

UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2017

Patrik Lopauer

Univerzita Pardubice

Fakulta elektrotechniky a informatiky

Informační systém autoservisu

Patrik Lopauer

Bakalářská práce

2017

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2016/2017

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Patrik Lopauer**
Osobní číslo: **I13170**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Informační systém autoservisu**
Zadávací katedra: **Katedra informačních technologií**

Z á s a d y p r o v y p r a c o v á n í :

Cílem této práce je vytvoření funkční aplikace, která bude obsahovat nástroje sloužící pro zajištění činnosti autoservisu. Aplikace bude umožňovat tyto funkcionality:

1. Registrace uživatelů systému.
2. Přístup uživatelů do systému podle jejich práv.
3. Evidence zaměstnanců a zákazníků servisu včetně adresy, telefonu, e-mailu a dalších nutných údajů.
4. Evidence objednávek zákazníků.
5. Evidence opravovaných automobilů a údajů potřebných pro jeho servis.
6. Vytváření denních/týdenních rozpisů práce pro servisní techniky.
7. Generování sestav dle volitelně zadaných kritérií.

V úvodní části je nutno provést rešerši systémů, které se zabývají touto problematikou. Rešerši je nutné doplnit o porovnání s nově navrhovaným systémem, který bude předmětem této práce. Praktická část musí obsahovat analýzu navrhovaného řešení, která bude obsahovat popis použitých technologií, návrh databáze, ER diagram s využitím "Crow's Foot" notace entity-relationship a aplikační řešení. Pro vytvoření aplikace bude využit skriptovací jazyk PHP nebo JAVA a databáze MySQL nebo Oracle.

Rozsah grafických prací:

Rozsah pracovní zprávy: **30 - 40 stran**

Forma zpracování bakalářské práce: **tištěná**

Seznam odborné literatury:

LACKO, Luboslav. Oracle - Správa, programování a použití databázového systému. Brno: Computer Press a.s., 2007. 573 s. ISBN 978-80-251-1490-2.
GROFF, James R. a Paul N. WEINBERG. SQL kompletní průvodce. Brno: Computer Press a.s., 2005. 936 s. ISBN 80-251-0369-2. NARAMORE, Elizabeth, Jason GERNER, Scouarnec YANN LE and Timothy BORONCZYK. PHP 6, MySQL, Apache: Vytváříme webové aplikace. Brno: Computer Press a.s., 2009. 816 s. EAN:9788025127674. DRUSKA, P. CSS a XHTML - tvorba dokonalých webových stránek krok za krokem, Grada, 2006. 200 s. ISBN: 80-247-1382-9.

Vedoucí bakalářské práce: **Ing. Miloslav Macháček, Ph.D.**
Katedra informačních technologií

Datum zadání bakalářské práce: **31. října 2016**

Termín odevzdání bakalářské práce: **12. května 2017**



Ing. Zdeněk Němec, Ph.D.
děkan



Mgr. Josef Horálek, Ph.D.
pověřený vedením katedry

V Pardubicích dne 31. března 2017

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 30. 08. 2017

Patrik Lopauer

PODĚKOVÁNÍ

Chtěl bych poděkovat panu Ing. Miloslavu Macháčkovi, Ph.D. za odborné vedení.

ANOTACE

Tato práce popisuje problematiku implementace informačního systému určeného pro autoservis. Cíl práce je aplikace, která pokrývá všechny potřebné funkcionality pro provoz a správu autoservisu. V teoretické části je provedena analýza několika existujících řešení. Tyto řešení jsou hodnoceny podle několika faktorů a porovnané s vyvíjeným systémem. Praktická část obsahuje postup implementace v jazyce JAVA a v databázi MySQL. Tyto technologie jsou popsány v teoretické části.

KLÍČOVÁ SLOVA

Informační systém, autoservis, JAVA, MySQL

TITLE

Information system for car repair shop.

ANNOTATION

This work describes the implementation of the information system for the car service. Aim of work is an application that covers all the necessary functionality for operation and administration of the car repair station. In the theoretical part is analyzed several existing solutions. These solutions are evaluated by several factors and compared to the developed system. The practical part contains the implementation procedure in JAVA and in MySQL database. These technologies are described in the theoretical part.

KEYWORDS

Information system, car repair shop, JAVA, MySQL

OBSAH

0	Úvod.....	14
1	Teoretická část	15
1.1	Historické zákonitosti.....	15
1.2	Informační systém.....	15
1.3	Problematika systémů (aplikací, které navrhujete)	17
1.4	Požadavky na správnou funkčnost aplikace.....	17
1.5	Vývoj aplikačního SW	17
1.5.1	Historie relačních databází.....	18
1.5.2	Charakteristika relačních databází	19
1.5.3	SQL jazyk	20
1.5.4	Porovnání RDBMS	21
1.5.5	Historie programovacích paradigmat.....	23
1.5.6	Charakteristika Objektivě orientované programování	24
1.5.7	Porovnávání OOP jazyků.....	25
1.6	Alfasoft.....	27
1.6.1	Instalace	27
1.6.2	Prvotní nastavení.....	28
1.6.3	Základní objednávky.....	28
1.6.4	Výhody.....	28
1.6.5	Nevýhody.....	28
1.7	KelEXPRESS Autoservis.....	29
1.7.1	Instalace	29
1.7.2	Prvotní nastavení.....	29
1.7.3	Zakládání objednávky	30
1.7.4	Výhody.....	30
1.7.5	Nevýhody.....	30

1.8	ADMWin – Autoservis	30
1.8.1	Instalace	31
1.8.2	Prvotní nastavení.....	31
1.8.3	Zakládání objednávky.....	31
1.8.4	Výhody.....	31
1.8.5	Nevýhody.....	31
1.9	ProCad - AutoServis	32
1.9.1	Instalace	32
1.9.2	Prvotní nastavení.....	33
1.9.3	Založení objednávky.....	33
1.9.4	Výhody.....	33
1.9.5	Nevýhody.....	33
1.9.6	Carsystem.....	33
1.10	TEAS – AutoServis	34
1.10.1	Instalace	34
1.10.2	Prvotní nastavení.....	34
1.10.3	Zakládání objednávky.....	34
1.10.4	Výhody.....	35
1.10.5	Nevýhody.....	35
1.11	Tabulkové hodnocení existujících řešení	35
1.12	Informační systém popisovaný v této práci.....	36
2	Praktická část	37
2.1	Návrh systému.....	37
2.1.1	Použité technologie.....	37
2.1.2	Bezpečnostní mechanismy.....	37
2.2	Architektura aplikace	38
2.2.1	Rich picture.....	38

2.2.2	UML Use case diagram	39
2.2.3	UML Activity diagram	40
2.3	Uživatelské role.....	41
2.3.1	Mechanik	41
2.3.2	Přijímací technik	41
2.3.3	Vedoucí	41
2.3.4	Administrátor	41
2.4	Datový model.....	42
2.4.1	Tabulka Zakázka.....	43
2.4.2	Tabulka Vozidlo	43
2.4.3	Tabulky ModelVozidla a ZnackaVozidla.....	43
2.4.4	Tabulky KaroserieVozidla, BarvaVozidla.....	43
2.4.5	Tabulky MotorVozidla a Palivo	43
2.4.6	Tabulky ServisniIntervalVozidla a ProvedeniServniIterVoz	43
2.4.7	Tabulka Zakaznik, SoukromyZakaznik a FiremniZakaznik.....	43
2.4.8	Tabulky Adresa a Kontakt	44
2.4.9	Tabulky Zamestnanec, Uzivatel, Role.....	44
2.4.10	Tabulky StavZakazky a SeznamStavu.....	44
2.5	Instalační příručka.....	44
2.5.1	Instalace Javy.....	44
2.5.2	Instalace MySQL	44
2.5.3	Vložení databáze do MySQL.....	45
2.5.4	Prvotní spuštění.....	45
2.6	Uživatelská příručka.....	45
2.6.1	Přihlášení	45
2.6.2	Hlavní menu.....	45
2.6.3	Rozpis práce.....	46

2.6.4	Správa zakázek	46
2.6.5	Správa zákazníků	46
2.6.6	Správa vozidel.....	46
2.6.7	Správa zaměstnanců.....	47
2.6.8	Správa uživatelů.....	47
2.6.9	Reporty.....	47
3	Závěr	48
4	Použitá literatura	49
	Příloha A – Adresářová struktura aplikace	51
	Příloha B – Zdrojový kod reportu	52

SEZNAM ILUSTRACÍ A TABULEK

Obrázek 1 - Graf nejpoužívanějších jazyků, zdroj [11]	25
Obrázek 2 - Rich picture, zdroj [vlastní]	38
Obrázek 3 – Use case diagram, zdroj [vlastní]	39
Obrázek 4 - Activity diagram pro vytvoření zakázky, zdroj [vlastní]	40
Obrázek 5 – Datový model, zdroj [vlastní].....	42
Obrázek 6 - Přihlašovací okno, zdroj [vlastní]	45
Obrázek 7 - Hlavní menu, zdroj [vlastní]	46
Tabulka 1 - Nejpoužívanější RDBMS, zdroj [8]	21
Tabulka 2 – Tabulkové hodnocení RDBMS, zdroj [10].....	21
Tabulka 3 - Tabulkové hodnocení OOP jazyků, zdroj [17].....	25
Tabulka 4 - Tabulkové hodnocení existujících řešení, zdroj vlastní	35

SEZNAM ZKRATEK A ZNAČEK

OS Operační systém

IS Informační systém

DB Databáze

0 ÚVOD

Cílem této bakalářské práce je implementovat funkční informační systém, který je určen pro mále až středně velké autoservisy. Pokud bychom se zaměřili na velký autoservis, bylo by potřeba spravovat několik poboček v jednom datovém modelu. Datový model je připraven na možnost rozšíření o moduly jako jsou pneuservis, garanční prohlídky, či lakovna, anebo je možné napojit online kalendář, který by zákazníkům ulehčoval domlouvání termínů. Systém je určen pro ušetření práce mechaniků a jejich vedoucích. Systém je možné provozovat na Oracle technologii, či na MySQL technologii.

V teoretické části je popsán základní termín informační systém. Dále je provedena detailní analýza již existujících řešení. Tato analýza vyhodnocuje konkurenční software pomocí několika bodů, například multiplatformost, možnost síťového běhu, rychlost a použité technologie. Výsledek analýzy je prezentován jak písemně, tak i pomocí grafů. Celkový výsledek je vyhodnocen procentuálně. Následně je provedena další analýza, která porovnává výsledek předchozí analýzy s touto bakalářskou prací. Další část popisuje stručnou historii použitých technologií, jejich výhody a bohužel i nevýhody. Těmito technologiemi jsou objektově orientovaný programovací jazyk Java a relační databázový systém SQL, který se dále dělí na Oracle SQL a MySQL. Tyto jsou neznámější a také nejpoužívanější. Java je použita na část grafického rozhraní informačního systému. Databázová technologie je použita pro uchovávání dat na serveru a tím pádem jsou data dostupná z více aplikací.

V praktické části je rozebírána postupná implementace informačního systému. Nalezneme zde i ukázky použitých technologií. Dále je zde několik diagramů pro komplexní implementaci. Jako první je využit UML Case Diagram pro snadnější pochopení chování systému s jednotlivými uživateli. Následuje UML Activity Diagram zobrazující postup úkonu, které vedou k vytvoření objednávky a Rich Picture, který popisuje celou situaci okolo informačního systému. Jedním z hlavních diagramů je samozřejmě ER, v němž je zachycen způsob ukládání dat do databáze a jejich vzájemné vztahy. Informační systém umožňuje provoz na dvou různých SQL systémech, tím pádem budou samozřejmě uvedeny dva ER digramy, a to pro Oracle a také MySQL. Také tu najdeme popis rolí a k čemu jsou určené. Na konci jsou dvě příručky. Obě příručky jsou psané tak, aby byly co nejlehčí na pochopení. První příručka se zabývá instalací programu a prvotním nastavením, druhá je uživatelská příručka. V té je popsán seznam úkonů, které lze provést v programu.

1 TEORETICKÁ ČÁST

1.1 Historické zákonitosti

V roce 1885 vynálezce Gottlieb Daimler ve spolupráci s Wilhelmem Maybachem sestrojil první spalovací motor na benzín. Ještě ten rok tento motor zabudovali do dřevěné dvoukolky, se kterou byl proveden první test funkčnosti. I když pokračovali ve své práci a stále zlepšovali motor i ovládání, bohužel se stále jejich vynález nedal uznat jako automobil. Jako jedno z prvních vozidel se uznává automobil tvůrce Siegfrieda Marcuse vyroben v roce 1875. Ten splňoval vlastnosti automobilu, kterými jsou natáčecí přední náprava a motor uložen uvnitř karoserie. Za první automobil pro lid se dá považovat až Ford T, který zkonstruoval Henry Ford. Důvodů je několik. Jedním z hlavních je cena - průměrná cena vozidla v tomto období bylo okolo 2000 až 3000 dolarů, avšak cena Fordu byla pouze 500 dolaru, což je veliký rozdíl. Dalším důvodem je masová výroba mající za následek dvě věci - prvním je zlevnění vozu, druhou je dostatečné finanční ohodnocení práce zaměstnanců. Čím bylo více vozidel, tím byla potřeba více servisních středisek. Jelikož se jedná o období mezi světovými válkami, byla výpočetní technika teprve na začátku. Nejrozšířenější v této době byl děrnoštítkový stroj, který byl pro lidi velmi drahý. Během druhé světové války začala nová generace, elektromechanické počítače využívající většinou relé pracující na 100 Hz kmitočtu. Vývoj probíhal v několika státech najednou a hlavní hybnou silou byla ona druhá světová válka. V Německu probíhal vývoj počítačů, který měl na starosti německý inženýr Konrad Zuse. Na opačné straně se vyvíjel počítač s názvem Mark, jehož konstruktérem byl Howard Hathaway Aiken. Poté následovalo několik dalších generací. Poslední generace je nejdůležitější, jelikož jde o éru osobních počítačů, která dovolila rozšíření počítačů i do firem a do osobního vlastnictví. Počítače se tak začaly užívat k ulehčení a zrychlení práce zaměstnanců. Jedná se o postupné zavádění informačních systémů. [1]

1.2 Informační systém

Informační systém je celek technických prostředků, který se skládá z počítačového hardwaru a softwaru, dat a lidí, kteří tento celek využívají pro procesy zabývající se sběrem, editací, zpracováním a k následné reprodukci, šíření informací potřebných na další plánování, rozhodování a řízení. Z definice je vidět jasná myšlenka, a to získat data z reálného světa které zpracujeme pomocí aplikace. Z aplikace pak vystupuje souhrn informací v předem nastaveném rozsahu a podobě. Další vlastností je ukládání a zpřístupnění informací. Základní prvek je modularita. Každý proces má svoje unikátní potřeby, a proto se systém může zaměřovat pomocí

modulu. Například obchod. Každý obchod má jako základní myšlenku prodávat zboží. Ale v detailu však již můžeme vidět rozdíl. Pokud bude mít obchod jednoho zaměstnance tak bude vykonávat všechny procesy pouze on. Ve velkém obchodě bude několik zaměstnanců a každý z nich bude mít jiná přístupová práva do IS. Například vedoucí skladu bude mít přístup pouze do modulu sklad, ale nebude mít již přístup do modulu prodej. A naopak prodavačka. Z toho vyplývá základní charakteristika:

- Informační systém je celek procesů sjednocených do jednoho programu, který má plnit určité úlohy pro celou organizaci, nikoli jen pro jednoho uživatele.
- Všichni uživatelé vycházejí ze stejných dat, které jsou uloženy v centrální databázi. To znamená, že chyba jedince může ovlivnit práci jeho kolegů, kteří na jeho práci navazují.
- Komunikace probíhá v režimu klient-server. Na serveru jsou uložena data v databázi a klient vzdáleně na serveru s těmito daty pracuje.
- Uživatelů mohou být desítky, stovky ale i tisíce, a proto jsou nároky na IS velké. Také je potřeba stále udržovat data aktuální a bezchybná.

Informační systém dělíme na čtyři kategorie:

- Podnikové informační systémy
 - Tuto kategorii ještě můžeme dělit na další systémy, které mají svoje zaměření. Příklady: plánování podnikových zdrojů, řízení dodavatelského řetězce, řízení vztahů se zákazníky a další.
- Veřejné informační systémy
 - Určeno pro televizi, rozhlas, knihovny, zpravodajské agentury, informační instituce.
- Informační systémy veřejné správy
 - V této kategorii jsou systémy, které využívá stát pro zjednodušení práce úředníků.
Příklady: Administrativní registr ekonomických subjektů, Informační systém o informačních systémech veřejné správy, Informační systém o datových prvcích
- Osobní informační systém
 - sloužící pro potřeby jednotlivce. [2], [3]

1.3 Problematika systémů (aplikací, které navrhujete)

Každý informační systém by měl být co nejvíce modulární, aby mohl být lehce rozšiřitelný o další moduly. Další problém je zajistit lehké a intuitivní ovládání aplikace, ale zároveň poskytovat dostatečné množství funkcí ke správě aplikace. Dále je potřeba dodržovat odbornou terminologii, aby aplikaci každý rozuměl. Také je potřeba provést kvalitní analýzu jak z pohledů zákazníka, tak zaměstnanců, aby aplikace splňovala všechny procesy, které jsou v autoservisu prováděny. K tomu se váže určení, co který zaměstnanec provádí a k čemu má mít přístupová práva.

1.4 Požadavky na správnou funkčnost aplikace

Požadavky na správnou funkčnost se určují pomocí analýzy. Ta je součástí Metodiky vývoje softwaru. Používají se nejčastěji tři techniky, a to Prototypování, Spirála a Vodopád. Postup je u každého trochu odlišný, detaily jsou vidět na obrázku. Analýzu si můžeme dále rozdělit na několik dalších částí. Například analýzu již existujících řešení, Rich picture k pochopení problému navrhovaného systému, UML Use case diagram k zobrazení chování systému z pohledu rolí uživatelů a UML Activity diagram k znázornění kroků vytvoření zakázky.

Základní požadavky:

- Registrace uživatelů systému.
- Přístup uživatelů do systému podle jejich práv.
- Evidence zaměstnanců a zákazníků servisu včetně adresy, telefonu, e-mailu a dalších nutných údajů.
- Evidence objednávek zákazníků.
- Evidence opravovaných automobilů a údajů potřebných pro jeho servis.
- Vytváření denních/týdenních rozpisů práce pro servisní techniky.
- Generování sestav dle volitelně zadaných kritérií.

1.5 Vývoj aplikačního SW

Základní charakteristika informačního systému je pracovat s aktuálními daty, které jsou uloženy na jednom místě. Nejvhodnější řešení je relační databáze, jenž splňuje několik dalších podmínek.

1.5.1 Historie relačních databází

Pro práci s relačními databázemi se používá jazyk SQL, což je zkratka slova Structured Query Language, který je nástupcem jazyka Structured English Query Language se zkratkou SEQUEL. Historie relačních databází začíná okolo 1890, kdy Herman Hollerith sestrojil první automat na bázi děrných štítků na sčítání obyvatel v USA. Později v roce 1911 se spojil s další firmou a založili International Business Machines, dnes známou pod zkratkou IBM. Až do roku 1960 se stále používali digitální počítače ovládané děrnými štítky. Velká nevýhoda děrných štítků je možnost pouze sériového přístupu k datům, což je pro vyhledávání nevhodné. Během tohoto roku se začalo přecházet na magnetické disky, které tento nedostatek odstraňovaly. V následujícím roce byl Charlesem Bachmanem z General Electric představen první integrovaný datový sklad. S dalšími výzkumníky pak založil skupinu Codasyl, jež publikovala základní specifikaci pro práci s databázemi s libovolným programovacím jazykem. Codasyl databáze využívala síťový model dat, ale u IBM se používal hierarchický model dat. S tím ale nebyl spokojený zaměstnanec Ted Codd, a tak v roce 1970 prezentoval článek, ve kterém popisoval implementaci relačního modelu. K operaci s daty použil základní metody, jimiž jsou sjednocení, kartézský součin, rozdíl, selekce, projekce a spojení. Ostatní operace jsou již jen kombinacemi těchto šesti. To vedlo k vývoji ne-procedurálního jazyka. Vývoj se rozdělil na dvě větve. IBM tomuto návrhu nevěřilo, a také by zavrhl již hotový produkt IMS. I přes to se rozhodli vývoji věnovat a začali implementovat Systém-R. Druhá větev vznikla na University of California at Berkeley s názvem Ingres. U IBM byl vývoj rozdělen na dvě fáze. Během první byl prakticky ověřen relační model v další fázi se věnovali implementaci SQL jazyku, který se stal standardem. V druhé větvi byla důležitá dvojice Michael Stonebraker a Eugene Wong - vedoucí projektu. Ingres měl být použit pro uložení geografických dat. Používali jazyk QUEL. V roce 1980 byla vydána komerční verze. Mezi větvemi existovala spolupráce, během níž studenti z univerzity jezdili na stáže do IBM. První SQL databázi uvedla na trh společnost Oracle, která si vzala inspiraci ze System-R. Poté na trh uvedl svůj produkt DB2 i IBM. Bohužel zde můžeme vidět chybu managementu, která chtěla získat více financí z původního produktu, a tak oddalovala uvedení již nového produktu, a to i přes to, že IBM byla u zrodu SQL databází. Jako zlatý věk relačních databází jsou označována 80 léta. Během těchto let došlo ke standardizaci s normou ISO/IEC 9075. Poslední standard byl vydán v roce 2016 a to ISO/IEC 9075:2016. Standarty nám zaručují stejný přístup k datům. Vývoj Ingresu byl ukončen v roce 1982 a do dnešní doby se transformoval do PostgreSQL. [4]

1.5.2 Charakteristika relačních databází

Základem relačních databází jsou relace, tedy databázové tabulky - dvourozměrné struktury tvořené záhlavím a tělem. Sloupce se nazývají atributy a řádky záznamy. Sloupce určují datový typ a velikost, řádek se využívá k uložení dat. Další základní prvky jsou tři druhy klíčů. Primární klíč je určen k jednoznačné identifikaci řádku s daty. Nikdy nesmí být prázdný, můžeme ho plnit číslem nebo posloupností písmen. Při každém vložení řádku by se měl automaticky inkrementovat. Kandidátní klíč je atribut nebo skupina atributů, který také jednoznačně identifikuje řádek v tabulce. Tento klíč se může stát i primárním. Pokud však využijeme jiný primární klíč, stává se z něj alternativní klíč. Cizí klíč je používán k vyjádření relací mezi tabulkami. Vztahy mezi tabulkami se dělí na stupně, kardinalitu a parcialitu.

Stupně dělíme na několik druhů:

1. Unární vztah – relace je spojena sama se sebou. Například vztah mezi zaměstnancem a nadřízeným. Ve firmách je budovaná hierarchie zaměstnanců, kde má každý zaměstnanec svého vedoucí, až do majitele firmy.
2. Binární vztah – standardní vztah mezi relacemi.
3. Ternární vztah – vztah mezi třemi relacemi.
4. N-ární vztah – vztah mezi n-relacemi zároveň.

Další je kardinalita, která je také dělena na čtyři druhy:

1. Mezi záznamy není žádný vztah.
2. 1:1 znamená vztah mezi záznamem z jedné tabulky, jež odpovídá záznamu v druhé tabulce.
3. 1:N přiřazuje jednomu záznamu více záznamů z druhé tabulky.
4. M:N dovoluje jakémukoli záznamu přiřadit nespočet záznamů z jiné tabulky.

Poslední je parcialita. Ta vyjadřuje, zda musí být vztah povinný nebo volitelný. Během návrhu databáze se používá normální forma. Tento proces se nazývá normalizace a slouží ke zlepšení práce s daty, k zabránění redundanci a zvýšení konzistence dat. Normalizaci dělíme na šest normálních forem. Pro práci s relačními databázemi je používán jazyk Structured Query Language se zkratkou SQL. Tento jazyk vychází z předchůdce Structured English Query Language, který se měl podobat co nejvíce lidskému jazyku. [5], [6]

1.5.3 SQL jazyk

Příkazy jazyka SQL umožňují úplnou kontrolu nad celou databází. Dělí se na:

- Příkazy pro manipulaci s daty

Všechny tyto příkazy jsou určeny pro výběr nebo editaci dat a můžeme je spojit s příkazem WHERE. Toto klíčové slovo slouží k doplnění dotazu o podmínku, která upřesní řádky se záznamy. Také je můžeme spojit s příkazem ORDER BY určená k řazení výsledků, které můžeme řadit vzestupně, nebo sestupně podle abecedy. Označují se zkráceně DML – Data Manipulation Language.

- SELECT – pro výběr dat.
- INSERT – pro vkládání dat.
- UPDATE – pro editaci dat.
- MERGE – kombinace příkazů INSERT a UPDATE, pokud data nenaleze, tak provede INSERT dat a pokud ano, tak data pomocí UPDATE zedituje.
- JOIN – sloučený výběr dat z několika databází.
- DELETE – vymazání dat.
- EXPLAIN – zobrazí postup zpracování SQL příkazu.
- SHOW – zobrazí databáze, tabulky a jejich definice.

- Příkazy pro definici dat

Tyto příkazy vytvářejí strukturu databáze. Například tabulky, indexy, pohledy a další objekty. Označují se zkráceně DDL – Data Definition.

- CREATE – vytváření nových objektů.
- ALTER – editace existujících objektů.
- DROP – mazání objektů.

- Příkazy pro řízení dat

Tyto příkazy slouží pro nastavování přístupových práv a řízení transakcí. Označují se jako DCL – Data Control Language.

- GRANT – příkaz pro přidělení oprávnění uživateli.
- REVOKE – příkaz pro zbavení práv uživateli.
- START TRANSACTION – zahájení transakce.
- COMMIT – potvrzení transakce.
- ROLLBACK – zrušení transakce, návrat do původního stavu. [5], [6]

1.5.4 Porovnání RDBMS

RDBMS je Relational Database Management System (česky relační řídicí databázový systém), který pro práci s daty využívá jazyk SQL. Existuje několik řešení. V tabulce níže můžeme vidět TOP 5 systémů a poté graf podle skóre.

Tabulka 1 - Nejpoužívanější RDBMS, zdroj [8]

Název	Umístění			Skóre		
	Duben 2017	Září 2016	Duben 2016	Duben 2017	Září 2016	Duben 2016
Oracle	1.	1.	1.	1402	+2	-65
MySQL	2.	2.	2.	1364	-12	-5
Microsoft SQL server	3.	3.	3.	1204	-3	+70
PostgreSQL	4.	4.	4.	361	+4	+58
DB2	5.	5.	5.	186	+2	+2

Na prvním místě je Oracle s 1402 body. Oracle byl také první, kdo uvedl na trh svoje řešení. Na druhém místě je MySQL s 1364 body. Tento systém je také vyvíjen společností Oracle, ale je nabízen pod GPL licenci, i pod placenou licenci. Verze zdarma neobsahuje veškeré možnosti, jež má placená licence. Důvod, proč je na druhém místě právě ona licence zdarma, je možnost jakéhokoli použití. Z toho důvodu je využíván na osobní projekty, s webovou aplikací anebo na ní může být implementován celý informační systém. Na třetím místě je Microsoft SQL Server s body 1204, největší konkurent pro systém Oracle. Nevýhoda Microsoft SQL je běh pouze na operačním systému Windows, což znamená další finanční výdaj za licenci. V tabulce níže nalezneme shrnutí základních informací a obodování podle kritérií zapsaných v levém sloupci.

Tabulka 2 – Tabulkové hodnocení RDBMS, zdroj [10]

Název	Maximální skóre	DB2	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
Licence	10	5	5	10	5	10
Serverový OS	5	3	0,5	2,5	3,5	5
Datové schéma	2	2	2	2	2	2
Datový typ	5	5	5	5	5	5

XML podpora	2	2	2	2	2	2
Sekundární index	3	3	3	3	3	3
Podpora SQL	5	5	5	5	5	5
API a přístupové metody	5	5	3	3	4	5
Podporované programovací jazyky	10	5	4	7	10	3
Vestavěné programovací funkce	10	5	10	5	10	5
Trigger	2	2	2	2	2	2
Partitioning metody	5	2,5	5	5	2,5	0
Replikační metody	5	2	4	5	5	2,5
MapReduce	2	0	0	0	0	0
Kontrola konzistence	5	5	5	5	5	5
Cizí klíč	5	5	5	5	5	5
Databázová transakce	5	5	5	5	5	5
Řízení paralelního zpracování	2	2	2	2	2	2
Trvanlivost	2	2	2	2	2	2
Běh v paměti	10	0	10	10	10	0
Získané skóre	100	65,5	79,5	85,5	88	65,5

Podle tabulkového hodnocení vychází pořadí

1. Oracle,
2. MySQL,
3. Microsoft SQL Server,
4. PostgreSQL,

5. DB2.

Hodnocení vyhrál RDBMS systém od společnosti Oracle. Jednou z hlavních výhod je procedurální nadstavba jazyka SQL, který obsahuje kurzory, procedury, funkce a další. Na druhém místě se umístil MySQL. Ten má největší výhodu v licenci GPL, která umožňuje jakékoliv použití zdarma. Třetí v pořadí je Microsoft SQL Server, který má jako největší nevýhodu pouze běh na operačním systému Windows Server. Další nevýhodou je pouze základní kompatibility s ostatními výrobky, které nejsou ve vývoji od společnosti Microsoft. Zbývá PostgreSQL, u něž je výhodou licence zdarma a DB2 od společnosti IBM, ovšem chybí mu několik funkcí, které mají jeho konkurenti. I když se jazyk SQL snaží být co nejpodobnější lidskému jazyku, stále by bylo jeho využívání koncovým uživatelem příliš složité. Také z toho důvodu se vyvíjí grafické uživatelské rozhraní. To může prezentovat webová aplikace, nebo aplikační software. Pro webovou aplikaci je určeno několik technologií. Například HTML, CSS, JavaScript, PHP, ASP, ASP.NET, Ruby on Rails, Perl, Python. Pro aplikační software se používá Java, C, C++, C# a několik dalších. V dnešní době se nejvíce využívá programovací paradigma objektově orientované. Smyslem objektového programování je rozdělit program na menší programy, které sice bude vykonávat pouze jednu činnost, ale bude se moct opakovaně použít pro řešení konkrétního problému. Výhodou tohoto paradigmatu je snazší a rychlejší vývoj programů. Další z výhod je snadná údržba a zpětná rozšiřitelnost, jež je hlavní podmínkou informačního systému. [7], [9]

1.5.5 Historie programovacích paradigmat

Na začátku 60 let minulého století se objevily základní myšlenky OOP. První syntaxe programovacího jazyka byla začleněna do Simula 67. Účelem jazyka Simula 67 bylo programování diskrétní simulace. Tedy simulovat reálný, či virtuální svět plný objektů. A proto by bylo vhodné tuto myšlenku aplikovat na vývoj aplikací. Oprávněnost této myšlenky potvrdil dnešní vývoj. V 70 letech 21. století se společnost Xerox rozhodla založit tým vývojářů, kteří se touto myšlenkou začali zabývat. Simula 67 položila základy, ale stále se nejednalo o kompletní programovací paradigma. Začali tedy vyvíjet programovací jazyk Smalltalk, který se stal plnohodnotným objektově orientovaným programovacím jazykem. Smalltalk pomohl rozšířit toto paradigma mezi vědce na univerzitách. Bohužel rozšíření mezi ostatní vývojáře neprobíhalo tak, jak si představovali. Později Bjarne Stroustrup dostal za úkol naprogramovat simulaci telefonních soustav. Stroustrup se rozhodoval jaký jazyk použít, Simula mu přišla nedostatečně propracovaná na úkol těchto rozměrů. Jazyk C se stával standardem pro

systemové programování, byl rychlý, ale z důvodu nízkoúrovňového přístupu by byl vývoj příliš dlouhý, a tak i zbytečně náročný. Konečné rozhodnutí bylo použití upraveného jazyka C s třídami, ke kterému vytvořil sadu maker, které převáděly zpět do jazyka C. Tento přístup se ukázal jako velmi efektivní a díky tomu se začal rychle rozšiřovat. V roce 1983 byl přejmenován na jazyk C++ a byla odstraněna makra na převod zpět na jazyk C tím, že se vytvořil nový překladač. C++ začal nabírat na popularitě, a to z důvodu stejné syntaxe jako měl jazyk C, avšak vývojář mohl využít výhod nového jazyka. V 80 letech se jazyk C++ začal šířit jako lavina. Bohužel mnoho vývojářů špatně pochopilo význam OOP, a proto ho začali používat na místech, kde se nehodil. Vývojáři začali opouštět toto paradigma a vraceli se k strukturovanému programování. Dlouhá leta probíhaly konference a debaty ohledně jeho užitečnosti a využití, ale až v roce 1995 vyšla první kniha s názvem Design Patterns – Elements of Reusable Object-Oriented Software, která vysvětlovala, jak správně používat OOP. Na tuto knihu navazovalo několik dalších, ve kterých byly vysvětlovány konkrétní problémy. Tímto se znovu nastartovala éra OOP, a ta pokračuje do dnešních dnů. [15]

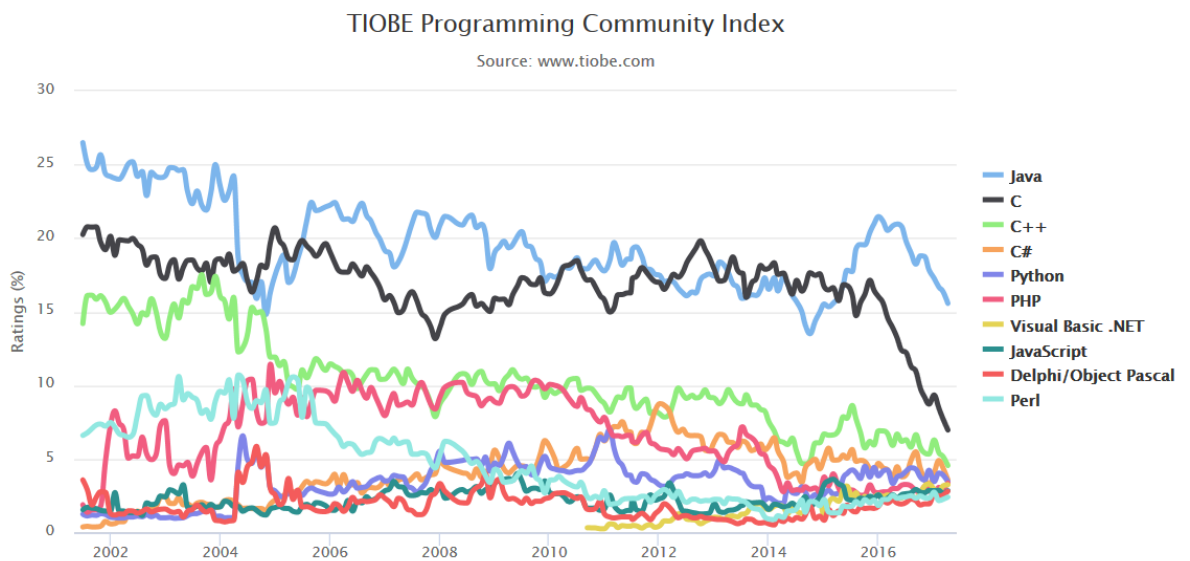
1.5.6 Charakteristika Objektivě orientované programování

Již z názvu je vidět, že se orientuje na objekty. Objekt je entita, která obsahuje data, i schopnosti. Například si vezmeme firmu. Firma obsahuje několik entit. Pod těmito entitami si můžeme představit zaměstnance, stroje, budovy. Každá entita je samostatně realizující činnost. Zaměstnanec vyrábí výrobek podle vstupních součástek. Celá výroba tedy funguje na interakcích mezi těmito entitami. Také je možné vzít zaměstnance a přesunout ho do jiné výroby. Takto funguje i OOP. Objekty popisujeme jejich vlastnostmi, ty dělíme na data a metody. V reálném světě nemají všichni zaměstnanci stejné vlastnosti, ale mají několik základních vlastností společných, podle kterých je můžeme dělit, což je pojmenované názvem Třída. Třída je množina objektu s určitými vlastnostmi. Přitom ale nedefinuje konkrétní data nebo funkce, to má na starost instance. Třída je tedy vzor, jak má objekt vypadat a instance pak už je specifický zaměstnanec se svými vlastnostmi. Každý zaměstnanec má základní vlastnosti jako je jméno, příjmení, datum narození, pohlaví a další, proto vytvoříme třídu Zaměstnanec. Začíná se lišit až v detailech, jako je pozice. Vytvoříme třídu Zaměstnanec – Dělník, která bude obsahovat všechny vlastnosti pozice dělníka. To se nazývá dědění. A tak může třída dědit vlastnosti z jejího rodiče. V dědění je nastaven hierarchický model. Na podobném principu funguje i skládání. Kdy každá třída má specifické vlastnosti a jejich složením získáme třídu se všemi potřebnými vlastnostmi. Použití je podle situace a účelu. U vlastnosti je potřeba nastavit

viditelnost. Jsou tři viditelnosti, a to veřejné, které je kdykoliv viditelné, soukromé, u nichž mohou používat jen jiné metody téhož objektu a chráněné, které jsou podobné jako soukromé, ale dědí se do odvozených tříd. Poslední charakteristika je zapouzdření. Ta slouží jako černá skříňka. Provede činnost, ale není známo, jakým postupem. [12], [13], [14], [16]

1.5.7 Porovnávání OOP jazyků

Mezi nejnámější objektově orientované jazyky patří C++, C Sharp a Java. Podle TIOBE Indexu jsou tyto jazyky v TOP 5 používaných jazyků.



Obrázek 1 - Graf nejpoužívanějších jazyků, zdroj [11]

Z obrázku je patrné, že v TOP 10 je devět jazyků objektově orientovaných a pouze jazyk C je strukturální. V následující tabulce najdeme porovnání výhod a nevýhod třech nejpoužívanějších programovacích jazyků pomocí skorového hodnocení.

Tabulka 3 - Tabulkové hodnocení OOP jazyků, zdroj [17]

Název	Java	C#	C++
Objektově orientované	5	5	10
Statické nebo dynamické typy	1	1	1
Generické třídy	0	0	5

Dědičnost	5	5	10
Refaktoring pro přejmenování	0	0	0
Přetěžování metod /funkci	10	10	10
Přetěžování operátorů	0	5	5
Garbage Collection	10	10	0
Statické metody / proměně	2	2	2
Reflexe	5	5	0
Modifikátory přístupu	8	10	8
Kontrakty kódu	0	0	0
Multijádrové	10	10	5
Regulární výraz	7	7	0
Aritmetika ukazatele	0	2	2
Integrované zabezpečení	5	5	0
Celkové skóre:	68	77	58

Z tabulkového hodnocení vyšel jako vítěz jazyk C#. Bohužel jeho největší nevýhoda je běh pouze na operačním systému, kde se dá nainstalovat .NET Framework. Tento framework není vyvíjen pro Linux, ale je ho možné nahradit pomocí frameworku mono. Z vlastních zkušeností bych ale rád upozornil na možné problémy s komptabilitou. Jazyk C# a Java využívají virtuální stroj pro běh aplikací. Virtuální stroj Javy je možné provozovat na všech známých operačních systémech, a to je jeho největší výhoda. Aplikace vyvíjena v C++ komunikuje napřímo s operačním systémem a pokud se budou volat systémové funkce, tak se musí odlišit

pro jaký OS je vyvíjen a také je potřeba pro každou platformu samostatná kompilace programu. Nevýhoda C++ je ruční práce s pamětí. [17]

1.6 Alfasoft

První konkurenční řešení se jmenuje Alfa-soft, vyvíjené panem Ing. Václavem Čubanem. Jsou nabízeny tři druhy licence. První licence s názvem samostatně. Ta obsahuje kniha zakázek s tiskem daňového dokladu k zakázce placené jak hotově, tak i převodem. Tato varianta neumožňuje evidovat zásobu na skladu. Sám autor uvádí že je to nejlevnější a také nejméně komplexní varianta využívaná hlavně malými autoservisy, nebo opravnou jiného druhu. Druhá varianta obsahuje nadstavbu Skladová evidence umožňující evidovat zásoby na skladě. Určeno pro autoservisy, které potřebují vydávat pouze jeden druh daňových dokladu. Poslední varianta je logicky nejdražší a umožňuje navíc vydávat více druhů daňových dokladu. Aplikace má nastaveny 5 dílny životní cyklus pro zakázku. Cyklus obsahuje přijmout, zahájit opravu, čerpat náhradní díly a práci, ukončit, vydat. Tento cyklus se doporučuje pro větší servis. Pro menší servis je určen zjednodušený cyklus obsahující přijmout, čerpat náhradní díly a práci, vydat. Je zde možnost dokoupit modul s názvem Úschovna pneu. Zde bohužel vidím problém v omezení 5 kusu pneumatik. Pokud by chtěl někdo použít tento modul v servisu nákladních vozidel, tak narazí na zásadní problém. Toto řešení nabízí jak lokální, tak i možnost práce v síti a k tomu využívá Microsoft SQL server.

1.6.1 Instalace

Instalace proběhla rychle. Po spuštění program zhlásil, že potřebuje balíček net Framework. Po ruční instalaci frameworku je potřeba upravit zástupce, aby se spouštěl jako správce. Po úspěšném spuštění programu se ukáže výběr instalace síťové nebo nesíťové verze. Nesíťová verze používá LocalDB od Microsoftu, síťová používá technologii SQL Server 2014. Po instalaci dat byl problém se založením tabulek do DB. Po neúspěšné instalaci lokální verze programu jsem požádal autora o přístup na RDP server, kde je již program nainstalován a připraven k použití. Autor nabízí vzdálenou pomoc s aplikací. Nevýhoda vzdálené pomoci je to, že je vždy placena. Pokud si autoservis chce vyzkoušet software, zakoupí jej a nepovede se jim instalace aplikace, budou chtít pomoci a autor jim za to naúčtuje hodinovou taxu, což je nevhodné jako prezentace služeb.

1.6.2 Prvotní nastavení

Prvotní nastavení bylo již na serveru provedeno, ale dalo se znovu spustit. Prvotní nastavení probíhá typickým vyplněním několika informací o autoservisu. Dále se vyplňuje, jak se bude účtovat a zda je autoservis firma, nebo fyzická osoba a podle toho lze doplnit několik dalších informací. Pote se ještě musí přenastavit model autoservis.

1.6.3 Základní objednávky

Zakládání objednávky jsou vcelku jednoduchý proces. V záložce Kniha zakázek je seznam již všech provedených zakázek. Po stisknutí tlačítka plus se přidá nový řádek. Ve spodní části okna je několik polí, které jsou třeba vyplnit. Pokud je vůz poprvé v autoservisu, tak se musí všechny údaje ručně vyplnit, ale pokud již jednou v servisu byl, tak se automaticky doplní podle SPZ z databáze. Dále je potřeba vyplnit pole s údaji zákazníka. Zde se bohužel vybírá podle kódu zákazníka, avšak při servisu, kde může být několik set zákazníků, je nesmyslné si pamatovat všechny kódy zákazníků. Pro výběr zákazníka ze seznamu je potřeba do pole s kódem zákazníka dát enter a poté se otevře další okno, ve kterém buď vybereme zákazníka, anebo ho pomocí tlačítka plus přidáme do seznamu. U stálých zákazníků to může znamenat urychlení, ale takových bude pouze menšina. Lepší by tak bylo dohledávání podle jakéhokoliv parametru zákazníka. Jedna z výhod tohoto programu je historie zakázek dohledatelná podle SPZ. Také je zde možnost založit novou zakázku podle již založené zakázky a jen upravit potřebné údaje. Po založení objednávky se vytiskne zakázkový list.

1.6.4 Výhody

- Automatické doplnění údajů podle SPZ již jednou servisovaných vozidel,
- evidence objednávek zákazníků,
- evidence opravovaných automobilů a údajů potřebných pro jeho servis,
- možnost síťové verze,
- skladové hospodářství,
- možnost vedení účetnictví.

1.6.5 Nevýhody

- Registrace uživatelů systému a přístup do systému podle jejich práv,
- instalace softwaru je velmi komplikovaná,
- vytváření denních/týdenních rozpisů práce pro servisní techniky,
- nelze generování sestav dle volitelně zadaných kritérií,

- pouze na platformě Windows,
- vyhledávání podle SPZ nebo kódu dodavatele.

1.7 KeIEXPRESS Autoservis

Další konkurenční řešení je KeIEXPRESS s modulem Autoservis vyvíjený společností KELOC CS, s.r.o. Společnost nabízí dvě možnosti licence, a to Autoservis a Autoservis Plus. Rozdíl mezi edicemi je v počtu modulů. Modul Autoservis je složen z Faktury vydané, Sklad, Kasa, Servis. Rozšířená licence se skládá z Faktury přijaté, Faktury vydané, Pokladna, Banka, Sklad, Objednávky, Kasa, Servis. Autoservis Plus je zaměřen na středné až velké autoservisy. I toto řešení nabízí lokální i síťovou komunikaci s SQL serverem. Využívá technologii MS SQL. U tohoto řešení je možnost pronájmu terminálového serveru, ve kterém poběží aplikace. To vidím jako velké plus pro malý autoservis, který si pořídí levnější PC, jež bude fungovat jako lehký klient pro připojení k terminálovému serveru. To znamená, že zákazník musí řešit v případě potíží jen připojení k internetu nebo technicky problém s tenkým klientem. Aplikace nabízí kompletní informace o průběhu servisu (popis závad, vyjádření technika), možnost zadání vlastních parametrů k identifikaci vozidla, přidělování náhradních dílů z číselníku přímo k zakázce (propojení s modulem Sklad), automatické vystavování výdejek ze skladu, možnost přidělení zakázky konkrétnímu mechanikovi nebo oddělení, vystavování faktur nebo paragonů přímo ze zakázky, sklad pneumatik - vhodný pro autoservis, který zákazníkům poskytuje službu uskladnění pneumatik mimo sezónu, EET (Elektronická Evidence Tržeb) a propojení s platebními terminály bank.

1.7.1 Instalace

Po spuštění se otevře instalační okno, kde se vyplní údaje jako připojení k databázi. Účetní jednotka, první uživatel. Po připravení databáze a startu kopírování dat do databáze aplikace spadla a zahlásila že potřebuje pro svůj běh NET framework. Po instalaci potřebného frameworku se instalace znovu pustila a pokračovala tam, kde skončila.

1.7.2 Prvotní nastavení

Prvotní nastavení bylo velmi složité. Jako první bylo nutné v nastavení modulu přenastavit modul objednávky a servis. Poté bylo potřeba vyplnit několik oken s informacemi o firmě, která bude tento software používat.

1.7.3 Zakládání objednávky

Zakládání zakázek je připraveno jako univerzální, nejdou tedy zadat přesné informace o vozidle, ale jen základní údaje. Při zakládání objednávky bylo nutné zadat formát generování čísla objednávky. Pote byla potřeba vytvořit nová řada objednávek, ve které bylo dalších několik oken s informacemi. Pokud byl zákazník nový, musel se vytvořit nový řádek s tímto zákazníkem a vyplnit. Poté následovalo kolo vyplňování několika oken plných informací. Pokud byl již zákazník vytvořen, tak se dal vybrat ze seznamu zákazníků, bohužel i toto bylo velmi nepřehledné. V seznamu se nedá nijak vyhledávat. Lze pouze řadit řádky podle abecedy nebo obráceně. Po založení objednávky se musí ručně vytisknout zakázkový list.

1.7.4 Výhody

- Možnost síťové verze,
- skladové hospodářství,
- možnost vedení účetnictví,
- registrace uživatelů systému,
- přístup uživatelů do systému podle jejich práv,
- obsahuje vytváření denních/týdenních rozpisů práce pro servisní techniky,
- nabídka běhu aplikace v terminálu.

1.7.5 Nevýhody

- Příliš složité pro uživatele, nutnost objednat si školení,
- neobsahuje evidenci opravovaných automobilů a údajů potřebných pro jejich servis,
- pouze na platformě Windows.

1.8 ADMWin – Autoservis

Program Autoservis je součástí programu ADMWin, který je dostupný ve dvou variantách, jimiž jsou AdmWin DE – daňová evidence nebo AdmWin PU – podvojný účetnictví. Software ukládá základní údaje o vozidle. Například SPZ, tovární značku – typ vozidla, měsíc/rok výroby, výbavu, VIN, stav tachometru a informace o prováděném servisu. Jedna z výhod tohoto softwaru je možnost plánování datumu provedení zakázek a určení technika, který provede servis v Termináři. V další části zakládání zakázky se zapisují použité náhradní díly, spotřebovaný materiál a vykonané práce. Se všemi těmito údaji se tiskne zakázkový list. Fakturace dílenských zakázek pro autoservis je již dále zpracována standardním postupem v ekonomickém software AdmWin. Před zápisem nové zakázky je možnost nahlédnout do

historie vozidla a předchozích servisů a podle toho naplánovat a upravit novou zakázku. Program také nabízí evidenci uskladněných pneumatik. Software využívá technologii Visual FoxPro.

1.8.1 Instalace

Instalace proběhla pořádku. Instalační program měl velikost pouze 8 Mb a žádné další soubory si z internetu nestahoval.

1.8.2 Prvotní nastavení.

Po prvotním spuštění se otevře program s oknem, ve kterém vyplníme údaje o společnosti, která bude tento software využívat. Na rozdíl od jiných programů se zde musí vše vyplnit ručně. Také se otevře okno s nápovědou, ve které je vysvětleno, jak program ovládat a základní principy. Poté bylo potřeba upravit modul s nastavením zakázek a objednávek.

1.8.3 Zakládání objednávky

Zakládání objednávek je vcelku jednoduché. V záložce objednávky je řádek se zakázkami, v němž se přes standardní tlačítko přidá nový řádek, který se vyplní v novém okně. Navíc zde jdou doplnit detailní informace o vozidle, ale ty nejsou povinné. Jde například o typ motoru, výkon. Tyto informace jsou ovšem důležité pro urychlení práce technika. Dále tu jsou nesmyslná pole s názvem počet pneu-počet. Je nelogické, aby do servisu dorazilo vozidlo se 3 pneumatikami a nejednalo se o závadu. Objednávku jde přiřadit zaměstnanci podle pracovního plánu. Před dokončením objednávky se přidá materiál, který bude potřeba pro provedení zakázky. Nakonec se vytiskne zakázkový list

1.8.4 Výhody

- Možnost vedení účetnictví,
- registrace uživatelů systému,
- přístup uživatelů do systému podle jejich práv,
- obsahuje vytváření denních/týdenních rozpisů práce pro servisní techniky.

1.8.5 Nevýhody

- Síťová verze aplikace funguje tak že se nasdílí celá složka s programem,
- velmi nepřehledné,
- nekompletní, pouze základní funkcionality,
- pouze na platformě Windows,

- při startu stále vyskakovalo okno s prvotním nastavením,
- nesmyslné informace, například počet pneu, do kterých se dalo zapsat slovo.

1.9 ProCad - AutoServis

Program řeší zakázkovou činnost od založení zakázky (včetně interních zakázek a garančních), přes výdej materiálu a výdej prací, až po ukončení daňovým dokladem (interním dokladem). Obsahuje kompletní skladové hospodářství, modul prodeje náhradních dílů, eviduje odpracované hodiny pracovníků, zachovává kompletní historii zakázek, vede přehled o vozidlech, které navštívily servis. V programu lze využívat i řadu dalších funkcí. Pro menší servisy je v demoverzi zapracována verze MINI 1, pro prodejce náhradních dílů verze MINI 2, pro pneuservisy MINI 3. K dispozici jsou verze pro jeden PC i verze síťové. Nejnovější verze přináší kromě změn respektujících změny v legislativě i řadu dalších změn a možností – například ověřování IČ firemních zákazníků a přenos jejich fakturační adresy ze systému ARES, vkládání razítka a podpisu v podobě grafického souboru, tvorbu podkladů pro hlášení INTRASTAT, rozšíření možností středisek, pokladny podle středisek a zaměstnanců atd. V současné době je prodáno přes 1000 instalací programů firmy ProCad, spol. s r.o. Program AutoServis je používán jak v nezávislých autoservisech, tak v desítkách autorizovaných servisů různých značek vozidel a u řady dalších firem. AutoServis není pevně svázán s účetními programy, naopak nabízí možnost importu dat do různých typů účetních SW. Umožňuje import a následné naskladnění dodavatelských faktur v podobě souborů od APM Kdyně, AUTO KELLY, ELIT, AUTOBENEX. Do programu jsou průběžně zapracovávány změny podporující specifické potřeby jednotlivých automobilových značek i neznačkových autoservisů. Na základě požadavků našich uživatelů byly připraveny úpravy pro značky DAF, Fiat, Ford, Land Rover, Kia, Nissan, Opel, Renault, Pneuservisy DUNLOP atd. K programu jsou také nabízeny moduly Plánování zakázek, Pneuservis a Alternativní pohon, Hromadné načítání materiálu z čtečky čárového kódu, Komunikace s Eshop Pneu B2B, Ceník pro přihrání pneu, Export elektronických faktur pro ČSOB. Je nabízena aplikace internetového obchodu plně propojeného s programem AutoServis. Automatická aktualizace vybraných karet, příjem objednávek, odesílání potvrzovacích e-mailových zpráv atd. Dále nabízí možnost převodů dat z jiných systémů (založení skladových karet).

1.9.1 Instalace

I zde vše proběhlo bez zádrhelů. Před instalací si instalační program stáhl nejnovější verzi a tu nainstaloval.

1.9.2 Prvotní nastavení

Prvotní nastavení u demoverze je již založené se zkušební firmou s názvem Cvičná firma spol. V záložce speciál je okno s konfigurací systémů, ve kterém lze změnit firmu. Údaje o firmě jsou uloženy v licenci, takže ruční zásah nelze provést a vše se musí řešit s autorem programu.

1.9.3 Založení objednávky

Zakládání objednávek je nepřehledné. Okno má nastaveno defaultní velikost a nedá se upravit. V manuálu je uvedeno, že pro výběr vozidla nebo zákazníka se zadá do textového pole * a zmáčkne se enter, ale v dostupné demoverzi se výběr vozidla uskutečňuje pomocí tlačítka, které obsahuje tři tečky. O vozidle se zadají pouze základní informace. Detailní informace se dají také zadat, ale jejich obsah je až příliš detailní a zbytečný. Formulář obsahuje tlačítko náhodná SPZ, které doplní do SPZ 9místné číslo, po dokončení objednávky se ale místo SPZ objeví číslo objednávky, účel této funkce tedy zná zřejmě jen autor aplikace. Do typu vozidla se dá vybrat ze seznamu vozidel, po vložení se do textového pole se vypíše název vozidla. Zápis do databáze hotové objednávky je dlouhý.

1.9.4 Výhody

- Registrace uživatelů systému,
- přístup uživatelů do systému podle jejich práv,
- obsahuje vytváření denních/týdenních rozpisů práce pro servisní techniky.

1.9.5 Nevýhody

- Síťová verze aplikace funguje tak, že se nasdílí celá složka s programem,
- velmi nepřehledné,
- pouze na platformě Windows,
- zastaralý manuál,
- historie zakázek je nefunkční.

1.9.6 Carsystem

Carsystem je vyvíjen společností Carsys s.r.o. Tento informační systém je určen pouze pro autoservis. Předchozí systémy však měli Autoservis pouze jako rozšiřující modul, v čemž vidím výhodu, zejména z důvodu kvalitnější a obsáhlejší implementace. Carsystem je nabízen ve dvou variantách, a to Easy service a Full service. Podle názvu je možné odvodit, že Easy service je určen pro menší servis, zjednodušuje práci a obsahuje jen to nejzákladnější. Skládá se pouze ze 4 základních modulů (Adresář, Autoservis, Vozidla a Sklad). Nabízí úkony a základní funkce

pro vedení a fakturaci servisních zakázek, vedení katalogu prací s ceníkem, vedení zákazníků a vozidel, evidenci uskladněných pneumatik a reporty. Rozšířená varianta Full service se skládá z Adresář, Autoservis, Vozidla, Sklad a Vydané doklady a nabízí navíc tyto funkce: evidence výkonu mechaniků, Skladové hospodářství, možnost evidence poškození vozidel, kompletní přehled všech faktur a plateb, nástroje pro fakturaci pojišťovným a garančních oprav v rámci jedné zakázky. Největší výhodou je certifikace od dealerské společnosti Ford Motor Company s.r.o. v České Republice, což je další ukazatel kvality toho softwaru. I tato aplikace využívá MS SQL. Bohužel společnost nenabízí demoverzi softwaru, a proto jsem nemohl provést hodnocení.

1.10 TEAS – AutoServis

Poslední porovnávaný software je AutoServis od společnosti TEAS spol. s r.o. Nabízí všechny klasické funkce, které jsou potřeba pro kvalitní evidenci zakázek a vozidel. Předností toho softwaru je automatické napojení na další online systémy což je jedna z hlavních vlastností informačního systému. U ostatních programu uživatel musí zadat všechny údaje o vozidle ručně, zde stačí zadat VIN kód vozidla a program si sám ověří data na portálu Caris a stáhne si zbytek údajů. V tom vidím velkou úsporu času zaměstnance, a hlavně i malou šanci chybovosti zadávaných údajů. Další napojení je na informačním systému veřejné správy, kde se ověřují a stahují údaje podle IČO o zákazníkovi. Velkým nedostatkem aplikace je pouze nesíťová verze.

1.10.1 Instalace

Instalace proběhla bez problému. Vše potřebné si instalační balíček nainstaloval sám. Poté si stáhl licenční klíč a nabídl možnost aktivování jednotlivých modulů. Aplikace je napsaná ve jazyce FoxPro, který je určen jen pro Windows.

1.10.2 Prvotní nastavení

V této aplikaci nebylo potřeba provést žádné prvotní nastavení, vše potřebné již bylo v licenci.

1.10.3 Zakládání objednávky

Aplikace zobrazí okno, ve kterém jsou zašedlá okna, do kterých nejde nic doplnit a vedle nich je pouze tlačítko pro výběr vozidla ze seznamu servisovaných vozidel, anebo zákazníků. Pokud vozidlo není v seznamu dá se vytvořit pomocí tlačítka nový. Zde je možné vyplnit údaje ručně, nebo se zadá VIN kód a aplikace sama doplní skoro všechny informace o vozidle. Poté vybereme majitele vozidla, který se musí také ručně vyplnit a založit. Bohužel služba doplnění podle VIN kódu je placená, ovšem dokáže ušetřit tolik času, že si brzy na sebe vydělá. Poté je

možné ještě vybrat jiného objednavatele, nežli je majitele vozidla. Ten se ale také musí nacházet na seznamu zákazníku. Poté je možné upravit již provedený servis a upravit ho podle aktuální zakázky. V dalším kroku se vyplní karta zakázky, ve které jsou uvedeny informace o vozidle, majiteli, objednavateli, zakázce. Jako poslední se doplní seznam předběžných prací a doplní se do plánovače úkolu pro zaměstnance.

1.10.4 Výhody

- Doplnění údajů o vozidle podle VIN kódu,
- doplnění údajů podle ICO u objednavatele nebo majitele,
- evidence objednávek zákazníků,
- evidence opravovaných automobilů a údajů potřebných pro jeho servis,
- skladové hospodářství – několik cen pro různé činnosti,
- možnost vedení účetnictví,
- registrace uživatelů systému a přístup do systému podle jejich práv,
- velmi jednoduchá instalace,
- vytváření denních/týdenních rozpisů práce pro servisní techniky,
- možnost přijmout zakázku do autoservisu přes chytrý telefon nebo tablet.

1.10.5 Nevýhody

- Síťová verze aplikace funguje tak, že se nasdílí cela složka s programem,
- pouze na platformě Windows,
- občasné delší odezvy aplikace.

1.11 Tabulkové hodnocení existujících řešení

Tabulka 4 - Tabulkové hodnocení existujících řešení, zdroj vlastní

Název	Maximální skóre	Alfa-soft	KelEXPRESS	ADMWin	ProCad	TEAS
Instalace	5	1	4	5	5	5
HW nároky	5	4	4	3	4	5
Síťová možnost	10	10	10	10	5	5

Rozšiřitelné pomocí modulů	5	2	3	2	4	5
Jednoduchost práce	10	6	5	6	7	8
Multi platformost	5	0	0	0	0	0
Cena	5	5	5	4	3	3
Autoservis jako hlavní činnost	5	0	0	0	5	5

Maximální skóre, které mohla aplikace získat, je 55 bodů. To žádná aplikace nezískala.

Podle tabulkového hodnocení vychází pořadí

1. TEAS,
2. ProCad,
3. KeIEXPRES,
4. ADMWin,
5. Alfa-soft.

1.12 Informační systém popisovaný v této práci

Informační systém popsán v této bakalářské práci se v základech příliš neliší od již nabízených softwarových produktů. Nabízí ukládání informací o zákaznících, vozidlech, zaměstnancích, zakázkách. Aplikace je přístupná podle práv. Například mechanik si může zobrazit pouze report s rozpisem práce.

2 PRAKTICKÁ ČÁST

2.1 Návrh systému

2.1.1 Použité technologie

Použité technologie jsou vybrány podle jejich největších výhod. Pro grafické rozhraní je vybrán jazyk Java, který je možný spustit na jakémkoliv operačním systému. Pro běh aplikace je třeba nainstalovat daný operační systém. Další výhodou jazyku Java je jeho rozšiřitelnost. Pro ukládání dat je využívána technologie MySQL, která byla zvolena z důvodu běhu na operačním systému Windows, Linux a další. Další výhodou je možnost pronajmu MySQL, databáze na internetu a mít tak databázi dostupnou online a přístupnou odkudkoliv po celém světě. Všechny použité technologie již byly popsány v teoretické části.

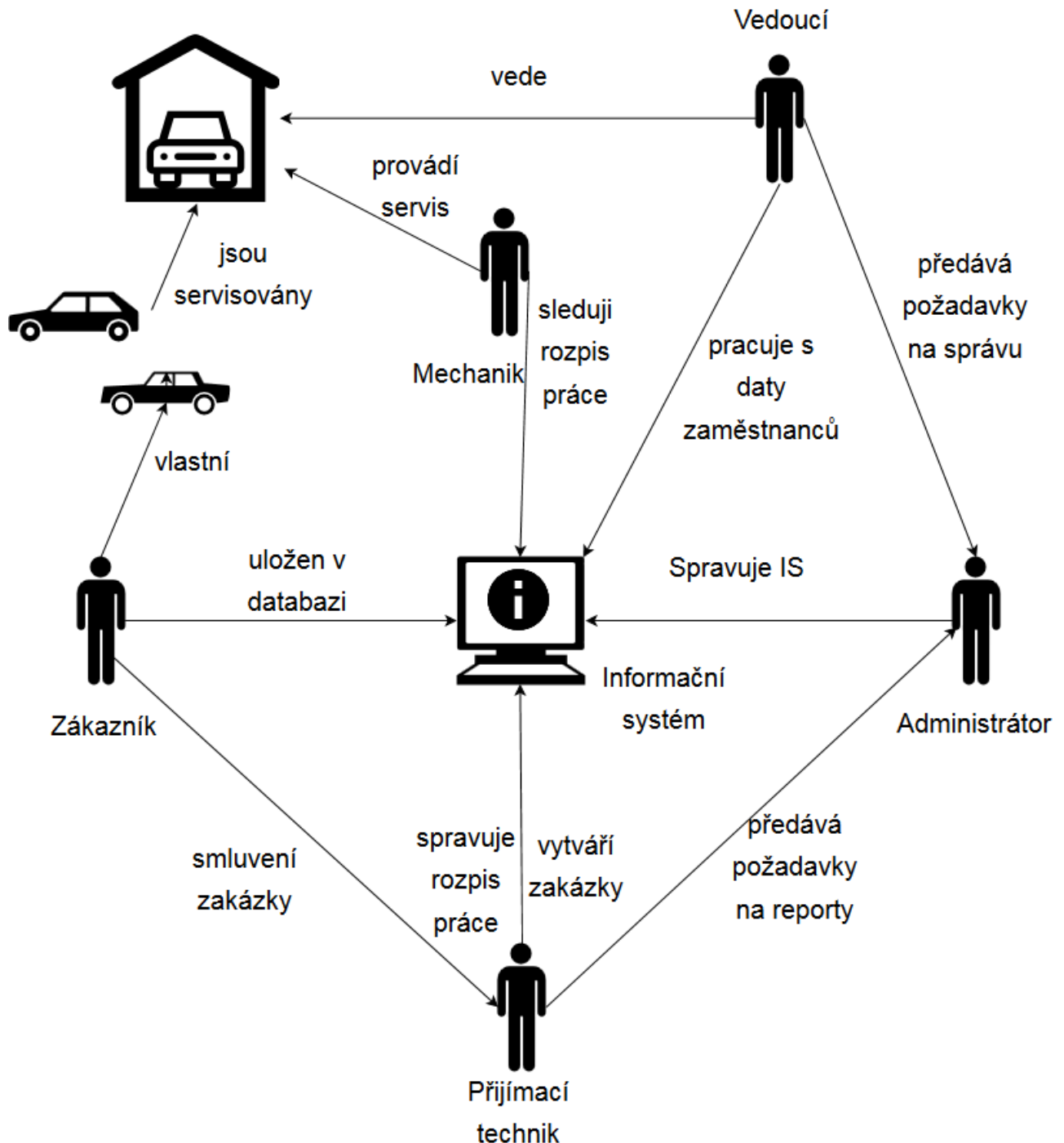
2.1.2 Bezpečnostní mechanismy

Jeden z bezpečnostních mechanismů, které MySQL obsahuje, je komunikace pomocí šifrovaného protokolu SSL. Pro připojení k databázi je použit speciální účet, který je možný v případě možnosti kdykoliv zablokovat. Další mechanismus je ukládání hesel do databáze ve formě hashe hesla. V komunikaci je možné pouze odchytnout hash hesla, pokud je tedy vypnut SSL šifrovací protokol.

2.2 Architektura aplikace

2.2.1 Rich picture

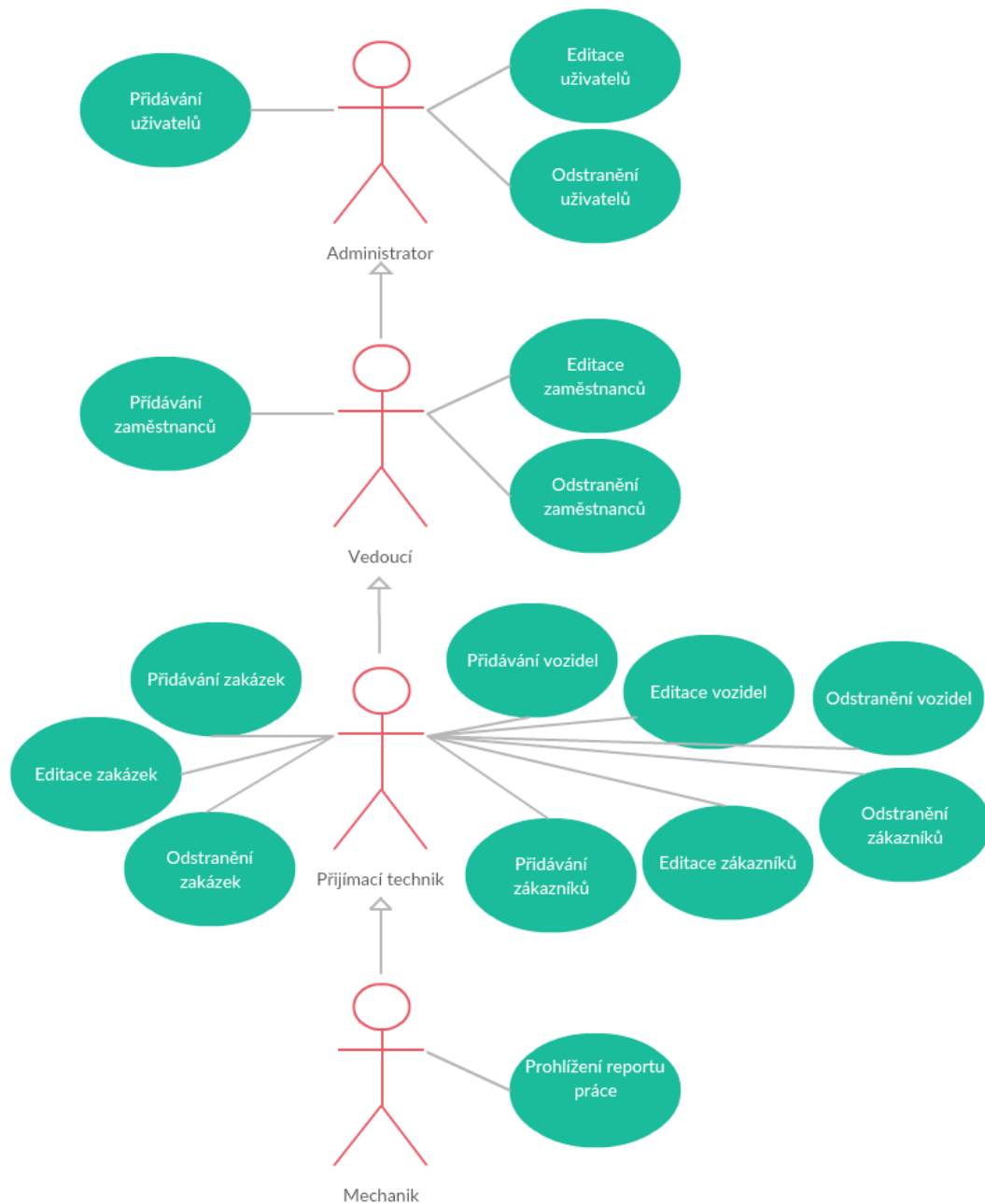
Rich picture je obrázek, který slouží k pochopení problému navrhovaného systému. Znárodnuje základní prvky a funkce systému, účastníky projektu a jejich interakce se systémem.



Obrázek 2 - Rich picture, zdroj [vlastní]

2.2.2 UML Use case diagram

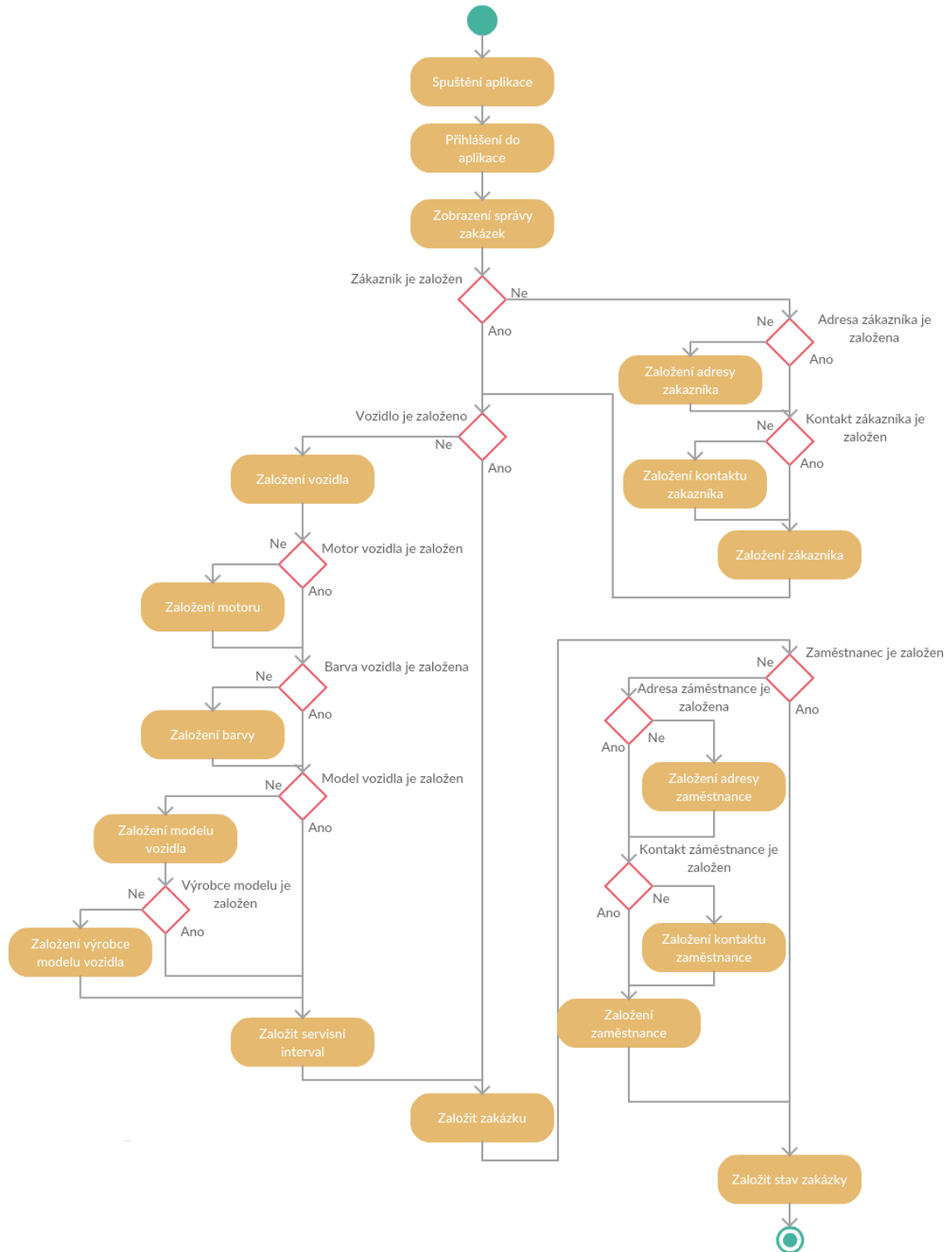
Use Case Diagram (česky diagram případů užití) zobrazuje použití systému tak, jak ho vidí uživatel, který ho bude používat ke své práci. Účelem diagramu je popsat funkcionalitu systému, tedy co od něj klient nebo my očekáváme. Diagram popisuje, co aplikace bude umět a provádět, ale již nepopisuje, jak se to bude provádět. Měl by být většinou jako první diagram, který se při návrhu informačního systému vytváří. Je důležité se nejprve shodnout na tom, co má systém umět a až potom se má řešit, jak to provést a dosáhnout toho.



Obrázek 3 – Use case diagram, zdroj [vlastní]

2.2.3 UML Activity diagram

Diagram aktivity popisuje chování aplikace. Diagram níže popisuje založení zakázky, nejsložitější funkci v programu.



Obrázek 4 - Activity diagram pro vytvoření zakázky, zdroj [vlastní]

2.3 Uživatelské role

Informační systém obsahuje čtyři role – mechanik, přijímací technik, vedoucí, administrátor. Role jsou hierarchicky řazené, to znamená, že vyšší role obsahuje všechny práva předchozí role.

2.3.1 Mechanik

Mechanik má pouze přístup do reportu práce, ve které zjistí časový harmonogram své denní nebo týdenní práce. Tato role nemá žádnou možnost editace, pouze prohlížení a zápis reálného startu a dokončení práce.

2.3.2 Přijímací technik

Přijímací technik má na starost kompletní práci se zakázkou, což znamená zakládání zakázky, ve které vybírá nebo přidává vozidlo, které bude servisováno a zákazníka. Dále má přístup do kompletní administrace vozidel. Zde je možné přidávat vozidla, editovat vozidla anebo vozidla smazat. Také má přístup do kompletní administrace zákazníků. Zákazníci se dělí na soukromé osoby a firemní zákazníky. I zde jde přidávat zákazníky, editovat zákazníky nebo zákazníky mazat.

2.3.3 Vedoucí

Vedoucí technik má přístup do všech předešlých funkcí, co obsahují nižší role. Navíc má možnost přidávat zaměstnance. Tato funkce automaticky zakládá i uživatele. Editace zaměstnanců a také mazání zaměstnanců.

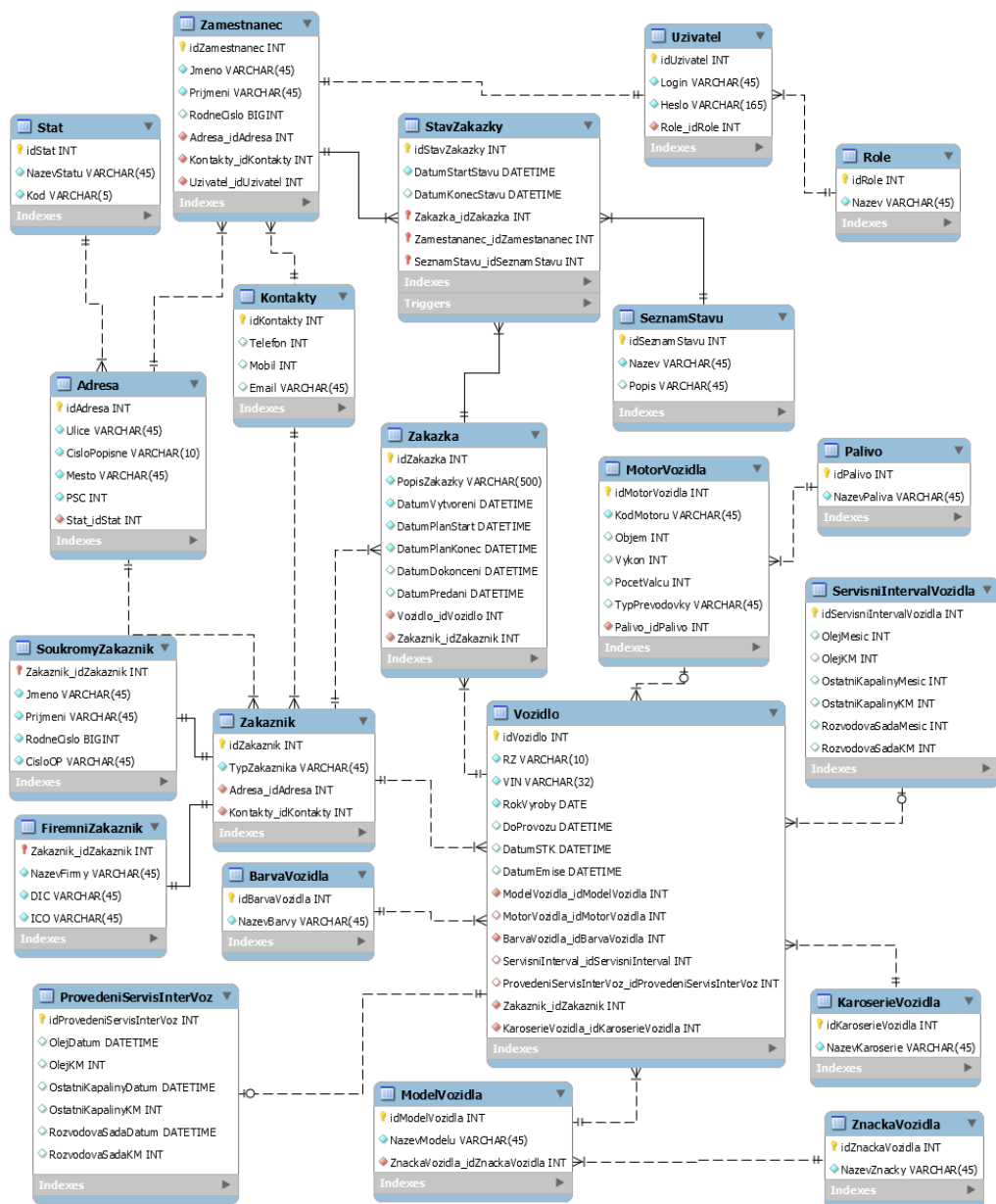
2.3.4 Administrátor

Administrátor je role, která má všechny funkce. Jako jediná má povolenou kompletní administraci uživatelů. Pod administrací uživatelů jsou funkce Přidávání uživatelů, Editace uživatelů, změna hesla, smazání uživatelů. Administrátor je doporučen pouze pro velmi znalého uživatele, anebo pro majitele autoservisu. Funkce přidávání uživatelů je automaticky obsažena při zakládání zaměstnance, ale lze vložit i uživatele bez zaměstnance. Například náhradního administrátora nebo jako testovacího uživatele. Editace uživatelů je určena pro změnu práv uživatele, uživatel se automaticky zakládá s nejnižším právem a tím je mechanik. Také je možné upravit přihlašovací jméno, kterým je automaticky přijmení zaměstnance. Další funkce je změna hesla. Defaultně je heslo nastaveno jako jméno zaměstnance. A poslední je smazání

uživatel. Je doporučeno, aby se smazal pouze přihlašovací uživatel a zaměstnanec se zachoval kvůli konzistenci zakázek a stavu.

2.4 Datový model

Na následujícím obrázku je datový model navrhované aplikace. Tento datový model zobrazuje způsob, jakým budou data ukládána do databáze a jak budou navzájem provázána. Model se skládá z 19ti tabulek, ve kterých jsou všechny údaje důležité pro běh aplikace a s tím spojený autoservis. Dále je popsána každá tabulka zvlášť. Každá tabulka obsahuje primární klíč, který nebudu popisovat.



Obrázek 5 – Datový model, zdroj [vlastní]

2.4.1 Tabulka Zakázka

Do PopisZakazky se zapisuje popis závad a potřebného servisu v zakázce. Dále je zde několik datumů, kdy DatumVytvoreni je zápis dat do databáze, DatumProvedeni zadává přijímací technik jako plánovaný start práce na zakázce a předpokládaný konec zakázky je vložen do DatumPredDokonceni - tyto dva datумы jsou klíčové pro plánování a generování rozpisu práce. Do DatumDokonceni se zadává reálné datum dokončení zakázky a do DatumPredavani se vkládá datum předání vozidla zpět zákazníkovi. Dále tu jsou dva cizí klíče pro vozidlo a zákazníka.

2.4.2 Tabulka Vozidlo

Tabulka vozidlo obsahuje zásadní informace o vozidle jako jsou registrační značka, VIN, rok výroby vozidla, datum uvedení do provozu, datum poslední technické kontroly a datum poslední kontroly emisí. Poté už obsahuje pouze cizí klíče na tabulky motor, karoserie, model, barva, servisní interval a provedený servisní interval.

2.4.3 Tabulky ModelVozidla a ZnackaVozidla

Tabulky obsahují pouze název modelů a k těm patřící značky.

2.4.4 Tabulky KaroserieVozidla, BarvaVozidla

V těchto tabulkách jsou uloženy vlastnosti vozidla - typ barvy vozidla a typ karoserie.

2.4.5 Tabulky MotorVozidla a Palivo

V tabulce MotorVozidla jsou informace o motoru daného vozidla. Tyto informace se skládají z kódu motoru, objemu, výkonu, počtu válců, typu převodovky a cizího klíče na palivo. Paliva jsou uložena v tabulce palivo, která obsahuje pouze název paliva.

2.4.6 Tabulky ServisniIntervalVozidla a ProvedeniServniIterVoz

V těchto tabulkách jsou údaje se servisním intervalem a o provedení těchto servisních intervalů.

2.4.7 Tabulka Zakaznik, SoukromyZakaznik a FiremniZakaznik

Tabulka Zakaznik obsahuje pouze primární klíč, typ zákazníka a dva cizí klíče pro tabulky Adresa a Kontakty. Na tuto tabulku jsou navázané tabulky SoukromyZakaznik a FiremniZakaznik, ve kterých jsou informace o zákazníkovi podle typu.

2.4.8 Tabulky Adresa a Kontakt

Tabulka Adresy slouží k evidenci adres zaměstnanců a zákazníků, tabulka Kontakt má stejnou funkci, ale obsahuje kontakty typu email, telefonní číslo a mobilní číslo.

2.4.9 Tabulky Zamestnanec, Uzivatel, Role

Do tabulky Zamestnanec se ukládá jméno, příjmení a rodné číslo zaměstnance. Po vložení zaměstnance se automaticky vytvoří uživatel, který je uložen v tabulce Uzivatel. Ta obsahuje Login, Heslo a cizí klíč na Role. V tabulce Role jsou 4 role, a to:

- Administrátor,
- Vedoucí,
- Přijímací technik,
- Mechanik.

2.4.10 Tabulky StavZakazky a SeznamStavu

V tabulce StavZakazky jsou sloupce DatumStartuStavu, do kterého se vkládá aktuální datum založení nového stavu a do sloupce DatumKonecStavu se ukládá aktuální datum při ukončení. A tři cizí identifikační klíče na tabulky Zakazka, Zamestnanec a na SeznamStavu, v němž je uložen seznam stavů, které mohou u zakázky nastat.

2.5 Instalační příručka

Pro běh aplikace je potřeba softwarové vybavení Java Runtime a MySQL server. Tyto soubory naleznete na přiloženém CD. Tato instalační příručka je určena pro systém Windows.

2.5.1 Instalace Javy

Podle operačního systému vybereme, jakou verzi Javy budeme instalovat. Po spuštění instalačního souboru se zobrazí okno, kde se klikne na tlačítko Install a poté proběhne instalace. Na konci instalace se klikne na Close a tím je instalace Javy dokončena.

2.5.2 Instalace MySQL

Instalace MySQL se provede pomocí utility MySQL Installer, která je umístěna na přiloženém CD. Po spuštění instalace se instalátor zeptá, zda chceme aktualizovat instalační soubory, což odsouhlasíme. Po aktualizaci se objeví hlavní instalační okno. Prvním krokem je přijetí licenčních podmínek. V dalším kroku se vybírá typ instalace. Pro tento případ vybereme možnost Server only. Poté jsou zobrazeny požadavky pro běh. Zde stačí kliknout na potřebný požadavek, následně na tlačítko Execute a provést instalaci požadavku. Další obrazovka

zobrazuje seznam instalovaných komponentů. Klikneme na Execute. Pokud instalace neproběhla korektně, tak je třeba ručně spustit instalaci souboru Visual Studio 2013 Redistributable také podle operačního systému. Po instalaci začne část s konfigurací. Zde vše necháme defaultní a zapamatujeme si heslo, které slouží k připojení k DB. Tímto máme MySQL server nainstalován.

2.5.3 Vložení databáze do MySQL

Po instalaci spustíme MySQL 5.7 Command Line Client – Unicode, kde se zobrazí terminálové okno, do kterého zadáme heslo z konfigurace. Poté do okna napíšeme SOURCE, za toto slovo vložíme mezeru a přetáhneme soubor autoservis.sql do okna. Zmáčkeme enter a skript se spustí. Po doběhnutí skriptu můžeme okno zavřít. Tím máme vše potřebné připravené.

2.5.4 Prvotní spuštění

Po prvotní přihlášení se použije administrátorský účet admin s heslem pokus. Poté je potřeba založit zaměstnance s rolí administrátor a smazat účet admin.

2.6 Uživatelská příručka

2.6.1 Přihlášení

Po spuštění aplikace se zobrazí okno s přihlašovacím formulářem. K přihlášení je potřeba přihlašovací jméno a heslo. V případě špatného uživatelského jména nebo hesla se zobrazí chybová zpráva.

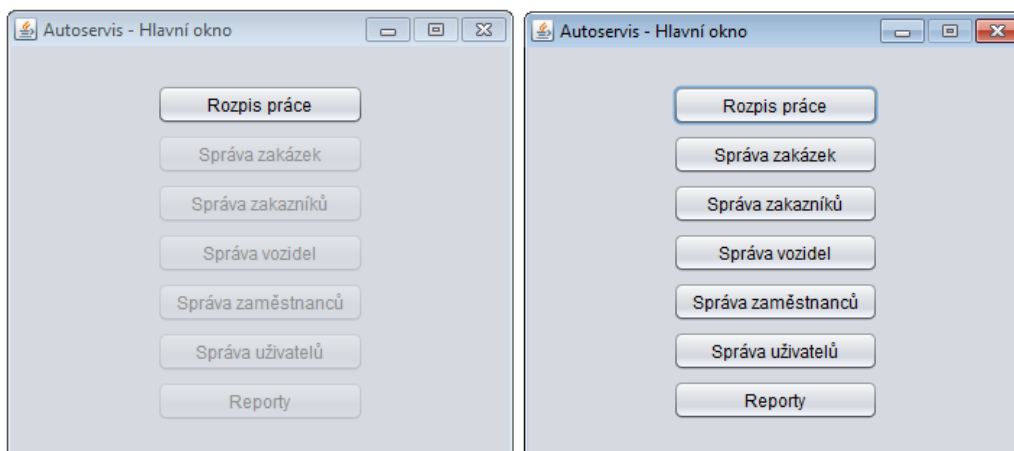


Obrázek 6 - Přihlašovací okno, zdroj [vlastní]

2.6.2 Hlavní menu

Po úspěšném přihlášení uživatele se zobrazí okno, ve kterém je hlavní menu. Tlačítka se odblokují podle role uživatele. Role mechanik má přístup pouze do Rozpisu práce, příjmací

technik má povoleno vstoupit do správy vozidel, zakázek a zaměstnanců, vedoucí má navíc možnost správy zaměstnanců a administrátor může přidávat uživatele.



Obrázek 7 - Hlavní menu, zdroj [vlastní]

2.6.3 Rozpis práce

V okně se zobrazuje naplánovaná práce pro zaměstnance. Rozpis se dá zobrazit v týdenním a denním rozpisu od určitého datumu. Dále okno obsahuje tlačítko pro výpis rozpracované práce a popisové pole ve kterém si zaměstnanec přečte co je potřeba provést na vozidle. Mechanik může změnit pouze stav zakázky, pokud má nějakou zakázku ve stavu probíhá tak se zobrazí tato zakázka a přepínací prvky jsou zablokovány.

2.6.4 Správa zakázek

Zde se zakládá a přiřazuje zakázka mechanikovi. Potřebné informace pro založení je potřeba vybrat plánovaný začátek provádění zakázky, plánovaný konec provádění zakázky, vozidlo, které se bude servisovat a zákazníka. Po založení je potřeba zakázku přiřadit zaměstnanci.

2.6.5 Správa zákazníků

Ve správě zákazníků se vybírá, zda je zákazník soukromý anebo firemní podle této volby se pak vyplňují odlišné údaje. Společné údaje jsou adresa a kontakty.

2.6.6 Správa vozidel

Správa vozidel se skládá ze čtyř oken. V hlavním okně se zakládá a upravuje vozidlo podle základních parametrů vozidla jako je registrační značka, VIN, datum technické kontroly a emisích, rok výroby a datum uvedení do provozu. Poté je potřeba vybrat majitele vozu ze zákazníků. Je možné přidat detailní informace o motoru vozidla a servisní interval o vozidle který je potřeba pro report potřebného servisu.

2.6.7 Správa zaměstnanců

V tomto okně jde přidávat zaměstnanec s potřebnými údaji jako jsou jméno, příjmení, rodné číslo, kontakt a adresa. Po vložení zaměstnance se automaticky přidá i do uživatelů kde je přihlašovací jméno příjmení a heslo je křestní jméno se základní roli mechanik. Vedoucí pak může ve správě uživatelů změnit pozici a heslo.

2.6.8 Správa uživatelů

Do správy uživatelů smí pouze role administrátor a vedoucí. Vedoucí je ještě omezen na akce změna heslo a role. Administrátor má vše povolené.

2.6.9 Reporty

Okno reporty obsahuje dvě tlačítka, každé tlačítko obsahuje jiný report. Po kliknutí na daný report si uživatel vybere, na které vozidlo nebo zakázku chce report vytvořit.

3 ZÁVĚR

V teoretické části byla provedena analýza konkurenčních systémů, na základě této analýzy bylo zjištěno že většina nabízených konkurenčních informačních systémů je příliš složitá a nejsou multiplatformní. Kvůli těmto důvodům je tento informační systém implementován technologií, která je multiplatformní. Další z výhod tohoto softwaru je využití technologie MySQL pro databázovou část. Tuto technologii nabízí v provozu několik webhostingů, takže jde aplikaci používat odkudkoliv kde je dostupný internet, tuto vlastnost jde využít, pokud budou data uložena v online databázi v webhostingu. Tuto vlastnost nenabízí žádný ze systémů porovnávaných v úvodní analýze. Tato aplikace bylo vyzkoušena na několika datech a všechny testy proběhly bez problému. Po instalaci aplikace se automaticky vloží ilustrační data pro rychlejší osvojení programu. Do budoucna je v plánu rozšířit aplikaci o evidenci použitých náhradních dílů, sklad pneumatik, tisk papírových reportů a online rezervace zakázek z webové stránky. Díky bakalářské práci jsem si zlepšil znalosti programování v jazyce Java. Prohloubil jsem si znalost databáze MySQL a jazyka SQL. A osvojil jsem si MySQL Workbench který je velmi užitečný pomocník při návrhu a úpravě databáze.

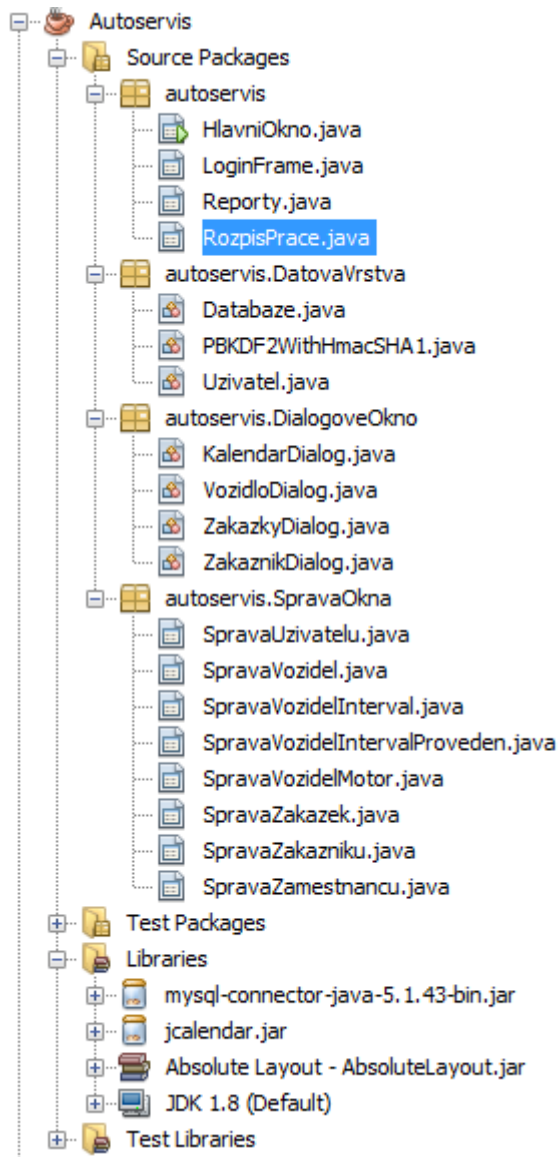
4 POUŽITÁ LITERATURA

- [1] Kdo byl první? - Gottlieb Daimler. *Eurooldtimers.com* [online]. 2017 [cit. 2017-08-21]. Dostupné z: <http://www.eurooldtimers.com/cze/historie-clanek/793-kdo-byl-prvni-gottlieb-daimler.html>
- [2] ARES. *Administrativní registr ekonomických subjektů* [online]. 2013 [cit. 2017-08-28]. Dostupné z: <http://www.info.mfcr.cz/ares/ares.html.cz>
- [3] Informační systém o informačních systémech veřejné správy. *IS o ISVS* [online]. 2015 [cit. 2017-08-21]. Dostupné z: <https://www.sluzby-isvs.cz/ISOISVS/Views/Public/Login.aspx?action=get>
- [4] Historie relačních databází. *ROOT.CZ* [online]. 2001 [cit. 2017-08-21]. Dostupné z: <https://www.root.cz/clanky/historie-relacnich-databazi/>
- [5] ISO/IEC 9075-1:2016: Information technology -- Database languages -- SQL -- Part 1: Framework (SQL/Framework). *International Organization for Standardization* [online]. 2016 [cit. 2017-08-21]. Dostupné z: <https://www.iso.org/standard/63555.html>
- [6] Základy relačních databází, jejich využití v programování webu. *Vysoká škola ekonomická v Praze* [online]. [cit. 2017-08-21]. Dostupné z: <http://gml.vse.cz/data/oppa-webdesign/zaklady-db.html>
- [7] RDBMS, Relational Database Management System. *Svět sítí* [online]. [cit. 2017-08-21]. Dostupné z: <http://www.svetsiti.cz/slovník.asp?hid=RDBMS-Relational-Database-Management-System>
- [8] DB-Engines Ranking of Relational DBMS. *DB-Engines* [online]. c2017 [cit. 2017-08-21]. Dostupné z: <https://db-engines.com/en/ranking/relational+dbms>
- [9] Top 10 Enterprise Database Systems of 2017. *ServerWatch* [online]. 2017 [cit. 2017-08-21]. Dostupné z: <http://www.serverwatch.com/server-trends/slideshows/top-10-enterprise-database-systems-to-consider-2015.html>
- [10] System Properties Comparison DB2 vs. Microsoft SQL Server vs. MySQL vs. Oracle vs. PostgreSQL. *DB-Engines* [online]. [cit. 2017-08-21]. Dostupné z: <https://db->

engines.com/en/system/DB2%3BMicrosoft+SQL+Server%3BMySQL%3BOracle%3BPostgreSQL

- [11] TIOBE Index for August 2017. *TIOBE* [online]. c2017 [cit. 2017-08-21]. Dostupné z: <https://www.tiobe.com/tiobe-index/>
- [12] Základní principy a pojmy objektově orientovaného programování. *PéHáPko.cz* [online]. [cit. 2017-08-21]. Dostupné z: <http://pehapko.cz/oop/uvod>
- [13] Objektově orientované programování. *Jak se naučit programovat* [online]. 2004 [cit. 2017-08-21]. Dostupné z: <http://jaksenaucitprogramovat.py.cz/cztutclass.html>
- [14] Programming Language Comparison. *Jason Voegele's Cyberhome* [online]. [cit. 2017-08-21]. Dostupné z: <http://www.jvoegele.com/software/langcomp.html>
- [15] Objektově orientované programování a evoluce vývoje softwaru. *ITnetwork.cz* [online]. [cit. 2017-08-21]. Dostupné z: <https://www.itnetwork.cz/navrhove-vzory/objektove-orientovane-programovani-a-evoluce-vyvoje-softwaru>
- [16] Objektově orientované programování? Co to je? *Objektově orientované programování a jeho výuka* [pdf]. 2009 [cit. 2017-08-21]. Dostupné z: http://vyuka.pecinovsky.cz/prispevky/2009_PS_OOP_-_Co_to_je.pdf
- [17] How to Comparing C# to C++ and Java. *EComputerNotes* [online]. [cit. 2017-08-21]. Dostupné z: <http://ecomputernotes.com/csharp/cs/comparing-c-to-c-and-java>

PŘÍLOHA A – ADRESÁŘOVÁ STRUKTURA APLIKACE



PŘÍLOHA B – ZDROJOVÝ KOD REPORTU

```
int idZakazky = ukazZakazky();
    if (idZakazky != 0) {
        String vytisk = "";
        try {
            PreparedStatement pst = db.CreatePreparedStatement(" SELECT
"
                + " `seznamstavu`.`Nazev`,"
                + " SEC_TO_TIME(SUM(TIME_TO_SEC(DatumKonecStavu)
- TIME_TO_SEC(DatumStartStavu))) AS Cas,"
                + " `stavzakazky`.`SeznamStavu_idSeznamStavu`"
                + " FROM `autoservis`.`stavzakazky`"
                + " left join autoservis.seznamstavu on
stavzakazky.SeznamStavu_idSeznamStavu = seznamstavu.idSeznamStavu "
                + " WHERE Zakazka_idZakazka = ? Group by
SeznamStavu_idSeznamStavu");
            pst.setInt(1, idZakazky);
            ResultSet rs = pst.executeQuery();
            rs.next();
            vytisk += "ID zakázky je " + idZakazky + ". ";
            for (int i = 0; i < 5; i++) {
                vytisk += "Stav: ";
                vytisk += rs.getString("Nazev");
                vytisk += " probíhal ";
                vytisk += rs.getString("Cas");
                vytisk += " ";
                rs.next();
            }

        } catch (SQLException ex) {
            Logger.getLogger(SpravaZamestnancu.class
                .getName()).log(Level.SEVERE, null, ex);
        }
        JOptionPane.showMessageDialog(null, vytisk, "Výsledek reportu",
            JOptionPane.INFORMATION_MESSAGE);
    } else {
        JOptionPane.showMessageDialog(null, "Nebylo vybrána žádná
zakázka", "Chyba reportu", JOptionPane.ERROR_MESSAGE);
    }
}
```