

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

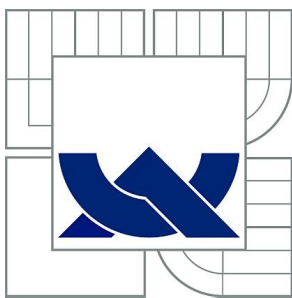
IS PRO PODPORU ZÁKAZNÍKŮ

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

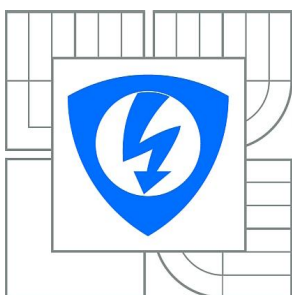
Bc. TOMÁŠ DOLEŽAL

BRNO 2010



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ**

**ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

## IS PRO PODPORU ZÁKAZNÍKŮ

IS FOR CMR

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

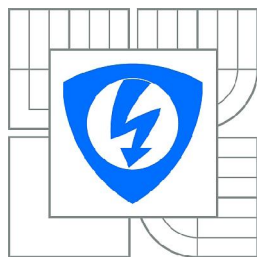
**Bc. TOMÁŠ DOLEŽAL**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. RADOVAN HOLEK, CSc.**

BRNO 2010



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí techniky

# Diplomová práce

magisterský navazující studijní obor  
**Kybernetika, automatizace a měření**

**Student:** Bc. Tomáš Doležal

**ID:** 83843

**Ročník:** 2

**Akademický rok:** 2009/2010

**NÁZEV TÉMATU:**

## IS pro podporu zákazníků

### POKYNY PRO VYPRACOVÁNÍ:

1. Navrhněte koncepci informačního systému pro podporu zákazníků
2. Proveďte datovou a funkční analýzu informačního systému, navrhněte E-R diagramy a diagramy datových toků pro jednotlivé role v systému. Zabývejte se problematikou životních cyklů.
3. Navrhněte procesy pro jednotlivé role v IS.
4. Navrhněte design jednotlivých formulářů pro realizaci procesů
5. Realizujte Informační systém a ověřte správnost navrženého řešení. Realizace se předpokládá nad databází MySQL za pomoci PHP.

### DOPORUČENÁ LITERATURA:

Maslakowski M., Naučte se MySQL za 21 dní. Praha: Computer Press, 2001. 478 s. ISBN 80-72226-448-6.

Brázda J., PHP 5 začínáme programovat. Praha: Grada Publishing a.s., 2006. 244 s. ISBN 80-247-1146-X.

Dle pokynů vedoucího práce a vlastního výběru.

**Termín zadání:** 8.2.2010

**Termín odevzdání:** 24.5.2010

**Vedoucí práce:** Ing. Radovan Holek, CSc.

**prof. Ing. Pavel Jura, CSc.**

*Předseda oborové rady*

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## Prohlášení

„Prohlašuji, že svou diplomovou práci na téma IS pro podporu zákazníků jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne: **24. května 2010**

.....  
podpis autora

## Poděkování

Děkuji vedoucímu diplomové práce Ing. Radovan Holek, CSc. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: **24. května 2010**

.....  
podpis autora

### **ABSTRAKT**

Tato diplomová práce se zabývá návrhem a zpracování MySQL databáze použitím programovacího jazyka PHP5. Cílem práce je IS na podporu zákazníků. K dosažení svých cílů používá PHP Framework Symfony. Pro každou roli v systému jsou prezentovány E-R diagramy, diagramy datových toků, stejně tak jako procesní diagramy. Diplomová práce se také zabývá životním cykly entit. Celá práce má za výsledek komplexní funkční informační systém, který obsahuje administraci a uživatelské rozhraní.

### **KLÍČOVÁ SLOVA**

Databáze MySQL, PHP, E-R diagramy, wramework, symfony, webdesign, role, procesy

### **ABSTRACT**

This dissertation is concerned with design and elaboration of MySQL database using PHP5 programming language. Goal of this work is customer support IS. To achieve it's goal uses Symfony PHP framework. E-R diagrams, dataflow diagrams as well as processes for each role are presented for this information system. Dissertation is also concerned with entities life cycles. Whole work results in the working information system, which contains administration and user frontend.

### **KEYWORDS**

MySQL database, PHP, E-R diagram, framework, symfony, webdesign, role, processes, information system

### Bibliografická citace

DOLEŽAL, T. *IS pro podporu zákazníků*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 63 s. Vedoucí diplomové práce Ing. Radovan Holec, CSc.

<b>1. ÚVOD .....</b>	<b>7</b>
<b>2. POŽADAVKY NA INFORMAČNÍ SYSTÉM.....</b>	<b>8</b>
2.1 Obecné požadavky .....	8
2.2 Požadavky na informační systém .....	8
<b>3. SOFTWARE VYBAVENÍ .....</b>	<b>9</b>
3.1 MySQL.....	9
3.2 PHP .....	10
3.3 Apache.....	10
3.4 CSS.....	10
3.5 XAMPP .....	11
3.6 SYMFONY .....	11
3.6.1 Instalace symfony .....	13
3.6.2 Stromová struktura souborů framework symfony .....	16
<b>4. E-R DIAGRAMY .....</b>	<b>22</b>
4.1 Relace 1:N .....	22
4.2 Relace M:N.....	23
<b>5. ŽIVOTNÍ CYKLUS .....</b>	<b>24</b>
5.1 Životní cyklus faktury .....	24
5.2 Životní cyklus smlouvy .....	26
5.3 Životní cyklus služby .....	28
5.4 Životní cyklus uživatele .....	29
5.5 Životní cyklus zákazníka.....	30
5.6 Životní cyklus ostatních entit .....	31
<b>6. ROLE.....</b>	<b>33</b>
<b>7. MODULY .....</b>	<b>34</b>
<b>8. PROCESY .....</b>	<b>36</b>
8.1 Procesy práce se zákazníkem .....	36
8.1.1 Proces zákazník .....	37
8.1.2 Procesy přiřazení služeb.....	39
8.1.3 Proces smlouva.....	40

8.1.4	Proces zákazník – login.....	42
8.2	Procesy fakturace.....	43
8.2.1	Fakturace služeb .....	44
8.2.2	Úprava faktury.....	45
8.2.3	Hromadné generování faktur ze smluv .....	47
8.3	Proces správa zAMĚSTNanců .....	48
8.4	Proces správa daně .....	48
8.5	Proces smluvní podmínky .....	49
8.6	Proces typů služeb.....	50
8.7	Proces správa služeb .....	50
8.8	Report zobrazení.....	51
<b>9.</b>	<b>DESIGN.....</b>	<b>52</b>
<b>10.</b>	<b>ZÁVĚR.....</b>	<b>56</b>
<b>11.</b>	<b>SEZNAM LITERATURY .....</b>	<b>57</b>
<b>12.</b>	<b>SEZNAM ZKRATEK, SYMBOLŮ A PŘÍLOH.....</b>	<b>58</b>

## 1. ÚVOD

Využívání internetu se v dnešní době stalo naprostou běžností. Proto se i mnoho firem snaží nějakým způsobem prezentovat na internetu. Jednoduché webové stránky s kontaktem a popisem své činnosti se mění na rozsáhlé databázové systémy umožňující všestrannou komunikaci. Proto se zabýváme návrhem informačního systému pro podporu zákazníků. Tento systém zpřehledňuje zákazníkovi své aktivity. Naopak firma tímto systémem dostává větší přehled o zákazníkovi. Tuto práci budeme realizovat na databázi MySQL za podpory PHP. Navrhne koncepti informačního systému. Navrhne E-R diagramy a diagramy datových toků pro jednotlivé role v systému. Sestavíme procesy pro jednotlivé role a graficky ztvárníme webdesing.

## 2. POŽADAVKY NA INFORMAČNÍ SYSTÉM

### 2.1 OBECNÉ POŽADAVKY

Postupným rozvojem sítě WWW vznikalo a vzniká řada internetových prohlížečů. Jeden z obecných požadavků je, aby ve všech běžně používaných prohlížečích byly stránky zobrazovány téměř stejně. Za nejpoužívanější lze považovat IE7, IE8, Firefox 3, Opera, Gogole Chrome. Je otázkou, zda je vhodné optimalizovat stránky například i na straší verze IE nebo touto „neoptimalizací“ donutit uživatele k vyšší, stabilnější, rychlejší a hlavně i bezpečnější verzi.

Webdesing by měl být především přehledný a přitom estetický.

### 2.2 POŽADAVKY NA INFORMAČNÍ SYSTÉM

Pro návrh IS pro podporu zákazníků jsme si na začátku stanovili tyto základní požadavky:

- Systém by měl být univerzální.
- Nabízené služby lze rozdělit do kategorií.
- Jedna služba může mít více kategorií.
- Zákazník se sám nemůže registrovat.
- Každému zákazníkovi lze přidělit přístupové práva (login a heslo).
- Zákazník si může prohlížet své faktury a smlouvy.
- Faktury lze reklamovat
- Služby se zobrazují bez přihlášení
- U faktur a smluv se musí zaznamenávat, který zaměstnanec s touto položkou naposledy pracoval (jak a kdy).
- Systém využívá MVC přístup.
- Data v databázi se nemažou – vyřešit za pomoci stavů
- Na jedné web stránce lze editovat jen jednu entitu (tabulku v databázi)

## 3. SOFTWARE VYBAVENÍ

### 3.1 MYSQL

MySQL je systém správy relačních databází podnikové úrovně. Pro navýšení rychlosti na server je vícevláknový (multithreaded). Komunikace probíhá pomocí jazyka SQL. Systém plně odpovídá standardu ANSI SQL92. MySQL byla od počátku optimalizována na rychlost i za cenu některých zjednodušení. Nyní je nejnovější verze 5, která přináší mimo jiné pohledy, triggerů a uložení procedury. Ve vývoji již je nová verze, kde se můžeme dočkat například podpory cizích klíčů nejen pro tabulky InnoDB či úložiště Falcon. Produkt je poskytován pod licencí GNU GPL. [1]

Kódování ANSI SQL92 je soubor standardů pro strukturovaný datový jazyk SQL, sjednaný v roce 1992 v Americkém institutu pro národní standardy. [2]

#### Historie

MySQL vznikl v dílně švédské firmy TcX v roce 1996. Jeho další rozvoj měla na svědomí firma MySQL AB. V červnu 2000 byl oznámen vstup dvou investorů Progress Software a VA Linux. [3]

#### Relační databáze

Model relační databáze byl vyvinut v sedmdesátých letech minulého století panem E. F. Coddem. [1] Relační databáze napodobuje způsob myšlení lidí. Člověk seskupuje podobné objekty do systematických celků a složité objekty rozkládá na jednodušší. I relační databáze seskupuje podobná data do jedné tabulky a složité datové struktury rozkládá do více logicky seskupených tabulek. Jednotlivé tabulky jsou k sobě vázány. Jinými slovy: databáze sestavená z řady tabulek, jejichž sloupce jsou vázány na sloupce v jiných tabulkách.

### 3.2 PHP

PHP je serverový skriptovací jazyk navržený pro potřeby webových stránek. Probíhá na straně server a generuje HTML (či jiný) výstup, který se zobrazuje na straně uživatele. Je to Open Source nezávislá na platformě. [2]

#### Historie

V roce 1994 Rasmus Lerdorf vytvořil jednoduchý systém evidence přístupů k web. Se zvyšujícími se požadavky vzniká Personál Home Page Tools a následně Personál Home Page Construction Kit. [5]

- PHP/Fi - Vytvořen nástroj pro podporu SQL dotazů. PHP/FI 2.0 bylo oficiálně uvedeno v roce 1997.
- PHP 3 - Brzy po oficiálním uvedení PHP/FI 2.0 vytváří Andi Gutmans a Zeev Surasky novější verzi.
- PHP 4 - Verze 4 byla uvedena v roce 2000. Byla rozšířena například o podporu pro mnoho WWW serverů, http sessions, buffering výstupu, zvýšena bezpečnost zpracování vstupů uživatele.
- PHP 5 - Vydána v červnu 2003. Nejvýraznější změny jsou v objektovém modelu.

### 3.3 APACHE

Apache HTTP Server je otevřený Open Source pro Linux, BSD a Windows platformy. Jedná se o softwarový webový server. Tento server se stal velmi oblíbený, od roku 1996 se stal nejpoužívanějším serverem na internetu. [4]

### 3.4 CSS

Kaskádový styl (Cascading Style Sheets) je jazyk pro popis způsobu zobrazení web stránek napsaný v jazyce HTML, XHTML, XML. Jeho hlavní výhodou je oddělení vzhledu od obsahu stránek. Tvorba stránek se stane přehlednější, nezatížena mnoha značkami, které se používají v HTML pro vzhled. Úpravou tohoto stylu se lehko změní celkový vzhled stránek. Způsob formátování je

přístupnější a snáze se vytvoří požadovaný vzhled. Při načítání stránek webovým prohlížečem dochází ke zrychlení (CSS se načítá pouze jednou, pokud nedojde ke změně). [5]

### 3.5 XAMPP

XAMPP je multiplatformní balíček obsahující Apache, PHP a MySQL. Tento balíček slouží především pro snadnou a rychlou instalaci potřebného vybavení pro práci s databází MySQL. Veškerý instalovaný software je přednastaven dle nejčastějších požadavků. Součástí balíčku je phpMyAdmin, fileZilla FTP server, merkury Mail Transport Systém a OpenSSL. [4]

PhpMyAdmin je nástroj, který prostřednictvím webového rozhraní umožňuje jednoduchou správu databáze MySQL.

XAMPP nainstalovaný na osobním počítači je velmi vhodný pro vytváření a ladění databázových systémů. Přímá úprava souborů na osobním počítači (localhost) ušetří mnoho času při ladění. Odpadá tak zdlouhavé kopírování přes ftp na vzdálený server.

### 3.6 SYMFONY

Symfony je komplexní webový framework postavený na PHP 5 vycházející z návrhového vzoru MVC, distribuovaný pod licencí MIT, nijak neomezující komerční použití. Symfony obsahuje nástroje pro zabezpečení formulářových prvků, které jsou největším bezpečnostním rizikovým faktorem webových aplikací, cache pro zrychlení generování webových stránek, testovací nástroje a abstraktní vrstvu pro práci s databázemi. [6]

Celý framework je z velké části inspirován jinými webovými aplikačními frameworky jako Ruby On Rails, Django a Spring. Vznikl původně pod názvem Sensio Framework jako odnož projektu Mojavi3-DEV, který měl v sobě integrován ORM vrstvu Propel.

Nejaktuálnější stabilní verze je symfony 1.4, na které je postavena tato diplomová práce. V této verzi je integrována ORM vrstva Doctrine 1.2 (zle nadále používat i Propel).

Momentálně autor symfony pracuje na verzi 2, která slibuje až 60% zrychlení a dle beta testerů opravdu takového navýšení rychlosti dosahuje. Symfony 1.4 už je v této verzi dostatečně rychlá a spolehlivá, což dokazuje velké množství web aplikací běžících na tomto systému.

Symfony podporuje tyto databáze: MySQL, PostgreSQL, Oracle a Microsoft SQL Serveru.

*Pozn.: název symfony se píše malými písmeny, protože francouzský autor Fabien Potencier nemá rád velké písmena.*

### **Framework**

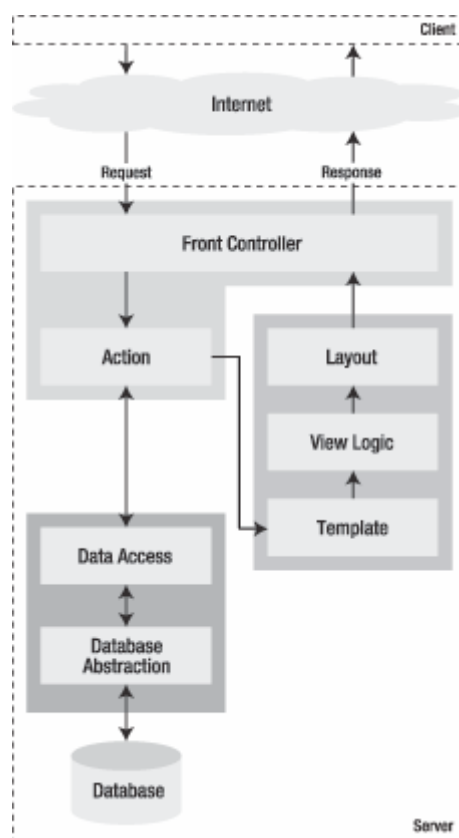
Framework je softwarová struktura, která slouží jako podpora při programování a vývoji a organizaci jiných softwarových projektů. Může obsahovat podpůrné programy, knihovnu API, návrhové vzory nebo doporučené postupy při vývoji.[7]

Cílem frameworku je převzetí typických problémů dané oblasti, čímž se usnadní vývoj tak, aby se návrháři a vývojáři mohli soustředit pouze na své zadání. Symfony této definici odpovídá.

### **MVC**

Model-view-controller je softwarová architektura, 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. [6]

- Model představuje informaci, podle které aplikace pracuje
- Pohled (view) vykresluje model do webové stránky v závislosti na interakci uživatele.
- Kontroler (controller) reaguje na akce uživatele a vyvolává vhodné změny modelu nebo pohledu.



Obr. 1: Architektura MVC v symfony [6]

Symfony pro zobrazení jedné web stránky prochází 7 skriptů: model (databázová abstrakce, přístup k datům), pohled (pohled, šablona, rozložení), kontroler (front kontroler, akce).

### 3.6.1 Instalace symfony

Nejprve je nutné nainstalovat webový server (např. Apache), databázový engine (v tomto případě MySQL) a PHP 5.2.4 nebo novější. Pak teprve můžete přejít k instalaci samotné symfony.

Obrovskou výhodou framework symfony je možnost použití příkazové řádky. Příkazový řádek umožňuje vygenerovat celou adresářovou strukturu našeho projektu. Má příkazy pro připojení k databázi a následnou úpravu databáze atd.

## Postup vytvoření projektu IS pro podporu zákazníků

Vytvoření adresáře, kde bude celý projekt IS pro podporu zákazníků:

Linux: `$ mkdir -p /home/sfprojects/dip $ cd /home/sfprojects/dip`

Win: `c:\> mkdir c:\sfprojects\dip c:\> cd C:\sfprojects\dip`

Instalace knihoven symfony

Knihovny pro symfony lze instalovat globálně (doporučeno) nebo pro každý projekt zvlášť (do stejného adresáře jako projekt). Doporučuji instalovat přímo na hlavní disk `C:\symfony`, u Linux `lib/vendor/symfony` Symfony lze stáhnou přímo na oficiálních stránkách. [6]

Ověření instalace:

Linux: `$ cd ../../ $ php lib/vendor/symfony/data/bin/symfony -V`

Win: `c:\ php C:\symfony\data\bin\symfony -V`

Vytvoření projektu:

v adresáři, kde se má projekt vytvořit použijete příkaz:

Linux: `$ php lib/vendor/symfony/data/bin/symfony generate:dip`

Win: `php C:\symfony\data\bin\symfony generate:dip`

Tímto je vytvořena základní struktura celého projektu (obr. 2). V kořenovém adresáři projektu byl vytvořen zástupce *symfony* na knihovny symfony. Ve Windows je třeba přepokopírovat `symfony.bat` do složky projektu. Pro další příkazy využijeme tohoto zástupce.

Vytvoření aplikace

`$ php symfony generate:app admin`

Povolení přístupu do adresáře `cache` a `log`

`$ chmod 777 cache/ log/`

Vygenerování schéma.yml z SQL:

```
$ php symfony doctrine:build-schema
```

Tento příkaz použijete pokud již databázi máme vytvořenou.

Propojení databáze s projektem:

```
$ php symfony configure:database "mysql:host=localhost;dbname=dip"  
root heslo
```

Máme vytvořené *schéma.yml* popisující databázi a zaponoví Doctrine funkci vygenerujeme SQL.

```
$ php symfony doctrine:build --sql
```

Vytvoření tabulek v databázi:

```
$ php symfony doctrine:insert-sql
```

Generování PHP třídy, která mapuje entitu:

```
$ php symfony doctrine:build --model
```

Tento příkaz generuje soubory ve složce *lib/model*. Vytváří tři soubory pro každou entitu. Příkladem bude entita ADRESA:

Adresa – objekt této třídy reprezentuje jeden záznam v tabulce ADRESA. Třída je prázdná ve výchozím nastavení.

BaseAdresa – rodič třídy Adresa. Při každém spuštění příkazu uvedeného výše se třída přepíše. Tento soubor programátor neupravuje. Při přegenerování programátor ztrácí veškerý svůj napsaný zdrojový kód v tomto souboru.

AdresaTable – třída definuje metody, které vrací objekt. Třída je prázdná ve výchozím nastavení.

Vytvoření forms a validators

```
$ php symfony doctrine:build --all --no-confirmation
```

Tento příkaz provede všechny dosavadní popsané příkazy a následně vytvoří knihovny s třídami formulářů a zároveň k nim příslušné validators. Vygenerované

rodičovské třídy je třeba podědit a upravit. I když je nutné je skoro celé přepisovat (protože nejsou dostačující), je vhodné je používat.

#### Testovací data

V adresáři *data/fixtures* do *yml* souborů se vepisují testovací data, které se nahrají příkazem:

```
$ php symfony doctrine:data-load
```

#### Vytvoření modulu:

```
$ php symfony doctrine:generate-module --with-show --non-verbose-templates admin ADRESA Adresa
```

Tento příkaz vytvoří základní akce a šablony. Vytváří šablony *index*, *edit*, *new* a *show* a k nim příslušné akce. Tímto je vytvořen dobrý základ pro naprogramování modulu.

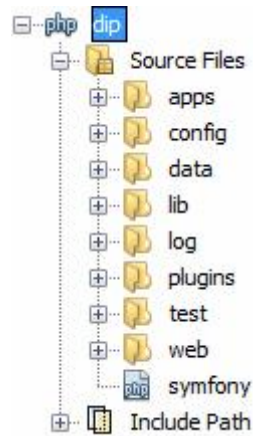
#### Vymazání cache

```
$ php symfony cc
```

Používá se při instalaci nových plugin či před nahráním na veřejný server.

### 3.6.2 Stromová struktura souborů framework symfony

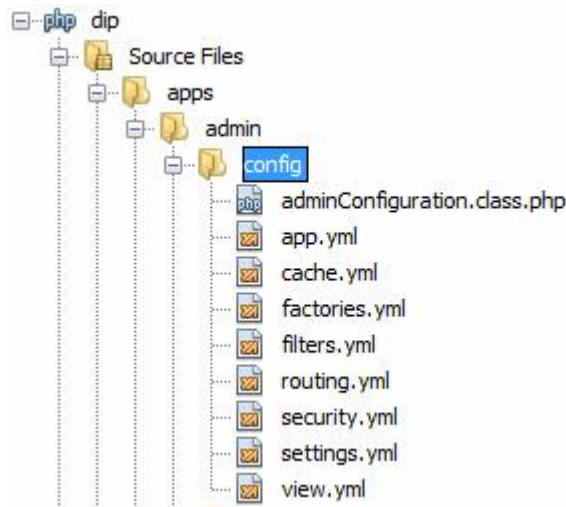
Na první pohled se může zdát, že je symfony velmi nepřehledné v záplavě složek a souborů, ale vše je logicky uspořádané a na druhý pohled i velmi přehledné. Na obr. 2 je hlavní adresářová struktura. Dále jsou popsány jednotlivé složky.



Obr. 2: Hlavní adresářová struktura projektu

### Složka apps

Projekt se rozděluje na aplikace a ty se dále dělí na jednotlivé moduly, které dále mohou být rozšířeny o plugins. V jednom projektu může být více aplikací (v IS pro podporu zákazníků jsou dvě: admin a frontend).



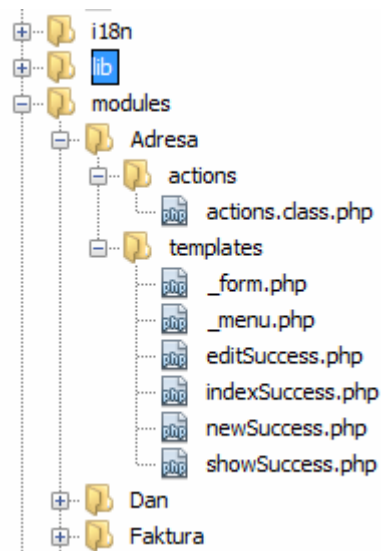
Obr. 3: Zobrazení složky config

Každá aplikace má svou vlastní konfiguraci, která je rozdělena do více yml souborů, jak ukazuje obr. 3. Pokud chceme pro jeden nebo více modulů mít jinou konfiguraci než pro celou aplikaci, vytvoří se v příslušném modulu soubor se stejným názvem jak v hlavní aplikaci a požadované změny se v tomto souboru modifikují. Tento soubor si symfony automaticky načítá. Co lze vše nastavit je podrobně popsáno v dokumentaci na oficiálních stránkách [6]. Ve stručnosti ve view

se definují javascripts, css, metas... Setings – předefinování prud, dev, test. Security – zapnutí nutnosti přihlášení. Routing – cesty a přesměrování. Cache zapínání, vypínání cache, velikost cache (pro programování i za cenu zpomalení se nechává vypnutá).

### Složka modules

Na obr. 4 je ukázán model adresy.



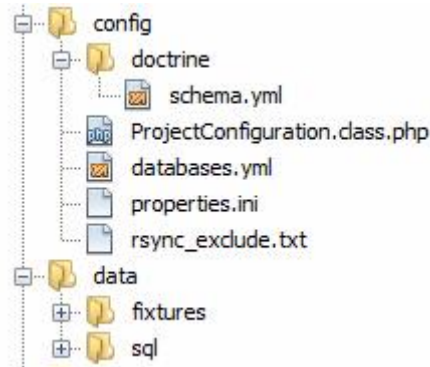
Obr. 4: Modul adresa

Každý modul se skládá z akcí (actions) a šablon (templates) . Akce se dále dělí na jednotlivé akce pro daný modul a na komponenty (components.class.php) pro daný modul(y). Šablony se pojmenovávají podle akcí, pokud se akce jmenuje index, tak odpovídající šablona je indexSuccess případně indexError.

Příkazem `$this->setTemplate('navez_šablony');` lze pro více akcí použít jednu šablonu.

### Složka config

V této složce jsou základní nastavení pro identifikaci databáze a její struktury. Z těchto souborů se přes příkazovou řádku vytváří moduly a třídy (obr. 5).

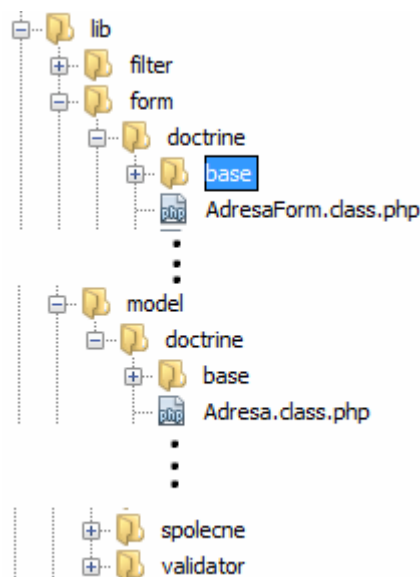


Obr. 5: Složka config

- schem.yml – model celé databáze (entity).
- ProjectConfiguration.class.php – konfigurace, volání plugins
- databases.yml – cesta, heslo k databázi, kódování...
- fixtures – obsahuje yml soubory obsahující testovací data

### Složka lib

Do složky lib se vkládají téměř veškeré třídy používané v projektu (obr. 6).



Obr. 6: Složka lib

### **Složka form**

Knihovny s třídami formulářů, které na základě parametrů generují html formulář. Za pomocí těchto knihoven můžeme konfigurovat html formuláře a dodržet MVC architekturu.

### **Složka modul**

Popisuje jednu z částí ORM mapování na databázových tabulkách (entita) . Zajišťuje možnost rozšíření třídy co generuje mapování o další parametry a metody, které zpracovává data před načtením (uložením) do (z) finálních objektů pro php.

### **Složka spolecne**

Společné knihovny pro všechny moduly (není v základu vytvořeno přes příkazový řádek).

### **Složka validator**

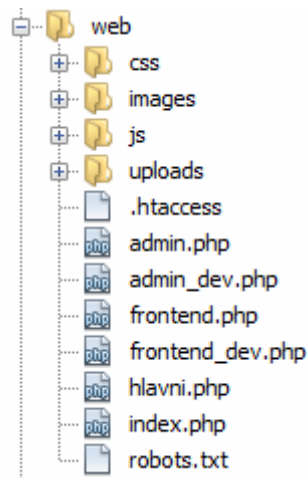
Pokročilé validátory, které nelze definovat přímo ve složce form. Využito především pro kontrolu více tříd formulářů ve vzájemném vztahu (není v základu vytvořeno přes příkazový řádek).

### **Složky base**

Vytvořená základní struktura pro form a modul. Tyto soubory se dědí a modifikují dle potřeb programátora. (Nelze je měnit přímo, protože symfony je při každé změně modelu databáze maže a nahrazuje aktuálními -> ztráta vlastního zdrojového kódu.)

### **Složka web**

Jediná část projektu přístupná z internetu. Rapidní zvýšení bezpečnosti. Co složka obsahuje vidíme na obrázku číslo 7.



Obr. 7: Složka web

Index.php a hlavni.php jsou soubory pro produkční použití. Pro testování a programování jsou soubory obsahující `_dev`. Jsou přístupné pouze z localhost. Nahrávají JavaScripty a další nástroje pro debugging. Na produkčním serveru lze tyto soubory zaměnit, dle specifického postupu a tím obejít nastavení serveru a zpřístupnit si debugging (není plnohodnotný). Což může velmi dopomoci při nasazování systému do „ostrého“ provozu. Na závěr je nutno tyto soubory smazat.

Dále zde nalezneme kaskádové styly, JavaScript (js), obrázky atd.

## 4. E-R DIAGRAMY

Celý návrh E-R diagramů viz příloha A.

### 4.1 RELACE 1:N

Relace 1:N je sdružení dvou tabulek, u kterých hodnota primárního klíče každého záznamu primární tabulky odpovídá hodnotě shodného pole nebo polí mnoho záznamů související tabulky. [3]

V tab. 1 je soupis vytvořených entit, které mají vazbu typu 1:N. U každé entity jsou uvedena popisem charakterizující uložená data.

Název entity	Popis
ADRESA	Adresy zákazníků
DAN	Číselník - hodnota daně
FAKTURA	Faktury zákazníků
KONTAKT	Záznam kontaktu dle typu kontaktu
MODUL	Modul
NEVYFAKTUROVANE_POLOZKY	Služby, které nebyly fakturovány
POLOZKA_FAKTURY	Služby, které jsou přiřazeny na fakturu
POLOZKY_SMLOUVY	Služby, které jsou přiřazené ke smlouvě
PRECHOD	Práva pro změnu stavu
ROLE	Role
SLUZBA	Nabízené služby
SMLOUVA	Smlouvy zákazníků
SMLUVNI_PODMINKY	Číselník - smluvní podmínky
STAV	Stav
TYP_KONTAKTU	Číselník - typ kontaktu
TYP_SLUZBY	Číselník - typ služby
UZIVATEL	Login
ZAKAZNIK	Zákazník
ZAMESTNANEC	Zaměstnanec

Tab. 1: Seznam entit

## 4.2 RELACE M:N

Je to spojení dvou tabulek, kde mnoho záznamů v tabulce souvisí s mnoha záznamy druhé tabulky. Tato vazba se vytváří za pomoci dvou relací 1:N a spojovací tabulkou mezi nimi. Toto uspořádání usnadňuje správu, zvyšuje pružnost a řídí se důsledně pravidly normalizace. [3]

Seznam vazebních tabulek použitý pro IS pro podporu zákazníků je s popisem, které entity spojují v tab. 2.

Název entity	Popis
ROLE_MODUL	Přiřazení modulu k roli
ROLE_PRECHOD	Přiřazení role k právu přechodu
ROLE_UZVATEL	Přiřazení role k uživateli
SLUZBA_TYP_SLUZBY	Přiřazení služby k typu služby
SMLOUVA_SM_PODMINKY	Přiřazení smluvních podmínek ke smlouvě

Tab. 2: Vazební tabulky pro relace M:N

## 5. ŽIVOTNÍ CYKLUS

Většina entit má své životní cykly. Cyklus určuje v jaké fázi se daný záznam v entitě nachází. Podle tabulky (entity) přechodu se určuje do jakého stavu se záznam může dostat a kdo pro tuto změnu má oprávnění. Určujeme tři typy stavů. Při vzniku nového záznamu v entitě je záznam v počátečním stavu a začíná „žít“ v životním cyklu. Dále se může dostat do průchozích stavů. Životní cyklus končí dosažením koncového stavu.

### 5.1 ŽIVOTNÍ CYKLUS FAKTURY

Faktura nabývá 6 stavů, z toho je jeden počáteční a dva koncové. V tab. 3 jsou jednotlivé stavy vypsány.

Stav	Název	Typ
1	Rozpracovaná	počáteční
2	Vystavená	
3	Reklamovaná	
4	Nepřijatá reklamace	
5	Zaplacená	konečný
6	Vyřazená	konečný

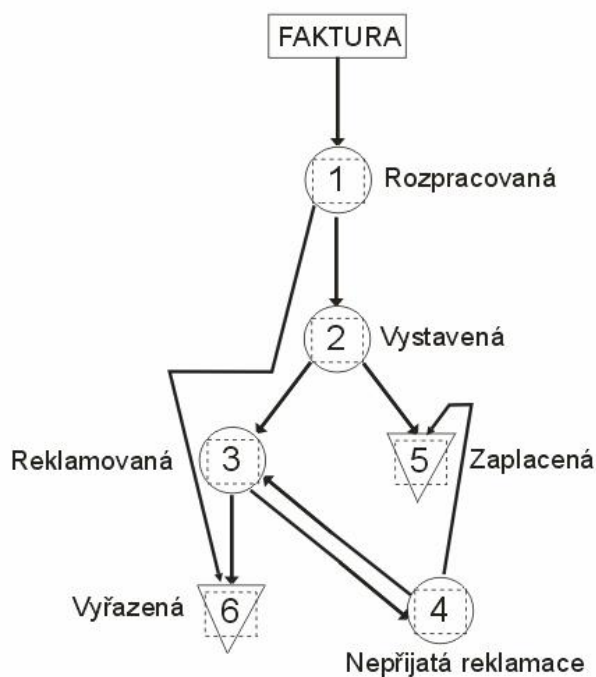
Tab. 3: Tabulka stavů faktury

V následující tabulce (tab. 4) jsou vypsány všechny možné přechody a k nim je přiřazena vždy role, která má oprávnění tento přechod uskutečnit. Každý přechod má své specifické požadavky, které musí být splněny. Např. nelze provést přechod 2-3 bez uvedení důvodu reklamace.

z	do	Role	Podmínky změny
1	2	Ucetni sluzba, Ucetni smluva	Vyplnění nezbytných údajů k faktuře
1	6	Ucetni smluva	Důvod vyřazení
2	3	Ucetni reklamace	Důvod reklamace
2	5	Ucetni zaplacení	Datum zaplacení
3	4	Ucetni rek. vyřízení	Datum, vyjádření
3	6	Ucetni rek. vyřízení	Datum, vyjádření, důvod vyřazení
4	3	Ucetni reklamace	Důvod reklamace
4	5	Ucetni zaplacení	Datum zaplacení

Tab. 4: Tabulka přechodů mezi stavy faktury, jejich oprávnění a podmínky přechodu.

Grafické vyjádření životního cyklu je velmi přehledné, ale postrádá některé informace obsažené v tabulce:



Obr. 8: Životní cyklus faktury

Faktura začíná svůj životní cyklus ve stavu rozpracovaná. Rozpracovanou fakturu lze vystavit po splnění náležitých podmínek, které jsou dány zákonem. Při chybě účetní vyplní důvod vyřazení a fakturu může vyřadit. Vystavená faktura v nejlepším případě může být zaplacená a dostává se do koncového stavu. Je zde samozřejmě počítáno i s tím, že příjemce faktury bude nespokojen a může ji i opakovaně reklamovat. Při oprávněné reklamaci se faktura vyřadí v opačném případě se požaduje opětovné zaplacení.

## 5.2 ŽIVOTNÍ CYKLUS SMLOUVY

V následující tabulce (tab. 5) jsou sepsané jednotlivé stavy smlouvy. Obdobně jak faktura tak i smlouva má své přechody, které může provádět jen oprávněná role (tab. 6).

Stav	Název	Typ
1	Rozpracovaná	počáteční
2	Nabídnutá	
3	Uzavřená	
4	Výpověď	
5	Zrušena	konečný
6	Ukončená	konečný
7	Vypovězená	konečný

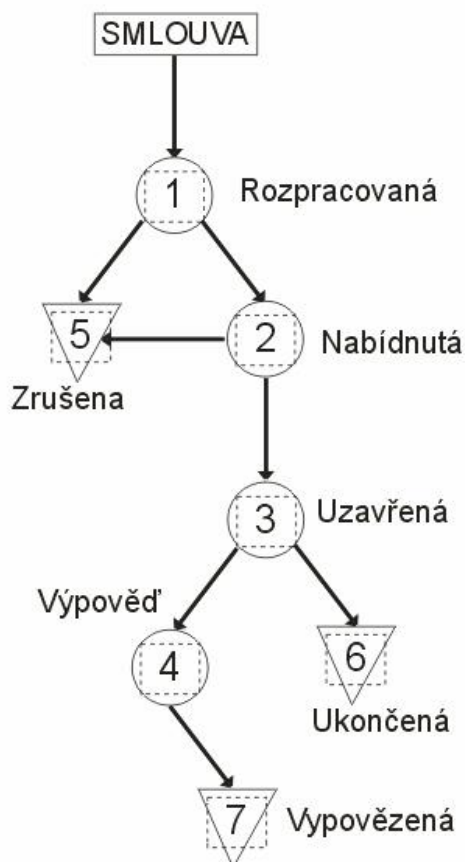
Tab. 5: Tabulka stavů smlouvy

Smlouvu vytváří role *Zamastnanec smlouva*. Při správném sestavení celé smlouvy ji posouvá do stavu nabídnutá, nebo ji může zrušit. Při fyzickém podepsání smlouvy zákazníkem se stává smlouva uzavřená a začíná běžet dle smlouvané doby. V opačném případě je zrušena. Při vystavení poslední faktury (to znamená, že smlouva došla v platném smluveném období) je smlouva ukončena. Zákazník kdykoliv v průběhu plnění smlouvy může smlouvu vypovědět (dostává se do stavu

4). Dle smluvních podmínek mu bude vystavena pokuta. Vystavením pokuty se smlouva stává vypovězená (stav 7).

z	do	Role	Podmínky změny
1	2	Zamestnanec smluva	Datum, periodicita, adresa
1	5	Zamestnanec smluva	Důvod zrušení
2	3	Zamestnanec smluva	Fyzické podepsání
2	5	Zamestnanec smluva	Důvod zrušení
3	4	Zamestnanec smluva	Osobní žádost zákazníka
3	6	Ucetní smlouva	Vystavena poslední faktura
4	7	Ucetní smlouva	Vystavena faktura s pokutou

Tab. 6: Tabulka přechodů mezi stavy smlouvy, jejich oprávnění a podmínky přechodu.



Obr. 9: Životní cyklus smlouvy

### 5.3 ŽIVOTNÍ CYKLUS SLUŽBY

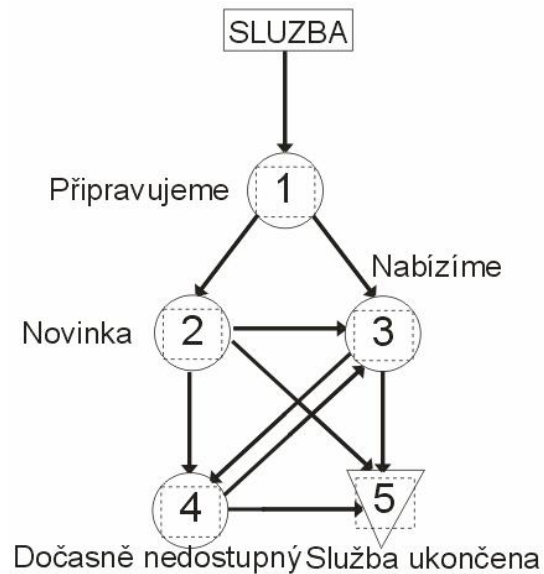
Z obchodního pohledu jsou služby velmi důležité. Kdyby nebylo služeb, nebylo by z čeho vytvářet faktury. Ale z pohledu databáze jsou stavy služby spíše informativní. Podstatnější jsou stavy 4 a 5, v tomto stavu se služba nemůže nabízet. Kompletní rozpis je v tab. 7 a tab. 8.

Stav	Název	Typ
1	Připravujeme	počáteční
2	Novinka	
3	Nabízíme	
4	Dočasně nedostupný	
5	Služba ukončena	konečný

Tab. 7: Tabulka stavů služby

Z	do	role
1	2	Manager sluzba
1	3	Manager sluzba
2	3	Manager sluzba
2	4	Manager sluzba
2	5	Manager sluzba
3	4	Manager sluzba
3	5	Manager sluzba
4	3	Manager sluzba
4	5	Manager sluzba

Tab. 8: Tabulka přechodů mezi stavy služby



Obr. 10: Životní cyklus služby

Do stavů připravujeme a novinka se záznam v entitě může dostat pouze jednou. Mezi stavy nabízíme (stav 3) a dočasně nedostupný (stav 4) se lze libovolně posouvat. Při dosažení stavu 5 (služba ukončena) celý životní cyklus končí.

#### 5.4 ŽIVOTNÍ CYKLUS UŽIVATELE

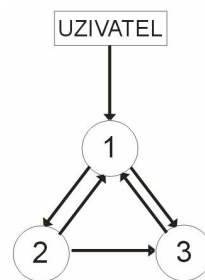
Uživatel může nabývat 3 stavů (obr. 9). Do stavu blokován se může dostat při zadávání špatného hesla při přihlašování (opakovaně), nebo při podezření na zneužití přístupových práv ho může blokovat role *admin ucty*.

Stav	Název	Typ
1	Aktivní	počáteční
2	Blokován	
3	Neaktivní	

Tab. 9: Tabulka stavů uživatele

z	do	role	podmínky změny
1	2	Admin ucty, neregistrovany	opakovaně špatně heslo, podezření na zneužití
1	3	Admin ucty	údržba
2	1	Admin ucty	změna hesla
2	3	Admin ucty	údržba
3	1	Admin ucty	změna hesla

Tab. 10: Tabulka přechodů mezi stavy uživatele, jejich oprávnění a podmínky přechodu



Obr. 11: Životní cyklus uživatele

## 5.5 ŽIVOTNÍ CYKLUS ZÁKAZNÍKA

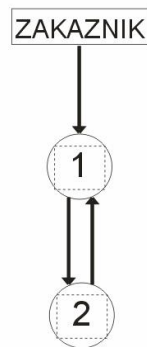
Zákazník může být aktivní nebo neaktivní.

Stav	Název	Typ
1	Aktivní	počáteční
2	Neaktivní	

Tab. 11: Tabulka stavů zákazníka

z	do	role	podmínky změny
1	2	Zamestnanec sluzba Zamestnanec smouva	Zákazník rozzával spolupráci
2	1	Zamestnanec sluzba Zamestnanec smouva	zákazník naváže spolupráci

Tab. 12: Tabulka přechodů mezi stavy zákazníka, jejich oprávnění a podmínky přechodu



Obr. 12: Životní cyklus zákazníka

## 5.6 ŽIVOTNÍ CYKLUS OSTATNÍCH ENTIT

Tímto jsou sepsané všechny entity se složitějším životním cyklem. Ostatní entity mají pouze dva stavy (aktivní/ neaktivní). Výjimku by mohla tvořit entita smluvní podmínky, zde se předpokládá psaní delšího textu a tím i přidání stavu rozpracovaná.

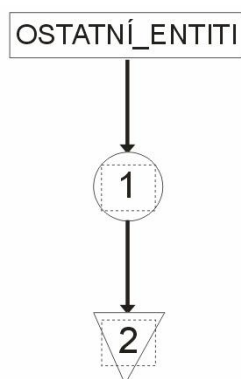
Stav	Název	Typ
1	Aktivní	počáteční
2	Neaktivní	koncový

Tab. 13: Tabulka stavů ostatních entit

V tab. 14 je seznam ostatní entit a k nim jsou přiřazeny příslušné role, které mají právo tento stav měnit. Stav 1 je počáteční stav. Stav 2 je koncový stav.

název entity	role
ADRESA	Zamestnanec sluzba Zamestnanec smluva Zamestnanec login
DAN	Ucetni dan
KONTAKT	Zamestnanec sluzba Zamestnanec smluva Zamestnanec login
NEVYFAKTUROVANE_POLOZKY	Zamestnanec uprav sl.
POLOZKY_SMLOUVY	Zamestnanec smluva
SMLUVNI_PODMINKY	Manager sm. podminky
TYP_KONTAKTU	Manager typ kontaktu
TYP_SLUZBY	Manager typ sluzby

Tab. 14: Tabulka entit a oprávněných rolí měnit stav



Obr. 13: Životní cyklus entit z tab. 14

## 6. ROLE

Každý uživatel pro práci s databází musí mít přidělenou roli. Podle přidělené role se zpřístupní část aplikace stahující se k této roli. Každý uživatel může mít více rolí. Seznam rolí a stručný popis procesů pro danou roli je v tab. 13. Podrobný popis procesů pro jednotlivé role viz. kapitola 8.

Role	Popis
Admin ucty	Založení, úprava zaměstnance, nastavení práv (role)
Zamestnanec sluzba	Správa zákazníka, objednávka služeb
Zamestnanec smlouva	Správa zákazníka, vytváření, úprava smluv
Zaměstnanec login	Povolení zákazníkům do systému
Ucetni dan	Správa daně
Ucetni sluzba	Fakturace služeb
Ucetni smluva	Hromadné vytváření faktur ze smluv
Ucetni reklamace	Přijímání reklamací
Ucetni rek. vyřízení	Vyřizování reklamací
Ucetni zaplacení	Potvrzuje zaplacené faktury
Manager sluzba	Vytváření, úprava služeb
Manager sm. podmínky	Vytváření, úprava smluvních podmínek
Manager typ kontaktu	Vytváření, úprava typu kontaktu
Manager typ služby	Vytváření, úprava kategorie služby
Zaměstnanec uprav sl	Opravy chybně přiřazených služeb
Zakaznik	Zobrazení informací o zákazníkovi

Tab. 15: Tabulka rolí v systému

## 7. MODULY

Modul je skupina formulářů, které spolu logicky souvisí. Jednotlivé formuláře spravují společná data. V tab. 17 je seznam použitých modulů pro rozhraní admin a k nim jsou přiřazeny role, které s tímto modulem pracují a tab. 16 pro rozhraní frontend.

Při programování nebyla důsledně dodržena souborová struktura symfony. Do složky Faktura byly naprogramovány tři moduly. Tato chyba se později ukázala za dost podstatnou. Proto v teoretické části je struktura symfony podrobněji popsána z důvodu předejetí obdobných chyb u čtenáře. Z pohledu symfony to byl jeden modul. Programově tyto tři „moduly“ byly odděleny, ale pro provázanost společnými prvky už bylo problematické je oddělit.

Problém byl vyřešen přejmenováním všech modulů a rozdělením (pojmenováním) modulu Faktura. Modul Menu tedy nepoužívá názvy modulů, jak jsou definovány v symfony, ale používá přejmenované. Přejmenováno bylo tak, aby se tento nový název dal použít přímo do popisku tlačítka odkazujícího se na tento modul. Případně byl jenom název modulu zkrácen viz tab. 16 a 17 sloupeček česky/zkratka.

Modul	Česky / zkratka	Role
Info	info	nepřihlášen
Menu	menu	Nepřihlášen Zákazník
Sluzba	služba	nepřihlášen
Zakzanik	zobrazit	Zákazník
prihlaseni	prihlaseni	nepřihlášen všechny role

Tab. 16: Tabulka modulů aplikace frontend a role k nim přistupující

Modul	Česky / zkratka	Role
Adresa	adresa	Zamestnanec sluzba Zamestnanec smlouva Zamestnanec login
Dan	daň	Ucetni dan
Faktura	faktura	Ucetni sluzba Ucetni reklamace Ucetni rek.vyrizeni
Faktura	fakturace služeb	Ucetni sluzba
Faktura	fvy	Ucetni sluzba
FakturaR	fre	Ucetni reklamace
FakturaRV	frv	Ucetni rek. vyrizeni
FakturaSmlouva	fakturace smluv	Ucetni smluva
FakturaZ	fza	Ucetni zaplacení
FakturaZobraz	zobrazit fakturu	Zamestnanec sluzba Zamestnanec smlouva Zamestnanec login
Info	Info	všechny role
Kontakt	kontakt	Zamestnanec sluzba Zamestnanec smlouva Zamestnanec login
Menu	menu	všechny role
NevyfakturovanePolozky	zsl	Zamestnanec sluzba
NevyfakturovanePolozky	zup	Zaměstnanec uprav sl
PolozkySmlouvy	ps	Zamestnanec smlouva
Popisek	popisek	všechny role
Sluzba	služba	Manager sluzba
Smlouva	smlouvy	Zamestnanec smlouva
SmluvniPodminky	smluvní podmínky	Manager sm. podminky
TypKontaktu	typ kontaktu	Manager typ kontaktu
TypSluzby	typ služby	Manager typ sluzby
UzivZak	zlo	Zamestnanec login
Uzivatel	uzivatel	Admin ucty
Zakaznik	zakaznik	Zamestnanec sluzba Zamestnanec smlouva Zamestnanec login
Zamestnanec	zaměstnanec	Admin ucty
prihlaseni	prihlaseni	Nepřihlášen všechny role

Tab. 17: Tabulka modulů aplikace admin a role k nim přístupující

## 8. PROCESY

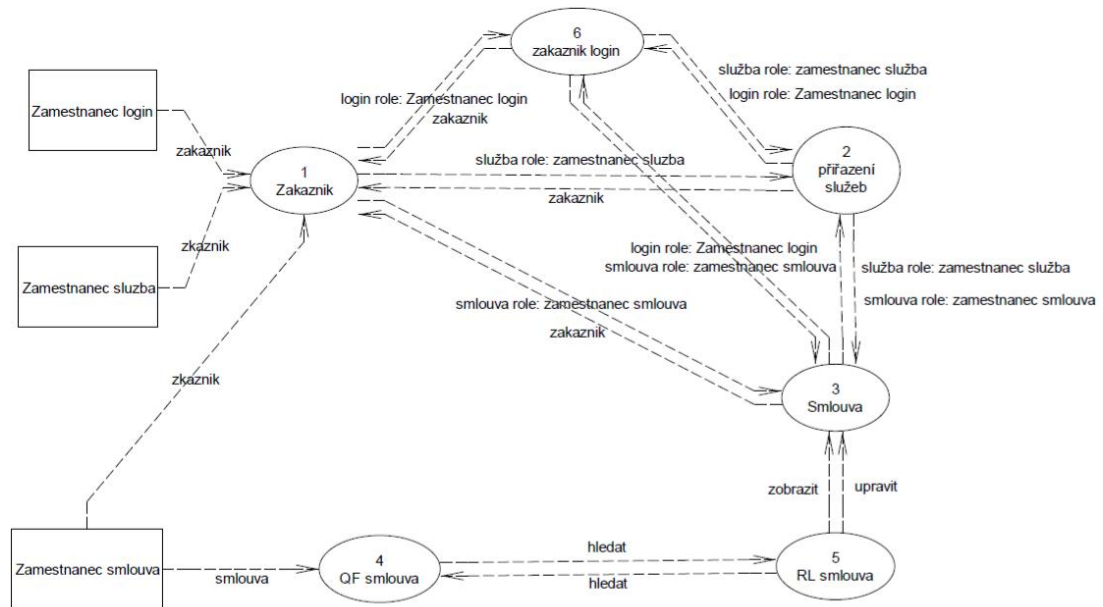
Systém je navržen tak, že pro každý proces je vytvořen modul. Mezi logicky navazujícími moduly lze přecházet bez navigace z menu.

Byly využity tyto formuláře:

- Query Form (QF) – zadávání filtru pro zobrazení filtrovaného RL
- Rekord List (RL) – zobrazení seznamu požadovaných dat
- View Form (VF) – zobrazení dat (detailní zobrazení záznamu)
- Edit Form (EF) – úprava dat (detailní editace záznamu)
- Insert Form (IF) – vytváření nových záznamů v databázi
- MROW – zobrazení jak u RL, ale s možností vývěru více dat (více řádků)
- MROW1 – zobrazení jak u RL, výběr jeden z mnoha (option)

### 8.1 PROCESY PRÁCE SE ZÁKAZNÍKEM

Tyto procesy má oprávnění spravovat role: *zamestnanec sluzba*, *zamestnanec smlouva*, *zamestnanec login* a *zaměstnanec uprav sl*. Je sestaven z 8 modulů: *zákazník*, *adresa*, *kontakt*, *zobrazit fakturu*, *zsl*, *zsm*, *zlo* a *zup*. Moduly *zákazník*, *adresa*, *kontakt*, *zobrazit fakturu* jsou společné pro všechny role (kromě role *zaměstnanec uprav sl*.) a následně pro každou roli je naprogramován jeden modul popsany níže. Procesy a jejich návaznosti jsou zakresleny na obr. 14.



Obr. 14: Procesy práce se zákazníkem

### 8.1.1 Proces zákazník

Do tohoto procesu vstupují role *zaměstnanec služba*, *zaměstnanec smlouva* a *zaměstnanec login*. Je to vstupní proces pro ostatní procesy. Zde se přes vstupní filtr dostaneme do seznamu zákazníků. Po výběru vyfiltrovaného zákazníka můžeme upravovat jméno, firemní údaje, kontakty, adresy, zobrazit jeho faktury, případně dle práv přejít do procesu pro přidávání žádané služby, vytváření smlouvy či založení login a hesla.

Kontakt může být například telefon, e-mail, ICQ či jiný, který předdefinuje role *manager typ kontaktu* ve svém procesu.

Firma nemusí mít jenom jednu adresu, ale může mít více poboček po celé republice. I běžný zákazník může mít trvalé a přechodné bydliště. Proto je databáze navržena pro téměř neomezené množství adres (omezeno technickými možnostmi HW). Program sám hlídá, aby vždy zůstala alespoň jedna adresa, kterou nedovolí odstranit.

Adresy a kontakty přes seznam se dají upravovat, nebo vytvářet nové. Faktury lze pouze prohlížet.

Vytváření nového zákazníka je provedeno ve 2 krocích. V prvním kroku se vyplní iniciály zákazníka a ve druhém kroku jeho adresa. Bez adresy se nový zákazník nevytvoří.

Celý proces je zobrazen na obr. 15, pro přehlednost nebyly zakresleny řídicí toky směřující na vstupní filtr (QF zákazník 1.1). Pro celkovou provázanost by se celý obrázek stal nepřehledný. Řídicí toky z ostatních procesů jsou naznačeny v „bublině“ číslo 1.16.



Obr. 15: Proces zákazník

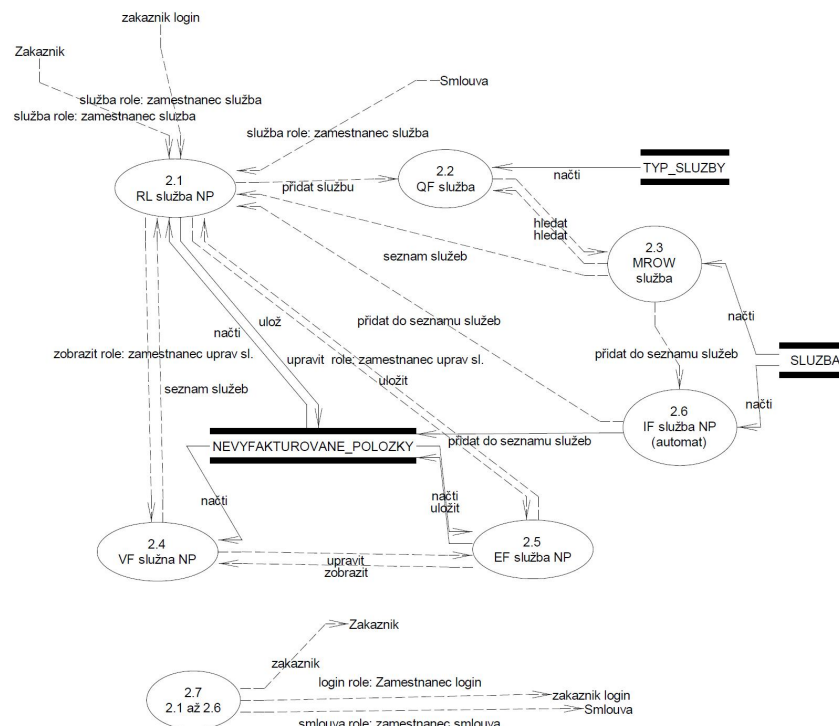
### 8.1.2 Procesy přiřazení služeb

Role *zamestnanac sluzba* (modul: *zsl*) začíná svůj proces na seznamu služeb NP (služby, které zákazník využil, nebo využívá a ještě pro ně nebyla sestavena faktura). Pro přidání další služby je záložka *přidat službu*. Pro snadnější orientaci je předřazen filtr a v následném okně se zaškrtnou vybrané služby, kde lze přímo i zadat množství odebrané služby. Při potvrzení se služby přesunou do seznamu (služba NP). V tomto seznamu je dovoleno množství navýšit, ale je zakázáno hodnotu snižovat. Jedním z požadavků pro sestavení této databáze je hlídat zaměstnance před podvodem (zaměstnanec se domluví se zákazníkem a před vystavením faktury množství sníží nebo smaže – tomuto je zabráněno).

Role *zamestnanec uprav sl.* (modul: *zup*) služby v seznamu *služba NP* má právo, přes příslušný formulář, množství snižovat, nebo službu úplně odstranit (proces *upravit přiřazenou službu*). Dále má možnost udělit slevu.

V těchto procesech je zaznamenáváno, kdy a kdo provedl poslední úpravu.

Procesy znázorněny na obr. 16. Pro přehlednost řídicí toky do ostatních procesů naznačeny na 2.7.



Obr. 16: Procesy přiřazení služeb

### 8.1.3 Proces smlouva

Tento proces (obr. 17 role *zaměstnanec smlouva* modul: *smlouvy*), při popisu je rozdělen na dvě části.

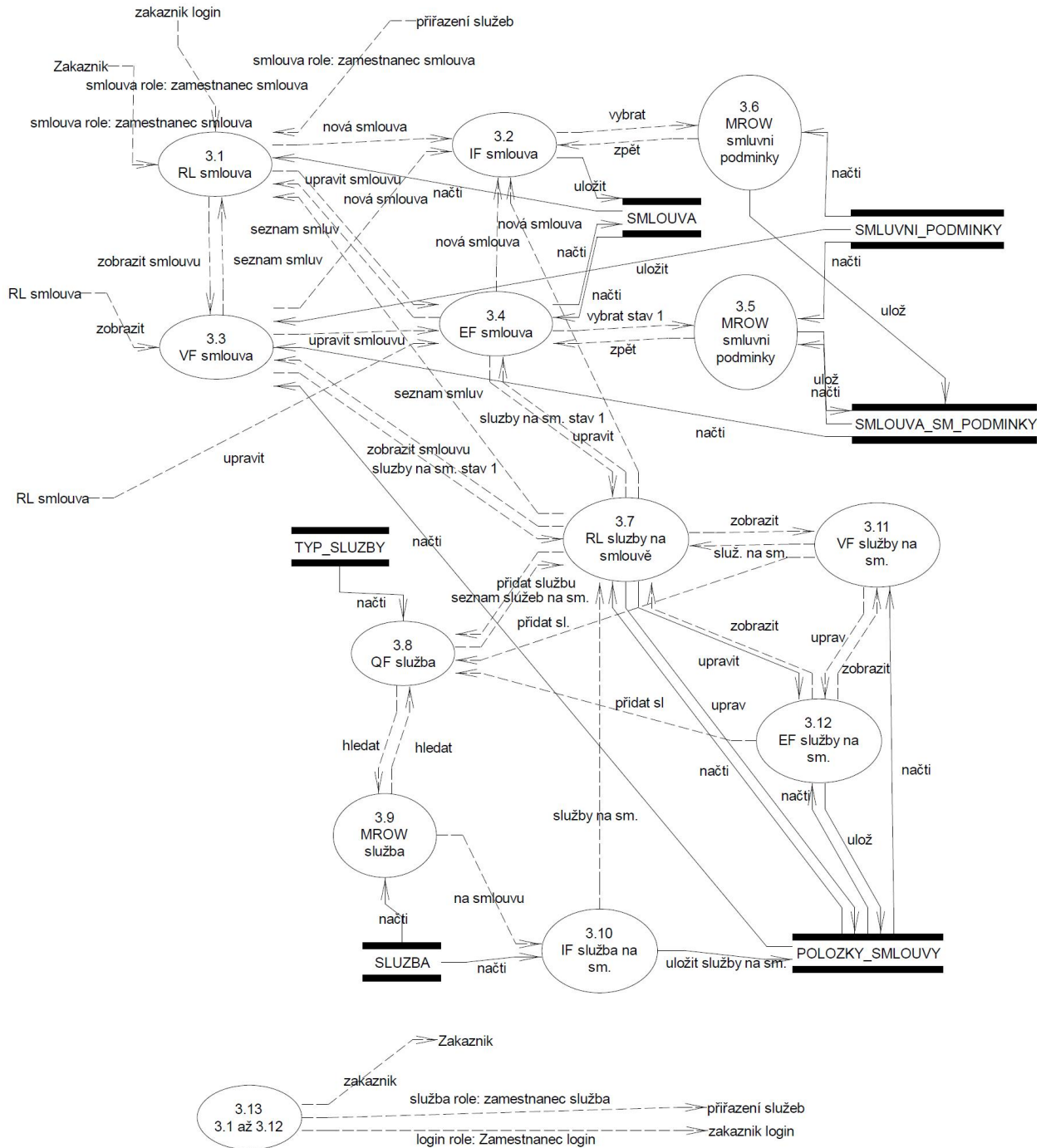
První část je samotné vytvoření smlouvy. Na formulář pro vytvoření smlouvy se lze dostat pouze z procesu *zákazníka* (prvně se musí najít zákazník, pro kterého se smlouva bude vytvářet). Pro vyhledávání a úpravu již vytvořených smluv se do tohoto procesu přistupuje přes navigační tlačítko v menu.

V detailu smlouvy se přiřazují smluvní podmínky (tj. obecně právní předpisy popisující sankce při neplnění smluvních závazků, postup a sankce při poškození cizí věci atd.). Smluvní podmínky sepisuje role *manager sm. podmínky*. Dále se zde musí zvolit ze seznamu adres fakturační adresa (velmi důležité pro automatickou fakturaci), sepisuje se datum začátku smlouvy a smluvní období (kolik měsíců smlouva trvá), případně další smluvní ujednání.

V druhé části se přiřazují samotné služby podléhající této smlouvě. Pro snadné nalezení služeb je připraven filtr (obdobně jako v procesu *přiřazení služeb*).

Po vytvoření smlouvy (změna stavu) proces povoluje pouze čtení (nikoliv úpravu), případně změnu stavu (dle životního cyklu) s popisem důvodu změny.

U smluv není evidováno, kdo přiřazoval jednotlivé položky do smlouvy, ale pouze kdo vytvářel smlouvu. Smlouvy se musí fyzicky podepisovat a tím i vznikne zákazníkem kontrola, zda je na smlouvě vše jak si přál. Před podepsáním je smlouva ve fázi přípravy a nevzniká žádná pohledávka jako u procesu *přiřazení služeb*.



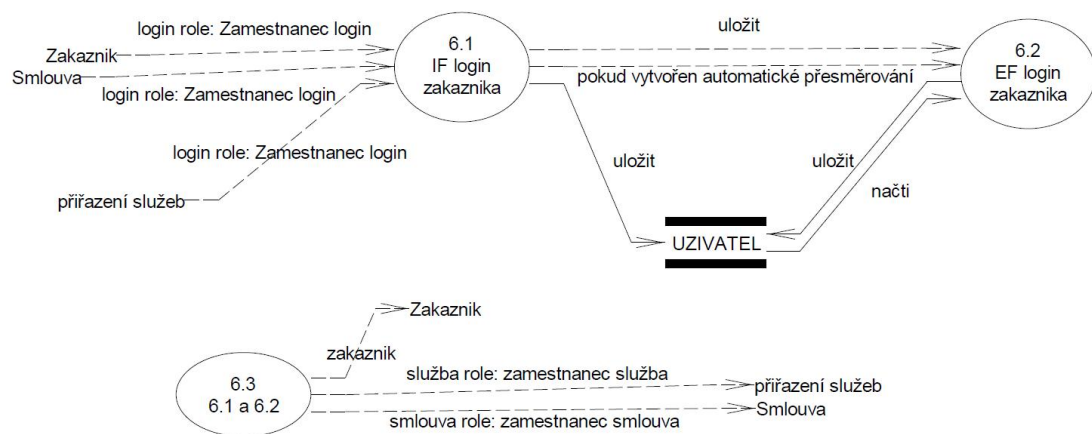
Obr. 17: Proces smlouva

### 8.1.4 Proces zákazník – login

Povolení zákaznickovy přístup do systému vytváří role *zamestnanec login*. Z procesu *zákazník*, přistupuje do procesu *zákazník – login*. Systém sám rozpozná, zda zákazník již přístup má povolen, nebo je třeba ho vytvořit a podle toho nabídne příslušný formulář.

Při vytváření se vygeneruje login podle následujícího předpisu: příjmení + jméno + (nuly doplňující na délku řetězce o velikosti 6) + dvojciferné číslo. Dvojciferné číslo udává kolik login již je zageristrováno se shodnými počátečními 6 znaky zvětšeno o jedna.

Je-li přístup vytvořen, nabízí se dvě možnosti: změnit heslo, nebo přístup zablokovat.



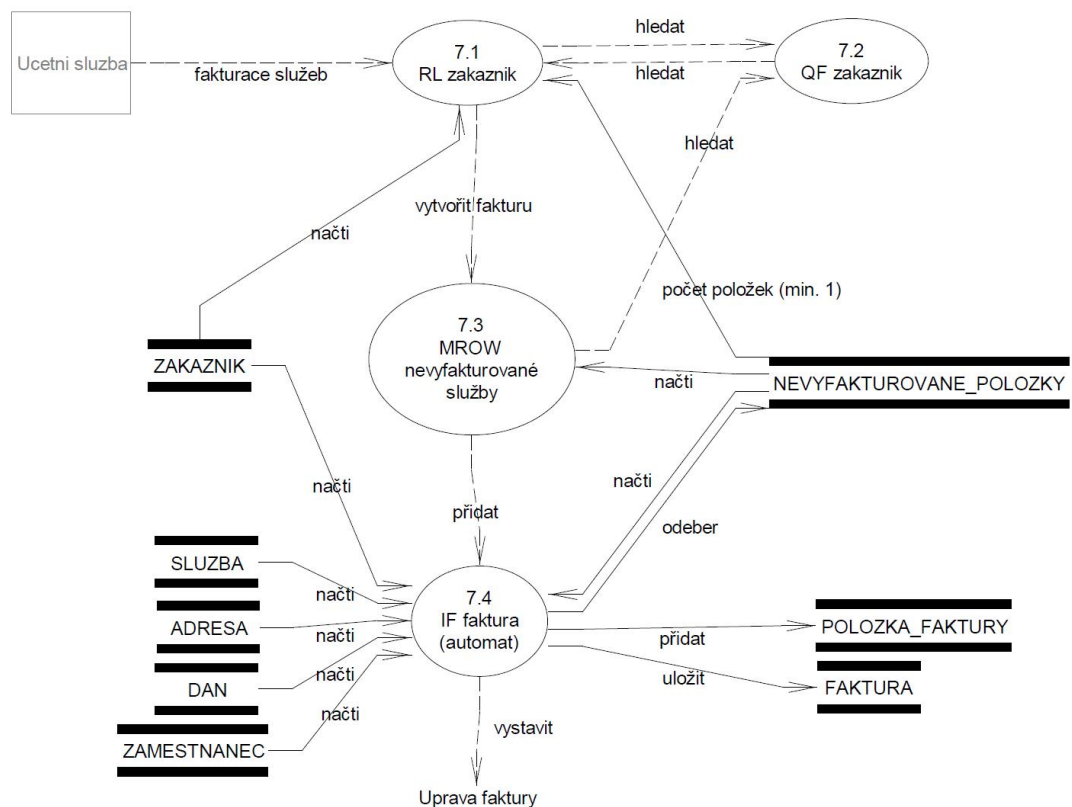
Obr. 18: Proces zákazník – login



### 8.2.1 Fakturace služeb

Proces pro vytváření faktur má na starosti role *Učerní služba* (modul: *fakturace služeb*). Tento proces (obr. 20) začíná v seznamu zákazníků, kteří mají jednu nebo více nevystavených položek. Přepnutím do filtru, lze tento seznam dodatečně filtrovat.

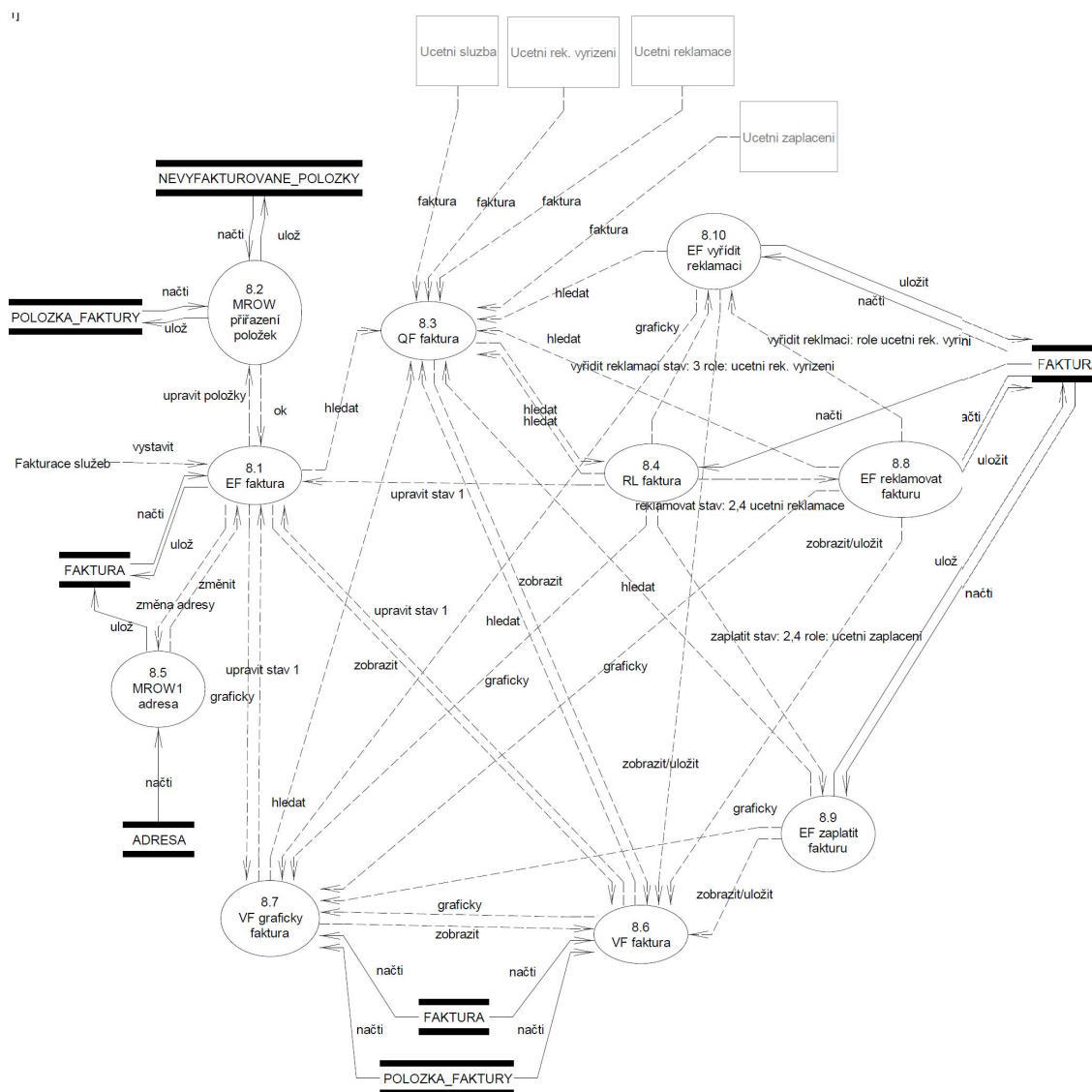
Následuje samotné vytvoření faktury. V seznamu MROW (7.3 z obr. 20) se vyberou požadované položky (minimálně jedna). Při potvrzení systém sám načte iniciály zákazníka, přiřadí první adresu a ke všem položkám přiřadí odpovídající daň. Tímto je vytvořena faktura ve stavu 1 (rozpracovaná). Dále následují procesy *uprava faktury* (kontrola, vystavení, úprava... viz. kapitola 8.2.2).



Obr. 20: Proces fakturace služeb

## 8.2.2 Úprava faktury

Pro úpravu faktury byly realizovány čtyři procesy: *vystavení, reklamace, vyřízení rek. a zaplacení*. Každý proces má na starosti jiná role. Úprava faktury se skládá z 5 modulů, kde modul *faktura* je společný pro všechny role. Na obr. 21 jsou znázorněny datové a řídicí toky pro úpravu celé faktury. U řídicích toků jsou poznámky za jakých podmínek se spustí jaký proces (např.: vyřídit reklamaci: stav 3 role: učetní rek. vyřízení).



Obr. 21: Procesy pro úpravu faktury

### 8.2.2.1 Vystavení faktury

Na proces *fakturace služeb* navazuje proces *vystavení faktury*. Proces spravuje stejná role: *ucerni sluzba* (modul: *fvy* a *faktura*). Proces má na starosti případnou změnu adresy nebo dodatečnou úpravu položek na faktuře. Úpravou se rozumí přesun do *POLOZKY\_FAKTURY* (tedy na fakturu) nebo zpět do *NEVYFAKTUROVANE\_POLOZKY*. Při správném vyplnění se faktura tlačítkem *vystavit* vystaví (změna stavu 1-2), případně vyřadí (záložka *vyřadit* současně s vyplněným důvodem vyřazení). Tímto proces končí a už se do něj faktura nikdy nedostane.

Do procesu (stav 1 – rozpracovaná) lze také dodatečně přistoupit přes navigační tlačítko *faktura*: filtr -> seznam -> zobrazit -> vystavit.

### 8.2.2.2 Reklamace faktury

Proces *reklamace faktury* (role *ucetni\_Reklamce*) začíná vstupním filtrem (navigace v menu: *faktura*), vhodným vyfiltrováním nalezne fakturu, kterou je třeba reklamovat. Zvolením tlačítka *reklamovat* se objeví příslušný formulář, kde se vyplní důvod reklamace. Na pozadí se doplní datum reklamace a jméno zaměstnance, který reklamaci přijal. Faktura se zobrazí jako reklamovaná a tímto proces končí a může začít nový.

### 8.2.2.3 Vyřízení rek. faktury

V tomto procesu se zpracovávají reklamace. Role: *ucetni rek. vyizeni*. Postupuje se obdobně jako při procesu *reklamace faktury*.

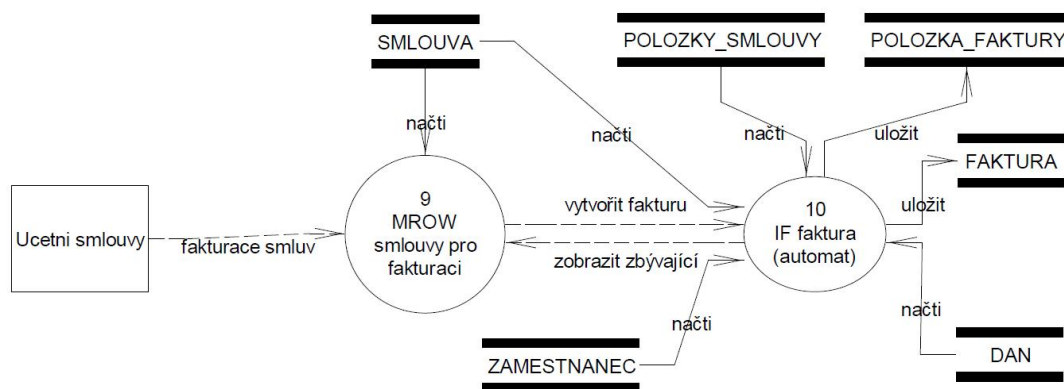
Vhodným vyfiltrováním (pouze zaškrtnout reklamovaná a zobrazí se všechny faktury pro tento proces) nalezne faktury pro vyřízení reklamace. Zvolením tlačítka *vyřídít reklamaci* se objeví příslušný formulář. Provede se záznam o způsobu vyřízení reklamace a podle toho se naloží s fakturou (vyřazení, opětovné požadování zaplacení).

### 8.2.2.4 Zaplacení faktury

Pro roli *Učerní zaplacení* byl navržen proces s názvem *zaplacení faktury*. Postup je opět obdobný jako předchozí: filtr -> seznam-> zaplatit. Ve formuláři se vyplní číslo účtu úhrady a uložením se doplní datum zaplacení. Tímto je faktura zaplacená. Tento proces lze v budoucnu dále rozšířit o import z bankovního systému.

### 8.2.3 Hromadné generování faktur ze smluv

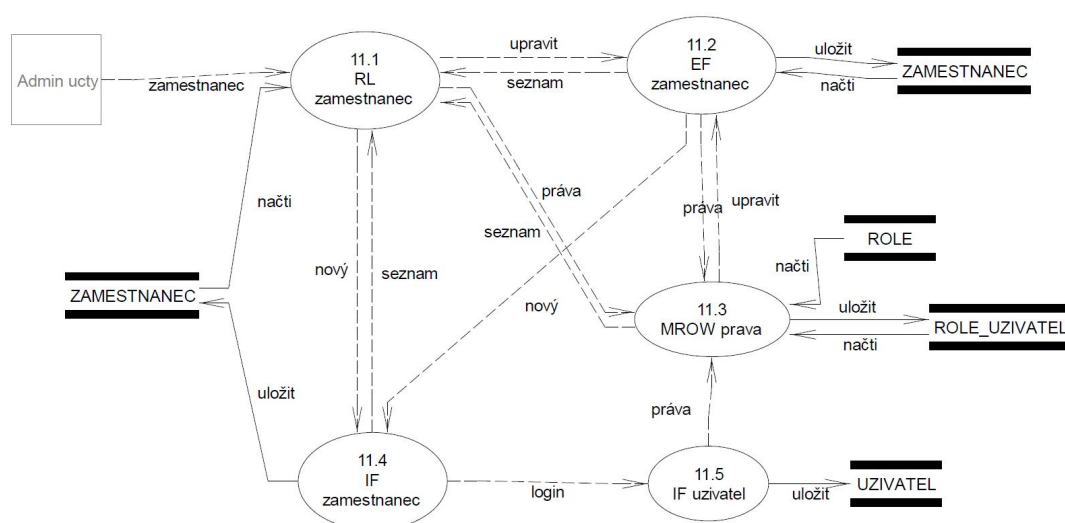
Proces spouští role *Ucetni\_Smlouvy*. Při spuštění tohoto procesu pro automatické generování faktur ze smluv se načnou všechny nevyfakturované smlouvy. Zde lze vybrat jen některé nebo označit všechny. Kontrola, aby nebyly vystaveny za stejné fakturační období dvě totožné faktury, je realizována pomocí uložení posledního data fakturace. Fakturuje se zpětně. Při stisknutí tlačítka *vytvořit fakturu* se načtou příslušné data z příslušných *data store* a vygeneruje se faktura ve stavu 2 (vystavená). Celý proces je zobrazen na obr. 22.



Obr. 22: Proces pro hromadné generování faktur ze smluv

### 8.3 PROCES SPRÁVA ZAMĚSTNANCŮ

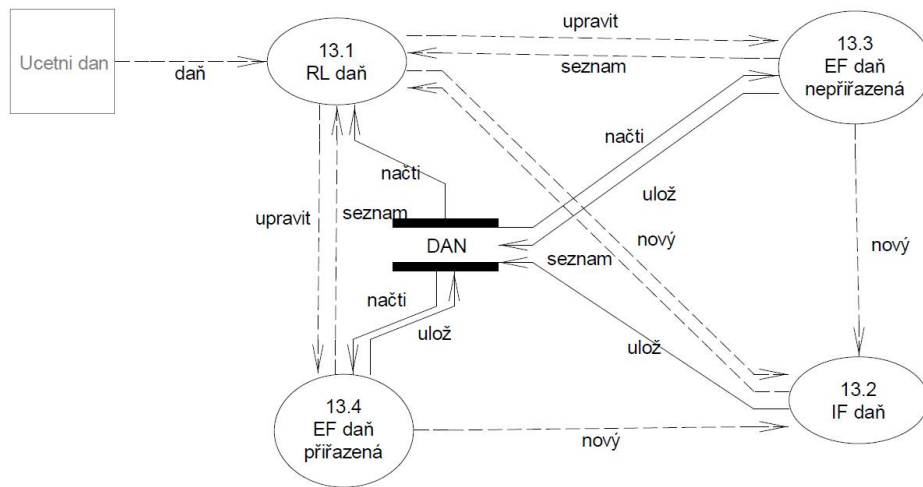
V tomto procesu role *admin ucly* vytváří a upravuje zaměstnance. Přiděluje jim jednotlivá práva (role), které určují co v systému mohou dělat. Při propuštění (stav propuštěn) se automaticky zruší všechna přístupová práva. Při změně práv je nutné se pro úplné načtení modulů odhlásit a opět přihlásit. Celý proces je znázorněn na obr. 23.



Obr. 23: Proces správa zaměstnanců

### 8.4 PROCES SPRÁVA DANĚ

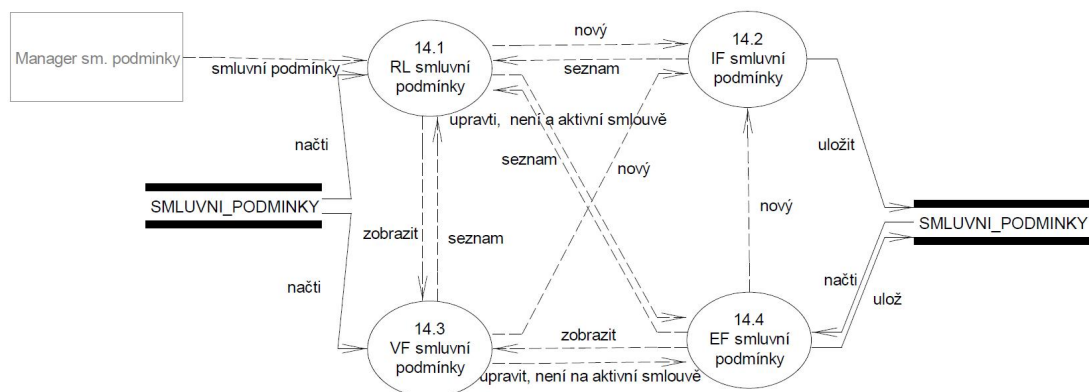
V tomto procesu role *ucetni dan* vytváří a upravuje číselník daň. Vystup do procesu je přes hlavní navigační tlačítko v menu. Začátek procesu je v seznamu daní. Lze vytvořit novou daň. Nebo stávající upravit. Daň nelze upravovat, pokud je přiřazená k některé službě (13.4 obr.24). Lze ji jen zakázat, aby se již nedala nově použít.



Obr. 24: Proces správa daně

## 8.5 PROCES SMLUVNÍ PODMÍNKY

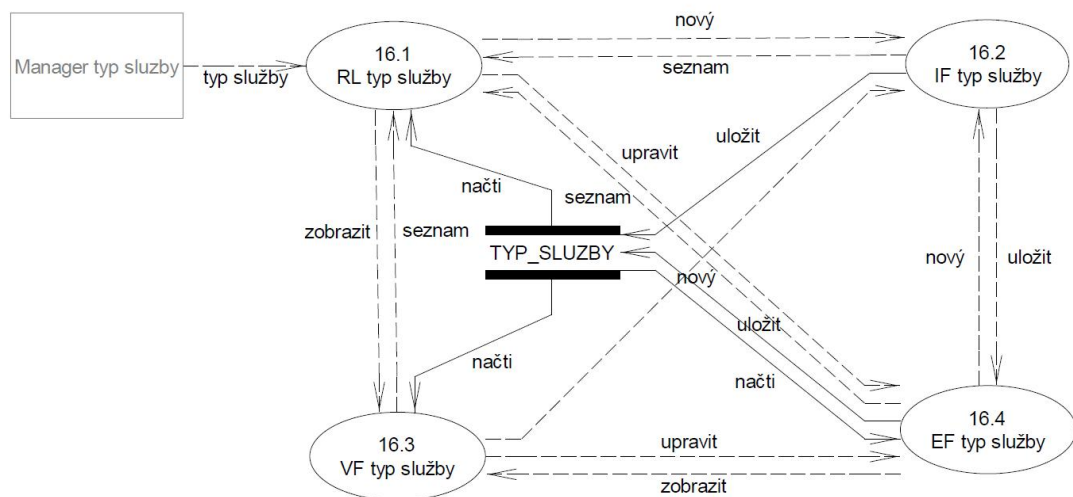
Proces je znázorněn na obr. 25. Oprávněná role je *manager sm. podmínky*. Proces dovoluje se volně pohybovat mezi seznamem, zobrazením a vytvořením nové smluvní podmínky. Povoluje upravovat smluvní podmínky jen takové, které nejsou použity na běžící smlouvě. Zakazuje jakoukoliv změnu smluvní podmínky, pokud je smluvní podmínka použita u běžící smlouvy.



Obr. 25: Proces smluvní podmínky

## 8.6 PROCES TYPŮ SLUŽEB

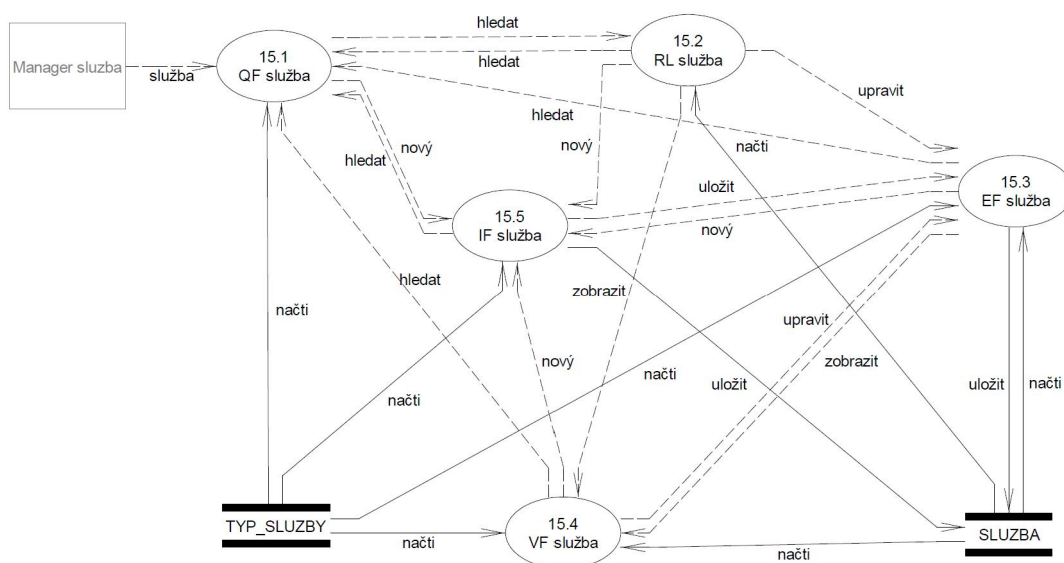
Oprávněná role je *manager typ služby*. Do procesu se přistupuje přes navigační tlačítko z hlavního menu. V procesu je dovoleno se volně pohybovat mezi seznamem, zobrazením a vytvořením nového typu služby. Úprava je omezena takto: typ služby lze upravovat, ale nelze zrušit pokud je přiřazena minimálně jedna služba.



Obr. 26: Proces typů služeb

## 8.7 PROCES SPRÁVA SLUŽEB

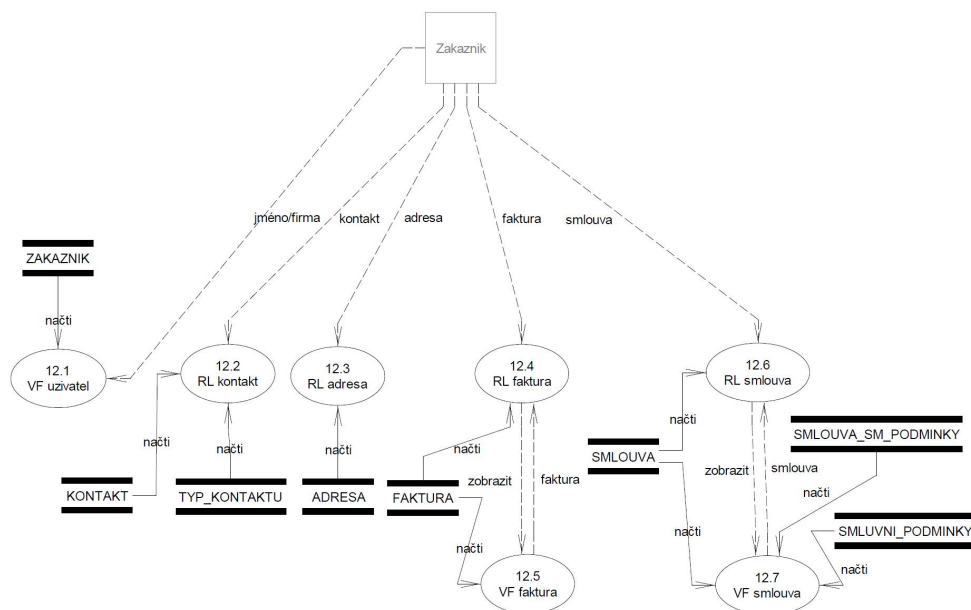
Do procesu se vstupuje přes hlavní navigační tlačítko v menu. Oprávněná role tohoto procesu je *Manager služba*. Začíná se vstupním filtrem. Následuje seznam služeb. Odtud za pomoci záložek se můžeme přesunout zpět na filtr, vytvořit novou službu, prohlédnout si službu, nebo ji upravit. Každá služba musí mít minimálně jeden typ služby. Celý proces je kompletně zakreslen na obr. 27.



Obr. 27: Proces správa služeb

## 8.8 REPORT ZOBRAZENÍ

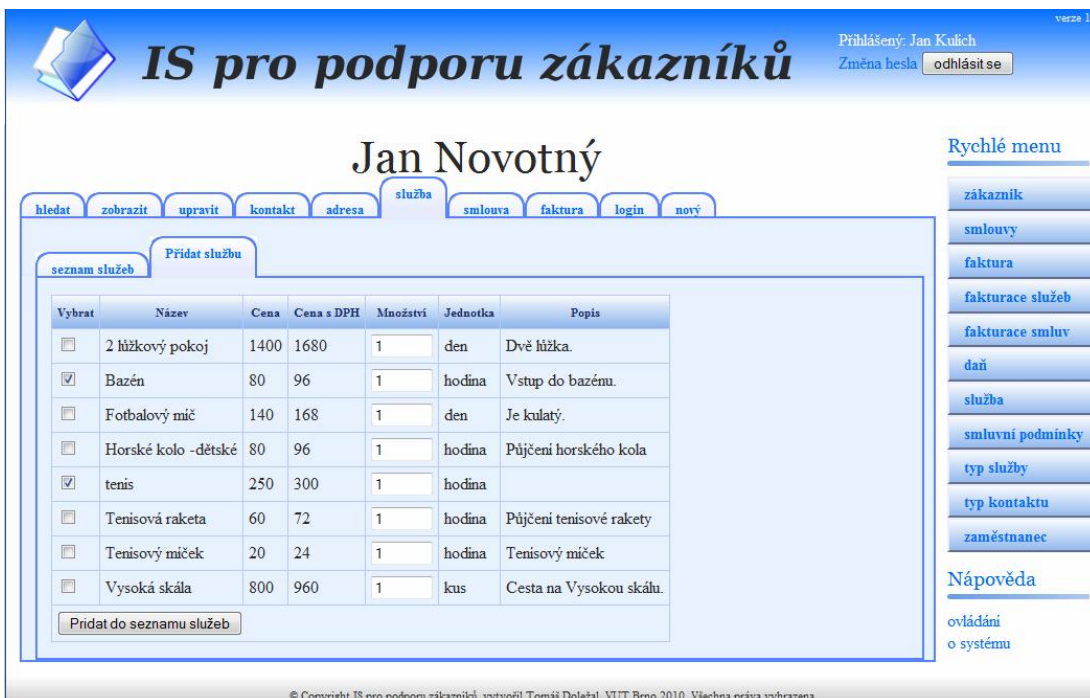
Oprávněná role je *zakaznik*. Tento proces (obr. 28) se stará o komunikaci mezi systémem a zákazníkem. Zákazník musí mít vygenerovaný login a heslo. Report umožňuje zobrazení: jméno/firma, kontakty, adresy, faktury a smlouvy.



Obr. 28: Report zobrazení

## 9. DESIGN

IS pro podporu zákazníků se skládá ze dvou aplikací pro které musel být navrhnout a realizován design. Při návrhu vzhledu pro admin prostředí byl brán velký ohled na to, že nejpodstatnější jsou data a jakákoliv velká barevnost a záplava obrázků by měla spíše rušivý efekt. Byla zvolena uklidňující modrá barva ve třech odstínech s jemnými přechody do bílé barvy. Rychlé menu bylo přesunuto doprava, kde na širokouhlém monitoru na sebe tolik nepoutá pozornost. Pro přehlednou navigaci v modulech jsou odkazy nastylizovány do tvaru záložek. Záložky se mohou řadit na patra. Design byl vytvořen za pomoci kaskádového stylu (kapitola 3.4). Na obr. 29 zobrazen formulář přiřazení služeb.



**IS pro podporu zákazníků** Přihlášený: Jan Kulich verze 1.0  
Změna hesla odhlásit se

**Jan Novotný**

hledat zobražit upravit kontakt adresa **služba** smlouva faktura login nový

**seznam služeb** Přidat službu

Vybrat	Název	Cena	Cena s DPH	Množství	Jednotka	Popis
<input type="checkbox"/>	2 lůžkový pokoj	1400	1680	<input type="text" value="1"/>	den	Dvě lůžka.
<input checked="" type="checkbox"/>	Bazén	80	96	<input type="text" value="1"/>	hodina	Vstup do bazénu.
<input type="checkbox"/>	Fotbalový míč	140	168	<input type="text" value="1"/>	den	Je kulatý.
<input type="checkbox"/>	Horské kolo - dětské	80	96	<input type="text" value="1"/>	hodina	Půjčení horského kola
<input checked="" type="checkbox"/>	tenis	250	300	<input type="text" value="1"/>	hodina	
<input type="checkbox"/>	Tenisová raketa	60	72	<input type="text" value="1"/>	hodina	Půjčení tenisové rakety
<input type="checkbox"/>	Tenisový míček	20	24	<input type="text" value="1"/>	hodina	Tenisový míček
<input type="checkbox"/>	Vysoká skála	800	960	<input type="text" value="1"/>	kus	Cesta na Vysokou skálu.

Přidat do seznamu služeb

**Rychlé menu**

- [zákazník](#)
- [smlouvy](#)
- [faktura](#)
- [fakturace služeb](#)
- [fakturace smluv](#)
- [daň](#)
- [služba](#)
- [smluvní podmínky](#)
- [typ služby](#)
- [typ kontaktu](#)
- [zaměstnanec](#)

**Nápověda**

- [ovládání](#)
- [o systému](#)

© Copyright IS pro podporu zákazníků, vytvořil Tomáš Doležal, VUT Brno 2010. Všechna práva vyhrazena.

Obr. 29: Admin prostředí – přiřazení služeb

Požadavkem bylo, aby jednotlivé formuláře si byli co nejvíce vzhledově podobný a stejná data na rozdílných formulářích (EF, VF, IF) byli pokud je to možné na stejných pozicích. Na obr. 30 je porovnání formulářů v modulu zákazník.



The image shows three side-by-side forms for customer management. The first form, 'zobrazit', displays data for customer ID 1. The second, 'upravit', shows the same data in an edit mode. The third, 'nový', is a blank form for creating a new customer. All forms share a common layout of fields for personal and company details.

zobrazit		upravit		nový	
Zákaznické číslo:	1	Titul		Titul	
Titul:		Jméno*	Jan	Jméno*	
Jméno:	Jan	Příjmení*	Novotný	Příjmení*	
Příjmení:	Novotný	Titul za		Titul za	
Titul za:		Datum narození	16.08.1981	Datum narození	
Datum narození:	16.08.1981	Firma		Firma	
Firma:		IČO		IČO	
IČO:		DIČ		DIČ	
DIČ:		Popis	Věrný zákazník	Popis	
Popis:	Věrný zákazník.	Stav	aktivní	Stav	aktivní
		uložit		uložit	

Datum založení:	15.05.2010 19:19
Založil:	Jan Kulich
Datum úpravy:	16.05.2010 20:02
Upravil:	Jan Kulich
Stav:	aktivní

Obr. 30: Admin prostředí – zobrazit (VF), upravit (EF), nový (IF) zákazník

Pro fakturu a smlouvu byl navržen i grafický design. Faktura na obr. 31.

hledat   zobrazit   **graficky**   reklamovat   zaplatit

### Faktura 1 - daňový doklad

<b>Datum:</b> Zdanitelného plnění: 15. 05. 2010 Vystavení: 15. 05. 2010 Splatnosti: 15. 06. 2010	<b>Odběratel:</b> Jan Novotný Krátká 2/18 59231 Nové Město na Moravě  IČO/DIČ: /
---	---

### Položky zdanitelného plnění

ID	Nazev	Cena/j. v CZK bez DPH	Mnozství	Jednotka	Cena celkem	Sazba DPH	Částka DPH	Cena celkem s DPH
1	tenis	250	2	hodina	500	20%	100	600
5	Tenisová raketa	60	2	hodina	120	20%	24	144
6	Tenisový míček	20	2	hodina	40	20%	8	48
9	Vysoká skála	800	1	kus	800	20%	160	960

Celkem bez DPH: 1460 Kč Celkem DPH: 292 Kč Celkem s DPH: 1752 Kč
--

Obr. 31: Grafické zobrazení faktury

Frontend prostředí nabízí kategorie služeb v pravém rychlém menu. Filtraci v horní části obrazovky. Pod filtrací seznam služeb s obrázkem viz obr. 28.

Nikdo není přihlášen.  
[Přihlásit se](#)
verze 1.0

## IS pro podporu zákazníků

### Nabízíme

Název:     Cena od:     Cena do:     Kategorie: \*Všechny

Připravujeme  
  Novinka  
  Nabízíme  
  Dočasně nedostupný  

<<< 1 2 >>> 9 záznamů - strana 1/2

**Rychlé menu**

- [\\*Všechny](#)
- [Rekreace](#)
- [Sport - prostory](#)
- [Sportovní potřeby](#)
- [Ubytování](#)
- [Výlety](#)
- [Výlety celodenní](#)

**Nápověda**

- [ovládání o systémů](#)

**obrázek není**

2 lůžkový pokoj  
Dvě lůžka.  
1680 Kč  
(1400 Kč bez DPH)  
Novinka  
[podrobnosti](#)

**obrázek není**

Bazén  
Vstup do bazénu.  
96 Kč  
(80 Kč bez DPH)  
Novinka  
[podrobnosti](#)

Fotbalový míč  
Je kulatý.  
168 Kč  
(140 Kč bez DPH)  
Nabízíme  
[podrobnosti](#)

Horské kolo  
Půjčení horského kola  
144 Kč  
(120 Kč bez DPH)  
Připravujeme  
[podrobnosti](#)

Horské kolo -dětské  
Půjčení horského kola  
96 Kč  
(80 Kč bez DPH)  
Nabízíme  
[podrobnosti](#)

© Copyright IS pro podporu zákazníků, vytvořil Tomáš Doležal, VUT Brno 2010. Všechna práva vyhrazena.

Obr. 32: Titulní strana frontend

IS pro podporu zákazníků byla testována na internetových prohlížečích: Internet Explorer 8, Firefox 3, Opera a Gogole Chrome.

Test prokázal plnou funkčnost na těchto internetových prohlížečích. Design u prohlížečů Firefox 3, Opera a Gogole Chrome byl až na drobné změny stejný. Internet Explorer 8 od firmy Microsoft Corporation měl problémy s vykreslením obloučku u záložek.

Doporučené rozlišení je 1024x768 nebo vyšší.

## 10. ZÁVĚR

Cílem této práce bylo navržení a realizování IS pro podporu zákazníků. Byl navržen model databáze tak, aby splňoval základní požadavky stanovené na začátku projektu. Z navrženého modelu se pro jednotlivé entity sestavil životní cyklus. Nejdůležitější a nejsložitější životní cykly jsou u entit faktura a smlouva.

Velká část této práce se zabývala návrhem procesů. Při zamyšlení nad tím, co vše IS pro podporu zákazníků musí umět, se teprve ukázala celková rozsáhlost této práce. Proto byly navrženy procesy tak, aby pokryly co nejvíce problematiku IS a přitom byly co nejjednodušší.

Samotná realizace byla postavena na framework symfony. Pro administrativní část, byla naprogramovaná aplikace admin. Pro uživatelskou část aplikace frontend. Při programování nebyla přesně dodržena struktura symfony, což kód poněkud znepřehlednilo. Pro další aplikace bych se snažil více využívat tříd ve složce lib. Naprogramovat pro každý proces vlastní třídu. Využívat více getterů a setterů pro formátování dat jak pro vstup do databáze tak i pro výstup.

Pro opakující se části zdrojového kódu v programu byly použity „částečníky“ (partial) a to především pro menu. Menu bylo programováno tak, aby se uživatel z jakéhokoliv místa v programu co nejdříve dostal k cíli. Veškeré vstupní data jsou v aplikaci kontrolována za pomoci validátorů.

IS pro podporu zákazníků byl naprogramovaný a pro správnou funkčnost odzkoušen. Na přiloženém CD je uveden odkaz a přístupová práva k systému umístěném na internetu.

## 11. SEZNAM LITERATURY

- [1] MASLAKOWSKI M., *Naučte se MySQL za 21 dní*. Praha: Computer Press, 2001. 478 s. ISBN 80-72226-448-6.
- [2] Brázda J., *PHP 5 začínáme programovat*. Praha: Grada Publishing a.s., 2006. 244 s. ISBN 80-247-1146-X.
- [3] STEPHENS K. Ryan, PLEW R. Donald, *Naučte se SQL za 21 dní*. Brno: Computer Press a.s., 2008, 581 s.
- [4] KAI 'Oswald' Seidler, *XAMPP* [online]. c2009 [cit. 2009-12-17]. Dostupný z WWW: < <http://www.apachefriends.org> >.
- [5] JANOVSKEÝ, D., *Jak psát web* [online]. c2009 [cit. 2009-12-17]. Dostupný z WWW: < <http://www.jakpsatweb.cz> >.
- [6] FABIEN P., *symfony* [online]. C2010[2010-05-19]. Dostupné z WWW: < <http://www.symfony-project.org>>.
- [7] GERNER, J., et al. *PHP5, MySQL, Apache: Vytváření webové aplikace*. Překlad Bohdan Kostka. 1. vyd. Brno: Computer Press a.s., 2006, 860 s. ISBN 80-251-1073-7.
- [8] JQuery UI Team. *JQuery user interface* [online]. c2010[2010-05-19]. Dostupný z WWW: <http://jqueryui.com/home>
- [9] Moxiecode Systéme AB. *TinyMCE: javascript WYSIWYG editor* [online]. c2003-2010[cit.2010-05-240]. Dostupný z WWW:<<http://tinymce.moxiecode.com/>>.

## 12. SEZNAM ZKRATEK, SYMBOLŮ A PŘÍLOH

Obr.	Obrázek
Tab.	Tabulka
CSS	Cascading Style Sheets
FTP	Dokument Object Model
HTML	Hypertext Markup Language
HTTP	HyperText Transfer Protocol
URL	Uniform Ressource Locater
MVC	Model View Controller
QF	Query Form
RL	Rekord List
VF	View Form
EF	Edit Form
IF	Insert Form
MROW	
MROW1	

### Přílohy

#### A. E-R diagramy

A.1 E-R diagram IS pro podporu zákazníků

A.2 Submodel „těla“ databáze

A.3 Submodel pro tvorbu služeb

A.4 Submodel pro vytváření faktur

A. 5 Submodel pro tvorbu smluv

A.6 Submodel pro vytvoření kontaktů zákazníků

#### B. Obsah přiloženého CD

Složka DP s textem této práce ve formátu pdf

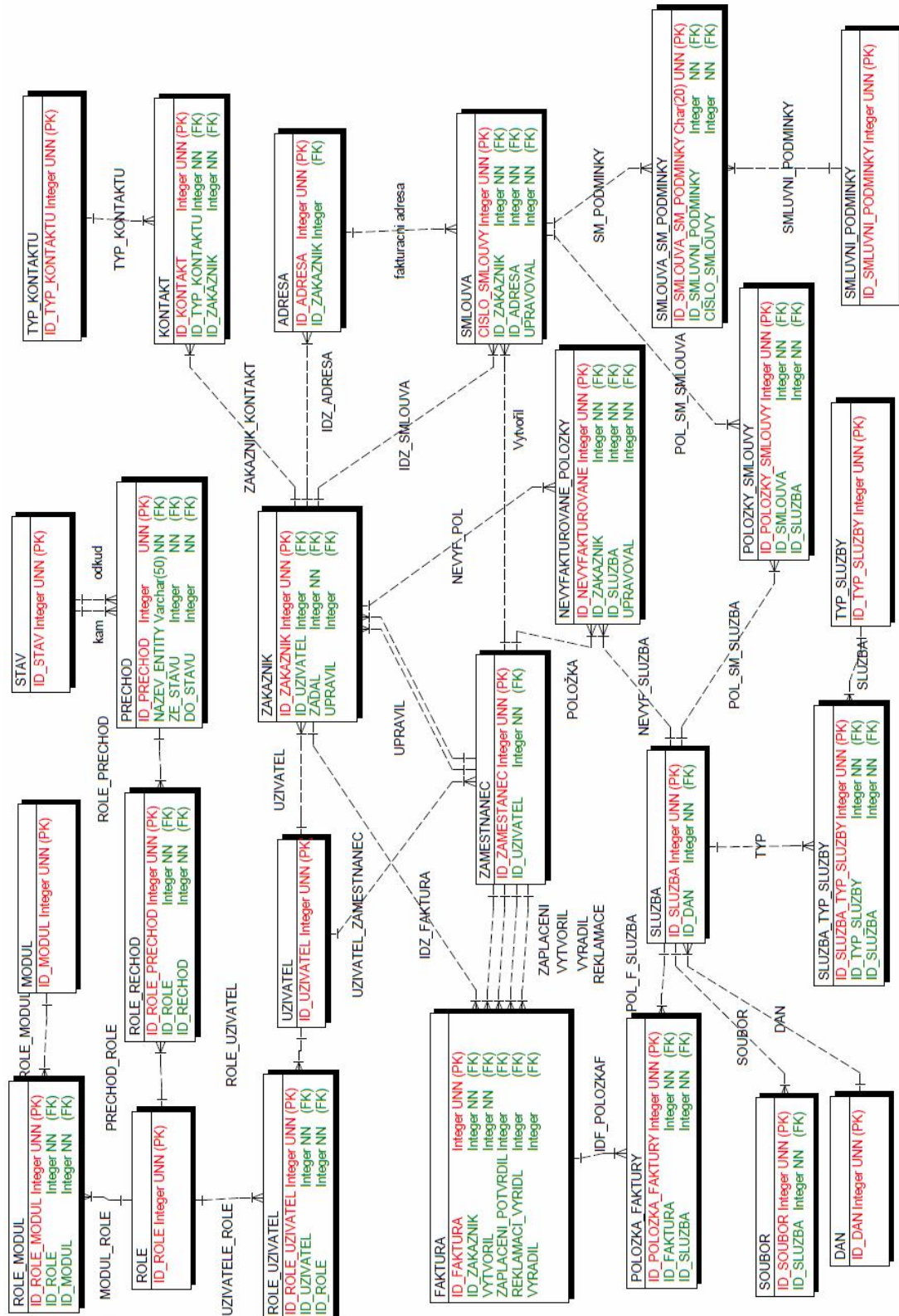
Složka DIP s aplikací IS pro podporu zákazníků

Složka symfony se symfony

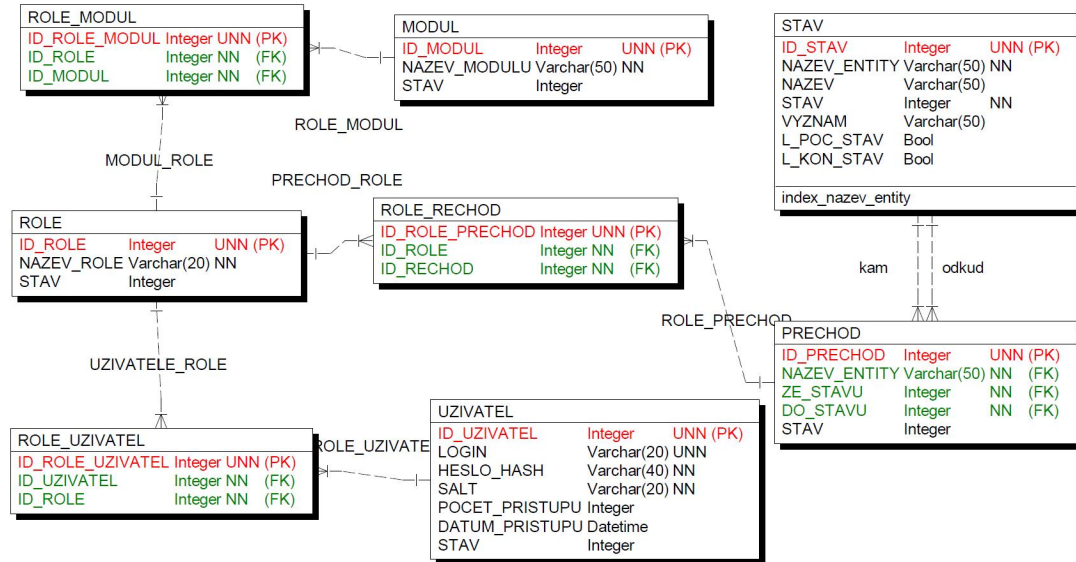
Složka SQL s SQL souborem pro založení databáze.

A. E-R diagramy

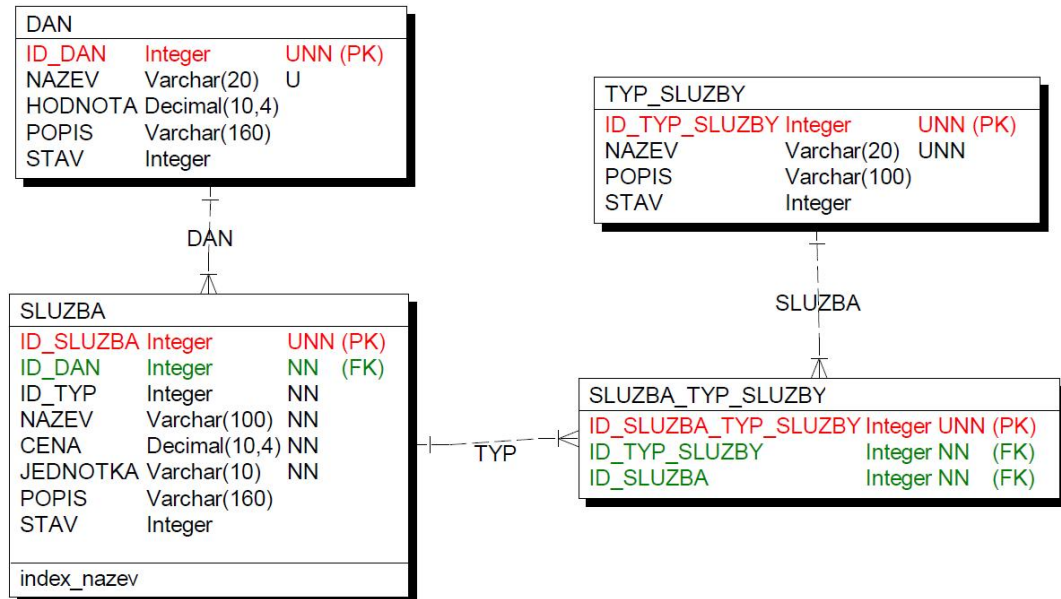
A.1 E-R diagram IS pro podporu zákazníků



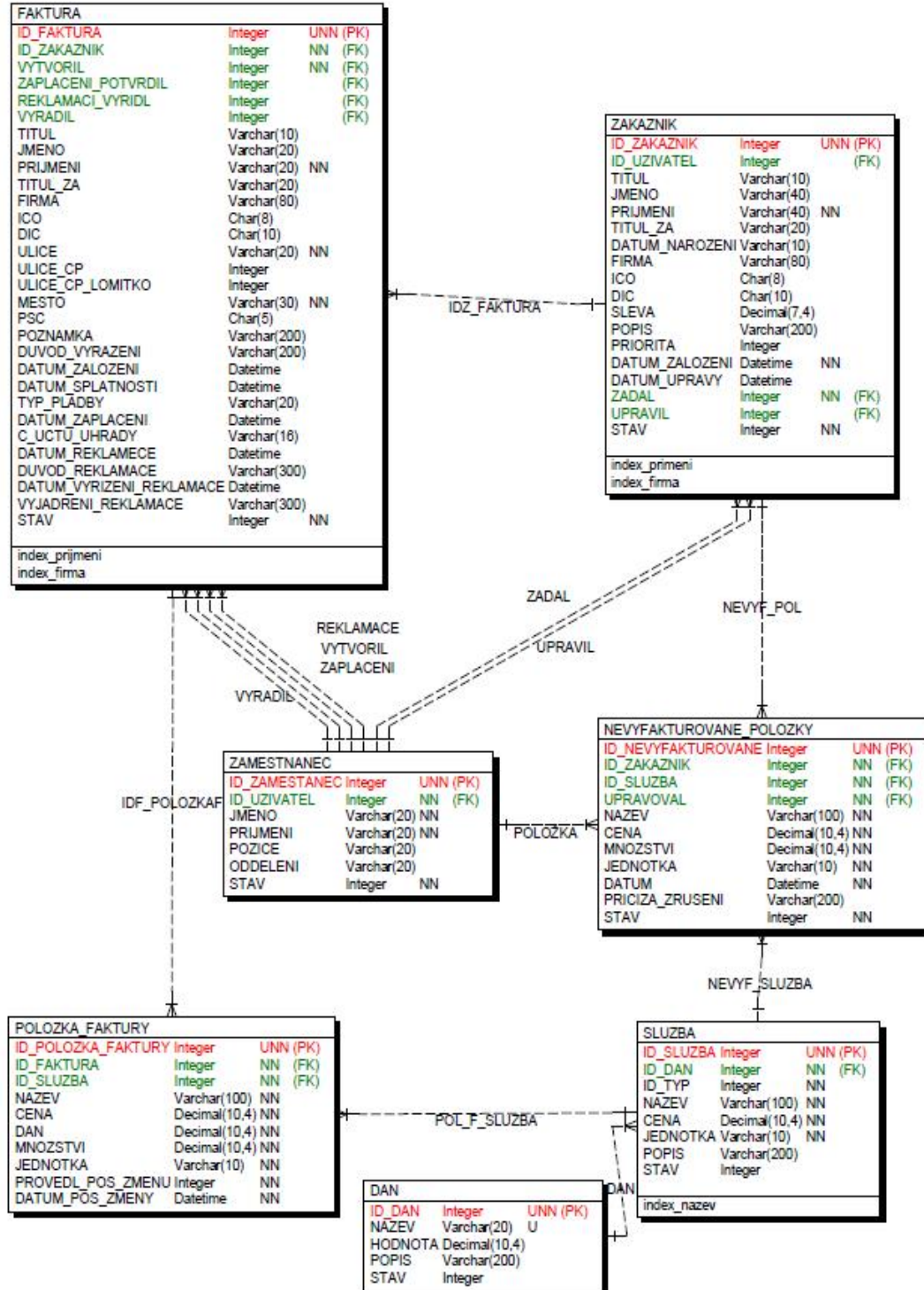
### A.2 Submodel „těla“ databáze



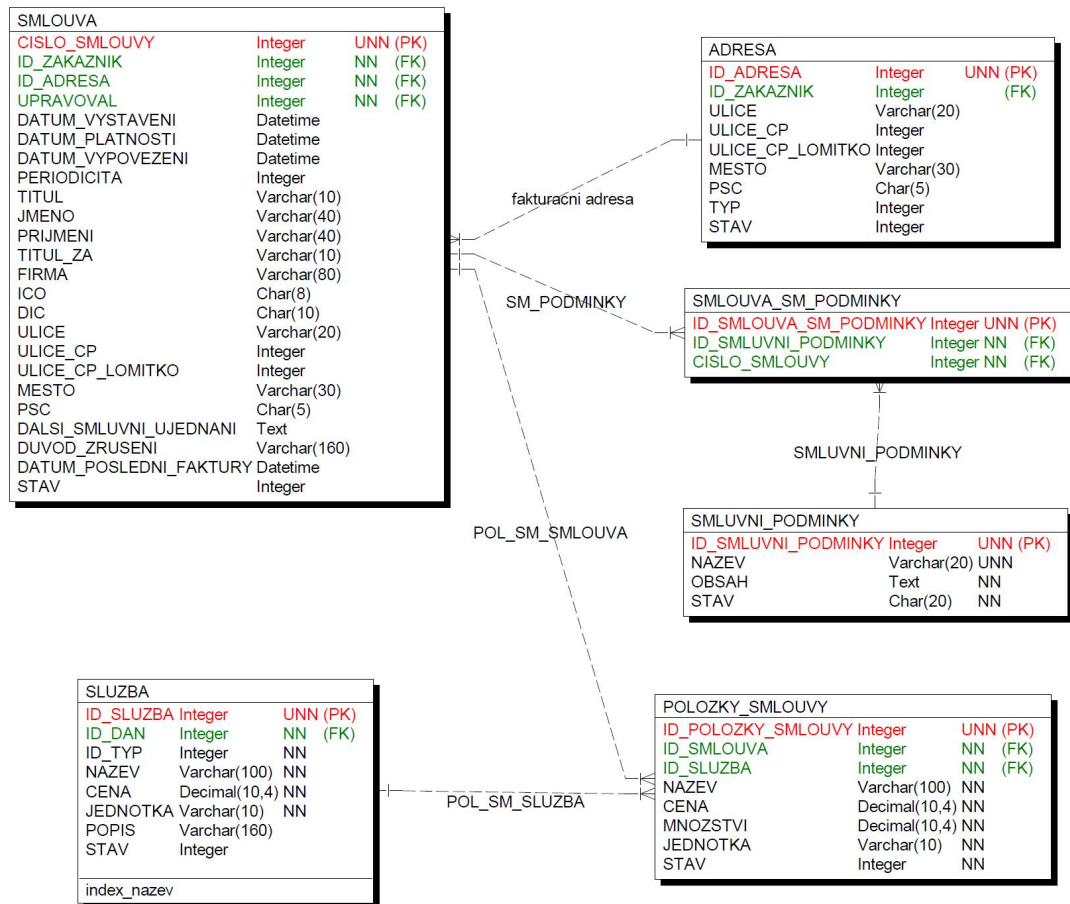
### A.3 Submodel pro tvorbu služeb



A.4 Submodel pro vytváření faktur



### A. 5 Submodel pro tvorbu smluv



### A.6 Submodel pro vytvoření kontaktů zákazníků

