť hranice systému a slúžia ako podklady pre odhady systému. Hlavnou úlohou je zachytenie aktérov, ktorí so systémom komunikujú a taktiež funkcie a služby, ktoré títo aktéri využívajú.[9]



Obr. 2: Use case zobrazenia funkcionality

Prihlásenie: Užívateľ navštívi obrazovku s prihlasovacími údajmi.

Užívateľ musí pre prihlásenie vyplniť svoj prihlasovací login a heslo. Prihlasovací login a heslo je skontrolované v databáze a ak sa zhoduje je užívateľ na základe jeho role presmerovaný na úvodnú obrazovku privátnej časti informačného systému. Prístup k jednotlivým zdrojom informačného systému je obmedzená na základe užívateľských rolí. Ak užívateľ zadal zlý prihlasovací login alebo heslo je na prihlasovacej obrazovke zobrazená chyba.

Správa profilu: Užívateľ je úspešne prihlásený a z menu aplikácie vyberie profil.

Užívateľ vidí svoj profil s ktorým vystupuje v systéme. Zmena profilu je umožnená každému užívateľovi systému, ktorý je prihlásený do systému. Každý užívateľ v systéme má iba jeden profil. Administrátor však vystupuje pomocou dvoch profilov. Prvým profilom je profil zákazníka a druhým je profil prekladateľa. Spolu s profilom môže prekladateľ a administrátor upravovať svoj úradný podpis alebo pečiatku, ktorý je zobrazený vo vygenerovaných faktúrach.

Zoznam faktúr a vytvorenie novej faktúry: Užívateľ je prihlásený a je v roli zákazník.

Užívateľ vyhledá zo zoznamu prekladov vyhovujúci preklad. Môže využiť vyhľadávaci filter pre nájdenie konkrétnejších záznamov. Po výbere prekladu je presunutý na vytvorenie novej objednávky. Po vyplnení novej objednávky je táto objednávka vytvorená a

čaká na schválenie.

Upload súborov: Užívateľ je prihlásený a je v roli zákazník a vytvoril objednávku.

Užívateľ spolu s objednávkou môže do systému nahrať súbor, ktorý bude zviazaný s objednávkou. Prekladateľ si po prijatí objednávky bude môcť tento súbor stiahnuť a pracovať na ňom.

Zobrazenie objednávok: Užívateľ je prihlásený ako zákazník, prekladateľ alebo administrátor.

Užívateľ si môže zobraziť skupinu objednávok, ktoré sú v rovnakom štádiu spracovania. Každý z užívateľov môže vidieť len definované stavy objednávok.

Schválenie objednávok: Užívateľ je prihlásený a je v roli administrátor.

Po výbere faktúry, ktorá čaká na spracovanie, vidí administrátor prehľad faktúry spolu so súbormi. Administrátor sa môže rozhodnúť či danú objednávku schváli alebo odmietne. Po schválení objednávky je potrebné vyplniť špecifické číslo. Špecifické číslo identifikuje objednávku pre prekladateľa aj zákazníka

Dokončenie objednávky: Užívateľ je prihlásený a je v roli prekladateľ.

Prekladateľ po zobrazení schválenej objednávky, môže do objednávky nahrať súbory, ktoré preložil a po dokončení objednávky môže vystaviť faktúru.

Vystavenie faktúry prekladateľom: Užívateľ je prihlásený a je v roli prekladateľ.

Prekladateľ vyplní faktúru základnými údajmi a pridá do faktúry položky faktúry. Túto faktúru môže zobraziť bez uloženie alebo ju môže dokončiť a tým vystaviť faktúru. Po dokončení objednávky už prekladateľ nemá prístup k danej objednávke. Prekladateľ po dokončení objednávky už nemôže meniť objednávku. Môže si iba zobraziť vygenerovanú faktúru vo formáte PDF.

Dokončenie kompletnej objednávky: Užívateľ je prihlásený a v roli administrátor.

Po dokončení objednávky prekladateľom a vystavení faktúry je táto objednávka presunutá k administrátorovi. Administrátor si v tomto stave objednávky môže zobraziť výslednú faktúru pre prekladateľa alebo môže prejsť k faktúre pre zákazníka. Administrátor pre zákazníka vystavuje novú faktúru s novými popismi a novými položkami faktúry. Administrátor dokončuje objednávku pre zákazníka a ukončuje ju. Objedávka je zobrazená užívateľovi v stave dokončená.

Správa jazykov: Užívateľ je prihlásený a je v roli administrátora.

Administrátor môže spravovať jazyky, z ktorých si zákazníci alebo prekladatelia vyberajú.

Správa nástrojov: Užívateľ je prihlásený a je v roli administrátor.

Administrátor môže spravovať nástroje, ktoré budú prekladatelia využívať.

Správa výkazov: Užívateľ je prihlásený a je v roli prekladateľ.

Pri zobrazení výkazov sú tieto výkazy zobrazené v tabuľke. Prekladateľ môže vytvoriť

nový výkaz, upraviť stávajúci alebo ho vymazať. Prekladateľ si môže zobrazit' prehľadný výpis výkazov vo formáte PDF za určité časové obdobie.

Vytvorenie prekladov: : Užívateľ je prihlásený a je v roli prekladateľ.

Prekladateľ si vedie portfólio svojich služieb a vytvára si ich v položke moje preklady. Pri vytváraní prekladov prekladateľ vytvára dvojice jazykov, ktoré dokáže preložit'. Vyberá jazyk z ktorého prekladá, jazyk do ktorého prekladá, oblasť prekladu, typ služby, cenu od do a či prekladateľ dokáže tento preklad notársky overit'.

3.6 Nefunkčné požiadavky

Nefunkčné požiadavky sú koncepčného charakteru a popisujú vlastnosti ako výkon, udržiava teľnosť, bezpečnosť, rozširiteľnosť taktiež definujú vlastnosti v oblasti chovania, požiadavky na spôsob riešenia (napríklad použitie konkrétnych technológií). Tieto požiadavky sa nevyužívajú pri analýze systému. Zohľadňujú sa až v etape návrhu informačného systému.

3.7 Bezpečnosť

Pretože náš systém by mal byt' prístupný z internetu, musíme systém navrhnuť tak aby bol čo najbezpečnejší. Taktiež pretože systém bude obsahovať osobné údaje každého z užívateľov. Po vyplnení prihlasovacieho mena a loginu sa odošle formulár na spracovanie. Tento proces sa tiež nazýva autentifikácia. Systém overí či v systéme existuje daný užívateľ. Ak v systéme užívateľ existuje. Načítajú sa jeho užívateľské role. Role definuje v akej úlohu bude prihlásený užívateľ vystupovať. Ide o takzvanú kontextovú autorizáciu, pri ktorej dokážeme zabezpečiť zdroje systému iba na základe rolí. Komunikácia medzi klientom a serverom by mala prebiehať skrz zabezpečené spojenie. Pre zabezpečenie komunikácie môžeme využit' https protokol.

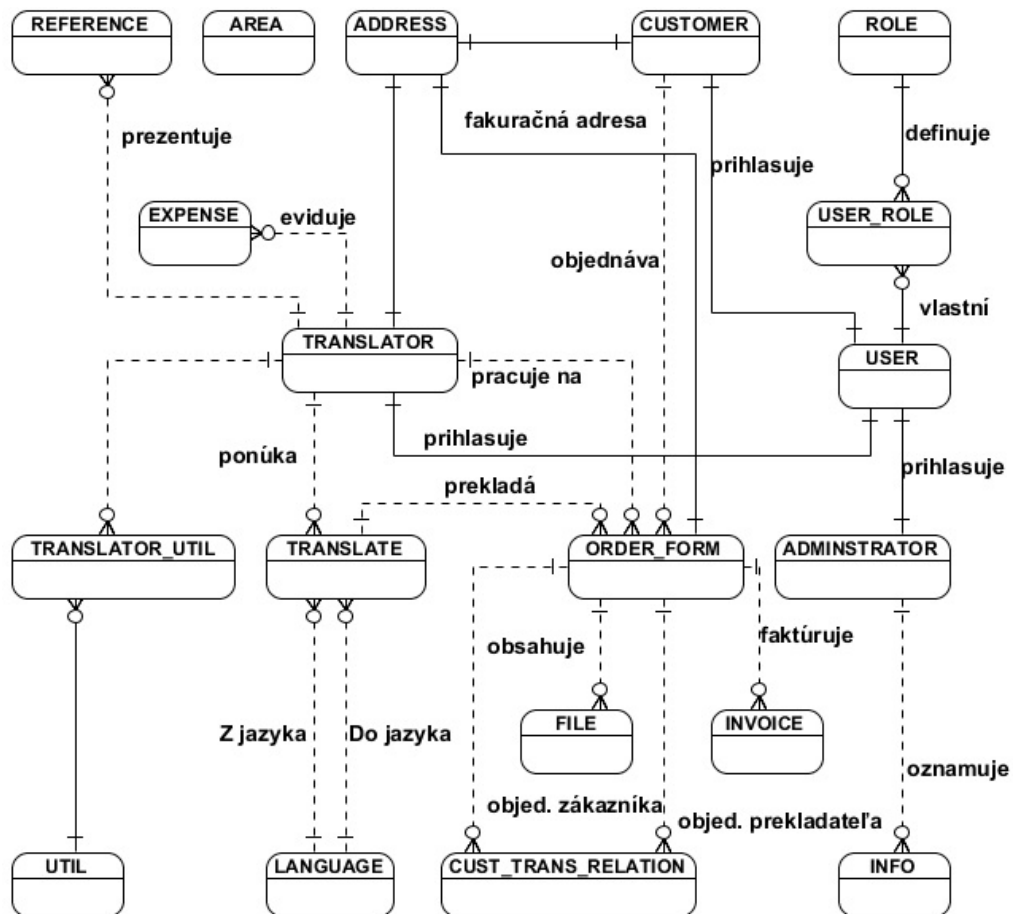
Ochrana pred zneužitím prihlasovacích údajov užívateľov, ktoré sú uložené v databáze je zabezpečená šifrovacím algoritmom SHA-512. Jednotlivé heslá v databáze sú zašifrované a pri porovnaní s prihlasovacím heslom sa tieto porovnávajú v zašifrovanej podobe.

3.8 Dátová analýza

Dátová analýza je základom analýzy informačného systému. Hlavnou úlohou dátovej analýzy je vytvorenie modelu, ktorý bude reprezentovať doménu informačného systému. Dátová analýza je veľmi dôležitá lebo v konečnom dôsledku nám definuje ako budeme v informačnom systéme pracovať s dátami a databázou. Definuje štruktúru databáze, väzby medzi jednotlivými entitami naše ho dátového modelu. Výsledkom takej analýzy je E-R diagram, ktorý zobrazuje konkrétne vzťahy medzi entitami dátového modelu. Lineárny zápis typu entít, ktorý určuje aké atribúty budú v databázových tabuľkách uložené, ktoré atribúty sú primárne kľúče a ktoré cudzie kľúče. Posledným vyjadrením dátovej analýzy je

dátový sloník, ktorý vyjadruje bližšie špecifikácie na štruktúru databáze a taktiež stručne popisuje jednotlivé atribúty konkrétnej databázovej tabuľky.

3.8.1 E-R diagram



Obr. 3: E-R diagram

3.8.2 Lineárny zápis typu entít

Language (*languageId*, *language*, *shortcut*, *type*)

Translator (*translatorId*, *userId*, *title*, *name*, *surname*, *email*, *telephone*, *dateCreate*, *lastLogin*, *ico*, *dic*, *pathCv*, *pathFoto*, *bankName*, *bankAddress*, *bankNumber*, *accountName*, *account*, *iban*, *bic*, *pathSignature*, *addressId*)

Customer (customerId,addressId,userId, name, surname, email, telephone, dateRegistration, lastLogin, comment, typeOfUser, nameOfOrganisation, infoAboutOrganisation, ico, dic)

Reference (id, title, text, translate)

Translate (translateId,fromLanguage,toLanguage, isFromMotherLanguage, isToMotherLanguage, fromArea, notarized, typeOfService, priceFor, priceTo, translator)

Orderfrom (orderId, dateCreate, dateWishcomplete, dateConfirmation, dateComplete, status, position, realPrice, contactName, contactSurname, contactEmail, contactTelephone, info, customerId, translate, translator,address, dueDate, subjectOrder, subjectInfo, tradeInfo, orderNumber)

User_Table (userId, login, password, enable)

Role_table (roleId, code)

Util (id, name, note, translate)

Address (addressId, street, city, postalcode, country)

Admin (adminId, userId)

Area (id, name)

Expense (id, name, type, numberintranslatebook, invoicenummer, date, paid, debit, cost, profit, translator)

File (fileId,orderId, name, prefix, fileToSave, info, owner, typeOfSendingFile, typeOfReceiveFile)

Info (infoId, title, message,dateCreate, designed,adminId)

Invoice (invoiceId, number, item, specificnumber, cost,orderId)

UserRole (userId,roleId)

TranslatorUtil (translator,id)

OrderFormFile (orderId,fileId)

CustTransTelation (id, orderId,orderId,date,specificNumber)

3.8.3 Analýza životného cyklu objednávky

Medzi hlavné funkcie naše ho informačného systému by mala byť správa objednávok. Od vytvorenia objednávky zákazníkom, schválenia objednávky administrátorom, vytvorenia novej objednávky pre prekladateľa až po dokončenie oboch objednávok a vystavenia faktúr. Cieľom tejto analýzy je vyjadriť procesy, ktoré vznikajú pri vytváraní objednávky v systéme. Pre vyjadrenie procesov v informačnom systéme môžeme využiť niekoľko štandardných prostriedkov. Prvým prostriedkom pre vyjadrenie stavu objektu je stavový diagram. Stavový diagram popisuje stavy objektu a povolené prechody medzi týmito

zamietne, prejde do stavu DECLINE. Ak objednávku schváli prejde táto objednávka do stavu PROCESS a v systéme je vytvorená nová objednávka pre prekladateľa, ktorá bude tiež v stave PROCESS. Tieto dve objednávky sú spojené špecifickým číslom a taktiež je vytvorený nový záznam v tabuľke CUST_TRANS_RELATION s týmito objednávkami.

DECLINE - Objednávka je zamietnutá a užívateľ ju môže iba vymazať.

PROCESS - Ak objednávka prejde do stavu PROCESS, prekladateľ na nej začal pracovať. Je umožnené aby si prevzal súbory objednávky a taktiež aby mohol nahrávať preložené súbory. Po dokončení objednávky môže prejsť k dokončeniu objednávky a vystaveniu faktúry. Po vytvorení faktúry ukončí objednávku a táto objednávka prejde do stavu DONE.

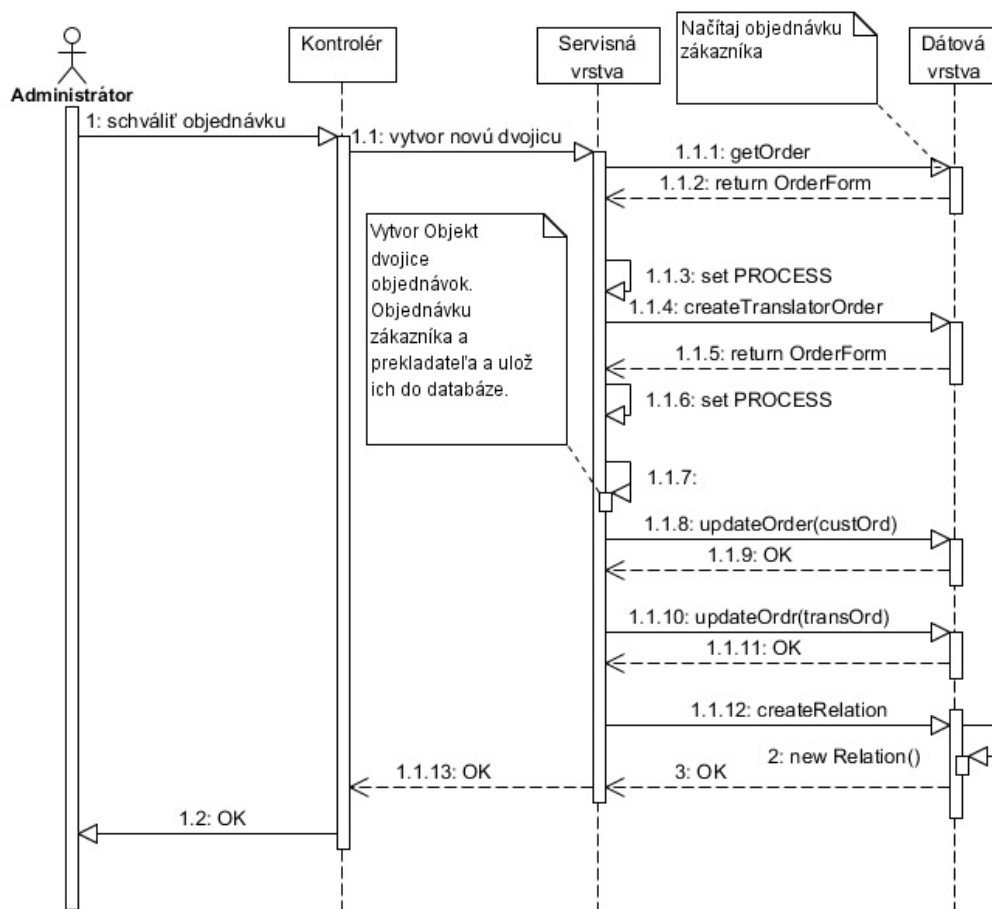
STORNO_PROCESS - Ak sa objednávka nachádza v stave PROCESS, užívateľ sa môže rozhodnúť, že už nechce dané služby a môže danú objednávku zrušiť. Takáto objednávka prejde zo stavu PROCESS do stavu STORNO_PROCESS. Pre prekladateľa je táto faktúra zobrazená v položke „Objednávka zrušená užívateľom“. Prekladateľ by mal ukončiť vykonávanie činnosti na objednávku a mal by vystaviť faktúru za vykonanú prácu, ktorú urobil ešte pred tým ako užívateľ objednávku zrušil. Po ukončení a vystavení faktúry táto objednávka prechádza do stavu DONE.

DONE - Objednávku v stave DONE vidí administrátor a môže vystaviť faktúru pre zákazníka alebo si zobrazí faktúru od prekladateľa, ktorú musí ako prevádzkovateľ systému uhradiť. Ak prejde k vytvoreniu faktúry, táto objednávka prejde do stavu PREPARE. Tento stav signalizuje, že administrátor na objednávke pre zákazníka začal pracovať. Po vystavení faktúry pre zákazníka prechádza objednávka pre zákazníka do stavu FINISH.

FINISH - Objednávka pre zákazníka je ukončená, zákazník si túto objednávku môže zobrazíť, prezvať si súbory objednávky a zobrazíť faktúru. Po prezretí objednávky táto prejde do stavu VIEW.

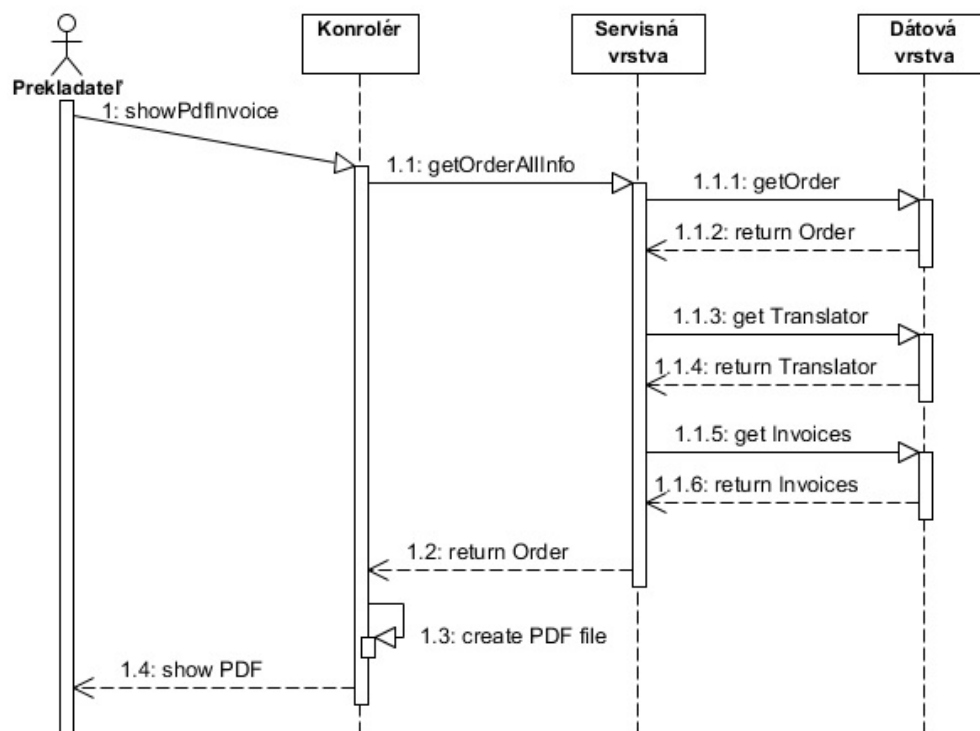
VIEW - Tento stav signalizuje administrátorovi, že zákazník si prezrel faktúru a je oboznámený, kedy ju má uhradiť a podobne.

Pre vyjadrenie procesov, ktoré vznikajú pri vytváraní faktúr a najmä dvojice faktúr prekladateľa a zákazníka som sa rozhodol popísať tento proces pomocou sekvenčného diagramu. Tento diagram zobrazuje procesy, ktoré sa vykonávajú sekvenčne. Sekvenčný diagram zobrazuje sekvencie správ, ktoré sa vymieňajú medzi úlohami, ktoré implementujú chovanie systému. Ukazujú tok riadenia medzi objektmi, ktoré sa zúčastňujú kontextu scenára.



Obr. 5: Sekvenčný diagram schvaľovania objednávky administrátorom

Ďalší proces, ktorého priebeh som sa snažil zachytiť pomocou sekvenčného diagramu je vygenerovanie faktúry do PDF. Tento proces pozostáva z niekoľkých krokov. Prvým krokom je overenie či je faktúra v stave DONE alebo FINISH. Ďalším je načítanie objednávky so všetkými informáciami o prekladateľovi spolu s adresou prekladateľa. Posledným krokom je načítanie položiek faktúry.



Obr. 6: Sekvenčný diagram generovanie výslednej faktúry

4 Implementácia IS

V nasledujúcej časti sa budem zaoberať implementáciou informačného systému. Ako som spomenul na začiatku mojej práce, systém bude vytvorený ako webový projekt. Pre tvorbu informačného systému som si vybral platformu Java. Platforma Java obsahuje množstvo webových rámcov pre tvorbu webových aplikácií. Ako framework pre prezentačnú vrstvu som si vybral webový framework spring MVC.

4.1 Architektúra

Pri návrhu informačného systému som sa snažil dodržať štandardy pri tvorbe viacvrstvovej J2EE aplikácií. Typická enterprise aplikácia sa skladá z 3 vrstiev: prezentačná vrstva, servisná vrstva, a dátová vrstva. Každá vrstva takéhoto informačného systému zodpovedá za určitú časť informačného systému a tieto vrstvy by mali byť striktne oddeliteľné. Jednotlivé vrstvy majú definované tieto vlastnosti.

PREZENTAČNÁ VRSTVA:

Hlavnou úlohou prezentačnej vrstvy je zapuzdrenie prezentačnej logiky, ktorá slúži na zobrazovanie výsledkov klientovi. Spracováva požiadavky, akcie ako prihlasovanie a odhlasovanie. Pre generovanie výsledkov klientovi prezentačná vrstva volá služby business vrstvy. Po volaní business vrstvy je prezentačná vrstva zodpovedná za vygenerovanie odpovede a odoslanie tejto odpovede klientovi.

SERVISNÁ VRSTVA:

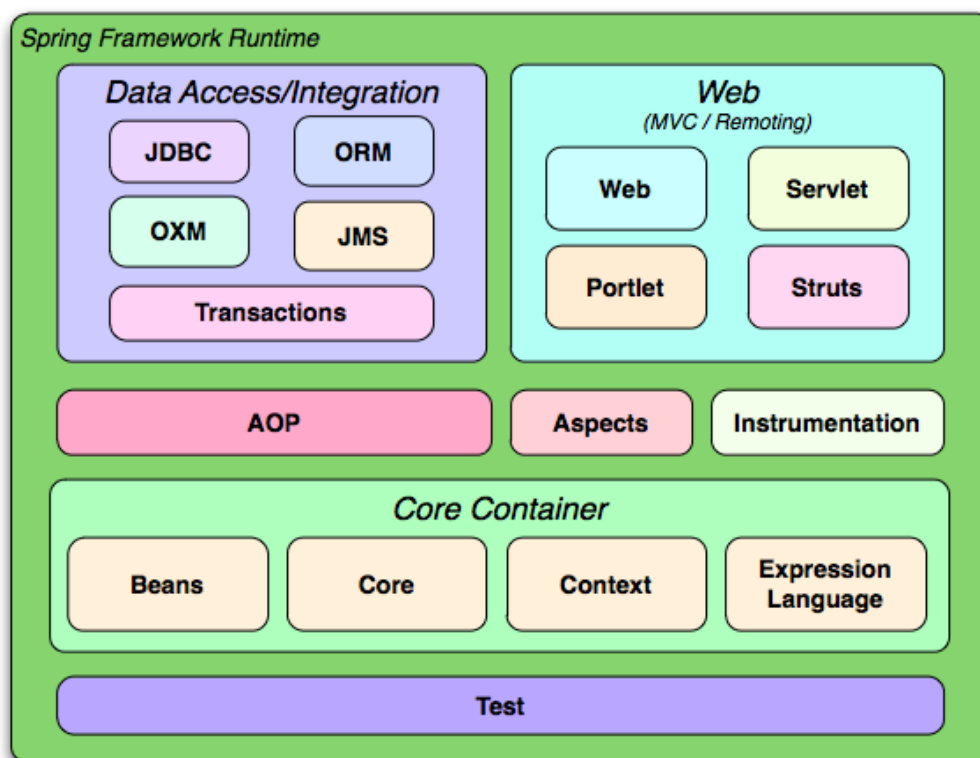
Táto vrstva obsahuje aplikačnú logiku a taktiež obsahuje dáta. Všetka logika aplikácie je centralizovaná do tejto vrstvy čo nám umožňuje ľahko meniť prezentačnú vrstvu bez zásadných zmien business vrstvy. Nie je závislá na prezentačnej vrstve ale využíva služby iných vrstiev akými je napríklad dátová vrstva.

DÁTOVÁ VRSTVA:

Táto vrstva obsahuje logiku pre prístup k dátam. Obecne môže ísť o ľubovoľné dáta. Väčšinou však ale uvažuje o prístupe k databáze. Hlavnou úlohou tejto vrstvy teda je získať z databázy alebo z ľubovoľného zdroja dáta a tie následne poslať vrstve, ktorá ich požaduje. Dátová vrstva je väčšinou závislá už len na použitej technológii pre prístup k dátam a taktiež na použitej databáze.

4.2 Spring Framework

Spring Framework[1, 3] je populárny open-source aplikačný rámec alebo framework pre vývoj J2EE aplikácií. Prvá verzia bola popísaná v knihe Roda Johnsona, ktorý sa zaoberá vývojom J2EE aplikácií a venoval sa problémom, s ktorými sa programátori stretávajú. Framework bol prvýkrát uvoľnený pod licenciou Apache 2.0 v roku 2003. Prvá verzia 1.0



Obr. 7: Architektúra frameworku spring

bola vydaná v roku 2004. Aktuálna verzia je 3.1. Spring sa stal veľmi populárnym a žiadaným rámcom pri tvorbe enterprise aplikácií pretože značne uľahčuje vývoj a následnú údržbu J2EE aplikácií. Stal sa alternatívou k Enterprise Java Bean (EJB), alebo ako jej nadstavba. Jadro frameworku je postavené na využití návrhového vzoru Inversion of Control alebo taktiež Dependency Injection. Spring môžeme taktiež označiť ako IoC kontajner. Tento návrhový vzor funguje na princípe presunutia zodpovednosti za vytváranie a previazanie objektov z aplikácie na framework. Objekty je možné získavať prostredníctvom vsadzovania závislostí. Objekty, ktoré sú vytvorené kontajnerom sú nazývané Beans. Objekty sú rámcom vytvorené typicky na základe načítania konfiguračného súboru vo forme XML, ktorý obsahuje definície týchto Beans. Framework sa skladá z niekoľkých modulov, ktoré môžeme použiť nezávisle. Vo svojej práci využívam niekoľko modulov ako Spring MVC 3, Spring Core, Spring Transaction, Spring Security.

4.2.1 Spring MVC framework

Spring framework sa skladá z niekoľkých modulov a jedným z nich je webový framework spring MVC. V dnešnej dobe ide o jeden z najpopulárnejších frameworkov pre tvorbu webových aplikácií nad platformou Java s využitím návrhového vzoru MVC. Spring

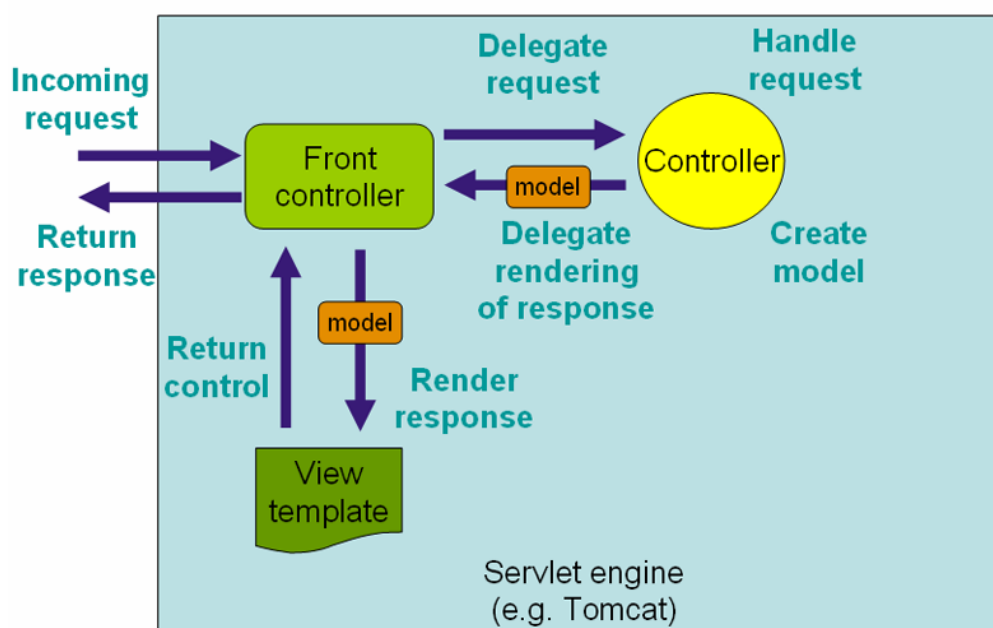
MVC framework je postavený nad štandardným Servlet API a ako prezentačnú vrstvu môže použiť niekoľko prezentačných technológií ako JSP, JSTP, Tiles atď.

4.2.2 MVC Model – View – Controller

Architektúra MVC[8] delí webovú aplikáciu na 3 logické časti tak aby tieto časti bolo možné upravovať samostatne a zmeny na jednej časti by nemali mať zásadný vplyv na ostatné časti. Tieto 3 časti sú Model, View, Controller. Model reprezentuje business logiku aplikácie. View zobrazuje užívateľské rozhranie a Controller má na starosti tok udalostí v aplikácii a obecne prepája prezentačnú a business vrstvu.

4.2.3 Architektúra Spring MVC

Spring MVC framework je rovnako ako mnoho iných webových MVC frameworkov riadený požiadavkou. Každá požiadavka smeruje na takzvaný centrálny servlet a ten ho následne odosiela na konkrétny kontrolér, ktorý ponúka ďalšiu funkcionálnu. Spring MVC je plne integrovateľný so Spring kontajnerom a tým umožňuje využívať všetky funkcie, ktoré framework ponúka.



Obr. 8: Architektúra Spring MVC

Jednou z hlavných komponent, ktoré využíva Spring MVC je DispatcherServlet. Pri DispatcherServlet ide o Front controller návrhový vzor, ktorý je často používaný pri

tvorbe J2EE aplikácií a patrí do skupiny enterprisne návrhových vzorov. Po nakonfigurovaní špeciálneho kontroléra DispatcherServlet, ktorý načíta aplikačný kontext webovej aplikácie je tento servlet schopný prijímať požiadavky a na základe konfigurácie spracovávať jednotlivé požiadavky. Tieto požiadavky na základe konfigurácie prepošle na konkrétny kontrolér, ktorý sa už postará o spracovanie požiadavky.

4.2.4 Implementácia kontrolérov

Hlavnou komponentov pre riadenie logiky aplikácie je kontrolér. Kontroléry definujú správanie aplikácie, ktoré zvyčajne definujeme cez servisné rozhranie. Kontroléry prekladajú užívateľské vstupy a transformujú ich do modelu, ktorý je následne zobrazovaný užívateľovi pomocou pohľadu. Spring implementuje kontroléry abstraktným spôsobom čo umožňuje vytvoriť širokú škálu kontrolérov. Od verzie 2.5 spring podporuje tvorbu kontrolérov pomocou anotácií. Predchádzajúce verzie umožňovali tvorbu kontrolérov implementáciou rozhrania `org.springframework.web.servlet.mvc.Controller` alebo zdedením vlastností z niektorého z preddefinovaných kontrolérov, ktoré už boli definované vo frameworku. Obrázok číslo 6 znázorňuje kontrolér, ktorý zobrazuje zobrazenie prekladateľov.

```
@Controller
public class TranslatorsController {

    @RequestMapping("/translators")
    public ModelAndView showAllTranslators() {
        ModelAndView mav = new ModelAndView();

        List<Translator> allTranslators = serviceTrans. listAllTranslators ();

        mav.setViewName("translatorsView");
        mav.addObject("translators", allTranslators );
        return mav;
    }
}
```

Výpis 1: Implementácia kontroléru v spring MVC

Ešte pred tým ako chceme vytvárať kontroléry s využitím anotácií, musíme v aplikačnom kontexte definovať xml element, ktorým povieme kontajneru aby reagoval a registroval anotácie, ktoré slúžia na definíciu a konfiguráciu kontrolérov. Elementom `context:component-scanbase-package="sk.translate.controllers"` hovoríme springu aby prehľadával a automaticky vytváral objekty, ktoré sú o anotované anotáciami ako `@Controller`, `@Service`, `@Component`. Atribútom `base-package` definujeme, v ktorom balíku ich má hľadať. Druhým elementom `mvc:annotation-driven` hovoríme springu aby reagoval na špecifické anotácie, ktoré slúžia na podporu pri vytváraní kontrolérov.

Po prijatí požiadavky, ktorá smeruje na url: `localhost:8080/app/translators` je tá to požiadavka presmerovaná na metódu `showAllTranslators`, ktorá získa modelové

dáta zo servisnej vrstvy, tieto dáta nastaví do objektu ModelAndView, ktorý reprezentuje dvojicu modelu a pohľadu. Po nastavení pohľadu je tento objekt vrátený z metódy a tým úloha kontroléra končí. Po vygenerovaní pohľadu s využitím dát z modelu je tento pohľad poslaný užívateľovi vo forme html. Spring MVC framework dodržiava MVC návrhový vzor a preto oddeľuje aplikačnú logiku od prezentačnej. Táto flexibilita nám umožňuje flexibilne meniť pohľady a taktiež technológie, ktoré pri generovaní pohľadov môžeme využiť. Z generovania čistého html kódu môžeme flexibilne prejsť na generovanie PDF pohľadov alebo xls súborov.

4.2.5 Prezentačná vrstva

Pre prezentačnú vrstvu som si ako prezentačnú technológiu vybral JavaServer Pages (JSP)[7, 6] a značkovaciu knižnicu JavaServer Pages Standard Tag Library (JSTL). JSP bola vyvinutá spoločnosťou Sun Microsystems, za účelom vývoja aplikácií na strane servera. JSP súbory sú vlastne HTML stránky, do ktorých sú vložené špeciálne tagy obsahujúce javovský zdrojový kód. JSP stránky by mali obsahovať iba prezentačnú logiku. Pre lepšie spracovanie modelu môžeme využiť knižnicu JSTL. JSTL definuje množinu tagov, ktoré uľahčujú bežné úlohy ako spracovanie XML, spracovanie podmienok, slučiek a internacionalizácie. JSTL poskytuje efektívny spôsob ako vložiť logiku v rámci JSP stránky bez priameho použitia java kódu.

Pre zefektívnenie práce s HTML a CSS som použil open source projekt bootstrap, ktorý vznikol pod záštitou spoločnosti twitter. Ide o sadu CSS štýlov a preddefinovaných layoutov, ktoré v plnej miere využívajú nové trendy pri tvorbe webových aplikácií. V dnešnej dobe sa projekt bootstrap stal pre mnohé komerčné projekty základ pre formátovanie a štylizáciu webových prezentácií. Bootstrap sa snaží v plnej miere tvoriť komponenty pomocou nových štandardov pre tvorbu webových prezentácií ako sú HTML5 a CSS3.

4.3 Servisná vrstva

Servisná vrstva by mala obsahovať väčšinu logiky aplikácie. Pri implementácii v prostredí J2EE existujú dva spôsoby vytvárania servisnej vrstvy. Prvou z nich je použitie Enterprise Java Beans (EJB). EJB pre svoj beh potrebuje aplikačný server, ktorý obsahuje EJB kontajner. EJB kontajner riadi hlavne komunikáciu so vzdialeným klientom, životný cyklus EJB bean a manažment transakcií. Hlavnou nevýhodou vytvárania servisnej vrstvy J2EE aplikácií pomocou EJB je testovateľnosť aplikácie, prenositeľnosť, rýchlosť spúšťania aplikačného servera, náklady na EJB server.

Druhým spôsobom vytvárania servisnej vrstvy je použitie springu s použitím čistých Plain Old Java Object (POJO) objektov. V krátkosti ide o objekty, ktoré nemusia pre svoju funkcionálnosť implementovať špecifické rozhranie. Použitie springu zjednodušuje návrh, čitateľnosť, udržiavateľnosť a testovanie projektu. Spring je schopný zaistiť deklaratívnu správu transakcií, správu databázových spojení, uľahčuje napojenie na databázovú vrstvu aplikácie a taktiež umožňuje deklaratívnu správu zabezpečenia. Jednou z hlavných výhod

však je, že spring nepotrebuje pre svoj beh aplikačný server ako JBoss a podobne. V plnej miere si vystačíme s webovým serverom ako Tomcat. Preto sa tento framework tiež označuje ako `lightweight` ľahký. Servisná vrstva by mala obsahovať nasledovnú funkcionálnu:

- **správa transakcií** – Pre správu a riadenie transakcií využívam podporné anotácie `@Transactional`. Ak chceme aby naše servisné metódy boli vykonávané v jednej transakcii, stačí zdefinovať takúto triedu s anotáciou `@Transactional` a v aplikačnom kontexte nakonfigurovať príslušného transakčného manažéra.
- **zabezpečenie servisnej vrstvy** – Servisná vrstva by mala zabezpečovať zabezpečenie aplikácie a centralizovaný prístup k jednotlivým zdrojom z databáze.

4.4 Databázová vrstva

Pre implementáciu dátovej vrstvy existuje niekoľko postupov. Ja som využil možnosti objektovo relačného mapovania. Pri využití objektovo relačného mapovania taktiež existuje niekoľko postupov.

Prvým z nich je manuálne mapovanie výsledkov z databáze na Java objekty a naopak. Tento prístup je podľa môjho názoru najrýchlejší z pohľadu výkonu databáze. Medzi databázou a aplikáciou nevzniká žiadna skrytá medzivrstva. Prakticky všetku kontrolu nad mapovaním záznamov z databáze na objekty má programátor pod kontrolou. Čo je však veľkou nevýhodou tejto metódy písania si vlastného objektovo relačného mapovania je škálovateľnosť takejto aplikácie. Pri zmene databázovej schémy musí programátor na každom mieste kde došlo k zmene manuálne meniť relačné mapovanie. Tento prístup je nevýhodný hlavne pri iteratívnom vývoji. Hodí sa na projekty, ktoré majú menej ako 30 tabuliek.

Druhou možnosťou je využitie niektorého z dostupných objektovo relačných nástrojov, ktorých je v dnešnej dobe veľké množstvo. Ja som využil rámec Hibernate. Hibernate[5] je open source knižnica pre programovací jazyk Java. Poskytuje rámec pre mapovanie doménových objektov do relačnej databáze a naopak. Ide o hlavnú funkciu, ktorú poskytuje. Umožňuje aplikácii byť nezávislou na databáze a zmeniť implementáciu databáze iba pomocou pár konfiguračných parametrov. Objektovo relačné mapovanie konfiguruje programátor prostredníctvom XML konfiguračných súborov alebo anotácií. Programátor definuje ako chce aby sa objekt namapoval do relačnej databázy a to tak, že určí aký atribút triedy má zodpovedať databázovému stĺpcu tabuľky. Umožňuje mapovanie 1:N, M:N na reprezentáciu kolekcii v doménovej triede. Programátor k dátam nepristupuje pomocou čistého SQL jazyka ale využíva takzvaný HQL (Hibernate Query Language). Programátor sa pri písaní HQL dotazov neodkazuje na databázové názvy tabuliek a stĺpcov ale využíva názvy objektov a atribútov modelových tried. Tento dotazovací jazyk vychádza z SQL preto je jednoduché sa ho naučiť. Z môjho pohľadu ide o komplexný nástroj, ktorý rieši mnoho problémov ale taktiež so sebou prináša niektoré problémy, na ktoré musí programátor myslieť pri vytváraní aplikácií pomocou tohto nástroja.

Výhody:

- Nezávislosť na databáze. Jednou z výhod, ktoré poskytuje je nezávislosť na databáze. Toto vyjadrenie sa môže zdať ako výhoda ale musíme si uvedomiť, že pri vývoji informačných systémom je prakticky jasné od začiatku návrhu na akej databáze bude systém prevádzkovaný.
- Programátor nemusí písať zložité SQL dotazy. Hibernate umožňuje ukladanie objektov pomocou jediného príkazu. Taktiež umožňuje získavanie objektov bez znalosti SQL. Obsahuje api pre vytváranie dynamických HQL dotazov za behu bez použitia konkatenácie reťazcov.
- Stará sa o závislosti medzi jednotlivými tabuľkami.
- Hibernate podporuje cachovanie objektov. Podporuje niekoľko poskytovateľov vrstvy pre cachovanie. V krátkosti ide o to, že medzi dátovou vrstvou a databázou je ešte medzivrstva, ktorá sleduje prístup k databáze. Ak napríklad užívateľ v krátkom časovom úseku chce načítať z databáze dva krát objekt s id 1. Hibernate po prvom načítaní objektu z databáze tento objekt umiestni do cache a ak chce užívateľ získať tento objekt z databáze druhý krát, hibernate nenačíta tento objekt z databázy ale z cache. Podpora cache je síce dobrý nástroj ale musíme byť veľmi opatrný pri konfigurácií.

Nevýhody:

- Ide o nástroj s množstvom API. Preto trvá pomerne dlhú dobu oproti čistému JDBC kým programátor pochopí všetky princípy a dokáže naozaj efektívne vytvárať databázové aplikácie s využitím tohto rámca.
- Hibernate vytvára medzivrstvu medzi dátovou vrstvou a databázou, ktorá je pre programátora skrytá. Táto vrstva zo sebou prináša spomalenie aplikácie oproti čistému JDBC.
- Pri zlej konfigurácii môže spôsobiť to, že jedným jednoduchým dotazom do databáze načítame všetky dáta z databáze čo je nežiaduce. Preto by sme mali pri konfigurácii mapovania uvažovať ako budeme naše dáta získavať. To zo sebou prináša taktiež nutnosť uvažovať pri navrhovaní databáze o použití ORM nástrojov.

Pri programovaní nášho informačného systému som sa zoznámil so základnými možnosťami tohto rámca. Z môjho subjektívne pohľadu môžem povedať, že hibernate je použiteľný nástroj, ktorý obsahuje množstvo výhod ale aj nevýhod, ktoré som objavil počas vývoja informačného systému a s ktorými musí programátor počítať pri navrhovaní aplikácií a pri získavaní dát z databázy. Medzi hlavné výhody, ktoré by som spomenul je jednoduché mazanie, ukladanie a upravovanie objektov.

Pri načítavaní objektov musí byť programátor veľmi opatrný pri konfigurácií. Využil som takzvaný prístup fetch, kedy som objekty v asociácií načítal pomocou jedného dotazu

z databáze a len tie, ktoré som potreboval. Snažil som sa všetky asociácie medzi objektmi nastavovať tak aby ich hibernate načítal až keď budú potrebné.

4.5 Prevádzka systému

Nasledujúce obrázky znázorňujú časť systému určenú pre administrátora, ktorá obsahuje správu objednávok. Ďalším obrázok znázorňuje informácie o objednávke spolu s možnosťou vygenerovanie PDF.

The screenshot shows the LaSOFT administrator interface. At the top, there is a navigation bar with the following items: LaSOFT, Profil, Objednávky, Použíatelia, Prekladatelia, Jazyky, Nástroje, Oblasť prekladu, and a user status 'Vitajte admin2 | Odhlásiť sa'. On the left side, there is a sidebar menu under the heading 'OBJEDNÁVKY' with the following items: Čakajúce na schválenie, Schválené, Dokončená, Pripravená na odoslanie, and a section 'DOKONČENÉ' with 'Uzavreté' and 'Prevzaté užívateľom'. The main content area displays a table of orders.

Akcie	Z jazyka	Skratka	do jazyka	Skratka	Oblasť	Meno	Priezvisko	Vytvorená	Dátum ukončenia	Typ služby
Vyber	Slovensky jazyk	SK	Bulharský jazyk	BUL	informatika	Peter	Trnka	22.04.2012	28.04.2012	TRANSLATE

Obr. 9: Ukážka prostredia pre administrátora

LaSOFT Profil Objednávky Použíatelia Prekladatelia Jazyky Nástroje Oblasť prekladu

MENU
 Objednávka
 Užívateľské súbory
 Preložené súbory

Objednávka			
Z jazyka	do jazyka	Type ponúkanej služby	Notársky overený
Slovensky jazyk	Bulharský jazyk	Preklad	Áno
Dátum vytvorenia	Želaný dátum ukončenia	Dátum porvrdenia	Dátum ukončenia
Ne 22.04.2012	So 28.04.2012	Ne 22.04.2012	Ne 22.04.2012
Prekladateľ			
Meno		Priezvisko	
Peter		Rafaj	
Skutočná cena			
120.0			
Zákazník			
Meno	Priezvisko	Email	Telefón
Peter	Trnka	customer@gmail.com	0908276979
Kontaktná osoba			
Meno	Priezvisko	Email	Telefón
j	j	j	j
Pozícia			
j			
Fakturačná adresa			
Ulica	Mesto	PSČ	Krajina
Bratislavská 120	Bratislava	09000	Slovensko
Bližšie info			
j			

Zobraz faktúru zákazníka Zobrazit faktúru prekladateľa

Obr. 10: Ukážka zobrazenia informácií o objednávke

5 Záver

Cieľom tejto bakalárskej práce bolo vytvorenie informačného systému pre podporu prekladateľskej činnosti prekladateľov a znalcov jazyka. V prvom bode som sa musel zoznámiť s činnosťou prekladateľov a prekladov. Ide o rozsiahlu problematiku pretože existuje množstvo druhov prekladateľov a znalcov. Tak isto prekladateľove služby môže využívať z právneho hľadiska široká skupina užívateľov od Policajných orgánov, súdov až po bežných užívateľov. S prekladateľskou činnosťou sú spojená potreba evidovať výkazy o vykonanej práci ale aj evidencia jednotlivých prekladov v prekladateľskom denníku. Systém umožňuje prekladateľom spravovať svoju prekladateľskú činnosť na jednom mieste.

Aby z prekladateľskej činnosti profitoval aj náš systém musela medzi prekladateľom a zákazníkom vzniknúť ešte ďalšia rola a tou je administrátor. Tým nám vzniká systém, kde sa zákazník obracia na systém, teda na administrátora a ten vytvára objednávku na prekladateľské služby pre prekladateľa. Po dokončení objednávky prekladateľom administrátor vystavuje novú faktúru pre zákazníka ale už z inou cenou akú obdržal od prekladateľa.

Vývoj informačného systému zahrňuje množstvo procesov a činností. Pri implementácii som sa snažil dodržiavať tieto štandardy. Počas implementácie som sa naučil používať Java jazyk a taktiež rámec spring pre tvorbu trojvrstvových webových aplikácií. Systém pre komunikáciu medzi klientom a serverom využíva zabezpečené spojenie čo napomáha bezpečnosti.

Informačný systém bol funkčne otestovaný. Avšak ako každý softvérový produkt, tie skutočné chyby sa objavujú až v produkčnom nasadení. Ako ďalšie rozšírenie systému vidím notifikáciu užívateľom emailom o zmenách v ich objednávkach alebo prepracovanú správu súborov, ktoré boli nahraté do systému.

6 Literatúra

- [1] Referenčná príručka spring MVC, *spring-framework-reference*, [online] 2004-2010 Dostupná z WWW: <http://static.springsource.org/spring/docs/3.0.x/spring-framework-reference/html/mvc.html>.
- [2] Moro Systems, *Černý kůň vývoje software*, [online] 2006-2010 Dostupný z WWW: <http://morosystems.cz/>.
- [3] Gary Mak, Josh Long, Daniel Rubio, *Spring Recipes Second Edition*, 2010, Apress, ISBN-13: 978-1-4302-2500-3
- [4] Peter Mularien, *Spring Security 3*, 2010, Packt Publishing, ISBN 978-1-847199-74-4
- [5] Gary Mak, *Hibernate Tutorials*, verzia preložená do slovenčiny, Dostupná z WWW: <http://ics.upjs.sk/novotnyr/java/hibernate-tutorial/hibernate-tutorial.pdf>.
- [6] Wikipedie, *otvorená encyklopédia*, JavaServer Pages Standard Tag Library, Dostupná z WWW: http://en.wikipedia.org/wiki/JavaServer_Pages_Standard_Tag_Library, 2012.
- [7] Wikipedie, *otvorená encyklopédia*, JavaServer Pages, Dostupná z WWW: http://en.wikipedia.org/wiki/JavaServer_Pages, 2012.
- [8] www.zdrojak.root.cz, *Úvod do architektúry mvc*, Dostupná z WWW: <http://zdrojak.root.cz/clanky/uvod-do-architektury-mvc/>, 2012.
- [9] <http://mpavus.wz.cz/uml/>, *Use case diagram*, Dostupná z WWW: <http://mpavus.wz.cz/uml/uml-b-use-case-3-2-1.php>, 2012.

A Príloha

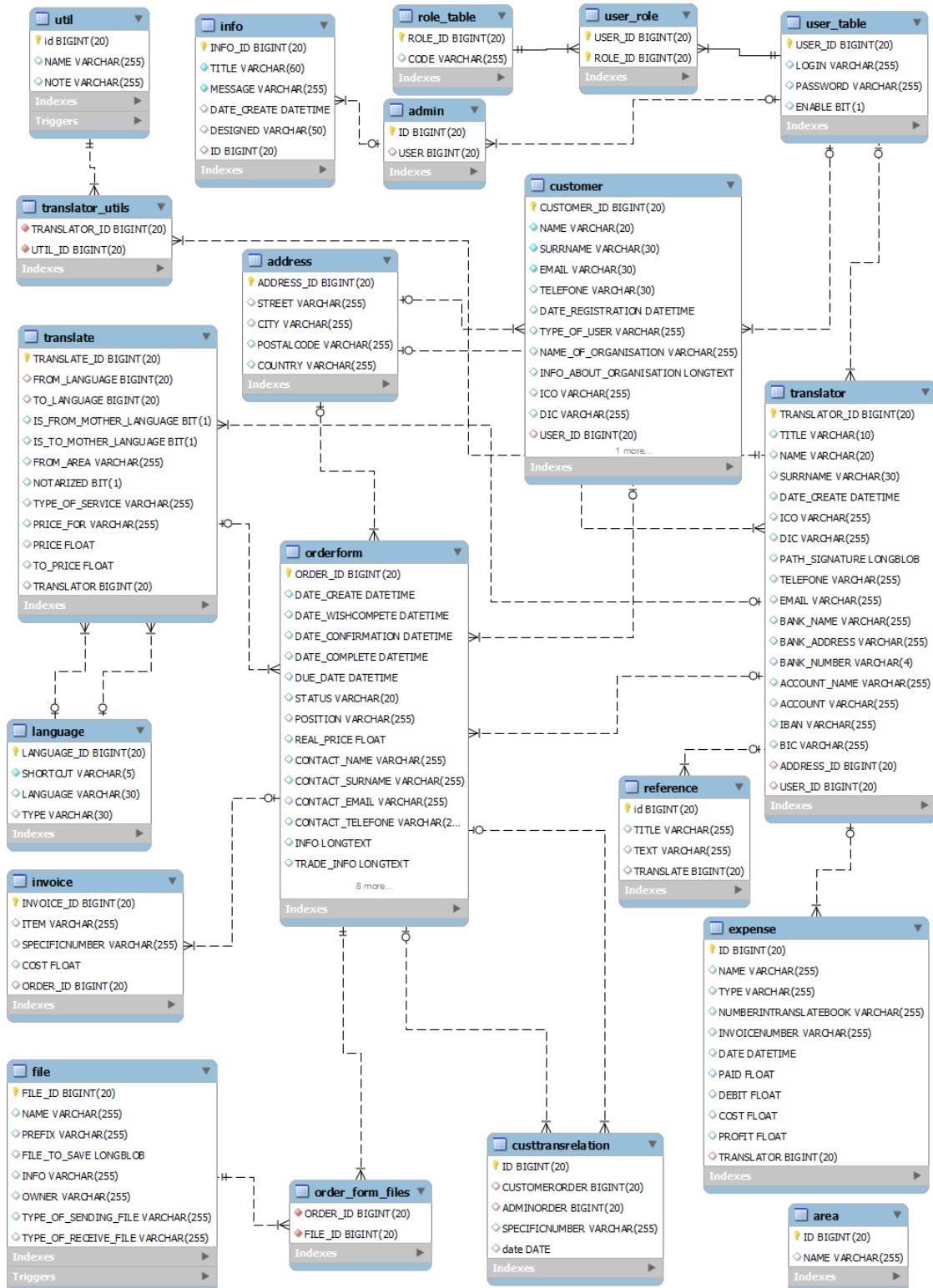
A.1 Obsah CD

Rozsiahle prílohy ako užívateľská dokumentácia a programátorská dokumentácia spolu so zdrojovými kódmi je obsiahnutá na priloženom CD. CD obsahuje nasledovné súbory:

- obraz operačného systému Windows XP s nastaveným webovým serverom Tomcat 7.0 a nainštalovanou databázou MySql
- Zdrojové súbory informačného systému v zložke translate
- programovú dokumentáciu vo forme PDF.
- užívateľskú dokumentáciu vo forme PDF.
- sql skript pre naplnenie databáze.

A.2 Schéma databáze

A.3 Dátový slovník



Obr. 11: Schéma databáze

nazov	typ	veľkosť	klúč	null	popis
languageId	INTEGER		PK	nie	jednoznačný identifikátor
language	VARCHAR	30	nie	nie	názov jazyka
shortcut	VARCHAR	5	nie	nie	definovaná skratka
type	VARCHAR	30	nie	nie	anglický názov jazyka

Tabuľka 1: Tabuľka Language

nazov	type	veľkosť	klúč	null	popis
translatorId	INTEGER		PK	nie	jednoznačný identifikátor
userId	INTEGER		FK	nie	jednoznačný identifikátor užívateľa
title	VARCHAR	10	nie	nie	titul
name	VARCHAR	30	nie	nie	meno
surname	VARCHAR	30	nie	nie	priezvisko
email	VARCHAR	30	nie	nie	kontaktný email
telephone	VARCHAR	30	nie	nie	kontaktný telefón
dateCreate	date		nie	nie	dátum vytvorenia
ico	VARCHAR	10	nie	ano	ico
dic	VARCHAR	10	nie	ano	dic
pathSignature	longblob		nie	ano	súbor s podpisom
bankName	VARCHAR	30	nie	nie	názov banky
bankNumber	VARCHAR	4	nie	nie	identifikačné číslo banky
bankAddress	VARCHAR	30	nie	nie	adresa banky
accountName	VARCHAR	30	nie	nie	názov účtu
account	VARCHAR	30	nie	nie	číslo účtu
iban	VARCHAR	30	nie	ano	medzinarodný identifikátor účtu
bic	VARCHAR	10	nie	ano	medzinarodný identifikátor banky
addressId	INTEGER		FK	nie	cudzí klúč adresy

Tabuľka 2: Tabuľka Translator

nazov	type	veľkosť	kľúč	null	popis
roleId	INTEGER		PK	nie	jednoznačný identifikátor role kod
code	VARCHAR	20	nie	nie	

Tabuľka 3: Tabuľka Role

nazov	type	veľkosť	kľúč	null	popis
id	INTEGER		PK	nie	jednoznačný identifikátor
title	VARCHAR	30	nie	nie	názov referencie
text	VARCHAR	100	nie	nie	rozšírené informácie o referenciách

Tabuľka 4: Tabuľka Reference

nazov	type	veľkosť	kľúč	null	popis
translator	INTEGER		FK,PK	nie	jednoznačný identifikátor prekladateľa
id	INTEGER		FK,PK	nie	jednoznačný identifikátor nástroja

Tabuľka 5: Tabuľka TranslatorUtils

nazov	type	veľkosť	kľúč	null	popis
customerId	INTEGER		PK	nie	jednoznačný identifikátor zákazníka
addressId	INTEGER		FK	nie	jednoznačný identifikátor adresy
userId	INTEGER		FK	nie	jednoznačný identifikátor užívateľa
name	VARCHAR	30	nie	nie	meno
surname	VARCHAR	30	nie	nie	priezvisko
email	VARCHAR	30	nie	nie	email
telephone	VARCHAR	30	nie	nie	telefón
dateRegistration	date		nie	nie	dátum registrácie
typeOfUser	VARCHAR	15	nie	nie	typ registrovaného užívateľa
nameOfOrganisation	VARCHAR	100	nie	ano	názov organizácie
infoAboutOrganisation	VARCHAR	255	nie	ano	bližšie informácie o organizácii
ico	VARCHAR	10	nie	ano	ico
dic	VARCHAR	10	nie	ano	dic

Tabuľka 6: Tabuľka Customer

nazov	type	veľkosť	klúč	null	popis
userId	INTEGER		FK,PK	nie	jednoznačný identifikátor užívateľa
role	INTEGER		FK,PK	nie	jednoznačný identifikátor role

Tabuľka 7: Tabuľka UserRole

nazov	type	veľkosť	klúč	null	popis
invoiceId	INTEGER		PK	nie	jednoznačný identifikátor objednávky
number	VARCHAR	10	nie	nie	špecifické číslo
item	VARCHAR	255	nie	nie	položka faktúry
cost	FLOAT		nie	nie	cena položky
orderId	INTEGER		FK	nie	identifikátor objednávky

Tabuľka 8: Tabuľka Invoice

nazov	type	veľkosť	klúč	null	popis
infoId	INTEGER		PK	nie	jednoznačný identifikátor správy
title	VARCHAR	30	nie	nie	nadpis správy
message	VARCHAR	255	nie	nie	správa
adminId	INTEGER		FK	nie	identifikátor administrátora
dateCreate	date		nie	nie	dátum vytvorenia
designed	VARCHAR	20	nie	nie	informácie pre koho je správa určená

Tabuľka 9: Tabuľka Info

nazov	type	veľkosť	klúč	null	popis
id	INTEGER		PK	nie	identifikátor výkazu
name	VARCHAR	20	nie	nie	meno výkazu
type	VARCHAR	20	nie	nie	typ výkazu
idTranslateBook	VARCHAR	10	nie	nie	číslo v slovníku
invoiceNumber	VARCHAR	10	nie	nie	číslo faktúry
date	date		nie	nie	dátum vytvorenia
paid	FLOAT		nie	nie	zaplatené
debit	FLOAT		nie	nie	čiastka, ktorú dlží
cost	FLOAT		nie	nie	cena služby
profit	FLOAT		nie	nie	zisk
translator	INTEGER		FK	nie	id prekladateľa

Tabuľka 10: Tabuľka Expense

nazov	type	veľkosť	klúč	null	popis
id	INTEGER		PK	nie	identifikátor okruhu
name	VARCHAR		nie	nie	okruh

Tabuľka 11: Tabuľka Area

nazov	type	veľkosť	klúč	null	popis
id	INTEGER		PK	nie	jednoznačný identifikátor administrátora
userId	INTEGER		FK	nie	jednoznačný identifikátor užívateľa

Tabuľka 12: Tabuľka Administrator

nazov	type	veľkosť	klúč	null	popis
adressId	INTEGER		PK	nie	jednoznačný identifikátor adresy
street	VARCHAR	40	nie	nie	uloča
city	VARCHAR	30	nie	nie	mesto
postalCode	VARCHAR	5	nie	nie	poštové smerové číslo
country	VARCHAR	20	nie	nie	krajina

Tabuľka 13: Tabuľka Address

nazov	type	veľkosť	klúč	null	popis
id	INTEGER		PK	nie	jednoznačný identifikátor v systéme
name	VARCHAR	50	nie	nie	názov použitého nástroja
note	VARCHAR	255	nie	nie	poznámka o nástroji

Tabuľka 14: Tabuľka Util

nazov	type	veľkosť	klúč	null	popis
translateId	INTEGER		PK	nie	jednoznačný identifikátor prekladu
formLanguage	INTEGER		FK	nie	identifikátor prekladaného jazyka
toLanguage	INTEGER		FK	nie	identifikátor preloženého jazyka
isFromMotherLang	boolean		nie	ano	rodinný hovorca prekladaného jazyka
isToMotherLang	boolean		nie	ano	rodinný hovorca preloženého jazyka
fromArea	VARCHAR	30	nie	nie	oblasť prekladu
notarized	boolean		nie	ano	preklad je notársky overený
typeOfService	VARCHAR	30	nie	nie	typ ponúkanej služby
priceFor	FLOAT		nie	nie	interval ceny od
toPrice	FLOAT		nie	nie	interval ceny do
translator	INTEGER		FK	nie	identifikátor prekladateľa

Tabuľka 15: Tabuľka Translate

nazov	type	veľkosť	klúč	null	popis
fileId	INTEGER		PK	nie	primárny klúč súboru
name	VARCHAR	100	nie	nie	názov súboru
prefix	VARCHAR	50	nie	nie	prefix súboru
fileToSave	BLOB		nie	nie	súbor
info	VARCHAR	255	nie	ano	informácie o súbore
owner	VARCHAR	10	nie	nie	vlastník súboru
typeOfSendingFile	VARCHAR	20	nie	ano	typ odoslania súboru
typeOfReceiveFile	VARCHAR	20	nie	ano	typ prijatia súboru

Tabuľka 16: Tabuľka File

nazov	type	veľkosť	klúč	null	popis
orderId	INTEGER		PK	nie	jednoznačné id objednávky
dateCreate	date		nie	nie	dátum vytvorenia
dateWithComplete	date		nie	nie	želaný dátum dokončenia
dateComplete	date		nie	ano	dátum dokončenia
dueDate	date		nie	ano	dátum splatnosti
statu	VARCHAR	15	nie	nie	status objednávky
position	VARCHAR	30	nie	ano	pozícia kontaktnej osoby
realPrice	FLOAT		nie	ano	výsledná cena
contactName	VARCHAR	30	nie	ano	meno kontaktnej osoby
contactSurname	VARCHAR	30	nie	ano	priezvisko kont. osoby
contactEmail	VARCHAR	30	nie	ano	email kont. osoby
contactTelephone	VARCHAR	20	nie	ano	telefón kont. osoby
info	VARCHAR	255	nie	ano	doplňujúce informácie o objednávke
tradeInfo	VARCHAR	255	nie	ano	živnostenské informácie
orderNumber	VARCHAR	15	nie	ano	číslo objednávky
subjectOrder	VARCHAR	100	nie	ano	nadpis objednávky
subjectInfo	VARCHAR	255	nie	ano	popisné informácie
specificNumber	VARCHAR	15	nie	ano	špecifické číslo
customerId	INTEGER		FK	nie	id zákazníka
translate	INTEGER		FK	nie	id prekladu
translator	INTEGER		FK	nie	id prekladateľa
address	INTEGER		FK	nie	id fakturačnej adresy

Tabuľka 17: Tabuľka OrderForm

nazov	type	veľkosť	klúč	null	popis
userId	INTEGER		PK	nie	jednoznačný identifikátor užívateľa
login	VARCHAR	20	nie	nie	login použitý pri prihlásení
password	VARCHAR	255	nie	nie	heslo užívateľa
enable	boolean		nie	nie	povolí užívateľa

Tabuľka 18: Tabuľka User

nazov	type	veľkosť	klúč	null	popis
orderId	INTEGER		PK,FK	nie	id objednávky
fileId	INTEGER		PK,FK	nie	id súboru

Tabuľka 19: Tabuľka OrderFormFile

nazov	type	veľkosť	klúč	null	popis
id	INTEGER		PK	nie	id relácie
customerOrder	INTEGER		FK	nie	id objednávky zákazníka
adminOrder	INTEGER		FK	nie	id objednávky administrátora
specificNumber	VARCHAR	15	nie	nie	špecifické číslo
date	date		nie	nie	dátum vytvorenia

Tabuľka 20: Tabuľka CustTransRelation