

# **Vývoj informačního systému pro firmy obchodující s geodetickým a stavebním zařízením**

**Bakalářská práce**

**Vedoucí práce:**

**Ing. Pavel Turčínek, Ph.D.**

**Miroslav Matzke**

**Brno 2015**

Zde prostor pro zadání práce

## **Poděkování**

Rád bych poděkoval vedoucímu této bakalářské práce Ing. Pavlu Turčínkovi, Ph.D. za ochotu a odbornou konzultaci.

### **Čestné prohlášení**

Prohlašuji, že jsem tuto práci: **Vývoj informačního systému pro firmy obchodující s geodetickým a stavebním zařízením** vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnici o zveřejňování vysokoškolských závěrečných prací*.

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

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

V Brně dne 21. května 2015

---

## **Abstract**

Matzke, M. Development of an information system for companies dealing with geodetic and construction equipment. Brno, 2015. Bachelor thesis. Mendel University in Brno.

This work is concerned with database design and implementation of a program of internal information system for the company TOPGEOSYS. The database was created by MySQL. To implement the program itself, it was used PHP programming language or the Nette framework, as well as CSS and HTML. The theoretical part describes the development of Web applications, market analysis and ERP systems, focusing on individual components. Also contains a description of imitation of databases and the overall system, its scalability and final evaluation of conformity-produced IS.

## **Keywords**

Information system, ERD, Use Case, Database, MySQL, Nette Framework, CSS, HTML

## **Abstrakt**

Matzke, M. Vývoj informačního systému pro firmy obchodující s geodetickým a stavebním zařízením. Brno, 2015. Bakalářská práce. Mendelova universita v Brně.

Tato práce se zabývá návrhem databáze a implementací programu vnitřního informačního systému pro firmu TOPGEOSYS. Databáze byla vytvořena za pomoci MySQL. Pro implementaci samotného programu je použita technologie programovacího jazyka PHP, respektive jeho framework Nette, dále také CSS a HTML. V teoretické části je popsán vývoj webových aplikací, analýza trhu a ERP systémy se zaměřením na jednotlivé komponenty. Také práce obsahuje popis implementace databáze a celkového systému, jeho rozšiřitelnost a závěrečné zhodnocení vytvořeného IS.

## **Klíčová slova**

Informační systém, ERD, Use Case, Databáze, MySQL, Nette Framework, CSS, HTML

# Obsah

<b>1</b>	<b>Úvod a cíl práce</b>	<b>9</b>
1.1	Úvod.....	9
1.2	Cíl práce.....	9
<b>2</b>	<b>Informační systémy</b>	<b>10</b>
2.1	Co zahrnuje informační systém .....	10
2.2	Rozdělení.....	10
2.3	ERP.....	12
2.4	Vývoj informačních systémů.....	14
2.4.1	Model vodopád.....	14
2.4.2	Model spirála .....	15
2.4.3	Model prototypování .....	16
2.5	Návrh, implementace, testování, údržba, užívání.....	17
2.5.1	Návrh.....	17
2.5.2	Implementace .....	18
2.5.3	Testování.....	18
2.5.4	Údržba .....	18
2.5.5	Užívání.....	18
2.6	Analýza trhu .....	18
<b>3</b>	<b>Metodika</b>	<b>20</b>
3.1	Dostupné technologie.....	20
3.2	Tvorba ERD.....	22
3.3	Tvorba Use Case.....	23
<b>4</b>	<b>Návrh databáze a systému</b>	<b>24</b>
4.1	Požadavky na databázi.....	24
4.2	Návrh databáze .....	24
4.2.1	ERD .....	24
4.3	Požadavky na funkce systému .....	30

---

4.4	Návrh systému.....	30
4.4.1	Use Case diagram .....	30
<b>5</b>	<b>Implementace</b>	<b>34</b>
5.1	Použité technologie .....	34
5.1.1	Nette Framework .....	34
5.1.2	CSS.....	37
5.1.3	HTML.....	38
5.1.4	MySQL.....	38
5.2	NetBeans.....	39
5.3	MySQL Workbench .....	39
5.4	Tisk objednávek do PDF .....	40
5.5	Problémy .....	41
<b>6</b>	<b>Diskuze</b>	<b>42</b>
6.1	Výsledky.....	42
6.2	Možnosti rozšíření .....	42
<b>7</b>	<b>Závěr</b>	<b>44</b>
<b>8</b>	<b>Reference</b>	<b>45</b>
<b>A</b>	<b>Projekt k prohlídce</b>	<b>47</b>
<b>B</b>	<b>Celý ERD</b>	<b>48</b>
<b>C</b>	<b>Celý Use Case</b>	<b>49</b>

## Seznam obrázků

<b>Obr. 1</b>	<b>Rozdělení velikosti firem (Náplava, 2012)</b>	<b>11</b>
<b>Obr. 2</b>	<b>Model vodopád (Rybička, 2009)</b>	<b>15</b>
<b>Obr. 3</b>	<b>Model spirála (Šmíd, 2002)</b>	<b>16</b>
<b>Obr. 4</b>	<b>Model prototypování (Rybička, 2009)</b>	<b>17</b>
<b>Obr. 5</b>	<b>Entity a jejich vazby</b>	<b>24</b>
<b>Obr. 6</b>	<b>Část ERD: doplňky, produkty a propojení</b>	<b>25</b>
<b>Obr. 7</b>	<b>Část ERD: zaměstnanci, zákazníci, komentáře a propojení.</b>	<b>27</b>
<b>Obr. 8</b>	<b>Část ERD: objednávka (nákup, zápůjčka, oprava) a propojení</b>	<b>28</b>
<b>Obr. 9</b>	<b>Use Case diagram funkcí systému</b>	<b>31</b>
<b>Obr. 10</b>	<b>Use Case: blok správy oprav</b>	<b>32</b>
<b>Obr. 11</b>	<b>NetBeans: orientace v souborech systému (presentery a templaty)</b>	<b>39</b>
<b>Obr. 12</b>	<b>ERD</b>	<b>48</b>
<b>Obr. 13</b>	<b>Use Case</b>	<b>49</b>



# 1 Úvod a cíl práce

## 1.1 Úvod

Informační systémy jsou dnes nedílnou součástí každé firmy, ať už se zabývá obchodem s geodetickými a stavebními přístroji nebo se věnuje jakémukoli odvětví. Velkou výhodou zavedeného elektronického informačního systému je nepochybně jednoduchá správa dat, ať už objednávek nebo zaměstnanců a oproti informačním systémům psaných na papír klesá riziko ztráty dat a naopak roste přehlednost, rychlost vyhledávání, jednoduchost spravování a prostorová nenáročnost. Informační systémy jsou závislé na datech, které jsou zadávány uživatelem, ty jsou dále využívány jako informace o produktech, zaměstnancích, zákaznících nebo například délce trvání záruční lhůty a pak jsou dále zpracovávány. Důležitou součástí je připojená databáze, jejíž struktura je navržena tak, aby dané firmě vyhovovala co nejlépe.

V této práci bude k vytvoření databáze použit databázový systém MySQL. Nad databází se pak budou provádět potřebné dotazy na vkládání, úpravu nebo čtení dat. Data budou do databáze ukládány pod různými typy (datum, text, číslice). Existují různé nástroje na ulehčení vytvoření databázových tabulek, není nutné je vypisovat SQL dotazy. V této práci bude na vývoj databáze použit program MySQL Workbench.

## 1.2 Cíl práce

Cílem této bakalářské práce je navrhnout a implementovat informační systém pro firmu obchodující s geodetickým a stavebním zařízením. Pro splnění tohoto cíle je potřeba nejdříve analyzovat současný stav ve firmě TOPGEOSYS. Na základě zjištěných informací bude možné navrhnout datový model tak, aby bylo možné uchovávat všechna potřebná data. Nad tímto datovým modelem pak bude navržen informační systém.

Logický model bude poté realizován pomocí relačního databázového systému. Vzniklé tabulky budou naplněny daty. Informační systém pak bude implementován jako webová aplikace splňující požadavky zadavatele.

Cíl bude zahrnovat sepsání postupu vytváření databáze a implementace celkové webové aplikace. Dále se práce bude zabývat technologií, která bude použita k vytvoření informačního systému tak, aby bylo možné jej nasadit do provozu.

## 2 Informační systémy

Informační systém, neboli zkráceně IS, je obecně sled lidí a technologií, díky kterým je umožněn sběr a zpracování dat za účelem získání informací. (Sodomka, 2011)

### 2.1 Co zahrnuje informační systém

Informační systém může být i v pouhé papírové podobě, tudíž informační systémy fungovaly i před příchodem novodobých technologií, ale s vývojem a příchodem nynějších prostředků jakožto počítačů, internetu, serverů a vývojových technologií (například MySQL a Framework Nette, které budou použity v této práci) se potenciál a využitelnost těchto systémů značně zvýšil, a to zejména proto, že práce s ním je jednodušší a existuje spousta druhů, které každé firmě nabídnou přesně to, co vyžaduje. Využitelnost spadá do všech odvětví a může jej využít jak velká firma, tak i jednotlivec.

IS zahrnuje databázi (datové uložení), data (v podobě uložených informací), software (samotná aplikace, GUI<sup>1</sup>, webová aplikace) a hardware (počítače, servery, routery, kabeláž). Hardware potřebný pro spuštění, údržbu, používání a posléze následné rozšiřování, se odvíjí od potřeb, odvětví a velikosti zákaznické firmy.

Malé firmy s ne velkým počtem dat mohou využívat svůj IS jen na běžných počítačích. Oproti tomu velké korporace na svůj provoz většinou musejí vlastnit rozsáhlé serverovny nebo mít servery propůjčené (pronajaté) s měsíčním nebo ročním poplatkem.

Stejně tak je tomu i se samotným softwarem a ne každý se hodí všem druhům podnikání a proto se dělí na několik druhů a rozsahových velikostí.

### 2.2 Rozdělení

Elektronické informační systémy se dělí do více skupin podle jejich:

- rozsahu funkcí
- funkčního zaměření
- použité technologie
- velikosti nasazení
- způsobu nasazení a používání

Dále se IS rozdělují zaměřením jejich využití:

- Manažerské
- Taktické

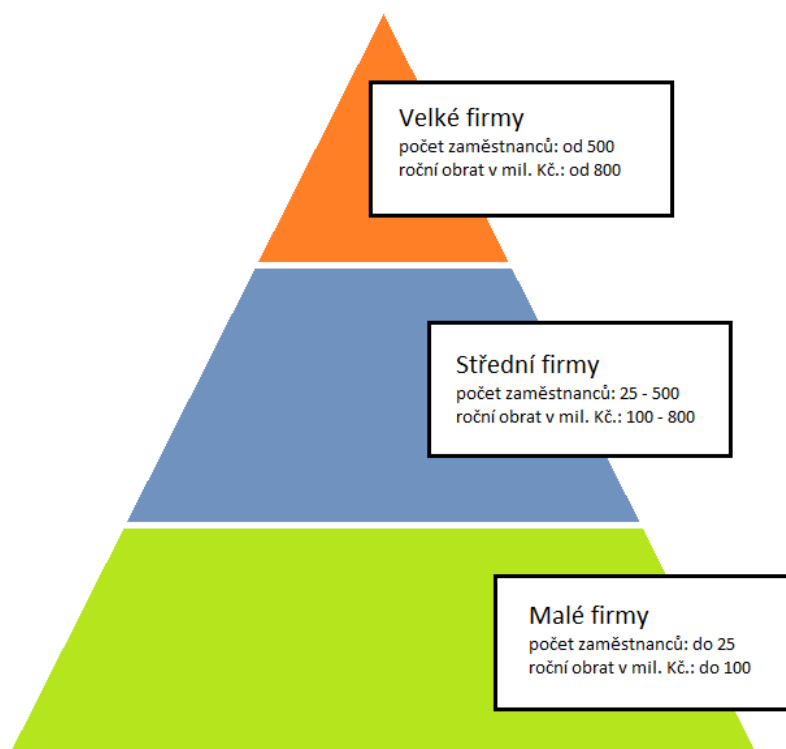
---

<sup>1</sup> Grafické uživatelské rozhraní pro snadné ovládání systému uživatelem přes počítač

- Kancelářské
- Expertní
- Operativní

Důležité dělení také spočívá dle velikosti nasazení informačního systému. Firma nasazující IS musí uvážit v potaz všechny faktory. Zda je firma malá, střední nebo velká, a to co do počtu zaměstnanců a stejně tak do velikosti ročního obrátu. Ale velikost není jediné, co při tomto kritériu rozhoduje, protože firma může mít perspektivní růst, a proto se také počítá s expanzí. Za předpokladu, že životnost informačního systému není neomezená, ale počítá se okolo pěti až deseti let, tudíž se z malé firmy během tohoto rozsahu může stát firma střední, tak se jako důležité kritérium bere vize a strategie.

Při výběru se počítá i finanční stránka, jakožto náklady na pořízení, jelikož obrat nemusí nutně znamenat zisk, tak by se mělo dobře promyslet, kolik prostředků se může uvolnit právě na pořízení informačního systému. Systém není nutný kupovat celý najednou, mohou se zakoupit jen některé moduly<sup>2</sup> a časem se podle plánu a strategií expanze rozšiřovat. (Náplava, 2012)



Obr. 1 Rozdělení velikosti firem (Náplava, 2012)

Nasazení informačního systému může být provedeno dvěma způsoby.

---

<sup>2</sup> Moduly informačního systému umožňují přidávat potřebnou funkcionalitu v průběhu chodu.

První varianta je vnitrofiremní nesazení. Probíhá tak, že systém je nainstalovaný přímo ve firmě u zákazníka a dodavatel k němu může mít pouze přístup vzdálený na údržbu a podporu. Tímto, ale neodpadává možnost vzdáleného přístupu uživatelů přes internet.

Druhá možnost je hostovaný systém, kdy si zákazník informační systém pouze pronajímá a má k němu jenom vzdálený přístup, a to odkudkoli. Avšak tu vzniká riziko v případném sporu o vlastnictví zaváděných dat a informací, jelikož jsou uloženy na straně dodavatele. Jsou tu však i výhody, zákazník nemusí vlastnit hardwarové vybavení k provozu systému, stačí mu pouze osobní počítač, přes který se na systém připojí. (Náplava, 2012)

## 2.3 ERP

ERP systémy neboli Enterprise Resource Planning jsou hlavní součástí jádra podnikových informačních systémů s účelem efektivního rozvržení zdrojů v podniku.

ERP systémy se ve většině případů zakupují po jednotlivých modulech, jako jsou:

- **Ekonomické moduly**  
Velmi důležitou součástí každé společnosti jsou finance. Tyto moduly se postarají o vše bez výjimek, starají se o finanční toky, komunikují s bankami, a to jak o výdajích, tak o příjmech a to ve všech měnových kurzech, tudíž zabezpečují i kurzovní lístek. Dále řídí pokladny, zabezpečují interní daně, starají se o postupy a procesy ve finančním odvětví firmy a vytvářejí výkazy pro jednoduchou kontrolu nad penězi, řídí rozpočty v celé firmě a další. Uchovávají všechny majetek, historické změny majetku a evidují nový a jeho hodnotu. Zavádí seznam, kdo má majetek zrovna na starosti (firemní dopravní prostředky, počítače, nástroje) a evidují odpisy. (qi.cz, 2015)
- **Moduly pro lidské zdroje**  
Dobrý chod firmy zajišťuje především kvalitní personál. Tyto moduly firmě pomohou již od počátku, a to s výběrem zaměstnanců. Pomohou při výběrových řízeních, ulehčí výběr podle předem stanovených kritérií, tak aby nový zaměstnanec co nejlépe zapadal přesně na svoje místo. Při práci pomohou s evidencí docházky jednotlivých zaměstnanců a to i automaticky. Mohou pomáhat při rozvoji jednotlivců v personálu, a to plánováním jejich školení a přezkoušení. Zavádí také hodnocení zaměstnanců a porovnává výkony jednotlivců, tudíž pomohou i při zkracování stavu firmy. Mohou obsahovat také plánování porad a schůzí prospívajících pro vedení a expanzi firmy. Dále pomáhají při vyplácení mezd jednotlivým zaměstnancům, a tak hlídají i finanční stránku. (qi.cz, 2015)
- **Marketing a obchod**  
Důležitou součástí obchodu jsou skladovací prostory, a proto jsou zde zahrnuty moduly pro řízení skladu, které pomohou při organizaci skladovaného zboží. Pomohou řídit doplňování nedostatkového zboží, spravují příjmy a výdaje

a monitorují pohyb mezi jednotlivými sklady firmy. U zboží evidují jeho parametry, mohou pracovat s čárovým kódem výrobku a ulehčit tak jeho evidenci. Moduly mohou obsahovat i práci s vratnými obaly prospěšnou zejména firmám zabývajících se potravinami. Moduly poskytují zpracovávání dokladů týkajících se obchodů, ať už prodeje nebo nákupu. (vario.cz, 2014) Pomáhají při tvorbě cen každého zboží a navazují na určité akční slevy nebo přirážky podle pohybu trhu výrobku. Plánují objednávání zboží a výrobu podle prodeje a úbytku. Pracují ve více jazycích a měnách, tudíž umožňují i expanzi firmy do zahraničí a komunikaci s Evropskou unií. Moduly dále mohou zahrnovat CRM<sup>3</sup>, pomáhat budovat a udržovat dobré vztahy s obchodními partnery a dále je dělit do skupit, což pak umožňuje komunikaci pouze pro určitou skupinu a podporuje cílený marketing. Uchovávají údaje o zákaznících, jako jsou telefon, e-mail a dávají tak možnost pro další komunikaci a rozšířený obchod. Důležitou součástí je také marketinkový modul, který pomáhá při pořádání seminářů, kampaní, školení a různých marketingových akcí, tak aby byly finančně a cíleně co nejvýhodnější. (qi.cz, 2015)

- Řízení firmy

Základní modely pro plánování práce zaměstnanců, řízení schůzí, zadávání úkolů všem zaměstnancům a v podstatě plánování času stráveného prací, jakozto řízení celého pracovního procesu, zavádějí do firmy pořádek. Dále moduly mohou pomoci při plánování projektů, a to tak, že si firma předem připraví projekt, například v podobě školící akce nebo organizace výplat, a tento projekt pak může být opakovaně využíván s přehledným souhrnem. Při takovéto akci mohou modely vypomoci i s jejím plánováním. (vario.cz, 2014) Mohou pomoci vedoucím oddělení tak, aby jejich zaměstnanci dosahovali kvalitnějších výsledků a také s organizací práce. Při řízení projektů pomáhají udržovat kapacity, reagovat na změny, sledovat stroje, zařízení a optimalizovat celý projekt. Dále mohou umožňovat plánování auditů, knih jízd a další. (qi.cz, 2015)

- Výroba

Mohou pomáhat ve výrobě, kde umožňují organizaci kapacity ve firmě, výrobě, marketingu, školení nebo ve skladech. Pomohou při rozdělování lidských zdrojů podle výroby a rozdělení provozů (ranní provoz, odpolední provoz, noční provoz, dvanáctihodinový provoz, dvacetičtyřhodinový provoz) a zefektivňují kapacitu personálu tak, aby výroba byla co nejvíce finančně výhodná. Dále mohou přímo podporovat samotnou technickou přípravu výrobku. Normalizují postupy, hlídají spotřebu materiálu, kontrolují jednotlivé kroky a postupy při výrobě a hlídají, aby samotná výroba byla zaznamenávána a kontrolována tak, aby docházelo k co nejmenší chybovosti. (qi.cz, 2015) Dalším rozšířením je plánování výroby, které úzce souvisí se samotnou výrobou. Plánování v podstatě podporuje správu všech strojů, směn, používaného materiálu,

---

<sup>3</sup> CRM neboli Customer relationship management je management pro řízení vztahů se zákazníky.

kritických technologií a další. Moduly se mohou zaměřit na řízení výroby, řízení jakosti při výrobě, jakožto hlídání kvality výrobku a poskytování statistik chybovosti pro reklamační servis. Jsou i moduly půjčoven, reklamačních oddělení, evidencí smluv, kusovníků a operací. (vario.cz, 2014)

- **Internetové moduly**

Internetové moduly nabízejí firmám internetové služby, jako je e-shop pro zákazníky s jednoduchým přístupem a snadným nákupem zboží. Dále jsou moduly pro portál, jakožto informační webová stránka, která firmě nabízí publikování svých služeb a výrobků na internetu a volnou komunikaci se zákazníkem. Existují i moduly pro mobilní aplikace tak, aby zaměstnanci mohli systém využívat ve svých mobilních zařízeních a zadávat tak data odkudkoli prostřednictvím internetu. (qi.cz, 2015)

- **Specializované moduly**

Další moduly se specializovanou funkcí mohou vypomáhat v mnoha odvětvích s různou činností mnoha firem. Například správy bytových družstev, domovní správy a všeobecné prostorové správy. Modul vodárenství poskytuje všem firmám, které nabízejí pitnou nebo užitkovou vodu, správu odběratelů a usnadňuje fakturaci vodného a stočného poplatku. Moduly na dopravu pomáhají počítat spotřebu, plánování tras, ujeté kilometry na přepravu zboží a pomáhají i s řešením všech financí a plánováním spojeným s přepravnou činností. Moduly na plánování servisu poskytují informace o opravovaných produktech, měsíční lhůtě jejich opravení a jejich stavu. Modul pokladního prodeje umožňuje evidenci zboží prodávaného na pokladně, počítá prodeje kusů, eviduje ceny a snadno nabízí akční slevy na zboží, které se na pokladnách tak často neobjevuje. (qi.cz, 2015)

## **2.4 Vývoj informačních systémů**

Vývoj informačního systému je dán stanoveným cílem. Jedním cílem může být vývoj IS, který se dále nabízí na prodej firmám v upravených verzích nebo po jednotlivých modulech. Jeho jádro se rozšiřuje za účelem nabídky více možností a zdokonalení.

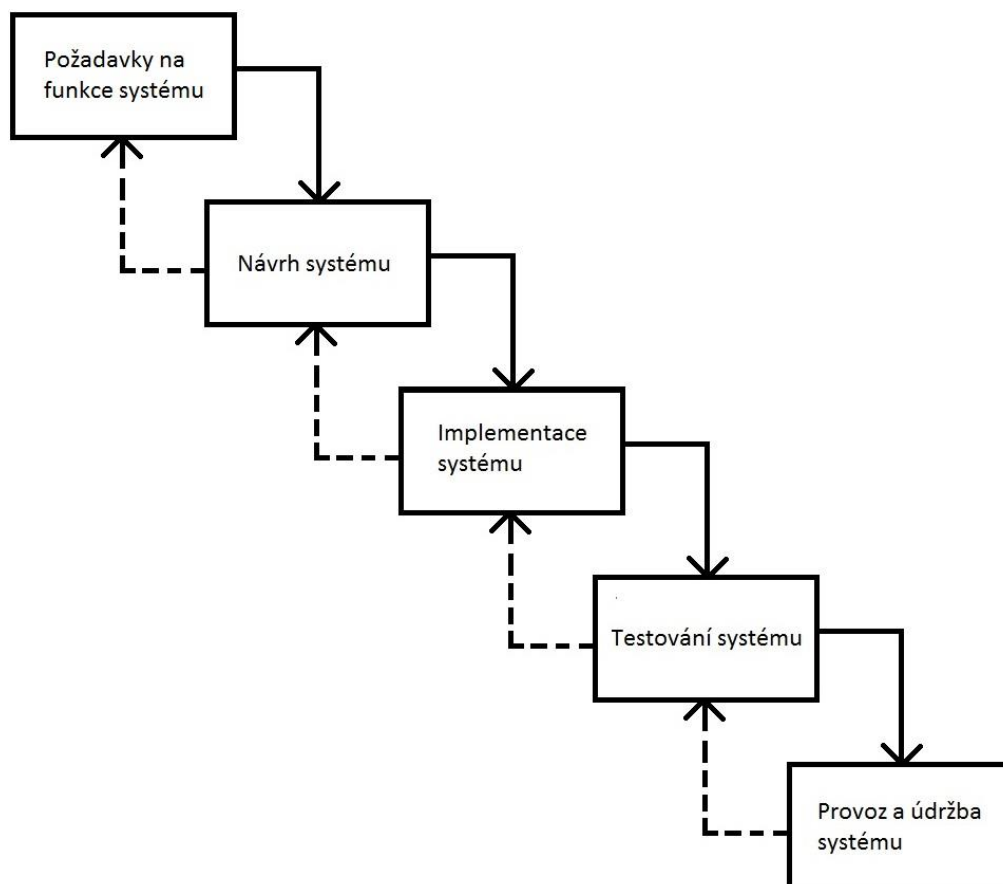
Nebo jako v této práci je cílem vytvořit informační systém na míru jedné firmy, a to podle jejich pokynů a požadavků.

Samotný vývoj informačního systému prochází několika životními cykly. Tyto cykly jsou popsány různými odborníky rozdílně. Také na vývoj existuje spousta metod a modelů.

### **2.4.1 Model vodopád**

Vývoj systému vodopádovým modelem probíhá po jednotlivých krocích a postupuje vždy na další krok až po úplném dokončení kroku předchozího. To z vodopádového přístupu dělá model, který je použitelný pouze za předpokladu, že zadavatel, zákazník, má předem přesně stanovené požadavky na funkcionalitu,

jelikož v průběhu vývoje se již tato funkcionalita nemění a první vyzkoušitelný návrh vyvstane až po několika krocích. Na druhou stranu vývoj pomocí tohoto modelu není příliš nákladný, jelikož se postupuje po krocích chronologicky a zpět už se nevrací. Avšak po nalezení chyby až v konečném produktu může být jeho oprava (přepracování) dražší a to hlavně proto, že musí projít opět všemi předcházejícími vrstvami modulu. (Rybička, 2009)



Obr. 2 Model vodopád (Rybička, 2009)

### 2.4.2 Model spirála

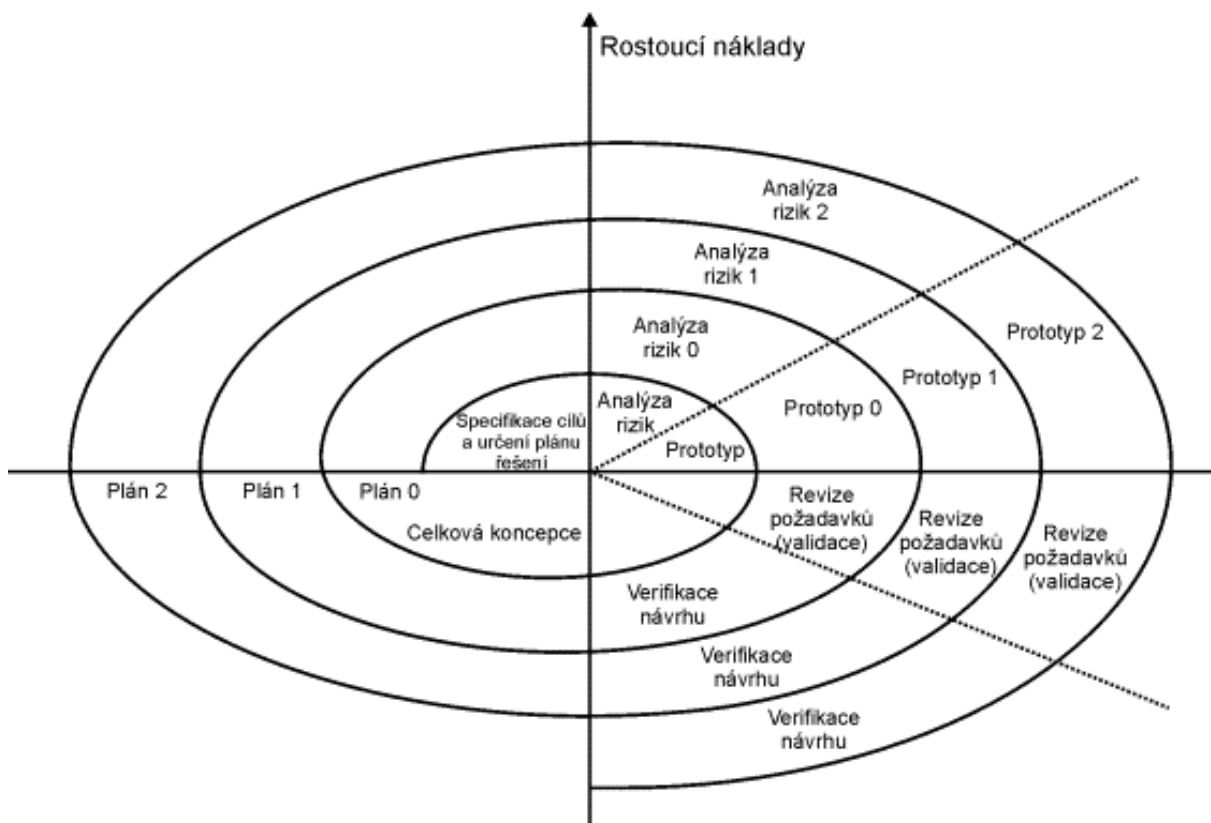
Spirálový model vymyslel Barry Boehm v devadesátých letech. Tento model zahrnuje v podstatě čtyři kroky, které se opakují dokola a nabalují na sebe předchozí vrstvy vyvíjeného systému tak, aby ve výsledku vznikl celý použitelný systém.

Tyto fáze jsou:

1. Stanovení cílů
2. Analýza
3. Vývoj a testování
4. Plánování

Tyto fáze se opakují neustále dokola, a to tak, aby na začátku projektu vyvstaly nějaké cíle. Následuje analýza systému a vytvoření prototypu. Pokračuje se validací, verifikací a posledním krokem je plánování na další kroky. V těchto následujících krocích se opět opakují všechny čtyři fáze.

Spirálový model se hodí na vývoj, při kterém je přístupný neustálý kontakt se zákazníkem. Jelikož s dalšími opakovanými fázemi roste časová i finanční náročnost, tak bez dohledu zadavatele model nabalí velká rizika v některých částech vývoje. U modelu nelze přesně stanovit čas ani jeho konečnou cenu. (Šmíd, 2002)



Obr. 3 Model spirála (Šmíd, 2002)

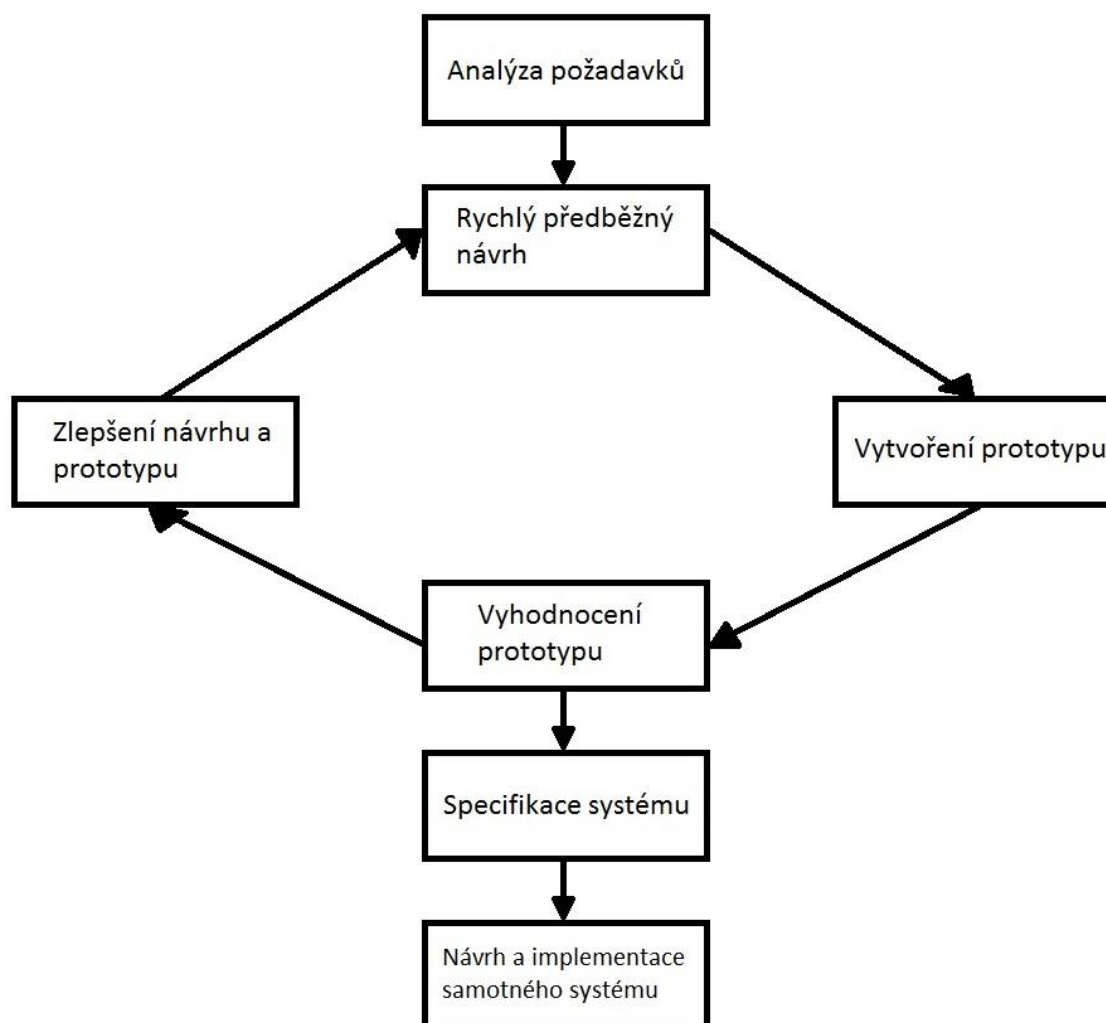
### 2.4.3 Model prototypování

Vývoj informačních systémů za pomoci prototypového modulu spočívá ve vytváření prototypu samotné aplikace. Tyto prototypy neobsahují všechny funkce, které by měla obsahovat finální podoba systému, pouze ilustrují a odlaďují nedostatky, které je potřeba dodělat.

Prototypy se zaměřují na různé prvky programu. Mohou být zaměřeny na grafickou stránku programu, na funkční stránku programu (ověřuje se správný chod funkcí a pochopení problému zákazníka). Dále existují prototypy pro ověření, zda požadovanou funkcionalitu je možné vyřešit (algoritmus existuje). Jsou i prototypy, které jsou vyvíjené stále dopředu, a nabaluje se na ně větší množství funkcí a mohou skončit jako finální verze aplikace. (Rábová, 2008)



Prototypování je využíváno tak, aby zadavatel mohl upřesňovat požadavky a každý následující prototyp se vždy přibližoval ke konečnému stavu. Je důležité, aby tyto prototypy byly vytvářeny co nejrychleji a mohl začít vývoj finální verze, který začíná až po schválení posledního prototypu koncovým uživatelem.



Obr. 4 Model prototypování (Rybička, 2009)

## 2.5 Návrh, implementace, testování, údržba, užívání

Vývoj informačního systému prochází jednotlivými fázemi od návrhu až po jeho vyřazení z provozu.

### 2.5.1 Návrh

Návrh informačního systému probíhá za spolupráce zadavatele (zákazníka) a jsou zde stanovena veškerá kritéria důležitá pro začátek tvorby, zejména rozsah projektu a finanční prostředky vyhrazené na IS. Je vytvořen entitně relační diagram na

stanovení rozsahu a funkčnosti databáze a Use Case diagram na stanovení funkcionality samotného systému. Jsou naplánované veškeré další kroky vývoje a je vytvořen konceptuální a implementační model.

### 2.5.2 Implementace

Samotná implementace systému probíhá podle určených pravidel a plánu. Také se nadále kontroluje, zda zákazníkovi vyhovuje tvořený koncept, a proto je možné předkládat nehotové projekty nebo prototypy na přezkoumání funkčnosti či designu.

### 2.5.3 Testování

Po implementaci systému nastává fáze zavádění, kdy se systém nainstaluje do zákaznickovy firmy. Instalace může být prováděna více způsoby, buď se starý systém zcela nahradí novým, nebo se ve fázi testování implementuje jen do části infrastruktury firmy a nebo se používá zároveň se starším provedením. Ve fázi testování se zkoumá, zda systém běží podle předchozích návrhů a zda vyhovuje pro chod firmy.

### 2.5.4 Údržba

Informační systém se pro správný chod firmy musí neustále udržovat ve stabilním stavu. Údržba zahrnuje aktualizace systému, čištění a zálohování databáze tak, aby se zabránilo případnému výpadku nebo vymazání a zpomalení chodu.

### 2.5.5 Užívání

Dobře udržovaný systém lze ve firmě používat dlouhou dobu, ale s technologickým vývojem a rostoucím chodem firmy je někdy nutné systém vyměnit za nový, který pak opět projde všemi fázemi vývoje.

## 2.6 Analýza trhu

Dnešní trh nabízí spoustu systémů s podobným i rozlišným využitím. Některé se hodí pro menší firmy, některé pro větší, ale všechny spojuje účel ulehčit zákazníkovi správu a vedení podniku.

### Co nabízí trh

Dnešní trh nabízí spoustu firem zabývajících se informačními systémy. Většina těchto systémů je navržena tak, že si uživatel může vybrat z několika vzájemně propojitelných komponentů nebo ustálených tematických celků.

Nejznámější firmy na trhu jsou:

- Helios
- Karat

- ABRA
- Money
- QI
- K2
- SAP
- Microsoft Dynamics

### **Řešení**

Firma TOPGEOSYS je firma zabývající se prodejem, opravami a půjčováním geodetických a stavebních zařízení. Firma sídlí v Brně a je autorizovaným dealerem japonské firmy TOPCON CORPORATION pro obchodní značky TOPCON a SOKKIA. Přesto je firma malá a nemá příliš velký počet zaměstnanců. Proto bylo lepší zvolit po finanční stránce individuální přístup řešení informačního systému a bylo tedy navrženo vlastní řešení informačního systému za konzultace přímo ve zmiňované firmě.

## 3 Metodika

### 3.1 Dostupné technologie

#### Nette Framework

Nette Framework je open sourceový framework vytvořený v České republice a slouží pro tvorbu webových aplikací založených na používání komponent a na bezpečnostní rizika. Je to framework pro programovací jazyk PHP, který usnadňuje práci s jazykem.

#### NVC

Model-View-Controller je softwarová architektura, která vznikla z potřeby oddělit u aplikací s grafickým rozhraním kód obsluhy (controller) od kódu aplikační logiky (model) a od kódu zobrazujícího data (view). Tím jednak aplikaci zpřehledňuje a usnadňuje budoucí vývoj a umožňuje testování jednotlivých částí zvlášť.

Model je datový a zejména funkční základ celé aplikace. Je v něm obsažena aplikační logika. Jakákoliv akce uživatele (přihlášení, vložení zboží do košíku, změna hodnoty v databázi) představuje akci modelu. Model si spravuje svůj vnitřní stav a ven nabízí pevně dané rozhraní. Voláním funkcí tohoto rozhraní můžeme zjišťovat či měnit jeho stav. Model o existenci view nebo kontroleru neví.

View, tedy pohled, je vrstva aplikace, která má na starost zobrazení výsledku požadavku. Obvykle používá šablonovací systém a ví, jak se má zobrazit ta která komponenta nebo výsledek získaný z modelu.

Controller je řadič, který zpracovává požadavky uživatele a na jejich základě pak volá příslušnou aplikační logiku (model) a poté požádá view o vykreslení dat. Obdobou kontrolerů v Nette Framework jsou presentery. (nette.org, 2015)

#### CSS

Kaskádové styly jsou jazyk, který popisuje zobrazení elementu (textu, nadpisu, tabulek, obrázku), a to na stránkách, které jsou psány značkovacím jazykem HTML. (Castro, 2011) Pomocí CSS jde nastavovat barva, velikost, font, podtržení, tloušťka a spousta dalších parametrů textu.

Dále veškeré nastavení barvy stránky, odsazení obrázků, velikost, spojení a centrování tabulek. Také jdou vytvářet různé kategorie zobrazení, zobrazení hypertextového odkazu a další.

#### HTML

HTML neboli HyperText Markup Language je značkovací jazyk, který umožňuje tvorbu internetových stránek. Jazyk se skládá z tagů (znaků), proto je označen za značkovací. (Hauser, 2006)

HTML umožňuje psát web s nejrůznějšími druhy tagů, například tagy pro tabulky, seznamy, texty, nadpisy, odkazy, obrázky a další.

Základní HTML struktura obsahuje tagy:

- html: označuje rozsah dokumentu
- head: hlavička dokumentu
- body: tělo dokumentu

## Composer

Composer je nástroj na správu závislostí v PHP. Umožňuje deklarovat libovolně složité závislosti jednotlivých knihoven a pak je nainstalovat do projektu. Základem projektu, který využívá Composer, je soubor composer.json, který obsahuje meta informace o projektu a jejich závislostech. (nette.org 2015) Po instalaci rozšiřuje možnosti terminálu (příkazového řádku) a to tak, že umožňuje instalaci add-onu, vytváření nových projektů rovnou s přidáním rozšířeními a knihovny. Composer je ovládaný přes příkazový řádek příkazem composer a následnými dovětky, které plní požadované funkce. (getcomposer.org, 2015)

## XAMPP

XAMPP je program, který nabízí jednoduchou instalaci a správu Apache a MySQL. Na správu MySQL využívá phpMyAdmin. To se dá snadno použít na zřízení domácího serveru využitelného na vývoj a odzkoušení webových aplikací. Dále program nabízí například FileZilla a Mercury nebo další konfigurace. (webzpravodaj.cz, 2013)

## MySQL

MySQL je multiplatformní databázový systém, na kterém komunikace probíhá pomocí jazyka SQL. (Ponkrác, 2012) MySQL databáze se dá vytvořit například v open sourceovém programu MySQL Workbench od stejného výrobce. Tvorba v tomto programu bude rozvedena dále.

## NetBeans

NetBeans je vývojové prostředí, které umožňuje snadný vývoj aplikací v různých programovacích jazycích, které je možno doinstalovat z velké škály rozšíření.

Nástroj zajišťuje přehlednost a jednoduchou orientaci v souborech systému a v jednotlivých kódech.

Kód odděluje barevně, tak aby byl co nejpřehlednější. Po instalaci rozšíření na určitý programovací jazyk se text zvýrazňuje právě podle struktury kódu použitého jazyka.

## MySQL Workbench

MySQL Workbench je open sourceový nástroj využívaný na grafické modelování databáze. Další možností využití je například synchronizace vytvořeného modelu.

Dále bude představena grafická tvorba Entitně relačního diagramu, která se dá provádět právě v tomto programu.

## Vizual paradigm

Vizual paradigm je softwarový nástroj na návrh a tvorbu diagramů, UML, BPMN, ERD, DFD, SysML a mnoho dalších.

Dále bude představena grafická tvorba Use Case diagramu právě tak, jak je možné jej vytvořit v tomto programu.

## 3.2 Tvorba ERD

Na tvobru ERD se dá využít program MySQL Workbench, který byl již uveden. Tvorba probíhá přidáváním entit, atributů a relací mezi entitami.

### Entita

Entita obsahuje název, atributy, které určují její vlastnosti, a jednotlivé entity mohou být propojeny relacemi. Entita by také měla mít primární klíč pro identifikaci jejího záznamu.

### Atribut

- Primární klíč (žlutý klíč): je to hodnota, která jasně definuje záznam entity
- Povinný atribut (modrý bod s výplní): důležité informace, které musejí být vyplněny při vytvoření (jméno, příjmení)
- Nepovinný atribut (modrý bod bez výplně): informace, které se při vytvoření entity mohou, ale nemusí vyplňovat.
- Cizí klíč (červený bod): je hodnota, která obsahuje primární klíč cizího záznamu entity, se kterou je entita v relaci.

### Datové typy

Datové typy používané u atributů určují, jaká data se do atributů budou moci ukládat. Mezi nejpoužívanější patří například:

- INT: pro ukládání číslíc
- VARCHAR: pro ukládání znakového řetězce (je zde možné určit délku)
- TEXT: pro ukládání textového pole
- DATE: pro ukládání data (rok, měsíc, den, čas)
- BOOLEAN: pro ukládání výstupu dvoustavového přepínače

## Relace

Spojení uvádějící, že jsou entity ve vztahu. Relace může být neidentifikovatelná (přerušovaná čára), nebo identifikovatelná (nepřerušovaná čára). Každá relace má vlastní kardinalitu, která uvádí, v jakém vztahu k sobě entity jsou.

- 1:N k záznamu jedné tabulky je přiřazeno více záznamů tabulky druhé
- 1:1 k záznamu z jedné tabulky je přiřazen opět jeden záznam z tabulky druhé
- M:N k více záznamům jedné tabulky je přiřazeno více záznamů z tabulky druhé, toto spojení se většinou realizuje přes pomocnou tabulku, která spojení rozčlení na dva spoje typu 1:N

## 3.3 Tvorba Use Case

Na tvorbu Use Case diagramu je možné použít nástroj Vizual paradigm, který byl již uveden dříve v této kapitole. Use Case, neboli diagram případů užití, znázorňuje případnou funkcionalitu systému.

Prvky používané při tvorbě tohoto diagramu jsou:

### Aktor

Aktor komunikuje s jednotlivými funkcemi systému (Use Case). Aktorem může být například uživatel, čas nebo administrátor. Aktor se znázorňuje figurkou s popisem pod ní. (itnetwork.cz 2015) Aktor se s Use Casy propojuje asociací.

### Use Case

Use Case (případ užití) je shluk akcí, které vedou k jednomu určitému výsledku. Use Case je tudíž funkcionalita, kterou by navrhovaný systém měl umět. Use Case se znázorňuje nejčastěji oválem s popisem uprostřed. (itnetwork.cz 2015)

Případy užití se propojují dvěma vazbami:

- Include: propojuje dva Use Case tak, že první zahrnuje druhý
- Extend: propojuje dva Use Case tak, že první rozšiřuje druhý

Use Case diagram může obsahovat ještě vymezené hranice systému, které se znázorňují čtvercovým ohraničením. (Kučerová, 2011)

## 4 Návrh databáze a systému

Návrh databáze a funkcionality systému byla konzultovaná a zadávaná firmou TOPGEOSYS.

### 4.1 Požadavky na databázi

Hlavním požadavkem na databázi bylo ukládání potřebných dat pro finanční chod firmy. Databáze by se měla soustředit na objednávky firmy, které se budou dále členit na prodej, výpůjčky a opravy.

V objednávkách bude zaznamenáváno, jaký zaměstnanec daný druh objednávky zpracoval a pro koho byla zakázka vyhotovena. Samotná objednávka bude obsahovat kupované, opravované a vypůjčované produkty a doplňky, které firma nabízí a další podrobnosti, jako je den vystavení, konec záruční lhůty a další.

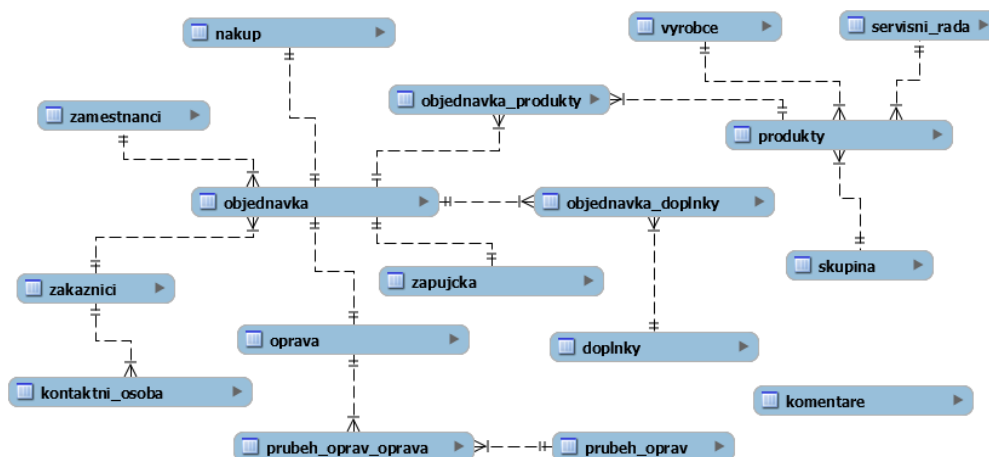
V databázi bude ukládáno vše o produktech (výrobce, cena, výrobní číslo, typ, servisní řada a další odborné informace), vše o doplncích (název, popis, cena), vše o zaměstnancích (jméno, příjmení a jejich kontakty), vše o zákaznících (ICO, DIC, kontaktní údaje, adresy) a vše o kontaktních osobách jednotlivých zákaznických firem (jejich jméno, příjmení, kontaktní údaje a další informace).

Pro návrh a vygenerování databáze byl využit databázový systém MySQL a jeho grafický nástroj MySQL Workbench.

### 4.2 Návrh databáze

Návrh databáze probíhal dle předem určených požadavků.

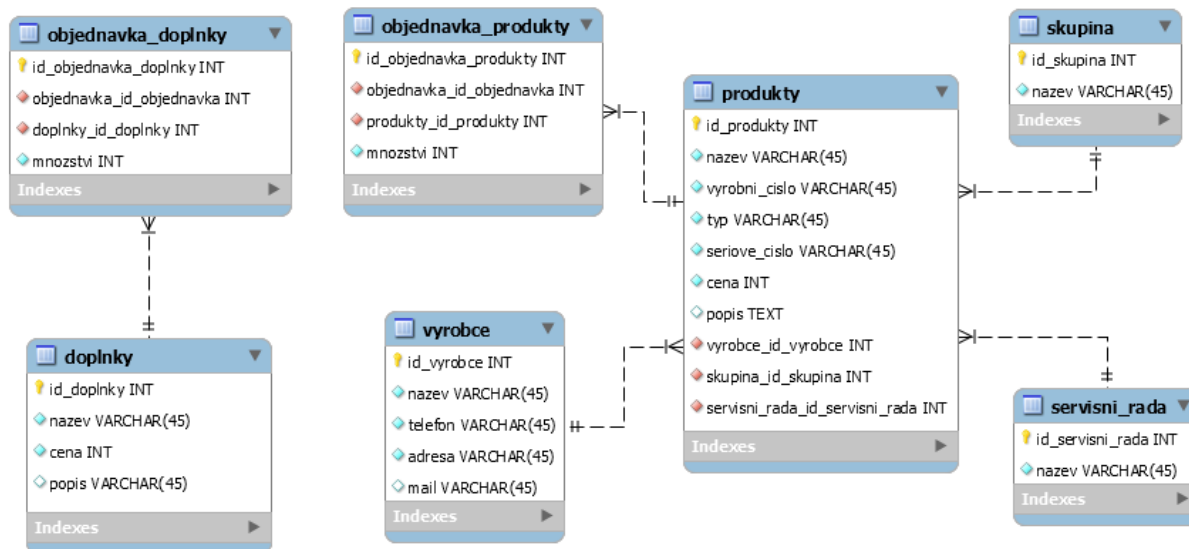
#### 4.2.1 ERD



Obr. 5 Entity a jejich vazby



ERD se skládá ze sedmnácti tabulek, entity a jejich vazby jsou zobrazeny na obrázku číslo pět nahoře. Dále bude rozebráno a popsáno celé ERD, které bylo rozděleno na tematické části pro lepší přehlednost a znázornění.



Obr. 6 Část ERD: doplňky, produkty a propojení

## Skupina

- Atributy tabulky: id\_skupina (INT), nazev (VARCHAR(45))
- Primární klíč: id\_skupina
- Popis (funkcionalita): tabulka skupin je připojena k tabulce produktů, aby určovala, v jaké skupině se daný přístroj vyskytuje

## Vyrobce

- Atributy tabulky: id\_vyrobce (INT), nazev (VARCHAR(45)), telefon (VARCHAR(45)), adresa (VARCHAR(45)), mail (VARCHAR(45))
- Primární klíč: id\_vyrobce
- Popis (funkcionalita): tabulka výrobců uchovává všechny výrobce produktů, které firma nabízí a rovněž zaznamenává jejich kontaktní údaje. Tabulka je také propojena s tabulkou produktů.

## Servisni\_rada

- Atributy tabulky: id\_servisni\_rada (INT), nazev (VARCHAR(45))
- Primární klíč: id\_servisni\_rada
- Popis (funkcionalita): podobný princip jako u předchozí tabulky skupin, je rovněž připojena na produkty a určuje jejich servisní řadu.

## Produkty

- Atributy tabulky: id\_produkty (INT), nazev (VARCHAR(45)), vyrobní\_cislo (VARCHAR(45)), typ (VARCHAR(45)), seriove\_cislo (VARCHAR(45)), cena (INT), popis (TEXT), vyrobce\_id\_vyrobce (INT), skupina\_id\_skupina (INT), servisni\_rada\_id\_servisni\_rada (INT)
- Primární klíč: id\_produkty
- Popis (funkcionalita): tato tabulka zaznamenává (ukládá) všechny produkty, které firma nabízí. Uchovává o nich všechny údaje (název, typ, cenu a další). Tabulka je napojena přes pomocnou tabulku M:N na tabulku objednávek tak, aby v objednávce byly zaznamenány tyto přístroje.

## Doplňky

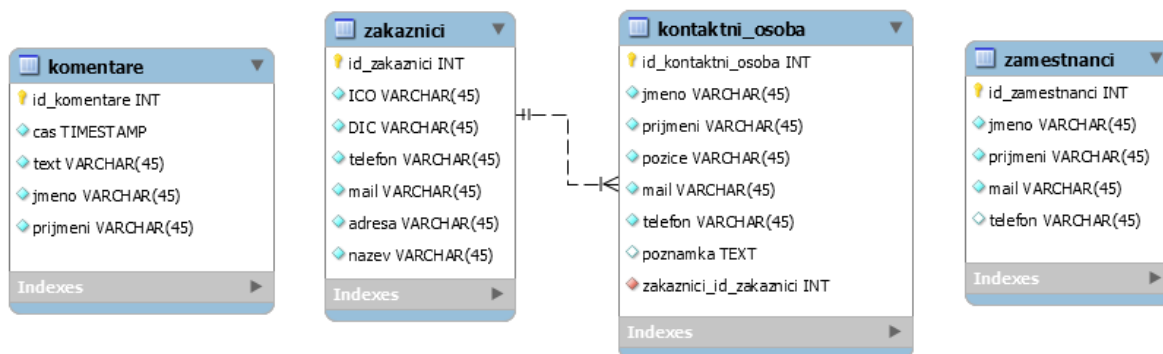
- Atributy tabulky: id\_doplňky (INT), nazev (VARCHAR(45)), cena (INT), popis (VARCHAR(45))
- Primární klíč: id\_doplňky
- Popis (funkcionalita): tabulka doplňků ukládá všechny informace o nabízených doplňcích firmy a ukládá jejich informace. Je rovněž napojena přes M:N na tabulku objednávek.

## Objednavka\_produkty

- Atributy tabulky: id\_objednavka\_produkty (INT), objednavka\_id\_objednavka (INT), produkty\_id\_produkty (INT), mnozstvi (INT)
- Primární klíč: id\_objednavka\_produkty
- Popis (funkcionalita): pomocná tabulka M:N spojující tabulku objednávek a tabulku produktů. Tabulka obsahuje primární klíče obou spojovaných tabulek a množství, které určuje počet objednávaných produktů. Během vývoje byl do tabulky přidán i samostatný primární klíč tabulky kvůli problémům s přidáváním do objednávek.

## Objednavka\_doplňky

- Atributy tabulky: id\_objednavka\_doplňky (INT), objednavka\_id\_objednavka (INT), doplňky\_id\_doplňky, mnozstvi (INT)
- Primární klíč: id\_objednavka\_doplňky
- Popis (funkcionalita): tato tabulka pracuje na stejném principu jako předchozí, jen s tím rozdílem, že do objednávky přidává místo produktů doplňky. Rovněž obsahuje primární klíče a množství. Stejně tak byl přidán samostatný primární klíč ze stejných důvodů.



Obr. 7 Část ERD: zaměstnanci, zákazníci, komentáře a propojení.

### Zamestnanci

- Atributy tabulky: `id_zamestnanci` (INT), `jmeno` (VARCHAR(45)), `prijmeni` (VARCHAR(45)), `mail` (VARCHAR(45)), `telefon` (VARCHAR(45))
- Primární klíč: `id_zamestnanci`
- Popis (funkcionalita): tabulka zaznamenává zaměstnance firmy, jejich jméno, příjmení a rovněž kontaktní údaje (telefon a e-mail). Také je napojena na tabulku objednávek a to, aby určovala, který zaměstnanec zpracovával danou objednávku.

### Zakaznici

- Atributy tabulky: `id_zakaznici` (INT), `ICO` (VARCHAR(45)), `DIC` (VARCHAR(45)), `telefon` (VARCHAR(45)), `mail` (VARCHAR(45)), `adresa` (VARCHAR(45)), `nazev` (VARCHAR(45))
- Primární klíč: `id_zakaznici`
- Popis (funkcionalita): tabulka uchovává všechny potřebné údaje o zákaznících, kteří kdy ve firmě měli zadanou jakoukoli objednávku. Tudíž je taktéž napojena na tabulku objednávek, aby určovala, kterému zákazníkovi objednávka přísluší. Dále je tabulka rozšířena napojením na tabulku kontaktních osob, aby zdražovala všechny osoby spadající pod jednoho zákazníka (zákaznickou firmu).

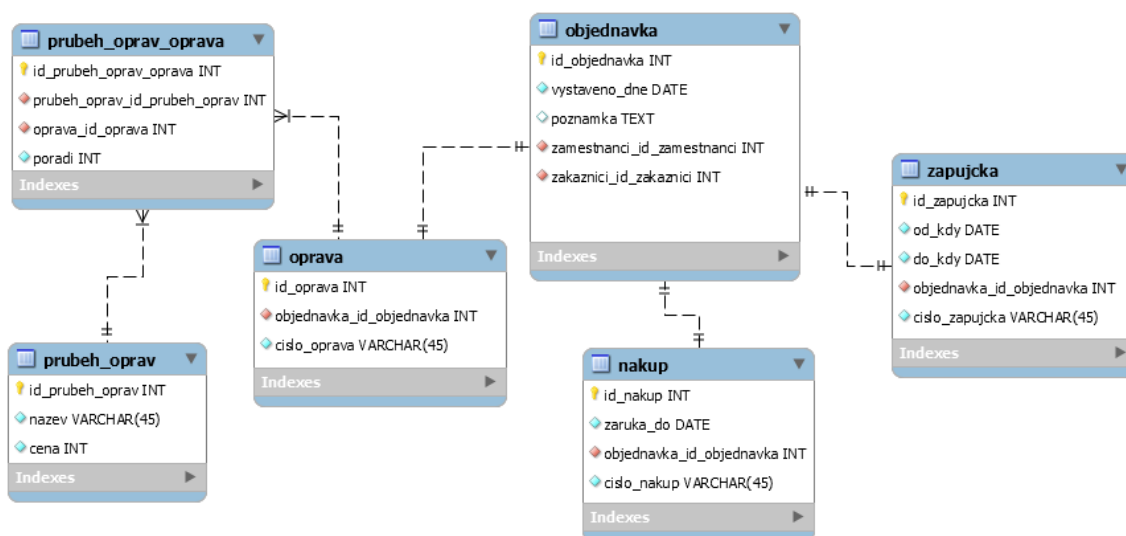
### Kontaktni\_osoba

- Atributy tabulky: `id_kontaktni_osoba` (INT), `jmeno` (VARCHAR(45)), `prijmeni` (VARCHAR(45)), `pozice` (VARCHAR(45)), `mail` (VARCHAR(45)), `telefon` (VARCHAR(45)), `poznamka` (TEXT), `zakaznici_id_zakaznici` (INT)
- Primární klíč: `id_kontaktni_osoba`
- Popis (funkcionalita): tabulka kontaktních osob rozšiřuje výše uvedenou tabulku zákazníků uchováním všech kontaktních osob a uchovává o nich po-

třebné údaje, jako jsou jejich jméno, příjmení, pozice, telefon, e-mail nebo případně poznámka.

## Komentare

- Atributy tabulky: id\_komentare (INT), cas (TIMESTAMP), text (VARCHAR(45)), jmeno (VARCHAR(45)), prijmeni (VARCHAR(45))
- Primární klíč: id\_komentare
- Popis (funkcionalita): tabulka komentářů je samostatně stojící tabulka uchovávající jméno, příjmení, čas (čas se zde automaticky generuje po napsání a odeslání zprávy) a text zprávy (komentáře). Komentář bude psaný na úvodní straně systému i bez přihlášení, proto tabulka stojí samostatně.



Obr. 8 Část ERD: objednávka (nákup, zápůjčka, oprava) a propojení

## Objednavka

- Atributy tabulky: id\_objednavka (INT), vystaveno\_dne (DATE), poznamka (TEXT), zamestnanci\_id\_zamestnanci (INT), zakaznici\_id\_zakaznici (INT)
- Primární klíč: id\_objednavka
- Popis (funkcionalita): tabulka objednávek obsahuje pouze atributy určující den vystavení objednávky a případnou poznámku. Nejdůležitější funkcí této tabulky je sdružení všech informací o objednávce z okolních tabulek, tudíž je připojena přes pomocné M:N tabulky k produktům a doplňkům. Dále je připojena vazbou 1:N k tabulkám zákazníků a zaměstnanců. Tabulka je používána pro sdružování informací. V samotném systému se uživatel k tabulce dostane pouze přes podružné tabulky nákupů, zápůjček a oprav, které jsou k objednávce připojeny přes netradiční vazbu 1:1 a přebírají veškeré údaje a přidávají některé vlastní.

## Nakup

- Atributy tabulky: id\_nakup (INT), zaruka\_do (DATE), objednavka\_id\_objednavka (INT), cislo\_nakup (VARCHAR(45))
- Primární klíč: id\_nakup
- Popis (funkcionalita): tabulka nákupů je podružná tabulka objednávky, která přebírá všechny její údaje a přidává vlastní (datum vypršení záruky a číslo nákupu využívané pro vyhledávání).

## Zapujcka

- Atributy tabulky: id\_zapujcka (INT), od\_kdy (DATE), do\_kdy (DATE), objednavka\_id\_objednavka (INT), cislo\_zapujcka (VARCHAR(45))
- Primární klíč: id\_zapujcka
- Popis (funkcionalita): tabulka zápůjček má stejnou funkcionalitu jako tabulka nákupů, rovněž přebírá informace z tabulky objednávek a přidává pár vlastních údajů potřebných pro zapůjčování produktů firmou, a to data od kdy a do kdy je produkt zapůjčen a rovněž číslo zápůjčky využívané pro orientaci a vyhledávání.

## Oprava

- Atributy tabulky: id\_oprava (INT), objednavka\_id\_objednavka (INT), cislo\_oprava (VARCHAR(45))
- Primární klíč: id\_oprava
- Popis (funkcionalita): tabulka oprav je poslední z podružných tabulek objednávek. Rovněž přebírá informace z objednávek a přidává vlastní, tentokrát pouze číslo opravy. Na rozdíl od výpůjček a nákupů jsou opravy propojeny ještě s jednou tabulkou obsahující průběhy oprav, a to přes pomocnou tabulku vazbou M:N, ze které berou informace, jak bylo postupováno při opravách.

## Prubeh\_oprav

- Atributy tabulky: id\_prubeh\_oprav (INT), nazev (VARCHAR(45)), cena (INT)
- Primární klíč: id\_prubeh\_oprav
- Popis (funkcionalita): tabulka sdružuje jednotlivé typy druhů oprav (čištění, kalibrace a další) a uchovává jejich cenu. Tabulka je připojena k pomocné tabulce vazbou M:N, aby tyto informace předala tabulce oprav.

## Prubeh\_oprav\_oprava

- Atributy tabulky: id\_prubeh\_oprav\_oprava (INT), prubeh\_oprav\_id\_prubeh\_oprav (INT), oprava\_id\_oprava (INT), poradi (INT)
- Primární klíč: id\_prubeh\_oprav\_oprava

- Popis (funkcionalita): tato tabulka je pomocná s vazbou M:N spojující tabulku průběhů oprav a opravy, tudíž obsahuje jejich primární klíče. Navíc obsahuje atribut pořadí určující, v jakém sledu byly vykonávány procedury při opravě. Stejně jako u tabulek spojujících produkty a doplňky s opravou byl i zde přidán samostatný primární klíč.

### 4.3 Požadavky na funkce systému

Hlavním požadavkem na funkcionální systém byla správa objednávek, jakožto prodej, opravy a zápůjčky. Důležitou funkcí v každé objednávce bylo přiřazování zákazníků, zaměstnanců, produktů a doplňků, popřípadě u oprav přidání průběhu opravy.

Důležitá byla jednoduchá správa všech položek v systému, jednoduché vyhledávání a přihlášení do systému, které se bude nacházet na hlavní straně ještě s případnými zprávami pro zaměstnance.

V systému bylo vyžadováno vytváření (přidávání) nových položek, zaměstnanců, zákazníků a všech objednávek. Stejně tak byla důležitá jejich úprava, jako je například úprava ceny u produktů, změny kontaktů zaměstnanců nebo zákazníků a změny dat záruk nebo prodloužení dob vypůjčení určitých kusů produktů.

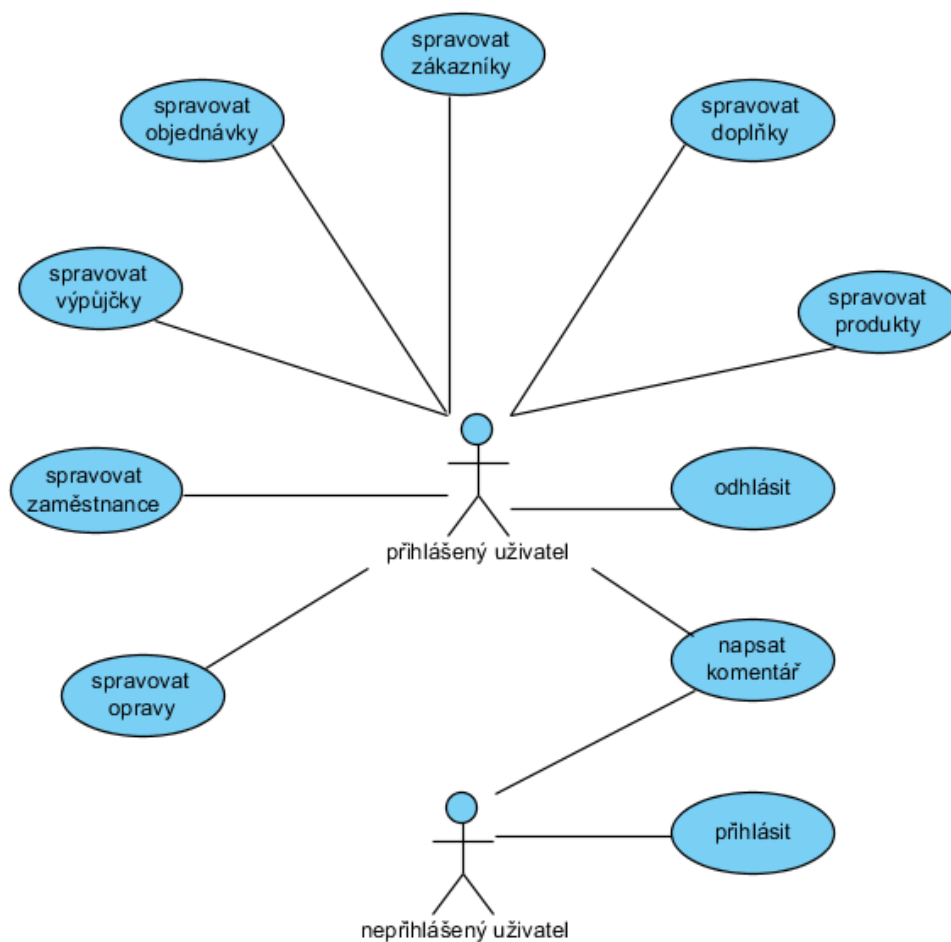
V neposlední řadě je vyžadován tisk objednávek ve formátu PDF, jako předávání fakturace zákazníkovi.

### 4.4 Návrh systému

Návrh systému probíhal dle předem určených požadavků firmy a pro jejich znázornění byl vytvořen Use Case diagram a na jeho tvorbu využit program Vizual paradigm.

#### 4.4.1 Use Case diagram

Funkcionalita systému byla znázorněna Use Case diagramem.



Obr. 9 Use Case diagram funkcí systému

Use Case diagram se skládá ze sedmi hlavních bloků a těmi jsou:

### **Správa doplňků**

Zahrnuje prohlídku všech nabízených doplňků firmy, jejich vyhledávání, přidávání a úpravu.

### **Správa produktů**

Obsahuje výpis všech produktů a vyhledávání jednotlivých produktů. U produktů je možné vytvořit nové nebo upravovat stávající skupiny, servisní řady a výrobce a vyplnit všechny informace. Vytváření a úprava samotných produktů pak navazuje na právě vytvořené vedlejší informace (skupiny, výrobce, servisní řady), které jsou zapotřebí vyplnit při zadávání parametrů.

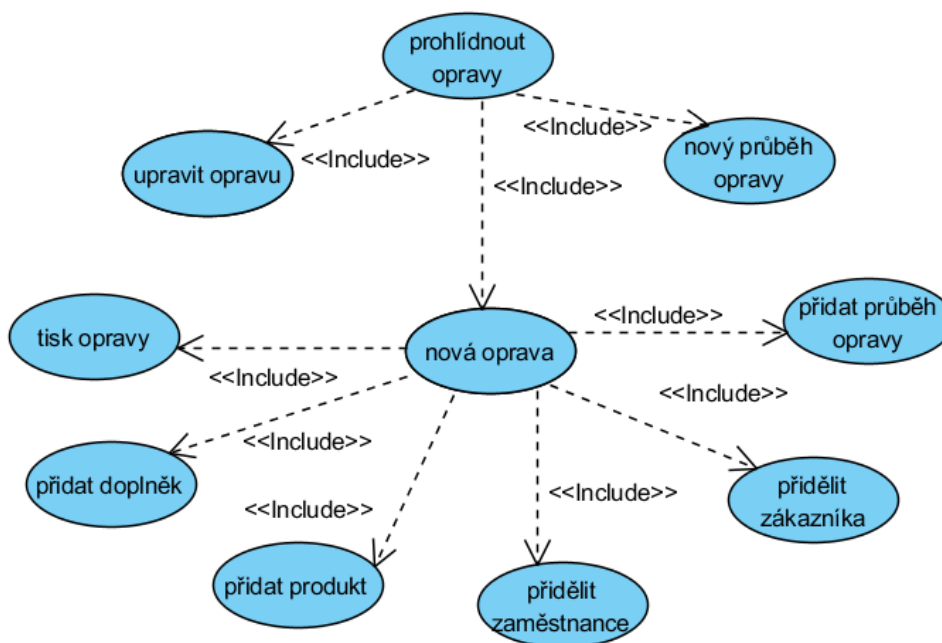
## Správa zaměstnanců

Umožňuje prohlídku zaměstnanců a jejich vyhledávání s možností úpravy stávajících (změna kontaktních údajů, jako je telefon nebo e-mail) a vytváření nově přijatých zaměstnanců.

## Správa zákazníků

Správa a prohlídka umožňuje vyhledání jednoho konkrétního zákazníka a zobrazit si všechny jeho podrobnosti i s výpisem všech přiřazených kontaktních osob pod něj spadajících. Je možné vytvářet nové, jak zákazníky, tak kontaktní osoby, a rovnou je přiřazovat k firmám. Součástí je i úprava všech údajů, stejně jako u vytváření samotných zákazníků i jejich kontaktních osob.

Nejdůležitější bloky funkcionality informačního systému jsou následující tři.  
Pro ukázkou rozepsaný Use Case diagram správy oprav.



Obr. 10 Use Case: blok správy oprav

## Správa nákupů

Správa nákupů je jedna ze tří nejdůležitějších funkcí systému (nákup, oprava, výpůjčka). Je zde možná prohlídka všech předchozích objednávek (nákupů) a vyhledání konkrétního podle čísla objednávky (nákupu). Vytváření nového nákupu a jeho úprava umožňuje vyplnění všech potřebných údajů, přiřazení zákazníka a zaměstnance, který objednávku zpracovával. Po vytvoření se zpřístupní přidávání objednaných produktů nebo doplňků do objednávky a závěrečný tisk ve formátu



PDF, ve kterém je vypočítaná celková cena a jsou na něm vyobrazeny všechny důležité údaje objednávky.

### **Správa oprav**

Správa a prohlídka oprav umožňuje uživateli vyhledávání podle čísla opravy a prohlídku jednotlivých detailů o opravě. Při vytváření a úpravě uživatel zadává všechna důležitá data opravy a přiděluje zákazníka a zaměstnance k opravě, stejně tak jak je tomu u nákupů. Po vytvoření se rovněž otevírá možnost do opravy přidávat servisované produkty a použité přípravky na jeho opravu. Na rozdíl od nákupů je zde ještě možnost vytvářet průběhy oprav, které jdou následně přidělit k opravě a nastavit jim pořadí, tak aby bylo zjevné, jaká činnost byla při servisu prováděna a to v přesném pořadí. Po zadání všech informací je tu opět možnost celou opravu vytisknout ve formátu PDF.

### **Správa výpůjček**

Správa výpůjček pracuje na stejném principu jako opravy nebo nákupy, tudíž je zde prohlídka všech výpůjček, jak už vrácených, tak čerstvě vypůjčených produktů. Vytváření a úprava rovněž zaznamenává veškeré informace o výpůjčkách, jako jsou data vypůjčení nebo vrácení a přidává zákazníka, pro kterého je výpůjčka vystavena a zaměstnance, který ji zpracoval. Po vytvoření se sem přidávají produkty a doplňky a po vyplnění všech údajů lze výpůjčku rovněž vytisknout, a to ve stejném formátu PDF.

### **Další možnosti systému:**

- Napsat komentář  
Komentáře neboli zprávy zaměstnancům se nacházejí na úvodní straně systému a po vyplnění jména a příjmení lze díky komentářům umístit na hlavní stranu vlastní text s přidaným datem kdy byl text odeslán.
- Přihlásit  
Nepřihlášený uživatel má v systému možnost pouze číst a zapisovat nové zprávy zaměstnancům nebo připomínky firmy. Proto je tu přihlašování, aby se zpřístupnily veškeré možnosti systému a zaměstnanec mohl využívat systém v celém jeho rozsahu odkudkoli hned po přihlášení.
- Odhlásit  
Po přihlášení do systému se z něj lze následovně odhlásit, a to zejména proto, aby bylo zachováno soukromí firmy, a to když se například zaměstnanec přihlásí na veřejném místě.

## 5 Implementace

Na implementaci samotného informačního systému bylo využito více technologií a to tak, aby co nejlépe splnily předem stanovený návrh funkcionality a splňovaly tak důležité požadavky.

### 5.1 Použité technologie

Základní použitou technologií je Nette Framework, na které stojí celý základ vytvořeného informačního systému.

Dalšími použitými technologiemi jsou HTML, CSS a již zmiňované MySQL na tvorbu databáze.

#### 5.1.1 Nette Framework

Tento projekt byl vytvářen ze základního Nette sandboxu, který již obsahoval všechna potřebná data pro správnou funkčnost frameworku.

Práce byla rozdělena na několik presenterů, které obsahují používané třídy a funkce v jednotlivých templatech.

Každý jeden presenter pracuje právě s jednou podsložkou template, která obsahuje více souborů s příponou latte a to:

- Homepage: obsahuje výpisy všech tabulek, které po rozkliknutí přesměrují uživatele níže na doplňky, objednávky, zákazníky a dále
- Doplnky: veškeré funkce ohledně přidávání úpravy a zobrazení doplňků
- Produkty: veškerá funkcionality produktů
- Zamestnanci: přidávání, zobrazení a úprava zaměstnanců
- Zakaznici: veškeré funkce pro tabulky zákazníků a kontaktních osob
- Objednavky: vše pro správu objednávek (nákupů)
- Vypujcky: vše pro správu výpůjček
- Opravy: vše pro správu oprav
- Error: práce s chybami
- Moje: stará se o funkce tisku PDF z objednávek (nákupů), výpůjček a oprav

Zde bude následovat ukázka a popis implementace správy doplňků, která je vyobrazena v Use Case diagramu v předchozí kapitole.

Základní render všech doplňků:

```
public function renderDoplanky() {
    $this->template->posts=$this->database->table('doplanky');
}
```

Vyhledávání doplňků, které je umístěno v presenteru Homepage, obsahuje formulář pro zadání názvu doplňku, který je předán do další funkce, která přesměruje uživatele na template vyhledávání a spustí render s regulárním výrazem názvu doplňku:

```
protected function createComponentVyhledaniDoplankyForm()
{
    $form=new Nette\Application\UI\Form;
    $form->addText('nazev')
    ->setDefaultValue('Vyhledání podle názvu')
    ->setRequired();
    $form->addSubmit('send','Vyhledej');
    $form->onSuccess[]=$this->VyhledaniDoplankyFormSucceeded;
    return $form;
}

public function VyhledaniDoplankyFormSucceeded($form)
{
    $values=$form->getValues();
    $this->redirect('doplankyVyhledej', array('data' => $values['nazev']));
}

public function renderDoplankyVyhledej($data)
{
    $this->template->posts=$this->database->table('doplanky')
    ->where('nazev LIKE ?', '%'.$data.'%');
}
```

Další částí jsou templaty. Doplňky mají dva templaty ve složce Homepage. Jeden je tu na zobrazení všech doplňků a druhý podobný na zobrazení vyhledaných doplňků.

Pak jsou další tři templaty v podsložce doplňky. První se stará o to, aby po kliknutí na jeden doplněk byly zobrazeny veškeré informace právě o rozkliknutém doplňku. Další dva jsou tu pouze proto, aby spouštěly funkce potřebné pro přidání a úpravu doplňků.

Funkce právě na tyto akce jsou následující:

**Render rozkliknutého doplňku:**

```
public function renderDoplanky($postId)
{
    $post = $this->database->table('doplanky')->get($postId);
    if (!$post) {
        $this->error('Doplněk nebyl nalezen.');
```

```
    }  
    $this->template->post=$post;  
}
```

### Formulář pro přidání (úpravu) doplňku:

```
protected function createComponentDoplankyForm()  
{  
    $form=new Nette\Application\UI\Form;  
    $form->addTextArea('nazev', 'Název:')  
->setRequired();  
    $form->addTextArea('cena', 'Cena:')  
->setRequired();  
    $form->addTextArea('popis', 'Popis:')  
->setRequired();  
    $form->addSubmit('send', 'Uložit');  
    $form->onSuccess[]=$this->doplankyFormSucceeded;  
    return $form;  
}
```

### Uložení doplňku do databáze:

```
public function doplankyFormSucceeded ($form)  
{  
    $values=$form->getValues();  
    $postId=$this->getParameter('postId');  
    if ($postId) {  
        $post=$this->database->table('doplanky')->get($postId);  
        $post->update($values);  
    } else {  
        $post=$this->database->table('doplanky')->insert($values);  
    }  
    $this->flashMessage('Doplněk byl uložen.', 'success');  
    $this->redirect('doplanky', $post->id_doplanky);  
}
```

### Funkce pro editaci doplňku, která přesune data z editovaného doplňku do formuláře pro úpravu:

```
public function actionEdit($postId)  
{  
    $post=$this->database->table('doplanky')->get($postId);  
    if (!$post) {  
        $this->error('Doplněk nebyl nalezen.');    }  
}
```

```
$this['doplukyForm']->setDefaults($post->toArray());  
}
```

### 5.1.2 CSS

V projektu byl použit CSS právě na úpravu zobrazení, například:

- Různé nápisy a nadpisy:

```
#napis{  
font-family: tahoma, Helvetica, sans-serif;  
font-size: 15px;  
font-weight: bold;  
color: #000000;  
padding-top: 10px;  
padding-left: 0px;  
margin-left: 0px;  
margin-right: 10px;  
}
```

- popřípadě tabulky:

```
table{  
border-width: 1px;  
border-color: #666;  
border-style: solid;  
font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;  
font-size: 12px;  
}
```

```
td{  
padding: 3px;  
border: 1px #ddd dotted;  
}
```

```
thead{  
background-color: #056597;  
color: #FFF;  
text-transform: uppercase;  
font-weight: bold;  
}
```

- nebo i pro možné schování textu:

```
#schovat {  
display: none  
}
```

### 5.1.3 HTML

V projektu byl použit HTML například v šabloně pro tisk PDF objednávek:

```
<html>
<head>
...
h1 {display: none;}
...
</head>
<body>
...
<h3>Objednané doplňky</h3>
{foreach $post->objednavka->related('objednavka_doplanky')
as $doplanky}
<div>Název: {$doplanky -> doplnky ->nazev}</div>
<div>Cena: {$doplanky -> doplnky ->cena}</div>
<div>Množství: {$doplanky -> mnozstvi}</div>
<h1> {$x=$doplanky -> mnozstvi} </h1>
<h1> {$y=$doplanky -> doplnky ->cena} </h1>
<h1> {$celkem=$celkem+($y*$x)} </h1>
{/foreach}
...
</body>
</html>
```

### 5.1.4 MySQL

Jak již bylo zmiňováno, MySQL bylo zvoleno jako systém použitý na tvorbu databáze v tomto projektu.

Stejně tak v projektu bylo použito pár SQL dotazů, ať už za zjednodušené pomoci Nette Frameworku, nebo čistým SQL.

Ukázka SQL dotazu na vložení hodnot do tabulky nákupů:

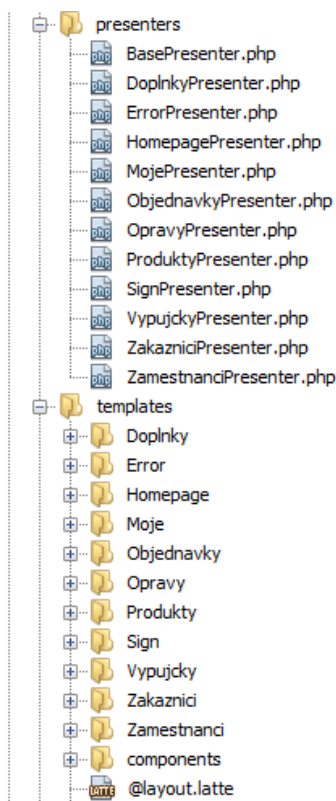
```
$post2=$this->database->query('INSERT INTO nákup(cislo_nakup,zaruka_do,objednavka_id_objednavka) VALUES ('. $cislo.', '. $do.', '. $idcko. ');');
```

Ukázka SQL dotazu vložení hodnot z formuláře do tabulky objednávek pomocí Nette Frameworku:

```
$post=$this->database->table('objednavka')
->insert($values);
```

## 5.2 NetBeans

Po čas vývoje informačního systému byl využíván nástroj NetBeans s přidáním rozšíření na programovací jazyk PHP, na kterém stojí Nette Framework, a to ve verzích 8.0.1 a 8.0.2.



Obr. 11 NetBeans: orientace v souborech systému (presentery a templaty)

## 5.3 MySQL Workbench

Na modelování databáze byl využit open source nástroj MySQL Workbench, který slouží ke grafické tvorbě databáze.

Zde si ukážeme kroky potřebné na tvorbu a synchronizaci databáze:

- Spustíme program MySQL Workbench
- Vytvoříme nový model
- Do vytvořeného modelu přidáme diagram
- Nyní pomocí boční nabídky přidáváme do diagramu tabulky, atributy, primární klíče, datové typy a jejich propojení
- Po dokončení diagramu můžeme databázi synchronizovat
- Nastavíme důležitá data, jako je hostovské jméno (IP adresa), port, jméno uživatele a další

- Po navázání spojení s databází zvolíme, jaké námi vytvořené schéma chceme přidat (v případě, že máme pouze jedno, vybereme jej)
- Po provedení nám program vypíše veškeré SQL dotazy všech přidávaných tabulek a nabídne nám provedení operace
- Po provedení již máme námi vytvořený diagram, uložený v podobě databáze na zadaném serveru. Dále databázi můžeme spravovat například přes php-MyAdmin nebo Adminer. Nebo můžeme databázi přehrát, upravit, vylepšit opět stejným postupem

Ukázka SQL dotazu, který program generuje z graficky vymodelovaného diagramu (tabulek):

```
CREATE TABLE IF NOT EXISTS `TOPGEO-
SYS2015.7.posledni`.`zakaznici` (
  `id_zakaznici` INT(11) NOT NULL AUTO_INCREMENT,
  `ICO` VARCHAR(45) NOT NULL,
  `DIC` VARCHAR(45) NOT NULL,
  `telefon` VARCHAR(45) NOT NULL,
  `mail` VARCHAR(45) NOT NULL,
  `adresa` VARCHAR(45) NOT NULL,
  `nazev` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id_zakaznici`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_general_ci;
```

## 5.4 Tisk objednávek do PDF

Na tisk objednávek do PDF byl použit add-on<sup>4</sup> do Nette Frameworku PdfResponse dev-master, který využívá knihovnu<sup>5</sup> mPDF.

Na instalaci add-onu byl využit Composer, který již byl představen v kapitole uvedené výše.

Příkaz pro instalaci použitého add-onu je:

```
composer require jkuchar/pdfresponse:dev-master
```

Po instalaci je možné využívat přídavek ve vlastním projektu. V tomto byl využit na tvorbu PDF z šablony objednávek:

```
use PdfResponse\PdfResponse;
```

---

<sup>4</sup> Add-on je přídavek (rozšíření) vytvářený uživateli

<sup>5</sup> Knihovna obsahuje funkce, které již byly naprogramované, a umožňuje jejich opětovné použití



Samotná funkce na vytváření PDF vypadá následovně:

```
public function actionPdfObjednavka($id)
{
    $appDir=Nette\Environment::getVariable('appDir');
    $template=$this->createTemplate();
    ->setFile($appDir."/templates/M/ pdf.latte");
    $template->post=$post=$this->database->table('nakup')
    ->get($id);
    $pdf = new PDFResponse($template);
    $this->sendResponse($pdf);
}
```

Na tvorbu byl využit upravený template, který byl použit v samotném systému. Bylo odstraněno menu, grafika stránky a další součásti, tak aby byl zachován pouze celý výpis objednávky a PDF bylo přehledné.

## 5.5 Problémy

Při implementaci systému vzniklo několik problémů a chyb, tyto problémy byly většinou vyřešeny nebo se je povedlo obejít.

Pár vzniklých problémů:

- Při zadávání nákupů, výpůjček a oprav bylo potřeba ukládat do databáze dvě tabulky z jednoho formuláře zároveň. Tento problém byl vyřešen SQL dotazem, ale při úpravě bylo nutno rozdělit úpravu na dvě samostatné části, kde v jedné se upravovala tabulka objednávek a v druhé tabulka nákupů, výpůjček nebo oprav.
- Dalším problémem bylo přidávání produktů a doplňků do objednávek přes M:N tabulky. Když vyplynula chyba při úpravě množství, přímá příčina této chyby nebyla nalezena, ale chybu bylo možno obejít přidáním primárního klíče do pomocných tabulek a použitím rozdílných mechanik.
- Problém byl také ve vyhledávání jednotlivých objednávek (nákupů, oprav a zápůjček) podle jejich čísla. Původně byla všechna čísla evidovaná v tabulce objednávek, to se ale nepodařilo spojit s podružnými tabulkami, jelikož vazba je zaznamenána právě tam. Tak byl problém vyřešen rozčleněním čísla objednávky na jednotlivá čísla (číslo nákupu, číslo opravy, číslo zápůjčky), která byla následně ukládána právě do podružných tabulek.

Většina chyb a problémů byla malého charakteru a byla vyřešena výpisem laděnky používané v Nette Frameworku nebo ochotou lidí odepisovat na dotazy položené na fóru zabývajícím se právě tímto frameworkem.

## 6 Diskuze

### 6.1 Výsledky

Výsledkem tohoto projektu je informační systém s databází obsahující sedmnáct tabulek pro uchování potřebných odborných informací pro firmu TOPGEOSYS zabývající se prodejem měřících přístrojů. Tabulky jsou zejména zaměřeny na vytvoření objednávek, jakožto nákupů, oprav a výpůjček z důležitých uložených informací o zaměstnancích, zákaznících, produktech a doplňcích, které je možno libovolně spravovat a upravovat.

Tabulky byly sestavovány dle konzultovaných informací s firmou a dodaných dat v podobě tabulek programu Excel, tak aby všechny atributy mohly být převedeny do databázové verze. Na vytvoření databáze byl zvolen systém MySQL a jeho grafický program MySQL Workbench.

Informační systém byl implementován za pomoci Nette Frameworku s přístupem do zmiňované databáze. Systém plní ekonomické požadavky na pořizovací cenu a převážnou část předem vytyčených cílů, tak aby bylo možné shromážďovat všechny údaje firmy, a z těch následně tisknout faktury objednávek ve formátu PDF.

### 6.2 Možnosti rozšíření

Na tento informační systém se nabízí navázání spoustou rozšíření do kompletního propracovaného systému, který by ještě více usnadnil správu a chod firmy. Mohly by se zde rozšířit vyhledávací možnosti jednotlivých objednávek, to by uživateli nabízelo různě seřazené výpisy, které by mohly obsahovat objednávky jednotlivých zákazníků nebo naopak veškeré objednávky vyřízené právě jedním zaměstnancem. To by mohlo sloužit k hodnocení zaměstnanců a kontrole pracovní vyváženosti mezi personálem. Vyhledávací možnosti by bylo možné rozšířit u všech důležitých tabulek produktů, doplňků, zaměstnanců, zákazníků, nákupů, oprav a výpůjček, které by umožňovaly rychlejší vyhledání, například podle data vystavení, jména zákazníka a dalších informací, což by mělo zapříčinit ulehčení vypátrání objednávek právě v případě, že zákazník ztratil fakturační údaje. Ukládané produkty by se daly rozšířit o fotografie, které by zajišťovaly lepší orientaci v přidávání do objednávek. Stejně tak by fotografie mohly být přiřazené k doplňkům nebo zaměstnancům. Systém by se mohl rozšířit tak, aby přístup měly i neoprávněné osoby, jakožto potenciální zákazníci, kteří by měli možnost nahlédnout právě do nabízených produktů a doplňků. Mohly by přibýt záznamy o jednotlivých prodaných kusech měřících přístrojů, které by ulehčily reklamační část, a bylo by možné zobrazit veškeré poruchy servisovaného produktu. Zaměstnanec by pak měl možnost zjistit, jaké úkony přístroj již prodělal, a bylo by možné snadněji přijít na to, kdy byla provedena například poslední kalibrace. Do systému by mohl být zaveden plánovací kalendář, který by měl za úkol spravovat události a připomínal personá-

lu, kdy vyprší lhůta na vrácení přístrojů, kdy je čas odeslání objednávek a umožňoval by zaměstnancům plánovat si osobní volno tak, aby na pracovišti vždy zůstal někdo z personálu a zajišťoval by se takto plynulý chod firmy. Možnost rozšířit systém na mobilní platformy, jakožto zjednodušení vzhledu a jednodušší vizualizaci na menších zobrazovacích prostředcích. Mobilní aplikace by umožňovala zaměstnancům připojit se na systém z kteréhokoliv místa s poskytnutím internetu a kontrolovat tak objednávky a chod firmy.

## 7 Závěr

V první části práce byly popisovány teoretické informace o systémech. Bylo popsáno vše, co spadá pod informační systém a jak se dělí podle jednotlivých kritérií. Také byly popsány modely pro tvorbu systému, zejména prototypováním, spirálou a vodopádem. Dále byly ukázány ERP systémy se zaměřením na komponenty dostupné na trhu a bylo představeno, co trh nabízí.

Před návrhem a tvorbou samotného systému byly nejprve vybrány technické prostředky, které byly následně v projektu používány. Pro návrh databáze byl zvolen program MySQL Workbench, tudíž na implementaci posloužil databázový systém MySQL. Pro implementaci systému se nejspíše nejvhodnějším jevil český Nette Framework a jako GUI byl použit NetBeans ve verzích 8.0.1 a později 8.0.2. Celý projekt byl vypracováván za pomoci domácího serveru XAMPP a k ladění chyb sloužila Nette Framework laděnka.

V první části tvorby bylo důležité navrhnout databázi, která pojme všechna důležitá data, která byla konzultována přímo ve firmě a nad kterou budou prováděny SQL dotazy. Důležitou částí bylo rozvrhnout datové typy všech atributů tabulek a propojení pro komunikaci mezi nimi. Na základě návrhu byla vytvořena funkční databáze, která se ale po čas průběhu implementace systému stále upravovala a ladila tak, aby bylo možné ji plně využívat systémem.

Druhou částí projektu bylo navrhnout a implementovat systém, který bude vytvořenou databázi používat. Jako první úkol bylo stanovit funkcionalitu systému za opětovné konzultace s firmou. Byl vytvořen návrh a Use Case diagram systému, ze kterého se při tvorbě vycházelo. Prvním úkolem bylo vytvoření funkcí, které by obstarávaly ukládání informací do připojené databáze. Vývoj pokračoval zobrazováním těchto dat a vytvořením výpisů objednávek a všech informací ve firmě. Poté byla do systému zakomponována úprava všech dat. V konečné fázi tvorby se přidávalo vyhledávání, tisk do formátu PDF a drobné dodělávky. Víím, že systém není plně dokonalý, ale základní, předem danou funkcionalitu, splňuje.

V práci je popsán celý návrh ERD databáze a jsou zde popsány jednotlivé funkce vyobrazené Use Case diagramem, které systém obstarává. Dále jsou v projektu ukázky kódů jednotlivých použitých technologií, které byly použity při samotné tvorbě. V příloze pak naleznete celý entitně relační diagram databáze, který byl v projektu rozkouskovan, a stejně tak kompletní Use Case diagram funkcí. Příloha také obsahuje adresu, kde je systém nahrán, a přístupové jméno a heslo k jeho vyzkoušení.

## 8 Reference

- CASTRO, E. *HTML, XHTML a CSS: Národní průvodce tvorbou WWW stránek*. Vyd. 6. Brno: Computer Press, 2011, 438 s. ISBN 978-80-251-1531-2.
- GETCOMPOSER.ORG *Composer documentation*. [online]. 2015. [cit. 2015-05-15]. Dostupné z: <https://getcomposer.org/doc/01-basic-usage.md>
- HAUSER, M., HAUSER, T., WENZ, CH. *HTML a CSS: Velká kniha řešení*. Vyd. 1. Brno: Computer Press, 2006, 912 s. ISBN 80-251-1117-2.
- ITNETWORK.CZ *itnetwork sociální síť pro IT profesionály*. [online]. 2013. [cit. 2015-05-16]. Dostupné z: <http://www.itnetwork.cz/uml-use-case-diagram>
- KUČEROVÁ, H. *Vývoj informačních systémů 2011/2012: Use Case model*. [online]. 2011. Dostupné z: <http://info.sks.cz/users/ku/PRI/usecase.htm>
- NÁPLAVA, P. *Rozdělení IS*. [online]. 2012. [cit. 2015-05-01]. Dostupné z: <http://czm.fel.cvut.cz/vyuka/Y16ISP/Download/Prednaska06.pdf>
- NETTE.ORG *Nette Framework dokumentace*. [online]. 2015. [cit. 2015-05-15]. Dostupné z: <http://doc.nette.org/cs/2.3/>
- PONKRÁC, M. *PHP a MySQL bez předchozích znalostí*. Vyd. 1. Brno: Computer Press, 2012, 224 s. ISBN 978-80-251-1758-3.
- QI.CZ *informační systém QI*. [online]. 2015. Dostupné z: <http://www.qi.cz/prohlidka-systemu/z-ceho-se-qi-sklada/>
- RÁBOVÁ, I. *Podnikové informační systémy a technologie jejich vývoje*. Vyd. Tribun EU, 2008, 139s. ISBN 80-739-9599-9.
- RYBIČKA, J. *Informační systémy*. [online]. 2009. [cit. 2015-05-03]. Dostupné z: <https://akela.mendelu.cz/~rybicka/prez/infosyst.pdf>
- SODOMKA, P., KLČOVÁ, H. *Informační systémy v podnikové praxi*. Vyd. 2. Brno: Computer Press, 2011, 504 s. ISBN 978-80-247-3046-2.
- ŠMÍD, V. *Životní cyklus informačního systému*. [online]. 2002. [cit. 2015-05-14] Dostupné z: <http://www.fi.muni.cz/~smid/mis-zivcyk.htm>
- VARIO.CZ *Moduly a funkce software Altus Vario*. [online]. 2014. Dostupné z: <http://www.vario.cz/moduly/>
- WEBZPRAVODAJ.CZ *online marketing, webdesign, rady a návody*. [online]. 2013. Dostupné z: <http://www.webzpravodaj.cz/234-xampp-navod-server-ve-vlastnim-pocitaci/>

# Přílohy

## **A Projekt k prohlídce**

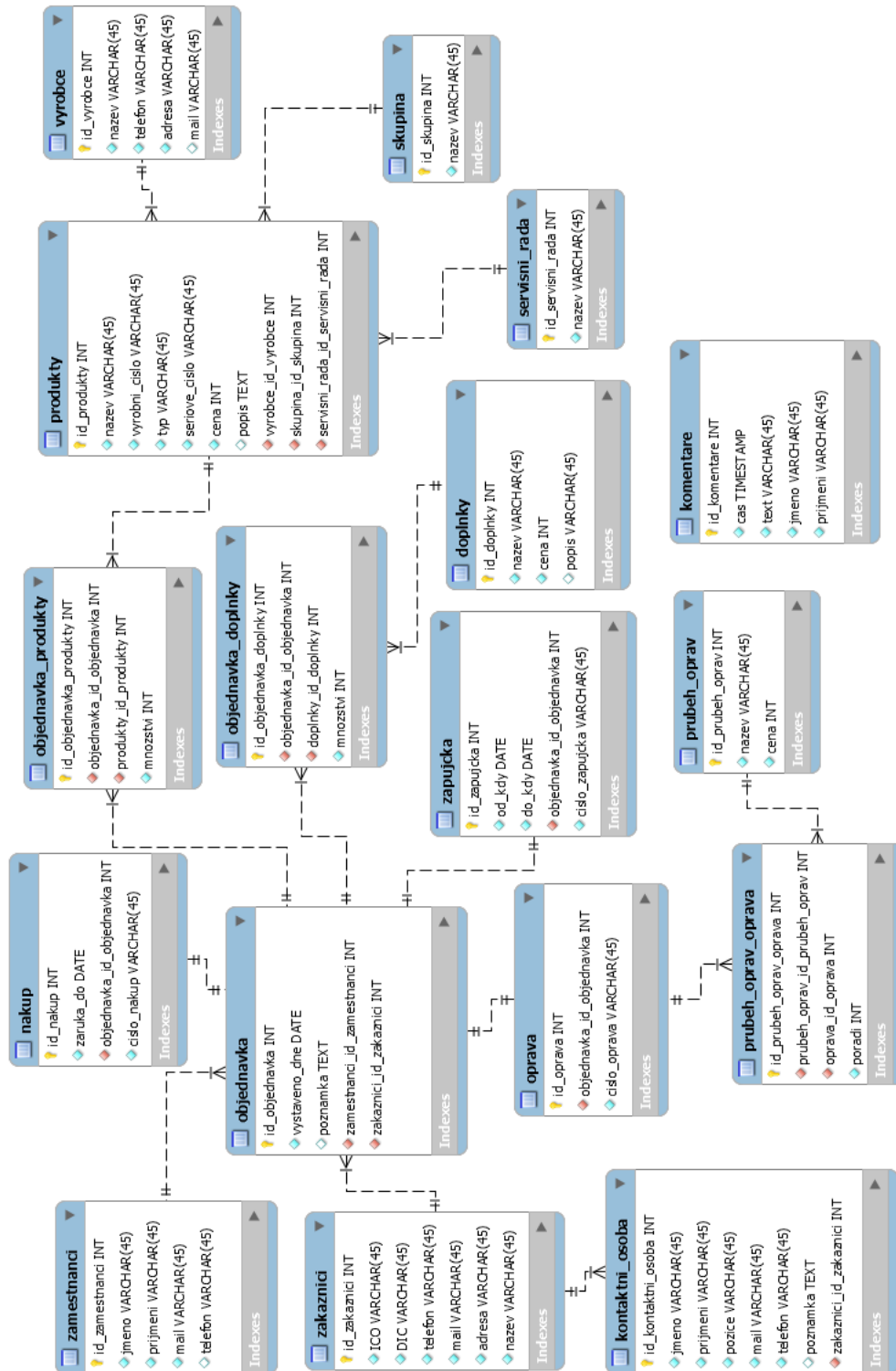
<http://matzke.javeismy.name/TOPGEOSYS/www/>

Údaje pro přihlášení:

Jméno: admin

Heslo: secret

# B Celý ERD



Obr. 12 ERD



