

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

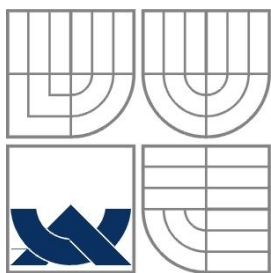
INFORMAČNÍ SYSTÉM PRO PŘEKLADATELSKÉ
SDRUŽENÍ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

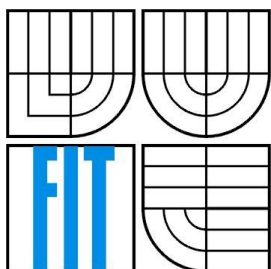
AUTOR PRÁCE
AUTHOR

JAN NEŘOLD

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

INFORMAČNÍ SYSTÉM PRO PŘEKLADATELSKÉ SDRUŽENÍ

INFORMATION SYSTEM FOR TRANSLATORS ASSOCIATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN NEŘOLD

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. BRONISLAV PŘIBYL

BRNO 2011

Abstrakt

Podniky jsou v dnešní době a ekonomické situaci vystavovány velkému tlaku na snižování nákladů a dosahování maximálního zisku a vysoké produktivity. K dosažení těchto požadavků velkou měrou přispívá informační systém, který umožňuje zefektivnit klíčové procesy podniku. Tato práce se zabývá analýzou požadavků, návrhem, implementací a testováním informačního systému pro překladatelské sdružení NK Translators s přihlédnutím k možnému využití systému jinými sdruženími. Analýza a specifikace požadavků byla provedena s využitím jazyka UML. K implementaci v podobě webové aplikace byly použity klasické webové technologie jako HTML, Javascript, PHP a databázový systém MySQL. Implementovaný systém byl nakonec úspěšně podroben uživatelskému testování.

Abstract

In today's economic situation, companies are exposed to pressure because of the demand for high quality services at low prices. The information system which makes key processes of a company much more efficient means an important support in reaching those demands. This thesis is concerned with the analysis of requirements, design, implementation, and testing of the information system for NK Translators, the translators association. The analysis and specification of the requirements were carried out with usage of UML. Web technologies like HTML, Javascript, PHP, and MySQL database system were used for implementation of the information system in the form of a web application. Finally, the implemented system was successfully submitted to a user testing.

Klíčová slova

Informační systém, překladatelské sdružení, relační databáze, Model-View-Controller, Unified Modeling Language.

Keywords

Information System, Translators Association, Relational Database, Model-View-Controller, Unified Modeling Language.

Citace

Neřold Jan: Informační systém pro překladatelské sdružení, bakalářská práce, Brno, FIT VUT v Brně, 2011

Informační systém pro překladatelské sdružení

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Bronislava Příbyla. Další informace mi poskytli překladatelé ze sdružení NK Translators. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jan Neřold
18. května 2011

Poděkování

Tímto bych rád poděkoval mému vedoucímu Ing. Bronislavu Příbylovi za odborné vedení, vstřícný přístup během celé práce a konzultací a za cenné rady pro uskutečnění práce. Rovněž děkuji Lukáši Kadidlovi a Pavle Vondrákové za spolupráci při specifikaci požadavků systému. Poslední dík patří překladatelské agentuře Abiturient, která poskytla informace o svém informačním systému pro inspiraci této bakalářské práce.

© Jan Neřold, 2011.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

1	Úvod.....	2
2	IS konkurenčních překladatelských sdružení	3
2.1	Agentura Abiturient	3
3	Metodologie a praktiky pro návrh IS	5
3.1	Návrh databázového schématu	5
3.2	UML.....	14
3.3	Architektura Model-View-Controller	20
4	Analýza a specifikace požadavků	23
4.1	Uživatelské akce	24
4.2	Uživatelské role	28
5	Návrh	29
5.1	Uživatelské role	29
5.2	Funkce systému	29
5.3	Struktura databáze.....	30
5.4	Logika systému	33
6	Testování.....	36
6.1	Metodika	36
6.2	Výsledky	37
6.3	Zhodnocení.....	39
7	Závěr	41

1 Úvod

Podniky v dnešní době a v dnešní ekonomické situaci jsou vystavovány velkému tlaku na snižování nákladů a dosahování maximálního zisku a vysoké produktivity. Pro dosažení těchto požadavků velkou měrou přispívá informační systém. Bez informačního systému pro vedení záznamů o aktuálně probíhající činnosti podniku, o jeho zákaznících, pracovnících a dalších aspektech, o kterých je potřeba mít přehled, je práce velmi náročná. Kvalitní informační systém znamená zvýšení efektivnosti a úspěšnosti podniku při dosahování vytyčených cílů.

Tato bakalářská práce se zabývá vytvořením informačního systému v podobě webové aplikace, která klade důraz na jednoduchost a efektivitu často prováděných úkonů, které usnadní práci manažera i překladatele v překladatelském sdružení. Jedná se o informační systém pro překladatelské sdružení NK Translators, který bude vytvořen přesně dle požadavků sdružení pro zjednodušení práce se správou projektů, překladatelů a zákazníků. Možnost využití systému jinými překladatelskými společnostmi však není vyloučena.

Následující kapitola bakalářské práce popisuje momentální situaci informačních systémů pro překladatele na českém trhu, její zhodnocení a zdůvodnění rozhodnutí o vytvoření vlastního systému. Další kapitola popisuje metodologie a praktiky využívané pro návrh informačních systémů. Následuje pojednání o specifikaci systému, jeho funkcionalitě a uživatelských rolích a jeho vlastní návrh pomocí vybraných technologií. Předposlední část je zaměřena na testování funkcionality a uživatelského rozhraní systému a následné zpracování poznatků a subjektivních dojmů získaných od uživatelů na vytvořenou aplikaci. Poslední část tvoří závěr a shrnutí bakalářské práce spolu se zhodnocením dosažených výsledků a uvážením dalšího vývoje implementovaného informačního systému v budoucnu.

2 IS konkurenčních překladatelských sdružení

V dnešní době existuje mnoho informačních systémů pro překladatelské společnosti, ale buď existují drahá komplexní řešení, která jsou pro překladatelské sdružení NK Translators cenově nedostupná, jsou zbytečně rozsáhlá a složitá a obsahují funkce, které bychom nevyužili, nebo existují cenově dostupnější varianty, které jsou pro nás již přijatelnější, ale vždy jim chybí některá důležitá funkce, kterou požadujeme.

Za účelem průzkumu stávajících řešení IS bylo osloveno více než 10 překladatelských sdružení se sídlem v Brně, ovšem kladně odpověděla pouze překladatelská agentura Abiturient, která poskytla informace o svém informačním systému a o svých dosavadních zkušenostech s IS.

2.1 Agentura Abiturient

V agentuře Abiturient jsou používány 3 informační systémy pro vedení společnosti. Všechny jsou tvořeny webovými aplikacemi. První je orientován na zákazníky a vytváření jejich objednávek, druhý pro administraci objednávek a obsahu webové prezentace a třetí pro správu projektů. První dva zmíněné IS si agentura vytvořila sama a systém třetí, pro správu projektů, je zakoupená aplikace a podobá se klasickému informačnímu systému pro obchodníky, které obsahují plánovače úkolů a kalendáře a právě tento se nejvíce podobá informačnímu systému, který byl implementován v rámci této práce.

I když jsem navštívil pouze jednu agenturu a získal zkušenosti pouze od pár překladatelů, kteří používají vlastní informační systém i systém komerční, získal jsem zkušenosti velmi bohaté, protože právě tato agentura při hledání vhodného IS vybírala zhruba z 30 IS. Agentura sama hledala vhodný informační systém přesně pro jejich účely a spousta systémů, které testovala, bylo zahraničních. Přesto si nakonec vytvořili 2 vlastní IS a s jedním komerčním systémem se spokojili.

Dle agentury Abiturient je největším problémem všech IS uživatelské rozhraní a doba potřebná k provedení jednoduché akce v rámci IS – například vytvoření nového úkolu. Při vytváření úkolu bylo v systémech pokryto veškeré možné nastavení pro nový úkol, což ale pak výrazně prodloužilo dobu potřebnou k vytvoření banálních úkolů, které si prý pak člověk raději zapíše na papír. Co se týče uživatelského rozhraní, dle agentury Abiturient ve většině IS chyběla jednoduchost. Byla zde snaha zobrazit veškeré možné informace ke všem akcím či úkolům, což nakonec působilo velmi nepřehledně a na první pohled chaoticky. Dalším problémem uživatelského rozhraní bylo prý neustálé přepínání mezi obrazovkami, např. při zobrazení podrobností projektu nebo editace projektu. Uživatel při častém přechodu z jedné obrazovky na jinou začne být zmatený a musí si pamatovat, odkud a kam se v systému pohybuje, což zneprůjemňuje práci se systémem. Podle agentury Abiturient

je lepším řešením nacházet se v IS při takovýchto akcích pořád na stejné obrazovce a místo zobrazení například podrobných informací na nové obrazovce informace zobrazí například ve vyskakovací bublině u příslušného projektu. Pak uživatel přesně ví, co si prohlíží a tím získá pocit jistoty při práci s IS. To samé platí pro editaci projektu – místo zobrazení editovatelných údajů o projektu na nové obrazovce je pro uživatele pohodlnější údaje editovat přímo „na místě“ bez jakéhokoliv přepínání obrazovky.

3 Metodologie a praktiky pro návrh IS

Tato kapitola pojednává o metodologiích a praktikách, které jsou běžně využívány během návrhu a implementace informačních systémů. Neklade si však za cíl být vyčerpávajícím zdrojem informací o metodologiích pro návrh IS.

3.1 Návrh databázového schématu

Relační databáze je databáze založená na relačním modelu. Relační databáze ukládá data ve formě vztahů, které uživatel vidí jako tabulky. Jejich sloupce se nazývají atributy nebo pole a řádky tabulky se nazývají záznamy. Atributy mají určen svůj konkrétní datový typ. Řádek je řezem přes sloupce tabulky a slouží k vlastnímu uložení dat. Pojem „relační databáze“ souvisí s teorií množin. Každá konkrétní tabulka totiž realizuje podmnožinu kartézského součinu všech potenciálně možných dat všech sloupců – relaci.

Návrh databázového schématu je klíčovým a velmi důležitým úkolem při tvoření databáze. Je velmi důležité udělat správný a kompletní návrh databáze, protože mnoho problémů spojených s databázemi je spojeno se špatným návrhem databáze. Jenom dobrý návrh databáze zajišťuje spolehlivou strukturu a integritu dat [1].

3.1.1 Normalizace DB modelu

Pojem normalizace je spjat s relačním modelem. V relačním modelu jsou data uložena v tabulkách, na které jsou kladeny jisté požadavky. Při splnění požadavků je tabulka označována jako normalizovaná. Pokud nejsou tyto požadavky splněny, jsou tabulky označovány jako nenormalizované a proces jejich převodu na tabulky se označuje jako normalizace.

Normalizace je odstranění redundantních (opakujících) se dat, omezení složitosti (rozložení složité relace na dvojrozměrné tabulky) a zabránění tzv. aktualizacím anomáliím (např. abychom smazáním všech knih autora nepřišli o data o autorovi), což by mělo vést k přehlednější, rozšiřitelnější a výkonnější databázi. Postup normalizace je rozdělen do několika kroků a po dokončení každého z nich se tabulka nachází v určité normální formě [2].

1. normální forma

Relace je v první normální formě, pokud každý její atribut obsahuje jen atomické hodnoty [2].

Hodnoty jsou z pohledu databáze dále již nedělitelné.

Příklad:

Jméno	Příjmení	Adresa	Telefony
Jan	Novák	Havlíčkova 2, Praha 3	125789654;601258987;789456123
Petr	Kovář	Svatoplukova 15, Brno	369852147;357951456;963852741
Ota	Pavel	Papalášova 25, Kocourkov	546789123;123456789;987456123

Tab. 3-1 Tabulka znázorňující údaje o osobách, sloupec Telefony obsahuje vícesložkový atribut.

Sloupec Telefony v Tab. 3-1 obsahuje vícesložkový atribut – více telefonních čísel. Aby tabulka byla v 1. NF, musíme buď rozdělit atribut telefon do více atributů (pouze za předpokladu, že se množství telefonních čísel bude vždy stejné nebo nižší než počet připravených sloupečků), nebo oddělit telefonní čísla do samostatné tabulky a tyto dvě tabulky propojit pomocí ID, což je podstatně flexibilnější řešení (primární klíče v tabulkách jsou značeny podtržením):

<u>ID</u>	Jméno	Příjmení	Adresa
1	Jan	Novák	Havlíčkova 2 Praha 3
2	Petr	Kovář	Svatoplukova 15 Brno
3	Ota	Pavel	Papalášova 25 Kocourkov

Tab. 3-2 Tabulka znázorňující osoby, sloupec Telefony vyjmut.

<u>ID_osoby</u>	Telefon
1	125789654
1	601258987
1	789456123
2	369852147
2	357951456
2	963852741
3	546789123
3	123456789
3	987456123

Tab. 3-3 Tabulka s telefony osob, která je s tabulkou osob propojená přes ID.

2. normální forma

Relace se nachází v druhé normální formě, jestliže je v první normální formě a každý neklíčový atribut je plně závislý na primárním klíči, a to na celém klíči a nejen na nějaké jeho podmnožině [2].

Druhou normální formu musí být řešena pouze v případě, že máme vícehodnotový primární klíč.

Příklad:

<u>Název</u>	<u>Výrobce</u>	Telefon výrobce	Cena	Množství
Mléčná čokoláda	Milka	+420123456789	30Kč	1000
Oříšková čokoláda	Milka	+420123456789	25Kč	2500
Mléčná čokoláda	Orion	+420897654321	20Kč	1500

Tab. 3-4 Tabulka znázorňující sklad, která nesplňuje 2. NF.

Klíčem relace v Tab. 3-4 je kombinace atributů Název a Výrobce. Telefon výrobce ovšem není závislý na celém klíči, ale pouze na atributu výrobce. To by vedlo k aktualizací anomálii a to k takové, že pokud by se vymazaly veškeré výrobky od výrobce Milka, smazalo by se i telefonní číslo na výrobce Milka, což není žádané. Řešením je opět rozpad na dvě tabulky:

<u>Název</u>	<u>ID_výrobce</u>	Cena	Množství
Mléčná čokoláda	1	30Kč	1000
Oříšková čokoláda	1	25Kč	2500
Mléčná čokoláda	2	20Kč	1500

Tab. 3-5 Tabulka skladu bez telefonu a jména výrobce.

<u>ID_výrobce</u>	Výrobce	Telefon
1	Milka	420123456789
2	Orion	420897654321

Tab. 3-6 Tabulka s odděleným jménem výrobce a jeho telefonním číslem.

3. normální forma

V této formě se nachází tabulka, splňuje-li předchozí dvě formy a žádný z jejich atributů není tranzitivně závislý na klíči [2].

Tranzitivní závislost je závislost mezi minimálně dvěma atributy a klíčem, kde jeden atribut je funkčně závislý na klíči a druhý atribut je funkčně závislý na prvním.

Příklad:

<u>ID</u>	Jméno	Příjmení	Město	PSČ	Funkce	Plat
1	Jan	Novák	Plzeň	73748	Ředitel	40000
2	Petr	Opařil	Polička	12345	Manažer	25000
3	Pavel	Sedláček	Brno	43432	Skladník	15000
4	Vladimír	Racek	Svitavy	10000	Manažer	25000

Tab. 3-7 Tabulka znázorňující zaměstnance, která nespĺňuje 3. NF.

Kromě závislosti všech atributů na klíči je v tabulce Tab. 3-7 ještě vidět závislost Platu na Funkci. Závislost $ID \rightarrow Funkce \rightarrow Plat$ je tranzitivní závislost Platu na klíči. Řešením problému je opět rozpad na více relací:

<u>ID</u>	Jméno	Příjmení	Město	PSČ	ID_funkce
1	Jan	Novák	Plzeň	73748	1
2	Petr	Opařil	Polička	12345	2
3	Pavel	Sedláček	Brno	43432	3
4	Vladimír	Racek	Svitavy	10000	1

Tab. 3-8 Tabulka, která splňuje 3. NF.

<u>ID_funkce</u>	Funkce	Plat
1	Ředitel	40000
2	Manažer	25000
3	Skladník	15000

Tab. 3-9 Tabulka, která doplňuje údaje předchozí tabulky.

Boyce-Coddova normální forma (BCNF)

Relace se nachází v BCNF, jestliže pro každou netriviální závislost $X \rightarrow Y$ platí, že X je nadmnožinou nějakého klíče schématu R [2].

Boyce-Coddovu normální formu a můžeme zjednodušeně označit jako "variaci" třetí normální formy. Boyce-Coddova normální forma je významná proto, že předepisuje chování funkčních závislostí jednak mezi klíčovými atributy, a pak také mezi klíčovými a neklíčovými atributy. Aby byla porušena BCNF musí být splněno několik podmínek a to poměrně specifických:

1. Relace musí mít více kandidátních klíčů.
2. Minimálně 2 kandidátní klíče musí být složené z více atributů.
3. Některé složené kandidátní klíče musí mít společný atribut.

Boyce-Coddova normální forma v podstatě říká, že mezi kandidátními klíči nesmí být žádná funkční závislost.

Příklad:

Město	Ulice	PSČ
Praha 10	Černokostelecká	10000
Jihlava	Žižkova	58601
Praha 10	Vrátkovská	10000
Brno	Dvořákova	58974
Praha 6	Chaloupeckého	16000

Tab. 3-10 Tabulka, která obsahuje 2 kandidátní klíče složené z více atributů.

V Tab. 3-10 platí dvě netriviální funkční závislosti: $\{\text{Město}, \text{Ulice}\} \rightarrow \text{PSČ}$ a $\text{PSČ} \rightarrow \text{Město}$. Protože neplatí $\text{Ulice} \rightarrow \text{PSČ}$ ani $\text{Město} \rightarrow \text{PSČ}$, tvoří dvojice $\{\text{Město}, \text{Ulice}\}$ klíč schématu. Klíčem je ale i $\{\text{Ulice}, \text{PSČ}\}$ platí totiž $\text{PSČ} \rightarrow \text{Město}$, nikoliv však $\text{PSČ} \rightarrow \text{Ulice}$. Tudíž je $\{\text{PSČ}, \text{Ulice}\}$ kandidátním klíčem schématu. Schéma má všechny atributy atomické a nemá žádný neklíčový atribut, tudíž je v 3. NF, ale není v BCNF. Tento fakt vede k tomu, že nelze evidovat města s PSČ bez znalosti Ulice a kromě toho jsou v relaci redundantní data, pokud by se evidovalo velké množství ulic v jednom městě, začal by to být problém.

Klasické řešení je rozpad na dvě tabulky. Vzhledem k tomu, že neplatí $\text{PSČ} \rightarrow \text{Ulice}$, musíme spojit PSČ a Ulice. Výsledkem tudíž budou relace Města (PSČ, Město) a Ulice (PSČ, Ulice):

Město	<u>PSČ</u>
Praha 10	10000
Jihlava	58601
Brno	58974
Praha 6	16000

Tab. 3-11 Tabulka s atributy Město a PSČ.

<u>Ulice</u>	<u>PSČ</u>
Černokostelecká	10000
Žižkova	58601
Vrátkovská	10000
Dvořákova	58974
Chaloupeckého	16000

Tab. 3-12 Tabulka s atributy Ulice a PSČ.

4. normální forma

Tabulka je ve čtvrté normální formě, je-li v BCNF a popisuje pouze příčinnou souvislost (jeden fakt) [2].

Čtvrtá normální forma se zabývá vztahy uvnitř složeného primárního klíče. Pokud je v tabulce složený primární klíč, může se stát, že některé hodnoty tohoto klíče jsou na sobě nezávislé, ale tím, že spolu tvoří klíč, vzniká falešná souvislost mezi těmito hodnotami a nemohou existovat nezávisle na sobě, což není v souladu s modelovanou realitou. 4. NF proto vyžaduje, aby klíč tvořily jen ty hodnoty, které mají vzájemnou souvislost.

Příklad:

<u>Zaměstnanec</u>	<u>Úkol</u>	<u>Kvalifikace</u>
Ing. Petr Pastyňák	Tvorba webu	Webdeveloper
Ing. Petr Pastyňák	Návrh databáze podnikového IS	Database Specialist
Eva Petrželová	Asistentka Ing. Pastyňáka	Psaní na stroji
Eva Petrželová	Asistentka Ing. Pastyňáka	ECDL
Pavel Mrkvička	Analytik podnikového IS	Aanalyst
Pavel Mrkvička	Analytik podnikového IS	UML

Tab. 3-13 Zaměstnanec a jeho úkol a kvalifikace.

Všechny atributy v Tab. 3-13 dohromady tvoří klíč schématu a neexistuje mezi nimi žádná funkční závislost, tudíž je v BCNF a všechno vypadá ideálně, ale není tomu tak. I když se dá předpokládat, že atributy Kvalifikace a Úkol jsou na sobě nezávislé, tak tabulka neumožňuje zachytit kvalifikaci zaměstnance, který nemá přiřazen žádný úkol a nelze ani úkolovat zaměstnance bez kvalifikace. Krom ztráty informací se rozkladem vyvarujeme i redundance dat. Tudíž je opět nutno tabulku rozdělit a to následovně:

<u>Zaměstnanec</u>	<u>Kvalifikace</u>
Ing. Petr Pastyňák	Webdeveloper
Ing. Petr Pastyňák	Database Specialist
Eva Petrželová	Psaní na stroji
Eva Petrželová	ECDL
Pavel Mrkvička	Aanalyst
Pavel Mrkvička	UML

Tab. 3-14 Rozpad předchozí tabulky na atributy Zaměstnanec a Kvalifikace.

<u>Zaměstnanec</u>	<u>Úkol</u>
Ing. Petr Pastyňák	Tvorba webu
Ing. Petr Pastyňák	Návrh databáze podnikového IS
Eva Petrželová	Asistentka Ing. Pastyňáka
Pavel Mrkvička	Analytik podnikového IS

Tab. 3-15 Rozpad předchozí tabulky na atributy Zaměstnanec a Úkol.

5. normální forma

Relace je v páté normální formě, pokud je ve čtvrté a není možné do ní přidat další atribut (skupinu atributů) tak, aby se vlivem skrytých závislostí rozpadla na několik dílčích relací [2].

Pátá normální forma se týká primárních klíčů, které jsou tvořeny nejméně třemi atributy. V případě, že mezi těmito hodnotami v klíči existují párové cyklické závislosti, tak je třeba tyto závislosti extrahovat do samostatných tabulek, ale původní tabulku je v některých případech třeba zachovat.

Příklad:

<u>Zástupce</u>	<u>Firma</u>	<u>Stát</u>
Antonín Bahel	Siemens	Německo
Antonín Bahel	Siemens	Rakousko
Ctírad Drba	Siemens	Francie
Ctírad Drba	Škoda Plzeň	Rakousko
Antonín Bahel	Škoda Plzeň	Norsko

Tab. 3-16 Obchodní zastoupení v Evropě.

V Tab. 3-16 je pátá normální forma porušena, neboť zde existují závislosti Zástupce -> Firma -> Stát -> Zástupce a to jsou párové cyklické závislosti. Mohlo by se stát, že s vymazáním obchodního zástupce, by se mohlo ztratit informace o tom, že firma prodává v zemi, kde jí zastupoval pouze ten smazaný zástupce, což je nežádoucí. Stejně tak odebrání firmy může způsobit ztrátu informace o působení obchodního zástupce v některé zemi. Takže musíme provést rozpad na tři relace, které nám pokryjí všechny vztahy:

<u>Zástupce</u>	<u>Stát</u>
Antonín Bahel	Německo
Antonín Bahel	Rakousko
Ctírad Drba	Francie
Ctírad Drba	Rakousko
Antonín Bahel	Norsko

Tab. 3-17 Rozpad tabulky Tab. 3-16.

<u>Zástupce</u>	<u>Firma</u>
Antonín Bahel	Siemens
Ctírad Drba	Siemens
Ctírad Drba	Škoda Plzeň
Antonín Bahel	Škoda Plzeň

Tab. 3-18 Rozpad tabulky Tab. 3-16.

<u>Firma</u>	<u>Stát</u>
Siemens	Německo
Siemens	Rakousko
Siemens	Francie
Škoda Plzeň	Rakousko
Škoda Plzeň	Norsko

Tab. 3-19 Rozpad tabulky Tab. 3-16.

V praxi se většinou normalizuje do třetí normální formy, vyšší normální formy je vcelku obtížné porušit a vyžadují relativně velké znalosti návrháře.

3.2 UML

Zkratka UML představuje souloví Unified Modeling Language. UML je v softwarovém inženýrství grafický jazyk pro vizualizaci, specifikaci, navrhování a dokumentaci programových systémů. UML nabízí standardní způsob zápisu návrhů systému včetně konceptuálních prvků, jako jsou business procesy a systémové funkce. Dále pak nabízí zápis konkrétních prvků, jako jsou příkazy programovacího jazyka, databázová schémata a znovupoužitelné programové komponenty.

UML podporuje objektově orientovaný přístup k analýze, návrhu a popisu programových systémů a nepopisuje způsob, jak se má používat, ani neobsahuje metodiky, jak analyzovat, specifikovat či navrhovat programové systémy.

UML se dá využít pro kreslení konceptu systému, kde vývojář zaznamená své myšlenky návrhu před implementací. Dále pro kreslení detailních návrhů, kde vývojář zachytí kompletní návrh či kompletní realizaci tak, aby byl programátor schopen vytvořit program bez velkého přemýšlení nad věcnou oblastí a nevznikala by potřeba konzultace. UML lze využít také jako programovací jazyk, kdy vývojář nakreslí UML diagramy, ze kterých se vygeneruje spustitelný kód. To však vyžaduje specializované nástroje a velmi přesné vyjadřování v UML diagramech [3][4][5].

Následuje popis vybraných diagramů UML, které jsou relevantní vzhledem k zaměření této práce.

3.2.1 Diagram případů užití

Diagram případů užití popisuje vazby mezi aktéry (Actor) a případy užití systému (Use Case). Diagram případů užití vymezuje hranice systému specifikací „funkční obálky“ systému, prostřednictvím níž systém komunikuje se svým okolím. Okolí modelovaného systému je popsáno pomocí aktérů. Detailní popis komunikace mezi aktéry a systémem není předmětem tohoto diagramu. Z vnějšího pohledu, tedy z pohledu aktérů, představuje diagram případů užití úplný popis funkčnosti systému [5].

Elementy diagramu případů užití

- **Případ užití (Use Case)**

Případ užití popisuje funkce, které systém poskytuje k užítku jednoho nebo více aktérů a je jednoznačně identifikován svým názvem. Případy užití jsou vždy iniciovány aktérem a jsou vždy popsány z pohledu aktéra. Důležitou součástí případu užití je popis funkčnosti, nejčastěji ve formě připojeného textu. Případ užití je v diagramu značen elipsou[5].

- **Aktér (Actor)**

Aktér vyjadřuje prvek okolí systému, který se systémem komunikuje, přijímá nebo poskytuje systému informace. Aktér je abstrakcí čehokoliv (uživatelé, skupiny uživatelů, spolupracující systémy, subsystémy, ...), co je považováno za externí entitu z pohledu modelovaného systému. Aktér bývá v diagramu případů užití označen symbolem osoby [5].

- **Relace generalizace (Generalization)**

Generalizace je vztah umožňující vyčlenit chování společné dvěma nebo více aktérům do nového rodičovského (nadřazeného) aktéra. Potomci rodičovského aktéra dědí od rodičovského aktéra veškeré případy užití a ty případy užití, které mají odlišné, jsou specifikovány přímo u potomků. To stejné platí pro případy užití. Vztah generalizace je značena orientovanou plnou čarou, kde orientace je vyjádřena šipkou ve tvaru trojúhelníka ve směru generalizace [5].

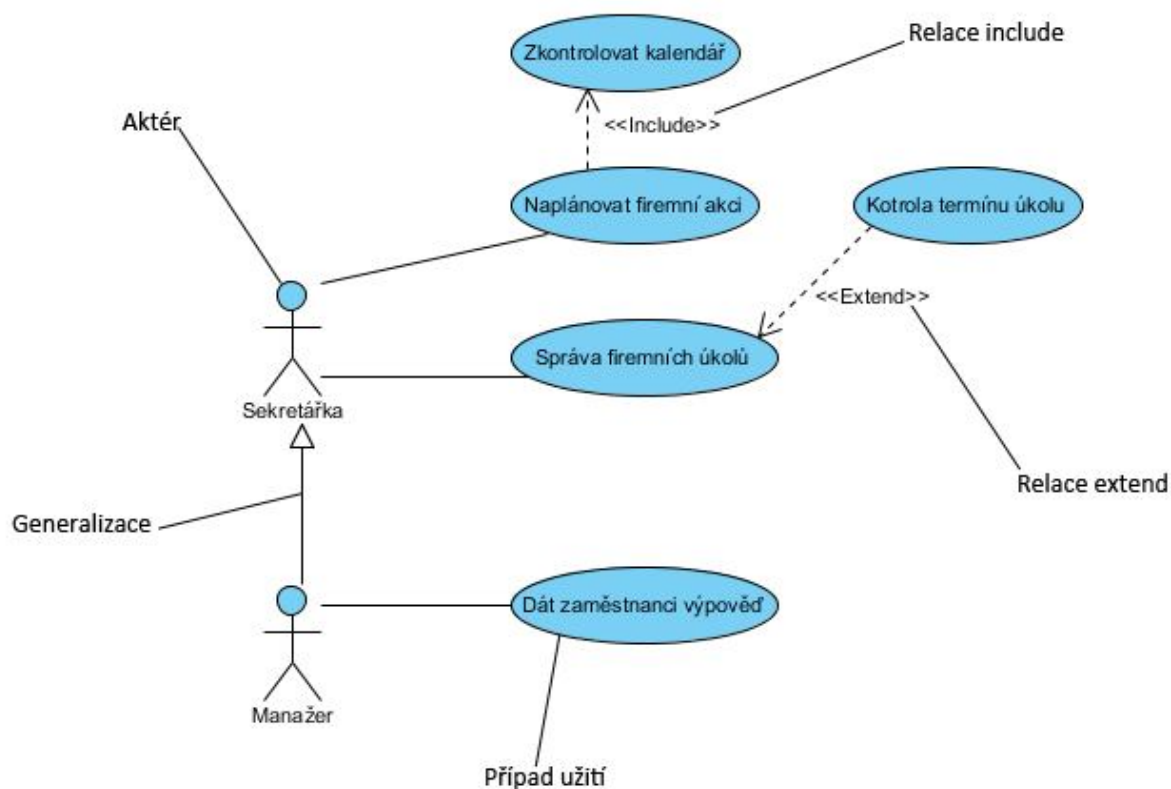
- **Relace include (Include Relationship)**

Relace include vyčleňuje kroky společné pro několik případů užití do samostatného případu užití, který je do příslušných případů užití následně zahrnut. Relace include je značena orientovanou přerušovanou čarou se stereotypem «include», kde orientace je vyjádřena šipkou k zahrnovanému případu užití [3].

- **Relace extend (Extend Relationship)**

Relace extend poskytuje způsob, jímž lze daný případ užití rozšířit o nové chování. Na rozdíl od konstrukce s relací include musí být rozšiřovaný případ užití smysluplný a proveditelný samostatně. Relace extend je značena orientovanou přerušovanou čarou se stereotypem «extend», kde orientace je vyjádřena šipkou k rozšiřovanému případu užití [3].

Na Obr. 3.1 je možno vidět zmíněné elementy diagramu případu užití spolu s relacemi mezi nimi.



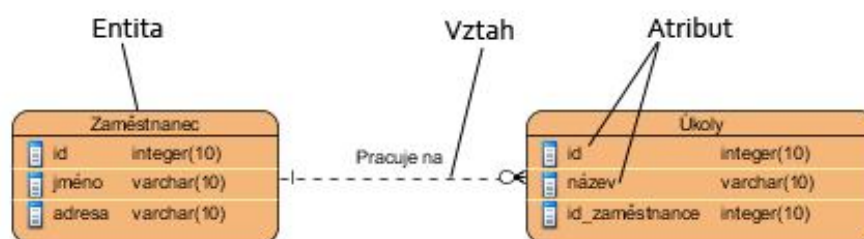
Obr. 3.1 Elementy diagramu případu užití a relace mezi nimi

3.2.2 ER diagram

ER diagram neboli entity-relationship diagram se v softwarovém inženýrství používá pro abstraktní a konceptuální znázornění dat. Je to grafická reprezentace entit, vztahů a omezení, které tvoří daný návrh. Stejně jako u jiných grafických návrhových metodologií, poskytuje grafický souhrn návrhu, který je velmi užitečný pro návrháře, nejen pro validaci správnosti návrhu, ale i pro diskuzi s kolegy a komunikaci s programátory, kterým je třeba problém vysvětlit. Bohužel neexistují standardní obecně platné konvence pro nákreš ER diagramu, a proto mohou existovat rozdíly a různé typy ER diagramů. Jeho základem je převedení komplexních struktur modelované skutečnosti do dvojrozměrných tabulek a nalezení vztahů mezi nimi. Efektivní datový model přesně a úplně popisuje a vyhovuje nárokům zadání a je použitelný pro tvorbu databáze, eliminuje redundanci dat a je nezávislý na hardwaru a softwaru. Právě lehký převod na tabulky ho dělá vhodným pro návrh relačních databází [4]. Pro návrháře to je velmi užitečné hlavně pro validaci správnosti návrhu. ER diagram obsahuje entity, atributy a vztahy:

- **Entita (Entity)** – objekt reálného světa.
- **Atribut (Attribute)** – vlastnost entity, která nás zajímá.
- **Vztah (Relationship)** – asociace mezi několika entitami.

Názorný příklad zmíněných součástí ER diagramu je možno vidět na Obr. 3.1.



Obr. 3.1 – Základní prvky ER diagramu.

ER diagram reprezentuje logické vztahy mezi entitami nebo objekty. Má nezastupitelnou úlohu nejen při návrhu databázových aplikací, ale také při jejich optimalizaci a odstraňování chyb. Při vytvoření ER diagramu je třeba rozhodnout, které entity, vztahy a omezení použijeme pro modelování fyzického systému. Speciální druhy vztahů představují agregace, kompozice, generalizace a specializace:

- **Agregace (Aggregation)** - je druh vztahu, kdy jedna třída je součástí třídy jiné.
- **Kompozice (Composition)** - je silnější druh agregace. Při kompozici je část přímo závislá na svém celku, zaniká smazáním celku a nemůže být součástí více jak jednoho celku
- **Generalizace (Generalization)** - druh vztahu, kdy jedna třída je všeobecným zavedením vlastností jiné třídy (jiných tříd). Kombinuje se několik množin entit, které sdílejí stejné rysy do množiny entit vyšší úrovně.
- **Specializace (Specialization)** - opak generalizace, tvoří se podskupiny v množině entit, které jsou různé od ostatních entit v množině. Tyto podskupiny se stávají množinami entit nižší úrovně, které mají atributy nebo jsou součástí množin vztahů, které se nepromítají do množiny vztahů vyšší úrovně.

3.2.3 Diagram tříd

Diagram tříd zobrazuje statickou strukturu systému. Primárně je určen k popisu tříd a jejich vztahů, avšak v diagramu tříd mohou být zobrazeny i rozhraní, složky, objekty a související vzájemné vazby. Výstižnější by byl proto název diagram statické struktury, avšak užívání termínu diagram tříd se již v tomto kontextu ustálilo. Je vytvářen v etapě analýzy a postupně je zpřesňován, je základem pro implementaci a nástrojem pro dokumentaci [5].

Elementy diagramu tříd

Třída (Class)

Třída je popisem množiny objektů sdílejících stejné atributy, metody, vztahy a sémantiku, je tedy elementem modelu s vyšší mírou abstrakce než element modelu objekt. Třída je jednoznačně určena svým názvem (v příslušném názvovém prostoru – složce). Pro třídu je možno definovat atributy a operace, jejichž názvy musí být jednoznačné v rámci třídy. Operace, které třída definuje, představují její chování nebo také zprávy, kterým třída rozumí. Atributy třídy jsou informace, které jsou o třídě uchovávané. Měly by být atomické a nedělitelné. Třída je značena obdélníkem při potlačení zobrazení atributů a operací nebo obdélníkem rozděleným na tři části pro zobrazení atributů a operací.

Vztahy mezi třídami

Asociace

Asociace znázorňuje vztahy mezi jednou nebo více třídami, které jsou abstrakcí množiny spojení mezi instancemi (objekty) těchto tříd. Jinak řečeno, asociace mezi třídami říká, že třídy mají mezi sebou „nějaký přímý vztah“. U asociací předpokládáme, že jsou obousměrné, pokud nejsou explicitně specifikovány jako jednosměrné. Jméno asociace implikuje směr a mělo by být v aktivní (slovesné) formě. Jako příklad je uvedena asociace mezi objektovou třídou *Autobus* a třídou *Seznam cestujících* na Obr. 3.2.

Agregace

Vazba typu agregace říká, že jedna třída je částí druhé třídy, např. *student* je součástí studijního kroužku, takže můžeme určit vazbu mezi třídou *Student* a třídou *Studijní kroužek* jako vazbu typu agregace. Agregace je specifitější vztah než výše popsaná asociace a volnější vztah než níže popisovaná kompozice. Zánik celku nemusí zapříčinit zánik obsahované komponenty – při zániku studijního kroužku bude *student* i nadále samostatně existovat. Agregace může být popsána stejnými prvky jako asociace, je však více specifická, než samotná asociace a může spojovat pouze dva prvky mezi sebou. Zápis se provádí opět nepřerušovanou linií a u agregovaného prvku zakončenou nevyplněným kosočtvercem.

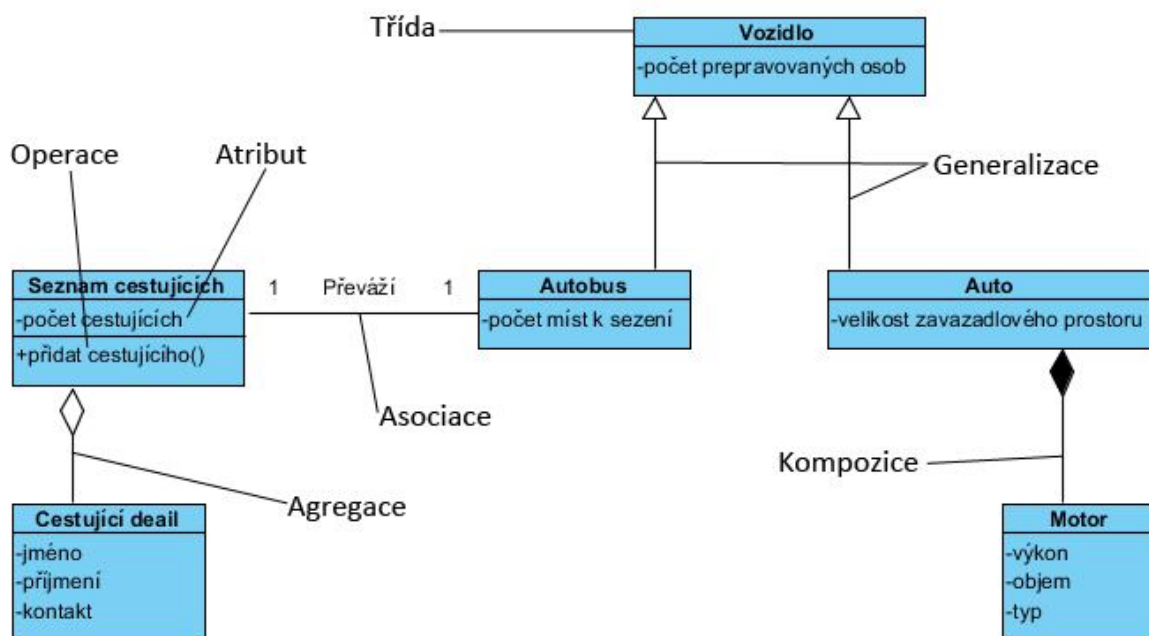
Kompozice

Kompozice představuje silnější vazbu, než agregace. Grafický zápis se provádí stejně jako u agregace s tím rozdílem, že kosočtverec je vyplněný. Prvky svázané kompozicí bez sebe nemohou existovat, nebo nemají žádný smysl. Je-li tedy odstraněn jeden prvek, je odstraněn i druhý. Tvoří tedy spolu kompozici v systému. Příklad je na Obr. 3.2, na kterém můžeme vidět třídu *Auto*, a dále pak třídu *Motor*, která specifikuje typ motoru v autě. V tomto případě třída *Motor* samostatně opodstatněnou existenci, což znamená, že s autem tvoří kompozici, nebo také silnou agregaci.

Generalizace / specializace

Generalizace je klíčový koncept objektivě orientované analýzy a návrhu. Pokud nacházíme v analýze tento typ vztahu, implementujeme ho v objektivě prostředí jako dědičnost. Princip dědění umožňuje objektivě třídám sdílet jejich charakteristiky včetně hierarchie dědění. Generalizace se používá především z důvodu opakovaného použití, aby se nemusely opakovat společné prvky. Příkladem může být například vozidlo, které má vlastnost počet přepravovaných osob a generalizuje osobní automobil s jeho vlastností velikost zavazadlového prostoru a například autobus s vlastností počet míst k sezení, viz Obr. 3.2. V UML se jedná o takzvaný supertyp (vozidlo) a subtyp (autobus, osobní automobil). Zápis je shodný s diagramem případu užití, tedy plnou čarou zakončenou nevyplněným trojúhelníkem. Specializace je opačný vztah.

Na Obr. 3.2 je možno přehledně vidět výše popsané elementy diagramu tříd a zmíněné vazby mezi nimi.



Obr. 3.2 Prvky diagramu tříd a vztahy mezi nimi

3.3 Architektura Model-View-Controller

Model-View-Controller (MVC) je velmi oblíbený způsob návrhu aplikace. Své kořeny má ve Smalltalku¹, kde se původně používal pro zakomponování tradičního postupu vstup–zpracování–výstup do programů s GUI². Neméně vhodný je ale i pro použití ve vícevrstvých webových aplikacích. Rozšířil se zejména při návrhu systému v jazyce Java a souvisejících technologiích. Jeho obecná definice jej ale umožňuje používat i v jakémkoliv jiném programovacím jazyce. Jedná se o softwarovou architekturu, která rozděluje datový model aplikace, uživatelské rozhraní a řídicí logiku do tří nezávislých komponent tak, že modifikace některé z nich má minimální vliv na ostatní. MVC je často chápán jako návrhový vzor, nicméně se týká architektury aplikací mnohem více než klasický návrhový vzor. Jeho základní myšlenky jsou velmi jednoduché, ale na druhou stranu není příliš konkrétní a tak jej lze snadno použít nesprávným způsobem. Existuje mnoho variant tohoto návrhového vzoru, které se zpravidla liší jen v některém paradigmatu [6][7][8].

Model (model)

Model je funkčním a datovým základem celé aplikace. Poskytuje prostředky jak pro přístup k datové základně a stavům aplikace, tak pro jejich ukládání a aktualizaci. Hovoří se o něm jako o softwarovém obrazu procesů reálného světa. Měl by být jako celek zapouzdřený a pro View (pohled) a Controller (řadič) nabízet přesně definované rozhraní. Nejčastěji je implementován pomocí objektových tříd, lze jej však (v přístupech odlišných od OOP) realizovat například i běžnými funkcemi.

View (pohled)

Slouží k vytvoření grafického rozhraní (GUI) pro interakci s uživatelem. Může se jednat například o webového klienta nebo o formuláře v aplikaci. Účelem této vrstvy je oddělení prezentační logiky od věcné a datové, což zvyšuje její udržitelnost a umožňuje jednodušší zapracování změn, které se objevují ve větší míře právě v prezentační vrstvě.

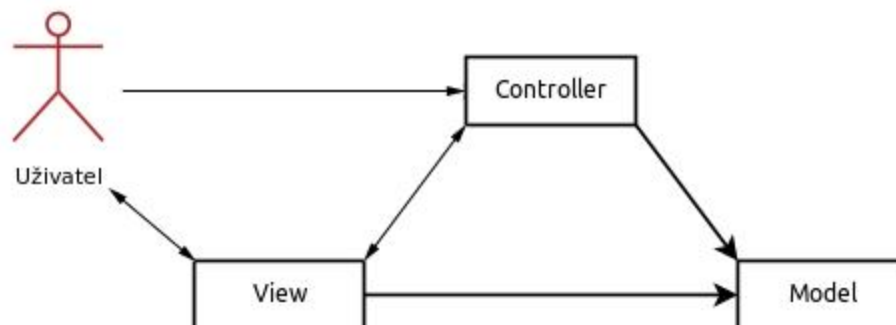
Controller (řadič)

Je vrstva mezi prezentační vrstvou (pohledem) a aplikační vrstvou (modelem). Reaguje na události pocházející od uživatele a zajišťuje změny v modelu nebo v pohledu. Řadič je jakousi ústřední výkonnou jednotkou, která se stará o celkové provázání funkčnosti aplikace. Definice řadiče není nijak striktní, což jeho implementaci poskytuje poměrně velký prostor pro jeho upřesnění. Právě pojetím řadiče se jednotlivé vzory MVC od sebe nejvíce liší.

¹ Smalltalk - interpretovaný, dynamicky a striktně typovaný, čistě objektový programovací jazyk [11]

² GUI – z anglického Graphical User Interface, je to grafické uživatelské rozhraní, které umožňuje ovládat počítač pomocí interaktivních grafických ovládacích prvků

Obr. 3.2 ilustruje návaznost jednotlivých komponent architektury MVC.



Obr. 3.2 – Návaznost komponent Model, View a Controller v architektuře MVC.

Ačkoliv může být koncept MVC realizován různým způsobem, obecně platí tento princip [7]:

1. Uživatel provede nějakou akci v uživatelském rozhraní (např. stiskne tlačítko).
2. Řadič obdrží oznámení o této akci z objektu uživatelského rozhraní.
3. Řadič přistoupí k modelu a v případě potřeby ho zaktualizuje na základě provedené uživatelské akce.
4. Doménová logika zpracuje změněná data. Některé aplikace užívají mechanismus pro perzistentní uložení dat (např. databázi). To je však otázka vztahu mezi doménovou a datovou vrstvou, která není architekturou MVC pokryta.
5. Komponenta pohled použije zaktualizovaný model pro zobrazení zaktualizovaných dat uživatelem. Komponenta pohled získává data přímo z modelu, zatímco model nepotřebuje žádné informace o komponentě pohled (je na ní nezávislý). Nicméně je možné použít návrhový vzor pozorovatel, umožňující modelu informovat jakoukoliv komponentu o případných změnách dat. V tom případě se komponenta pohled zaregistruje u modelu jako příjemce těchto informací. Je důležité podotknout, že řadič nepředává doménové objekty (model) komponentě pohledu, nicméně jí může poslat příkaz, aby svůj obsah podle modelu zaktualizovala.
6. Samotnému konečnému zobrazení výsledku uživateli ještě může u webových aplikací předcházet odpověď ze serveru na klienta, aby si ihned vyžádal obnovení stránky. Tím je zaručeno, že při obnovení stránky uživatelem nevyvolá na serveru požadovanou akci opakovaně, ale že se jedná pouze o obnovení pohledu, nyní už bez požadavku na změnu dat (modelu).
7. Uživatelské rozhraní čeká na další akci uživatele, která celý cyklus zahájí znovu.

V praxi se ukázalo, že jde o velmi užitečný přístup. Vývojáři si ověřili, že tato separace je nezbytná pro udržení přehledného kódu – obzvláště v případech, kdy na jedné aplikaci pracuje více lidí.

Výhody třívrstvé aplikace jsou patrné hlavně z dlouhodobého hlediska. Oddělení dat od jejich prezentace umožňuje vyšší úroveň zabezpečení, kdy koncoví uživatelé jsou zcela oddělení od datové vrstvy. Současně se také zvyšuje flexibilita, je možné používat servery pro rozložení zátěže, záložní servery, distribuovaná úložiště apod. Výhodné je i psaní aplikace, jelikož umožňuje týmový vývoj a jednotlivé části mohou být vyvíjeny samostatně.

Hlavní výhody MVC:

- Každou vrstvu můžeme lehce nahradit.
- Znovupoužitelnost kódu.
- Jednoduchá rozšiřitelnost.
- Minimalizace duplicitního kódu.

Nevýhody MVC:

- Složitější implementace.

4 Analýza a specifikace požadavků

V současné době na českém trhu existuje jediný informační systém pro překladatelské činnosti. Jedná se o informační systém jménem Quahill. Z malého průzkumu formou konzultací se správci informačních systému v konkurenčních překladatelských společnostech bylo zjištěno, že překladatelská sdružení a překladatelské společnosti buď informační systém nepoužívají vůbec (jedná se spíše o soukromíky nebo malá překladatelská sdružení), nebo několik málo společností používá informační systém Quahill. Nejčastěji mají však překladatelské společnosti svá vlastní řešení informačního systému, který je šitý na míru a mají své vlastní správce systému. To, že si většina společností nechá vyrobit informační systém na míru, vypovídá o tom, že vyhovující informační systém na trhu chybí. Informační systémy pro překladatele sice existují, ale jsou buď nedostačující svojí funkcí a chybí jim důležité prvky potřebné pro práci překladatele nebo jsou to naopak velmi rozsáhlé informační systémy, které obsahují funkce, které nejsou ani potřeba a práce s nimi je velmi složitá, zdlouhavá a vyžaduje zaškolení. To se velmi projeví na finanční stránce věci.

Komerční informační systémy by se daly klasifikovat do tří skupin:

- **IS pro velké firmy** – tyto IS jsou rozsáhlá a složitá řešení, která požadují zaškolení uživatelů, což je časově a finančně náročné. Samotné systémy jsou samozřejmě velmi drahá řešení.
- **Středně velká řešení** – IS, které jsou zaměřeny na obchodníky. Obsahují plánovače akcí, kalendář atd. Často již nepotřebují zaškolení, protože nejsou zdaleka tak rozsáhlé a také nejsou tak drahé. Pro NK Translators ovšem také nevhodné, protože chybí funkce pro správu terminologie překladů, o které je pojednáno v odstavci „Správa terminologie“ v podkapitole 4.1 Uživatelské akce.
- **Malé IS** – systémy, které obsahují malé množství funkcí, mají jednoduché intuitivní ovládání, jsou relativně levné.

Hlavním důvodem zvolení tématu informačního systému pro překladatelské sdružení je vytvoření přehledného a intuitivně ovladatelného překladatelského systému, který by mohly využít i jiné překladatelské společnosti.

Protože jsem členem překladatelského sdružení NK Translators, které začíná svojí působnost rozšiřovat a práce bez přehledného informačního systému se stává čím dál náročnější a nepřehlednější, rozhodl jsem se vytvořit informační systém pro toto sdružení, který bude odpovídat přesně jeho požadavkům.

Cílem mé práce je vytvořit přehledný, uživatelsky intuitivní systém, který bude usnadňovat práci v překladatelském sdružení NK Translators. Cílem je rovněž vytvořit systém, který nebude

obsahovat žádné zbytečné funkce navíc, ale pouze ty, které jsou pro řízení sdružení nezbytné. Díky tomu by se nemělo jednat o příliš rozsáhlou aplikaci, na kterou by jinak bylo třeba školení. Tomu by také mělo pomoci intuitivní ovládání, díky kterému by nemělo být třeba zaškolení uživatele.

Za dvouletou překladatelskou činnost jsem získal dost zkušeností s prací překladatele, abych byl schopen definovat, které vlastnosti a funkce bude systém muset obsahovat, aby co nejvíce usnadnil práci při vedení malého překladatelského sdružení. Dobře se orientuji v dané problematice, takže po vlastní analýze požadavků jsem svůj návrh předvedl kolegům a společně jsme systémové požadavky doplnili o jejich nápady a návrhy jednotlivých funkcí dopracovali do detailů na základě jejich dosavadních zkušeností s překladatelskou činností.

Navrhovaný systém, jakožto systém pro vedení malého překladatelského sdružení, bude poskytovat funkce jako např. správu projektů sdružení, správu externích překladatelů a zákazníků. Všechny funkce systému, které vzešly z analýzy požadavků, jsou podrobně popsány v následující kapitole.

4.1 Uživatelské akce

Na základě činností, které pracovníci sdružení provádí, vyplynuly požadavky na následující funkce:

Správa projektů

Pojem „správa“ značí čtyři základní operace používané při práci s daty - CRUD:

- C = Create – vytvoření nového záznamu.
- R = Read (někdy uváděné jako Retrieve) – čtení (zobrazení) záznamu.
- U = Update – úprava existujícího záznamu.
- D = Delete (někdy uváděné jako Destroy) – smazání (zničení) záznamu.

Společnost bude mít přehled nad všemi svými právě probíhajícími projekty. „Projekt“ zde znamená překlad textu, textovou korekturu nebo tlumočení. Uživatel tedy bude moci v systému vytvářet, prohlížet, editovat a rušit projekty (zakázky).

Správa zákazníka

Společnost bude vést záznamy o všech svých dosavadních zákaznících. Záznamy budou obsahovat osobní informace a kontaktní údaje zákazníka. Jedná se opět o 4 základní operace CRUD, tentokrát aplikované na zákazníka sdružení, vyjma operace Delete – zákazníka nebude možno smazat.

Správa externího překladatele

Budou vedeny také záznamy se všemi externími překladateli společnosti, které budou obsahovat osobní a kontaktní údaje. Jsou to operace CRUD aplikované na externího překladatele, opět vyjma operace Delete. Zde bude možno překladatele deaktivovat, pokud dojde k ukončení spolupráce z jakéhokoliv důvodu. Deaktivace znamená pouze grafické znázornění, že překladatel ve sdružení není aktivní a jeho účet nebude smazán. Deaktivovaného překladatele lze kdykoliv opět aktivovat.

Správa terminologie

Systém bude nabízet správu terminologie, což je důležitý prvek tohoto informačního systému. Je to asi hlavní rozdíl, který jej odlišuje od běžných IS. Často totiž nastane situace, kdy překlad nepřekládá pouze jeden překladatel. Aby sdružení bylo schopno dodat rozsáhlý překlad v termínu přijatelném pro zákazníka, musí jej často rozdělit mezi více překladatelů. Po spojení částí od jednotlivých překladatelů musí být použita terminologie v jednotlivých částech překladu maximálně shodná, aby výsledný překlad dával smysl a tvořil jednotný celek. Protože překladatelé našeho sdružení nepracují ve společných prostorách, ale svojí překladatelskou činnost vykonávají doma, nekonzultují často překlady osobně v průběhu překladu. Důkladná konzultace proběhne po dokončení překladu při jeho sjednocení. Aby sjednocení překladu bylo usnadněno a byl zkrácen čas při sjednocení terminologie, bude informační systém nabízet funkci přidávání terminologie ke každému překladu. Do této sekce systému bude překladatel při překladu přidávat důležité fráze, slova nebo výrazy, které se v překládaném textu vyskytují často. Ke každému vloženému výrazu navíc překladatel uvede značku signalizující jeho jistotu s daným překladem výrazu. Značky budou vyjadřovat pomyslnou stupnici, která říká buď „překlad výrazu je na 100% správný“ nebo na druhé straně „nevím, potřebuji poradit“. Stupnice bude obsahovat tři stupně. Překladatelé, kteří se na překladu podílí, mají k terminologii překladu přístup a můžou jednotlivé výrazy upravovat, pokud s nimi nesouhlasí a jsou si jistí s jiným překladem nebo mohou doplnit výraz, se kterým si jiný překladatel neví rady. Dále sekce terminologie nabízí funkci našeptávače, který bude nabízet všechny překlady výrazu, který již byl v minulosti přeložen (a to v kterémkoliv překladu). Pokud je text překládán do více jazyků najednou, bude terminologie překladu rozdělena podle cílových jazyků, aby se zabránilo smíchání dvou jazyků dohromady. Jak z popisu vyplývá, tato sekce bude použita pouze pro projekty typu překlad, pro textové korektury nebo tlumočení není sjednocení terminologie třeba.

Generování faktur

Po vytvoření nového projektu je ihned možné vygenerovat fakturu k projektu. Do faktury se vygenerují odpovídající údaje z databáze, které se týkají daného zákazníka a jeho zakázky. Faktura je generována ve formátu PDF a ukládána na FTP prostor webu, odkud je kdykoliv přístupná.

Seznam nezaplacených projektů

Po dokončení projektu se signalizace o existujícím projektu přesune ze seznamu aktuálních projektů do seznamu nezaplacených projektů, aby bylo jasně vidět, na které platby se čeká a jak dlouho. Vygenerované faktury mají splatnost 14 dní. Počítá se s tím, že jakmile je vygenerována faktura, je projekt spolu s fakturou v zápětí odeslán zákazníkovi. Jakmile je tedy projekt v seznamu nezaplacených projektů déle než 14 dní od data vygenerování faktury, změní se barva dané položky pro signalizaci o vypršení splatnosti faktury. Po přesunutí projektu do seznamu nezaplacených již projekt nemůže být editován, ale pouze prohlížen. Jsou zde k dispozici veškeré informace o projektu včetně příslušné terminologie a faktury.

Seznam nevyúčtovaných projektů

Jakmile sdružení obdrží peníze za projekt, manažer sdružení daný projekt označí jako zaplacený a projekt se přesune do seznamu, kde se nacházejí zaplacené projekty, u kterých obdržené finance nejsou rozděleny mezi překladatele, kteří se na projektu podíleli. Nejdůležitější prvek tohoto seznamu bude indikace toho, kdo se na projektu podílel a kterému překladateli náleží jak velká odměna, která byla zadána již při vytvoření projektu každému překladateli zvlášť.

Archív

Do archívu je projekt přesunut, jakmile je projekt vyúčtován mezi překladatele. Archív je rozdělen po kalendářních rocích. To, do kterého roku se projekt zařadí, rozhoduje datum zaplacení projektu, které je klíčové pro účetnictví. Archív bude obsahovat veškeré informace o projektu spolu s fakturou a terminologií.

Projekt, který v průběhu není zrušen, tedy prochází celkem čtyřmi stavy:

- Aktuální
- Nezaplacen (= hotov)
- Zaplacen, ale nevyúčtován mezi překladatele
- Projekt je vyúčtován (konečný stav projektu, kdy je přesunut do archívu)

Seznam zrušených projektů

Pokud dojde ke zrušení projektu, projekt není smazán z databáze, ale je projektu nastaven speciální stav, který přeskočí všechny ostatní stavy projektu při běžném životním cyklu, který končí přesunutím hotového a zaplaceného projektu do archívu. Tento stav bude indikovat zrušení projektu a způsobí jeho přesunutí do sekce Zrušené projekty. Každý projekt v této sekci bude obsahovat veškeré údaje, jako v kterékoli jiné sekci a navíc bude obsahovat krátký popis informující o důvodu jeho zrušení.

Správa uživatelských rolí

Tato sekce, která je přístupna pouze uživatelské roli Administrátor, nabízí prohlížení, vytváření a upravování uživatelských rolí. Každá uživatelská role může disponovat jistými oprávněními v rámci IS. U každé role je přiřazen seznam překladatelů, kteří danou roli zastupují. Seznam možných oprávnění je možno vidět v Tab. 4-1.

Oprávnění prohlížet	Oprávnění editovat	Ostatní
Prohlížet moje projekty	Editace zákazníků	Generování faktur
Prohlížet nevyúčtované projekty	Editace oborů	
Prohlížet nezaplacené projekty	Editace jazyků	
Prohlížet uživatelské role	Editace uživatelských rolí	
Prohlížet aktuální projekty	Editace projektů	
Prohlížet reference	Editace referencí	
Prohlížet terminologie	Editace terminologie	
Prohlížet externí překladatelé	Editace překladatelů	
Prohlížet jazyky	Editace typografie	
Prohlížet archiv		
Prohlížet obory		
Prohlížet zrušené projekty		
Prohlížet zákazníky		
Prohlížet můj profil		

Tab. 4-1 Seznam možných oprávnění v IS.

Správa jazyků

V této sekci je možné definovat jazyky, mezi kterými překladatelské sdružení provádí textové překlady, korektury a které tlumočí. Je zde možno provádět CRUD operace s jazyky a je zde znázorněno, který překladatel ovládá který jazyk.

Správa oborů

Každý překladatel má k sobě přiřazeny obory, ve kterých provádí překlady a tato sekce umožňuje jejich editaci. V této sekci je možné vidět, který překladatel ovládá jaké obory. Příklady oborů: obecný, technické texty, právní texty, zdravotnictví atd.

4.2 Uživatelské role

Uživatelské role určují pravomoci přihlášeného uživatele. Každý uživatel má po přihlášení do systému nějakou roli, která definuje, k jakým prvkům a sekcím má uživatel přístup. Každá uživatelská role disponuje množinou práv, která jsou rozdělena podle jednotlivých úkonů. Lze vytvořit novou uživatelskou roli, které mohou být přiřazena přesně ta práva, která jsou potřeba. V informačním systému budou existovat 3 uživatelské role – překladatel, manažer a administrátor. V rámci IS budou kromě popisku odlišeny také grafickými ikonkami.

4.2.1 Překladatel

Překladatel je role, která v systému nemá skoro žádná práva a nemá přístup k citlivým údajům firmy, které jsou přes informační systém dostupné uživatelským rolím manažer a administrátor. Uživatel s touto rolí má pouze přehled nad akcemi, které se ho týkají a na kterých se podílí. Například, pokud se překladatel podílí na překladu nebo textové korektuře, vidí pouze název projektu, jeho výdělek z daného projektu, do kdy musí být projekt dokončen, jména spoluúčastníků projektu a má přístup k terminologii daného projektu, kterou může editovat. Dále si může zobrazit svůj vlastní profil, kde může editovat své kontaktní údaje.

4.2.2 Manažer

Manažer disponuje všemi právy systému, kromě správy uživatelských rolí s jejich oprávněními. Manažer může prohlížet a editovat veškeré údaje co se týče externích překladatelů, zákazníků a projektů. Má přístup ke všem projektům, ať se na nich podílí nebo ne. Vytváří projekty nové a přiřazuje jim překladatele. Tímto má přístup také k seznamu externích překladatelů, který může spravovat. Komunikuje se zákazníky, takže má přístup ke všem jejich údajům. Má přístup do archívu sdružení, kde může prohlížet dokončené projekty spolu s fakturami. Nakonec může také spravovat jazyky, které překladatelské sdružení ovládá, co se týče překladů, korektur a tlumočení.

4.2.3 Administrátor

Uživatelská role administrátor má stejná práva jako manažer a navíc má možnost vytvářet a editovat uživatelské role a jejich oprávnění.

5 Návrh

V této kapitole je popsán návrh řešení informačního systému na základě předchozí analýzy a specifikace požadavků. Pro návrh systému byl využit jazyk UML, jehož diagramy přehledně a názorně ukazují, jak bude informační systém vypadat a jaké bude mít vlastnosti a funkce. Je ukázán diagram případů užití pro názorné specifikování funkcí systému ve vztahu k jednotlivým uživatelským rolím, ER diagram pro abstraktní a konceptuální znázornění dat uložených v databázi a diagram tříd pro naznačení datových struktur a souvislostí mezi objekty.

5.1 Uživatelské role

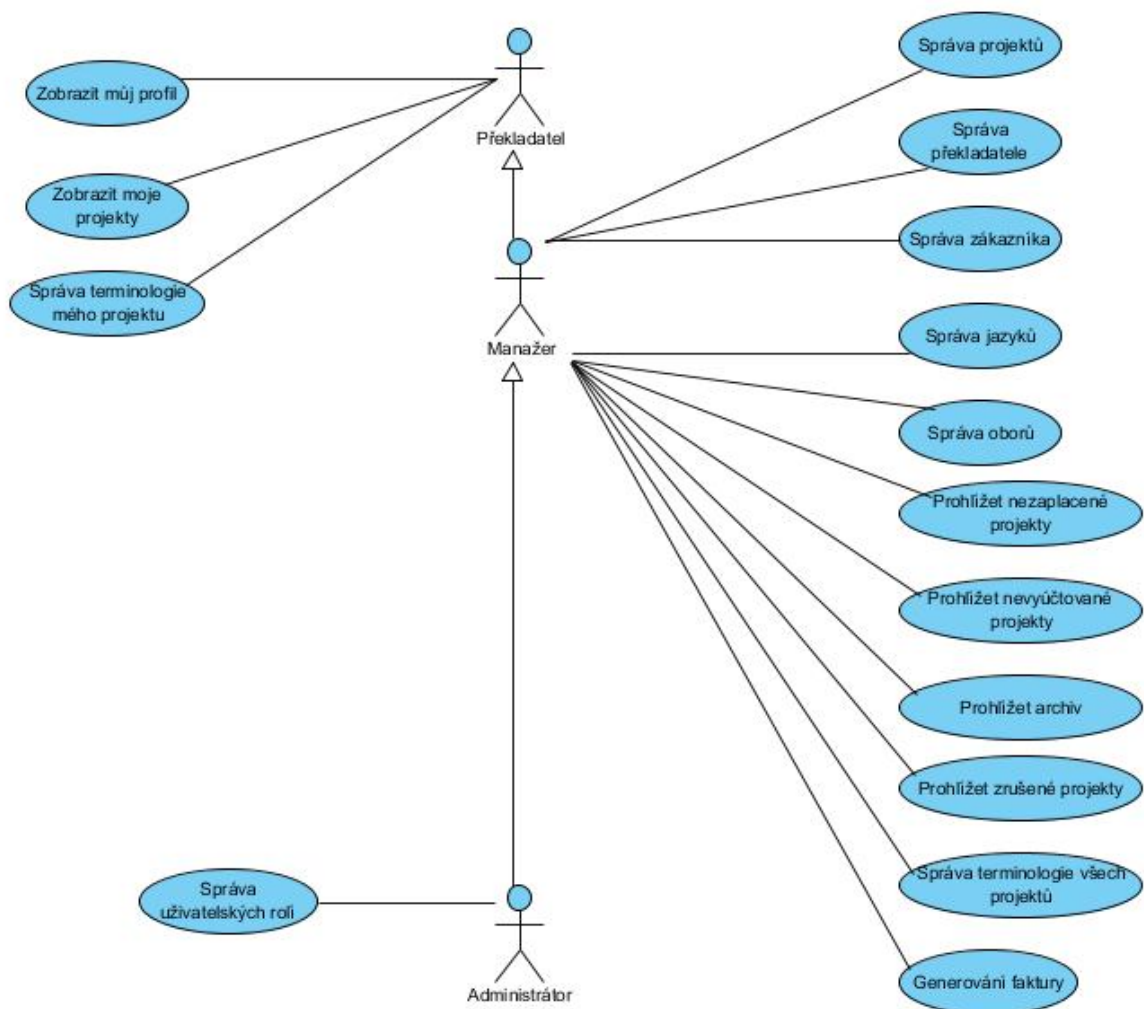
Uživatelské role navrhovaného IS byly vytvořeny dle specifikace požadavků a jsou následující:

- Překladač
- Manažer
- Administrátor

V rámci sdružení nebude s informačním systémem pracovat žádný uživatel, který by představoval roli jinou, než jednu z výše uvedených.

5.2 Funkce systému

Na Obr. 5.1, který zobrazuje use-case diagram, je možné vidět jednotlivá práva a akce jednotlivých uživatelských rolí v systému. Uživatelská role překladač má práva nejnižší a to pouze pro prohlížení projektů, na kterých se podílí spolu se správou terminologie, která se k nim vztahuje a prohlížení vlastního profilu. Uživatelská role manažer disponuje veškerými pravomocemi systému, kromě správy uživatelských rolí, což má navíc k dispozici pouze uživatelská role administrátor. Uživatelská role manažer dědí všechny případy užití uživatelské role překladač a uživatelská role administrátor dědí všechny případy užití role manažer. Všechny případy užití, které jsou pojmenovány jako „Správa“ zastupují tzv. CRUD operace, značící čtyři základní operace používané při práci s daty (CRUD operace jsou popsány v podkapitole 4.1 Uživatelské akce).



Obr. 5.1 – Use case diagram znázorňující uživatelská práva a uživatelské akce v systému

5.3 Struktura databáze

Databáze navrhovaného IS obsahuje následující entity:

Customers

Tato entita reprezentuje veškeré potřebné informace o zákazníkovi překladatelského sdružení. Hlavně kontaktní a fakturační údaje, které jsou dohledatelné v případě nutnosti kontaktování zákazníka nebo vystavení faktury.

Translators

Zde se nacházejí veškeré potřebné informace o externím překladateli, který se sdružením spolupracuje. Hlavně kontaktní, osobní a fakturační údaje, které jsou dohledatelné v případě nutnosti kontaktování překladatele.

Posts

Tato tabulka obsahuje všechny uživatelské role informačního systému. Každá uživatelská role má přesně definovaná oprávnění, která udávají její pravomoci v systému. Tabulka obsahuje název a popis role a je pomocí cizího klíče propojena s tabulkou Rights, kde jsou každému postu přiřazena oprávnění.

Rights

Tato entita obsahuje přístupová práva systému, která jsou patřičně přiřazena každé uživatelské roli. Přístupová práva rolí definují jejich omezení práce s IS. Tabulka obsahuje název a popis práva.

Fields

Zde jsou rozděleny jednotlivé překladatelské obory/odvětví, které jsou propojeny s překladateli, díky čemuž je možné zobrazit, kdo ovládá jaký obor. Tato funkce je využita při vybírání překladatelů pro zadaný projekt.

Invoices

Do této tabulky se ukládají názvy faktur vygenerované k projektům. Každý záznam uchovává informaci o roce, kdy byla faktura vygenerována, což určuje složku, ve které je uložena, dále jméno souboru a id projektu, ke kterému se vztahuje.

Projects

V tabulce Projects jsou zobrazeny veškeré údaje týkající se projektů překladatelského sdružení. Záznam udává informace o zákazníkovi, kdo se podílí na vypracování, mezi kterými jazyky překlad probíhá, jak je finančně oceněn a do kdy musí být vyhotoven.

Terminology

Do této entity se přidávají často opakované a důležité pojmy, se kterými překladatel v projektu (platí pouze u překladu textu) setká. Tabulka obsahuje daný pojem, jeho definici (překlad), ID překladatele, který pojem přidal, časové razítko a cílový jazyk pojmu a ID projektu, ke kterému se pojem vztahuje. Význam této sekce byl popsán v podkapitole 4.1 Uživatelské akce.

Languages

Entita Languages obsahuje názvy a zkratky jazyků, které sdružení ovládá – mezi kterými je schopno provádět překlady, textové korektury a tlumočení. Zároveň je propojena s entitou Translators, kde je vyznačeno, které jazyky daný překladatel ovládá.

5.4 Logika systému

Logika navrhovaného informačního systému je rozdělena do následujících tříd:

Customers

Tato třída poskytuje funkce pro získání údajů o všech dosavadních zákaznických sdružení, pro vytvoření nového zákazníka a pro změnu údajů o zákazníkovi.

Translators

Hlavní funkce této třídy jsou získání údajů o všech překladatelích sdružení, vytvoření nového překladatele a změna údajů o překladateli. Další funkce, které tato třída nabízí, jsou získání jazyků nebo oborů daného překladatele, aktivace a deaktivace překladatele nebo získání údajů o překladateli, který je právě přihlášen.

Rights

Tato třída nabízí 2 funkce – získání seznamu veškerých dostupných práv v informačním systému a získání seznamu oprávnění daného překladatele.

Fields

Tato třída slouží pro manipulaci s obory, ve které překladatelé ovládají v rámci překladů. Hlavní funkce této třídy jsou získání všech oborů, vytvoření, editace a smazání oboru. Další funkcí je získání seznamu překladatelů daného oboru.

Projects

Tato třída je nejobsáhlejší třídou informačního systému. Obsahuje hlavní funkce jako získání údajů o všech projektech, vytvoření nového projektu, editaci již vytvořeného projektu a také jeho zrušení. Další funkce, které tato třída nabízí je získání překladatelů projektu, vedoucího projektu, zákazníka projektu, zakladatele projektu nebo získání seznamu projektů, na kterých se podílí právě přihlášený překladatel.

Invoices

Třída Invoices obsahuje funkce pro práci s fakturou. Funkce nabízí získání údajů o projektu pro vytvoření nové faktury a dále samotné vytvoření nové faktury. Další funkcí je zjištění id poslední vygenerované faktury pro určení id nové faktury, které musí být jedinečné v rámci každého kalendářního měsíce a poslední funkcí je získání faktury daného projektu.

Terminology

Tato třída slouží pro práci s terminologií projektu a to pouze, pokud se jedná o překlad textu, při textové korektuře nebo tlumočení sdílení terminologie není třeba. Pokud je překlad překládán do více jazyků, existuje pro každý cílový jazyk zvláštní seznam terminologie, aby nebyly pojmy pro různé jazyky v jednom seznamu. Funkce této třídy jsou získání všech terminologických pojmů k danému projektu s daným cílovým jazykem, vložení nového pojmu a následně jeho editace a smazání.

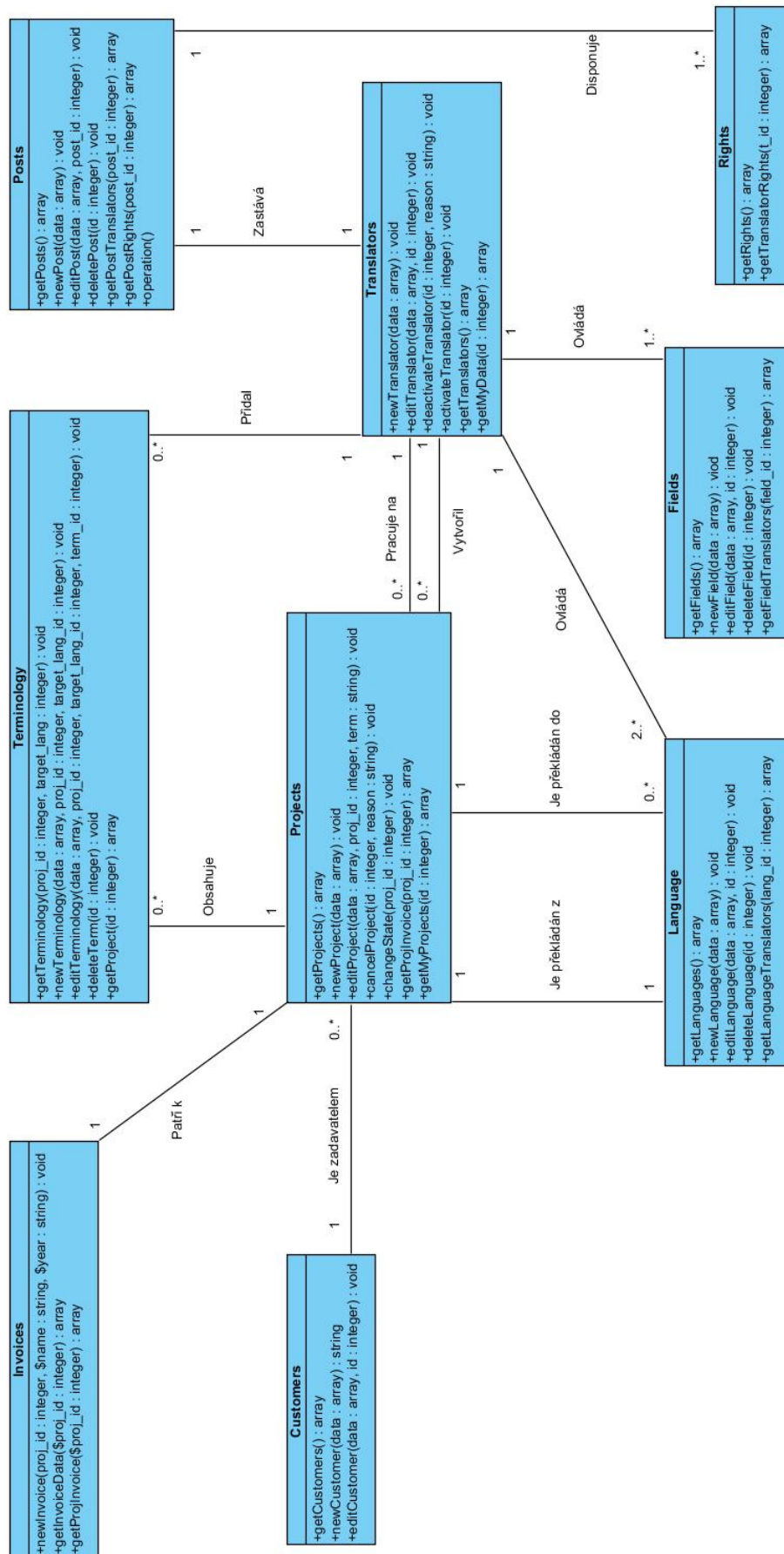
Languages

Třída Languages slouží pro práci s jazyky, se kterými překladatelské sdružení pracuje. Její hlavní funkce jsou opět získání seznamu všech jazyků, vytvoření nového jazyka, editace a smazání. Další funkce, které třída nabízí, jsou získání pouze jména daného jazyka a získání překladatelů k danému jazyku.

Posts

Třída Posts nabízí funkce pro práci s uživatelskými rolemi. Mimo funkce pro získání všech rolí, vytvoření nové role, její následné editace nebo smazání se zde nachází funkce získání všech oprávnění dané role a získání všech překladatelů dané role.

Na Obr. 5.3 je možno vidět diagram tříd (class diagram), který zobrazuje statický pohled na systém, tj. zejména třídy jako typy objektů, obsah tříd a statické vztahy, které mezi nimi existují. Znárodnuje datový model systému od konceptuální úrovně až po implementaci.



Obr. 5.3 – Diagram tříd vrstvy Model.

6 Testování

Před nasazením systému do provozu je třeba jej podrobit řádnému otestování na uživateli, kteří se s ním setkávají poprvé. Můžeme tak narazit na dosud nepovšimnuté nedostatky, které systém může mít, protože systém a hlavně jeho uživatelské rozhraní bylo implementováno pouze dle autora vlastního uvážení a uvážení jednoho člověka v rámci informačního systému, který bude používat desítky uživatelů, nestačí. Testování systému nejlépe ukáže, do jaké míry je vytvořený systém použitelný a efektivní. Dále je zjištěno, co je třeba ještě opravit či zdokonalit, aby práce s IS byla pro uživatele opravdu efektivní, pohodlná a nenáročná.

6.1 Metodika

Pro testování byly vybrány osoby různého věku, různého pohlaví a hlavně různého studijního nebo pracovního oboru, aby výsledky byly co nejvíce vypovídající o náročnosti používání systému. Osoby různého věku, pohlaví i oborů se systémem budou v praxi opravdu pracovat, protože překladatelská činnost nemusí být hlavním oborem daného člověka a také člověk, který se zabývá překlady textů, se nemusí zdatně orientovat při práci s počítačem. Je tedy důležité, aby zejména starší lidé a osoby, které na počítači nejsou příliš zblhlí, zvládli práci s implementovaným systémem bez větších problémů.

Uživatelské testování je rozděleno do dvou hlavních částí:

1. Provedení všech operací, které systém nabízí.
2. Vyplnění dotazníku, který se vztahuje k předchozí práci se systémem.

Lidé, kteří se s naším informačním systémem dosud nesešli, si vyzkoušeli práci se systémem a vykonali na něm veškeré operace, které systém nabízí. Testovací osoba systém testovala v rámci uživatelské role administrátor, která má veškerá oprávnění v systému. Testovací osoba dostala pouze bodový plán, kde jsou popsány jednotlivé úkoly, které má provést a předešlé seznámení se systémem bylo pouze minimální. Bodový plán se skládal z přihlášení do systému, vytvoření projektu, zákazníka, překladatele a jazyka a jejich editace a následné zrušení. Celý bodový plán je možno vidět v příloze, viz Příloha A.

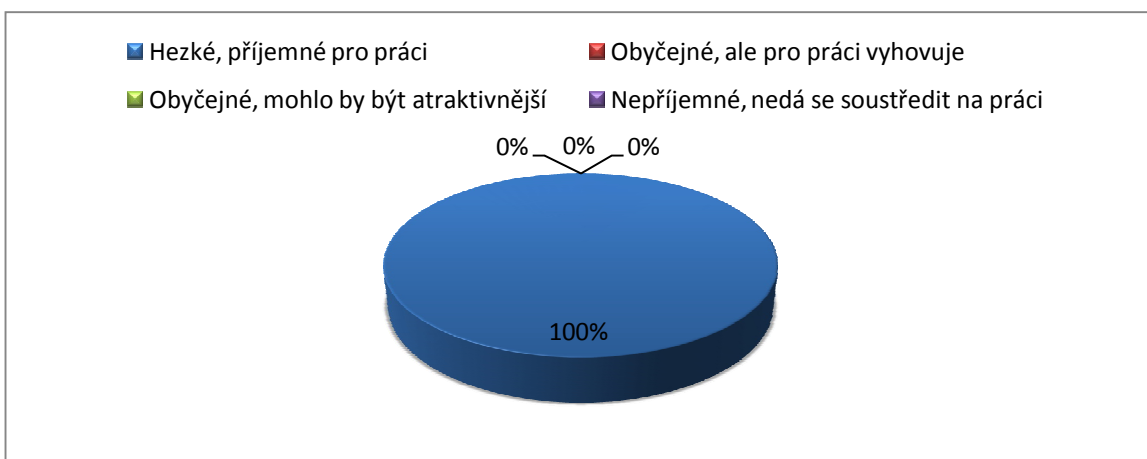
Během testování bylo sledováno, do jaké míry byly jednotlivé body splněny. Dále, jak dlouho a kde všude uživatel po obrazovce „tápal“ a přemýšlel, jak daný úkol splnit. Po dokončení úkolů dostala testovací osoba dotazník k vyplnění. I když většinu nedostatků, kterými systém disponuje, bylo vidět hned při testování, dotazník byl také přínosný, protože byl zaměřen na subjektivní dojmy ze systému. Dotazník je možno vidět v příloze, viz Příloha B.

6.2 Výsledky

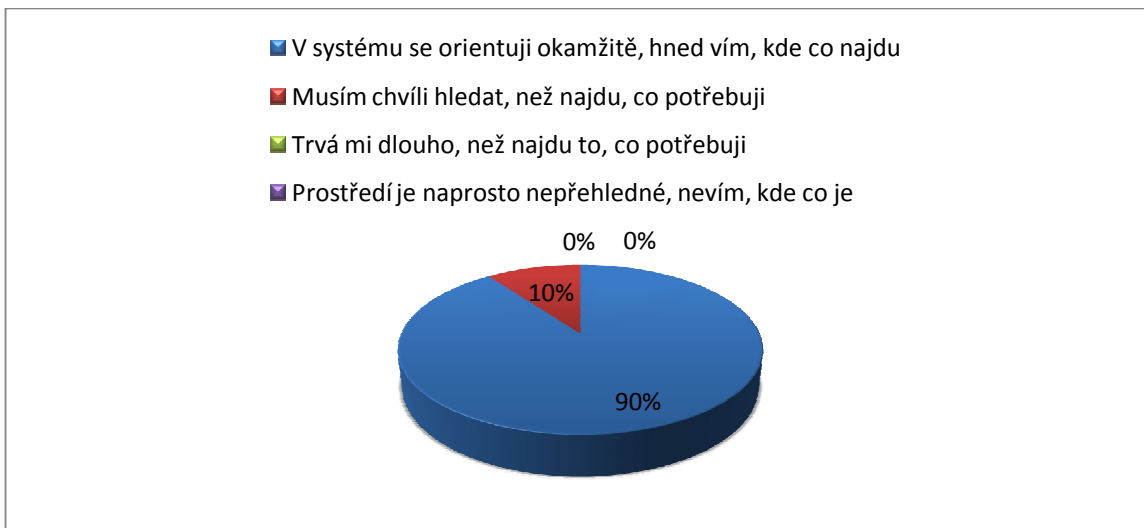
Na následujících diagramech jsou znázorněny výsledky testování implementovaného informačního systému. Testování se zúčastnilo celkem 10 lidí různého pohlaví, věku a pracovního či studijního zaměření. Konkrétně se jednalo o 4 muže a 6 žen ve věkovém rozsahu od 21 do 35 let. Mezi obory, ve kterých se dané osoby pohybují, patří například pedagogika, ekonomie, strojírenství nebo historie. Nebyla tedy mezi nimi žádná osoba, která se pohybuje v oblasti IT, což by do výsledků testování zahrnuje nerelevantní výsledky.



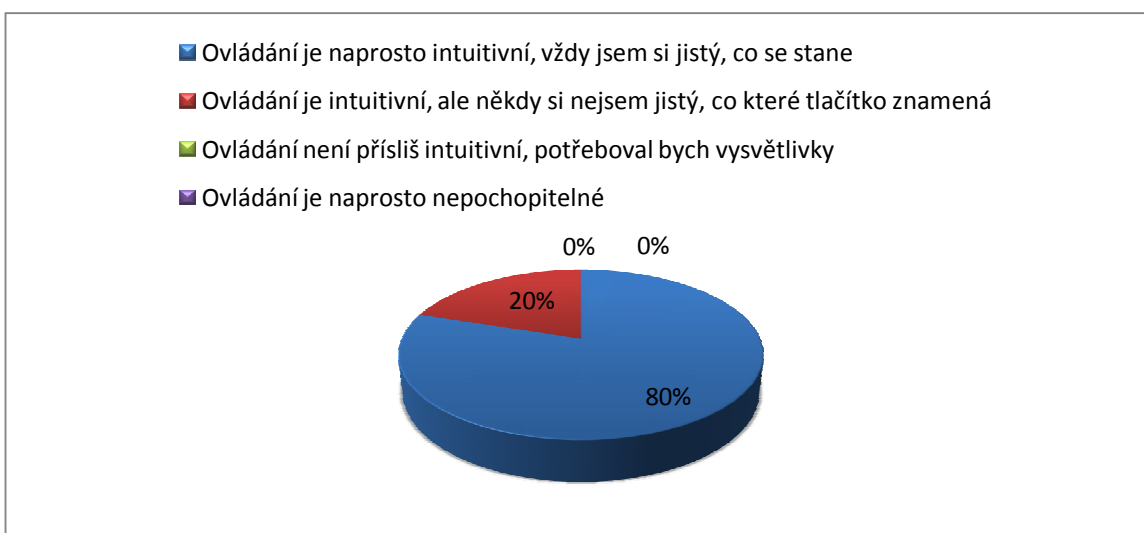
Obr. 6.1 Poměr případů, kdy během testování nastal problém.



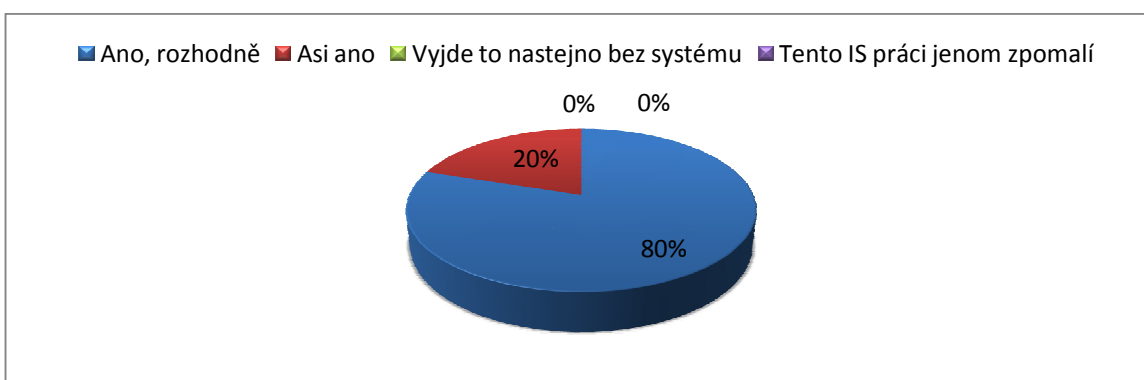
Obr. 6.2 Hodnocení uživatelského prostředí.



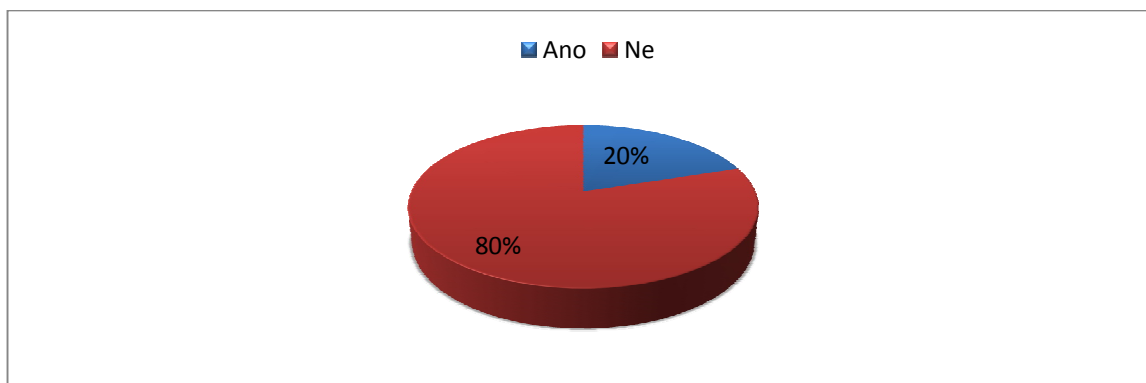
Obr. 6.3 Náročnost orientace v IS.



Obr. 6.4 Míra intuitivnosti ovládání IS.



Obr. 6.5 Může IS ulehčit práci?



Obr. 6.6 Je třeba zdokonalit/zefektivnit některou funkci IS?

6.3 Zhodnocení

Testování informačního systému dopadlo velmi pozitivně. Až na malé nedostatky, se kterými se uživatelé setkali, nenastal žádný větší problém, co se týče funkčnosti systému i pohodlnosti práce. 3 lidé z 10 se setkali během práce s IS s problémem. U všech tří se problém týkal filtrování položek. Filtr podle uživatelů buď nefungoval, nebo chvíli trvalo, než uživatel zjistil, jak ho použít. Problémem se ukázalo to, že filtr je tzv. „case sensitive“, neboli citlivý na velikost písmen. Čili když uživatel zadal jméno s malými písmeny a nebylo nic nalezeno, vyvolalo to dojem, že filtr nefunguje, i když opak byl pravdou. Když uživatel vyhledává nějaké položky, ať už na internetu nebo v aplikaci, na velikost písmen často nedbá, čili tento problém je potřeba vyřešit a upravit implementaci filtru tak, aby ignoroval velikost písmen. 2 lidé z 10 si nebyli vždy jistí funkčností tlačítka. Je to tím, že ne všechna tlačítka mají popisek své funkčnosti, ale mají pouze její grafické znázornění. Toto však nepovažují za závažný problém, protože popisek není u tlačítek, která se v IS vyskytují často a po jejich prvním použití už uživatel s jistotou ví, kterou funkci dané tlačítko reprezentuje. Jelikož bude každý uživatel před prvním použitím systému zaučen, nejedná se o problém. Někteří uživatelé uvedli, že jim v systému chybí některé funkce nebo poukázali na jisté nedostatky. Například ne u všech tlačítek se po najetí myši zobrazí titulek tlačítka, který popisuje jeho funkčnost, což je potřeba doplnit. Jako výborný nápad shledávám návrh na automatické zasílání e-mailů, které připomenou zákazníkovi platbu, pokud ji neprovedl v rámci splatnosti faktury, na což systém uživatele sice upozorňuje, ale automatickou zprávu zákazníkovi nezašle. Dalším zajímavým navrhovaným doplňkem je hromadné zasílání novinek a akčních slev všem zákazníkům, což také do budoucna uvážím. Dále bylo poukázáno na nedostatky, jako chybějící popisek u pole ve formuláři, což bude samozřejmě doplněno nebo bylo zmíněno, že některá pole jsou zbytečně povinná, což také bude zvaženo. Ostatní aspekty systému, jako je intuitivnost nebo příjemný vzhled, byly hodnoceny velmi pozitivně.

Myslím, že testování dopadlo velmi dobře. Na jednu stranu, hlavní funkce systému byly v pořádku a na druhou stranu bylo poukázáno a drobné nedostatky, které je potřeba doladit. Největším přínosem testováním pro implementovaný IS bylo navržení funkčního rozšíření systému o vhodné funkce, jako je automatické zasílání e-mailů, které upozorní zákazníka, aby zaplatil svoji zakázku, pokud vyprší splatnost faktury a peníze ještě nepřišly na účet. Tato funkce bude určitě implementována v budoucnosti jako rozšíření systému.

7 Závěr

Cílem této bakalářské práce bylo navrhnout a implementovat informační systém pro překladatelské sdružení NK Translators jako webovou aplikaci, která poskytuje přehled projektů, zákazníků a externích překladatelů sdružení a usnadňuje vedení překladatelského sdružení. Byl kladen důraz na jednoduché uživatelské rozhraní a intuitivní ovládání.

Klíčem k proniknutí do problematiky bylo seznámení se s návrhem informačních systémů pomocí studia materiálů zmíněných v kapitole 3 Metodologie a praktiky pro návrh IS. Následovalo seznámení se s požadavky kladenými na informační systém pro NK Translators. Jsem členem překladatelského sdružení, takže s požadavky IS jsem byl již obeznámen a bylo je pouze potřeba upřesnit se spolupracovníky. Zjištěné požadavky a specifikace byly podrobně analyzovány CASE nástroji, pomocí kterých byla znázorněna funkcionální systém a dále byla pomocí UML navržena databáze a logika systému. Před samotnou implementací byla provedena malá analýza českého trhu ohledně informačních systémů pro překladatelské agentury, mj. navštívením překladatelské agentury Abiturient, která informační systém používá, a která byla ochotna se podělit o své zkušenosti. Systém byl implementován pomocí jazyků MySQL, PHP a JavaScript, se kterými jsem se seznámil v dostupné literatuře [9][10][11] a využil je k implementaci systému. Jako jádro systému byla zvolena softwarová architektura Model-View-Controller. Funkčnost celého systému byla ověřena na vzorku uživatelů, kteří se se systémem setkali poprvé, a kteří s prací se systémem nebyli předem zaškoleni. Testování dopadlo velmi pozitivně. Byly odhaleny drobné nedostatky systému a zároveň byla navržena další rozšíření systému, která přijdou na řadu v budoucnu. Uživatelské rozhraní bylo pro uživatele intuitivní a rychle pochopitelné.

Myslím, že se podařilo implementovat informační systém, který je vhodný pro překladatelské sdružení a i když je implementovaný přesně na míru pro NK Translators, určitě by ho uvítaly i další překladatelské agentury. Tímto však vývoj systému nekončí. Systém bude ihned nasazen do provozu a myslím, že v průběhu času bude stále co zdokonalovat a budou vznikat návrhy na nové funkce a rozšíření, které dále usnadní vedení překladatelského sdružení.

Literatura

1. **HERMANDEZ, M. J.** *Návrh databází*. Praha 7: Grada Publishing, a.s., 2006. ISBN 80-247-0900-7.
2. **OPPEL, J. A.** *Databáze bez předchozích znalostí*. Brno: Computer Press, a.s., 2006. ISBN 80-251-1199-7.
3. **COCKBURN, A.** *Use Cases: Jak efektivně modelovat aplikace*. Brno: Computer Press, a.s., 2005. ISBN 80-251-0721-3.
4. **ARLOW, J. a NEUSTADT, I.** *UML 2 a unifikovaný proces vývoje aplikací*. Brno: Computer Press a.s., 2008. ISBN 978-80-251-1503-9.
5. **KANISOVÁ, H. a MÜLLER, M.** *UML srozumitelně*. Brno: Computer Press, a.s., 2007. ISBN 80-251-1083-4.
6. **BURBECK, S.** *Applications Programming in Smalltalk-80: How to Use Model-View-Controller (MVC)*. Softsmarts, Inc., 1987.
7. **Singh, I., Stearns, B. a Johnson, M.** *Designing Enterprise Applications with the J2EETM Platform, Second Edition*. Addison-Wesley Longman Publishing Co., Inc., 2002. ISBN 0-201-78790-3.
8. **Baray, C.** The Model-View-Controller Design Pattern. *Indiana University*. [Online] 1999. [Citace: 22. 04 2011.] <http://cristobal.baray.com/indiana/projects/mvc.html>.
9. **PONKRÁČ, M.** *PHP a MySQL bez předchozích znalostí*. Brno: Computer Press, a.s., 2007. ISBN 978-80-251-1758-3.
10. **LACKO, L.** *PHP 5 a MySQL 5 - Hotová řešení*. Brno: Computer Press, a.s., 2007. ISBN 978-80-251-1695-1.
11. **SUEHRING, S.** *JavaScript - Krok za krokem*. Brno: Computer Press, a.s., 2008. ISBN 80-251-2241-9.
12. **LOUNT, P. W.** What is Smalltalk? *Smalltalk dot org*. [Online] [Citace: 22. 04 2011.] <http://www.smalltalk.org/smalltalk/whatissmalltalk.html>.
13. **TIŠNOVSKÝ, P.** Squeak a Smalltalk - historie vývoje grafického uživatelského rozhraní. *ROOT.CZ*. [Online] 09. 09 2010. [Citace: 22. 04 2011.] <http://www.root.cz/clanky/squeak-a-smalltalk-historie-vyvoje-grafickeho-uzivatelskeho-rozhrani/>. ISSN 1212-8309.
14. **BERNARD, B.** Úvod do architektury MVC. *zdrojak.root.cz*. [Online] 05. 07 2009. [Citace: 27. 03 2011.] <http://zdrojak.root.cz/clanky/uvod-do-architektury-mvc/>. ISSN 1212-8309.

Seznam příloh

Příloha A – Bodový plán testování IS

Příloha B – Dotazník

Příloha C – DVD se zdrojovými soubory a souborem README

Příloha A – Bodový plán testování IS

Přejděte na [www: https://www.nk-translators.com/bp/](https://www.nk-translators.com/bp/)

- Přihlaste se do systému jako administrátor:
 - o login: admin@admin.cz
 - o heslo: admin
- Pokud jste se úspěšně přihlásili, na obrazovce vidíte aktuální projekty překladatelského sdružení NK Translators (překlady, textové korektury nebo tlumočení).
- Zobrazte detaily kteréhokoliv projektu.
- Zkuste použít filtr na kteroukoliv nabízenou položku.
- Založte nový projekt (dle vlastní fantazie).
- Změňte některý údaj ve Vámi vytvořeném projektu.
- Označte Vámi vytvořený projekt jako dokončený.
- Seřad'te projekty podle termínu.
- Přejděte do sekce Nezaplacené projekty.
- Najděte zde svůj projekt, vygenerujte mu fakturu a označte jako zaplacený.
- Přejděte do sekce Nevyúčtované projekty.
- Najděte zde svůj projekt a označte jej jako vyúčtovaný.
- Přejděte do archívu a ujistěte se, že se v aktuálním roce nachází Váš projekt.
- Přejděte do sekce Překladaelé.
- Vytvořte nového překladatele (životopis – životopis.pdf, fotka – fotografie.jpg).
- Změňte Vašemu překladateli kterýkoliv údaj.
- Vašeho překladatele deaktivujte, protože nedodržel pracovní podmínky.
- Seřad'te překladatele podle příjmení.
- Přejděte do sekce Zákazníci.
- Vytvořte nového zákazníka.
- Změňte Vašemu zákazníkovi kterýkoliv údaj.
- Přejděte do sekce Uživatelské role.
- Vytvořte novou uživatelskou roli v systému (ikona – role.png) s jistými oprávněními.
- Změňte nové uživatelské roli kterýkoliv údaj.
- Smažte novou uživatelskou roli.
- Přejděte do sekce Jazyky.
- Vytvořte nový jazyk (ikona – jazyk.gif).
- Změňte kterýkoliv údaj nového jazyka.
- Smažte nový jazyk.
- Přejděte do sekce Obory.

- Vytvořte nový obor (jedná se o obor překladatelský).
- Změňte kterýkoliv údaj nového oboru.
- Smažte nový obor.
- Prohlédněte si vlastní profil.
- Odhlaste se ze systému.

Příloha B - Dotazník

Věk:

Pohlaví: muž / žena

Obor (studijní nebo pracovní):

- Nastal nějaký problém během některého úkolu?
 - Ne
 - Ano:
- Jak na Vás působí prostředí informačního systému?
 - Hezké, příjemné pro práci
 - Obyčejné, ale pro práci vyhovuje
 - Obyčejné, mohlo by být atraktivnější
 - Nepříjemné, nedá se soustředit na práci
- Náročnost orientace v systému
 - V systému se orientuji okamžitě, hned vím, kde co najdu
 - Musím chvíli hledat, než najdu, co potřebuji
 - Trvá mi dlouho, než najdu to, co potřebuji
 - Prostedí naprosto nepřehledné, nevím, kde co je
- Je ovládání systému intuitivní?
 - Ovládání je naprosto intuitivní, jsem si vždy jistý, co se stane
 - Ovládání je intuitivní, ale někdy si nejsem jistý, co které tlačítko udělá
 - Ovládání není příliš intuitivní, potřeboval bych vysvětlivky
 - Ovládání je naprosto nepochopitelné
- Myslíte, že může tento informační systém ulehčit práci ve firmě?
 - Ano, rozhodně
 - Ano, trochu
 - Vyjde to nastejno
 - Tento IS práci jenom zpomalí
- Když se vžijete do správce projektů v překladatelském sdružení, chybí Vám nějaká konkrétní funkce v systému?
 - Ne
 - Ano:
- Narazili jste na funkci, kterou je potřeba zdokonalit/zefektivnit?
 - Ne
 - Ano: