

Návrh a implementace aplikace pro správu pracovních cest

Diplomová práce

Vedoucí práce:

RNDr. Zuzana Prišćáková, Ph.D

Bc. Gabriela Macáková

Brno 2015

Poděkování

Touto cestou bych ráda poděkovala vedoucí své diplomové práce RNDr. Zuzaně Prišćákové, Ph.D za cenné rady, připomínky a odborné vedení. Dále bych chtěla poděkovat především své rodině a přátelům, kteří mi v této důležité chvíli vycházeli tak obětavě vstříc.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Návrh a implementace aplikace pro správu pracovních cest** vypracovala 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 *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědoma, 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 15. prosince 2015

Abstract

Macáková, G. Design and implementation of application for business trip's administration. Diploma thesis. Brno: Mendel university, 2015.

The diploma thesis is focused on the design and implementation of application for business trips's administration. The aim of this thesis is to familiarize the readers with the issue of business trips from the economic and financial point of view, with the basics of information systems and with the designing of diagrams in UML's notation. From the practical point of view this thesis deals with the design of database and application which should be helpful for automation of the creation and approval business trip's process and also deals with the administration of all created business trips. The outcome of this thesis is the design of UML's diagrams and implementation of application for business trips's administration in C# programming language.

Keywords

.NET, administration, business trip, C#, design, desktop application, diagram, diploma thesis, information system, MySQL, UML

Abstrakt

Macáková, G. Návrh a implementace aplikace pro správu pracovních cest. Diplomová práce. Brno: Mendelova univerzita v Brně, 2015.

Diplomová práce se zabývá návrhem a implementací aplikace pro správu pracovních cest. Cílem práce je seznámit čtenáře s problematikou pracovních cest z hlediska ekonomického, se základy tvorby informačních systémů a s návrhem diagramů v notaci UML. Z praktického hlediska se práce zabývá návrhem databáze a aplikace, jež by měla napomoci automatizaci procesu tvorby a schválení pracovní cesty a celkové správě vytvořených pracovních cest. Výstupem diplomové práce je návrh diagramů UML a implementace aplikace pro správu pracovních cest v jazyce C#.

Klíčová slova

.NET, C#, desktopová aplikace, diagram, diplomová práce, informační systém, návrh, pracovní cesta, správa, MySQL, UML

Obsah

1	Úvod a cíl práce	13
1.1	Úvod.....	13
1.2	Cíl práce.....	13
2	Struktura práce	14
3	Teoretická východiska práce	15
3.1	Pracovní cesty z hlediska Zákoníku práce.....	15
3.1.1	Definice pracovní cesty.....	15
3.1.2	Definice zahraniční pracovní cesty.....	15
3.2	Náhrada výdajů v souvislosti s výkonem práce.....	16
3.2.1	Druhy cestovních náhrad.....	16
3.2.2	Náhrada jízdních výdajů.....	17
3.2.3	Náhrada výdajů za ubytování.....	18
3.2.4	Náhrada výdajů za stravování (stravné).....	18
3.2.5	Náhrada nutných vedlejších výdajů.....	19
3.3	Cestovní náhrady při zahraniční pracovní cestě.....	19
3.3.1	Druhy cestovních náhrad.....	19
3.3.2	Náhrada jízdních výdajů.....	20
3.3.3	Náhrada výdajů za ubytování.....	20
3.3.4	Náhrada výdajů za stravování (zahraniční stravné).....	20
3.3.5	Náhrada nutných vedlejších výdajů.....	21
3.4	Informační systémy.....	22
3.4.1	Základní pojmy.....	22
3.4.2	Architektura informačního systému.....	22
3.4.3	Globální architektura IS.....	23
3.4.4	Dílčí architektury IS.....	25
3.4.5	Vrstvená architektura IS.....	26
3.5	Návrh informačních systémů.....	27
3.5.1	Strukturovaný přístup.....	28

3.5.2	Objektově orientovaný přístup.....	28
3.6	UML notace	29
3.6.1	Charakteristika UML.....	29
3.6.2	Struktura a základní stavební bloky jazyka UML.....	30
3.6.3	Předměty (things).....	30
3.6.4	Relace (relationships)	31
3.6.5	Diagramy (diagrams).....	32
3.7	Implementační technologie informačních systémů.....	34
3.7.1	Softwaroví klienti.....	34
3.7.2	Programovací jazyky.....	36
3.8	Shrnutí kapitoly	39
4	Analytická část	41
4.1	Dostupná řešení.....	41
4.1.1	plusPortal	41
4.1.2	HELIOS Green	42
4.1.3	Informační systém Vema.....	42
4.1.4	Cesty ver.3.0 Perm	43
4.1.5	Wheel CRM & DMS.....	44
4.1.6	Ekonomický systém POHODA	44
4.2	Stručná charakteristika podniku.....	45
4.3	Informace o pracovní cestě	45
4.3.1	Cestovní příkaz.....	45
4.3.2	Proces pracovní cesty	46
4.4	Analýza požadavků.....	47
4.4.1	Funkční požadavky.....	47
4.4.2	Nefunkční požadavky	48
4.4.3	Požadavky na účastníky procesu	49
4.5	Srhnutí kapitoly	50
5	Vlastní řešení	51
5.1	Návrh aplikace pro správu pracovních cest.....	51
5.1.1	Diagram případů užití	51

5.1.2	Diagram tříd	53
5.1.3	Sekvenční diagram.....	53
5.1.4	Diagram aktivit.....	55
5.1.5	Stavový diagram	56
5.2	Implementace aplikace pro správu pracovních cest	58
5.2.1	Návrh databáze	59
5.2.2	Přihlášení do aplikace	63
5.2.3	Manažer správy pracovních cest.....	63
5.2.4	Nová pracovní cesta	64
5.2.5	Kontrola základních údajů.....	65
5.2.6	Vyplacení zálohy.....	66
5.2.7	Zpráva o výsledku pracovní cesty.....	67
5.2.8	Vyúčtování pracovní cesty.....	68
5.2.9	Doplnění pokladního dokladu.....	70
5.3	Testování aplikace	72
5.4	Shrnutí kapitoly	72
6	Diskuze	73
6.1	Vlastní řešení vs. dostupná řešení	73
6.2	Výhody a nevýhody vlastního řešení.....	74
6.2.1	Výhody.....	74
6.2.2	Nevýhody.....	75
6.3	Zhodnocení vlastního řešení a návrh vylepšení.....	75
7	Závěr	77
8	Literatura	78
A	Správa pracovních cest	82

Seznam obrázků

Obr. 1	Dílčí architektura informačního systému	24
Obr. 2	Diagramy UML	32
Obr. 3	Diagram případů užití pro proces tvorby a schválení pracovní cesty	52
Obr. 4	Diagram tříd	53
Obr. 5	Sekvenční diagram tvorby nové pracovní cesty	54
Obr. 6	Diagram aktivit pro založení nové pracovní cesty	56
Obr. 7	Stavový diagram procesu tvorby a schvalování pracovní cesty	57
Obr. 8	Model databáze	59
Obr. 9	Úvodní přihlášení uživatele do aplikace	63
Obr. 10	Manažer správy pracovních cest z pohledu zaměstnance	64
Obr. 11	Doplnění základních údajů o pracovní cestě	65
Obr. 12	Kontrola základních údajů o pracovní cestě	66
Obr. 13	Vyplacení zálohy na pracovní cestu	67
Obr. 14	Vytvoření pokladního dokladu	67
Obr. 15	Zpráva o výsledku pracovní cesty	68
Obr. 16	Zprávy o výsledku pracovní cesty	68
Obr. 17	Vyúčtování pracovní cesty	69
Obr. 18	Výběr dokladů nabytých během pracovní cesty	70
Obr. 19	Pokladní doklady k přeúčtování	71
Obr. 20	Doplnění pokladního dokladu	71
Obr. 21	Diagram tříd (vyšší rozlišení)	82

Obr. 22	Sekvenční diagram tvorby nové pracovní cesty (vyšší rozlišení)	83
Obr. 23	Diagram aktivit pro založení nové pracovní cesty (vyšší rozlišení)	84

Seznam tabulek

Tab. 1	Požadavky na účastníky procesu pracovní cesty	49
---------------	--	-----------

1 Úvod a cíl práce

1.1 Úvod

Pracovní cesty jsou problematikou, která byla řešena alespoň jedenkrát snad v každém větším existujícím podniku. Mnoho podniků své zaměstnance dokonce pravidelně vysílá na nejrůznější školení do velkých měst či do podnikových poboček v zahraničí.

Doposud byla v podnicích tato problematika řešena ručně, bez jakékoliv automatizace procesu. Zaměstnanci tak byli nuceni před pracovní cestou navštívit veškeré účastníky tohoto procesu, aby získali potřebné údaje do formulářů cestovních výkazů. Jednalo se o podpisy, různá schválení, apod. Tento proces tak mohl být velice zdoluhavý, což způsobovalo, že se zaměstnanec nemohl věnovat smysluplnější práci a podnik tak přicházel o zisk.

Doba však pokročila a podniky začaly modernizovat. V současné době lze vidět snahu v podobě zavádění automatizací napříč celým podnikem. Ty mají za úkol snižovat časové prodlevy jednotlivých zaměstnanců a zvyšovat zisk v podniku. Není tedy divu, že i doména pracovních cest prošla automatizací. Moduly pracovních cest jsou součástí mnoha velkých podnikových informačních systémů, které jsou nasazeny ve většině středně velkých či velkých podniků. Nicméně se stále jedná o řešení, ve kterých se do popředí dostává snaha vyhovět co nejširšímu spektru zákazníků (uživatelů).

Ale každý podnik je jiný, a proto každému vyhovuje jiné řešení. Z toho důvodu vznikla i tato diplomová práce, právě kvůli požadavku na aplikaci, která by byla šitá na míru podnikovým potřebám.

1.2 Cíl práce

Primárním cílem této diplomové práce je seznámit čtenáře s problematikou pracovních cest z ekonomického hlediska, dále se základy tvorby informačních systémů a s jejich návrhem za pomoci diagramů v notaci UML včetně nastínění použitelných technologií pro následnou implementaci.

Sekundárním cílem diplomové práce je navrhnout a implementovat aplikaci pro správu pracovních cest dle požadavků podniku, pro který je praktická část diplomové práce realizována. Aplikace bude následně podrobena testování, na jehož základě budou objektivně zhodnoceny její výhody a nevýhody, a diskutována případná budoucí rozšíření.

2 Struktura práce

Diplomová práce bude rozdělena do jednotlivých částí (kapitol), které budou popsány níže.

V teoretické části práce s názvem Teoretická východiska práce bude podrobně rozebrána teorie potřebná k naplnění cílů diplomové práce. Konkrétně se tak bude jednat o objasnění problematiky pracovních cest z hlediska příslušného zákona Zákoníku práce. V dalším kroku bude objasněna problematika informačních systémů. Konkrétně zde bude popsána jejich základní terminologie, jednotlivé typy architektur informačních systémů. Kapitola bude pokračovat návrhem informačních systémů, kdy budou představeny nejběžněji používané přístupy a s nimi související grafické notace. Podrobněji pak bude popsána právě notace UML, která poslouží pro praktickou část práce. Po objasnění návrhu bude přistoupeno k objasnění fáze tvorby informačních systémů, s čímž souvisí také typy softwarových klientů a konkrétní implementační technologie.

Po teoretické části bude následovat Analytická část, která se bude věnovat dostupným řešením stejné problematiky (tedy oblasti pracovních cest). Vybraná dostupná řešení pak budou představena a použita dále v práci. Analytická část bude obsahovat také stručnou charakteristiku podniku, pro který je praktická část diplomové práce realizována, a informace o tom, jakým způsobem jsou pracovní cesty v podniku řešeny. Podrobně tak bude představen cestovní příkaz a proces tvorby a schvalování pracovní cesty. Dále bude sepsán seznam požadavků na aplikaci ze strany podniku.

Další částí bude část Vlastní řešení, která se bude zabývat již praktickým řešením diplomové práce. Tato kapitola bude obsahovat fázi návrhu a implementace. Ve fázi návrhu budou představeny a popsány diagramy UML, které poslouží pro implementaci aplikace a také pro představu, jakým způsobem lze s touto notací pracovat. Ve fázi implementace pak bude zohledněna tvorba databáze a podrobně popsána funkcionalita celé aplikace včetně printscreenů aplikace.

V závěru práce pak bude implementovaná aplikace podrobena diskuzi, bude provedeno její srovnání s již existujícími řešeními, dále budou definovány výhody a nevýhody, a dojde také k celkovému zhodnocení aplikace, kdy budou navržena případná budoucí rozšíření.

3 Teoretická východiska práce

V první části této kapitoly bude podrobně nastíněna problematika pracovních cest z hlediska ekonomického. V další části pak bude nastíněna problematika návrhu informačních systémů za pomoci notace UML, a také problematika jejich následné implementace včetně zmínění nejběžněji využívaných implementačních technologií.

3.1 Pracovní cesty z hlediska Zákoníku práce

Aby bylo možné zrealizovat návrh a následnou implementaci požadované aplikace, bylo nejprve nutné nastudovat příslušné zákony, dle kterých probíhá přepočítání jednotlivých nákladových složek vyúčtování pracovní cesty. Problematikou pracovních cest a jejich náhrad se v České republice zabývá Zákoník práce. Před samotným přistoupením k jednotlivým paragrafům Zákoníku práce je nutno definovat, co to vlastně (zahraniční) pracovní cesta je.

3.1.1 Definice pracovní cesty

Pracovní cestou dle § 42 Zákoníku práce se rozumí „časově omezené vyslání zaměstnance zaměstnavatelem k výkonu práce mimo sjednané místo výkonu práce. Zaměstnavatel může vyslat zaměstnance na dobu nezbytné potřeby na pracovní cestu jen na základě dohody s ním. Zaměstnanec na pracovní cestě koná práci podle pokynů vedoucího zaměstnance, který ho na pracovní cestu vyslal“. (Andraščíková a kol., 2015)

„Vysílá-li zaměstnavatel zaměstnance na pracovní cestu k plnění svých úkolů do jiné organizační složky (k jinému zaměstnavateli), může pověřit jiného vedoucího zaměstnance (jiného zaměstnavatele), aby zaměstnanci dával pokyny k práci, popřípadě jeho práci organizoval, řídil a kontroloval; v pověření je třeba vymezit jeho rozsah. S pověřením podle věty první musí být zaměstnanec seznámen. Vedoucí zaměstnanci jiného zaměstnavatele však nemohou vůči zaměstnanci jménem vysílajícího zaměstnavatele právně jednat“. (Andraščíková a kol., 2015)

3.1.2 Definice zahraniční pracovní cesty

Zahraniční pracovní cestou dle § 154 Zákoníku práce se rozumí „časově omezené vyslání zaměstnance zaměstnavatelem k výkonu práce mimo sjednané místo výkonu práce. Zaměstnavatel může vyslat zaměstnance na dobu nezbytné potřeby na pracovní cestu jen na základě dohody s ním. Zaměstnanec na pracovní cestě koná práci podle pokynů vedoucího zaměstnance, který ho na pracovní cestu vyslal“. (Andraščíková a kol., 2015)

„Vysílá-li zaměstnavatel zaměstnance na pracovní cestu k plnění svých úkolů do jiné organizační složky (k jinému zaměstnavateli), může pověřit jiného vedoucího zaměstnance (jiného zaměstnavatele), aby zaměstnanci dával pokyny k práci, popřípadě jeho práci organizoval, řídil a kontroloval; v pověření je třeba vymezit jeho

rozsah. S pověřením podle věty první musí být zaměstnanec seznámen. Vedoucí zaměstnanci jiného zaměstnavatele však nemohou vůči zaměstnanci jménem vysílajícího zaměstnavatele právně jednat". (Andraščíková a kol., 2015)

3.2 Náhrada výdajů v souvislosti s výkonem práce

Po objasnění definice (zahraniční) pracovní cesty lze přistoupit k výběrům paragrafů a odstavců nezbytných pro propočet výdajů pracovních cest souvisejících s touto diplomovou prací.

Dle § 151 Zákoníku práce je zaměstnavatel, který vyslal zaměstnance na pracovní cestu, ze zákona povinen poskytnout vyslanému zaměstnanci náhradu výdajů, které mu v souvislosti s touto pracovní cestou vzniknou, v rozsahu a za podmínek, které budou vymezeny dále. (Andraščíková a kol., 2015)

§ 152 Zákoníku práce stanovuje, že za cestovní výdaje, které jsou zaměstnavatelem poté propláceny, se považují výdaje, které zaměstnanci vzniknou při:

- pracovní cestě,
- cestě mimo regulérní pracoviště,
- mimořádné cestě v souvislosti s výkonem práce mimo místo výkonu práce či pravidelného pracoviště,
- přeložení,
- dočasném přidělení,
- přijetí do zaměstnání v pracovním poměru,
- výkonu práce v zahraničí. (Andraščíková a kol., 2015)

3.2.1 Druhy cestovních náhrad

Dle § 156 Zákoníku práce jsou rozlišovány tyto druhy cestovních náhrad, za které je zaměstnavatel povinen poskytnout zaměstnanci při pracovní cestě náhradu:

- jízdní výdaje,
- jízdní výdaje při návštěvě člena rodiny,
- výdaje za ubytování,
- zvýšené stravovací výdaje (dále jen "stravné"),
- nutné vedlejší výdaje. (Andraščíková a kol., 2015)

Zaměstnavatel však může dle svého uvážení poskytnout zaměstnanci náhrady i za další výdaje, které nejsou uvedeny výše. Za cestovní náhrady jsou však primárně považovány ty, které jsou uvedeny již v dříve zmíněném § 156. (Andraščíková a kol., 2015)

3.2.2 Náhrada jízdních výdajů

Při pracovních cestách je pro zaměstnance důležitá zejména možnost náhrady jízdních výdajů, které mnohdy dosahují vysokých částek. Náhradu jízdních výdajů v Zákoníku práce ošetřuje § 157, který bude posán níže.

Při použití určeného prostředku hromadné dopravy pro dálkovou přepravu či taxislužby přísluší zaměstnanci náhrada jízdních výdajů v prokázané výši. Pokud se stane, že zaměstnanec k přepravě na místo výkonu práce využije místo určeného hromadného dopravního prostředku jiný dopravní prostředek včetně silničního motorového vozidla (s výjimkou vozidel poskytnutých samotným zaměstnavatelem), náleží mu náhrada jízdních výdajů v odpovídající ceně jízdného za využitý hromadný dopravní prostředek. (Andraščíková a kol., 2015)

Pokud zaměstnanec na žádost zaměstnavatele použije silniční motorové vozidlo (s výjimkou vozidla poskytnutého samotným zaměstnavatelem), náleží mu za každý 1 km jízdy následující základní sazby:

1. jednostopá vozidla a tříkolky – **1,00 Kč**,
2. osobní silniční motorová vozidla – **3,80 Kč**. (Andraščíková a kol., 2015)

Pokud zaměstnanec hodlá využít k silničnímu motorovému vozidlu i přívěs, zvýší se sazba základní náhrady za 1 km minimálně o **15 %**. Tato sazba se mění v závislosti na vývoji cen a je ošetřována § 189 Zákoníku práce. Při využití nákladního automobilu, autobusu či traktoru náleží zaměstnanci náhrada ve výši nejméně dvojnásobku sazby platící pro osobní silniční motorová vozidla (tj. 3,80 Kč/km). (Andraščíková a kol., 2015)

Náhrada za spotřebované pohonné hmoty se určuje jednoduchým výpočtem, kdy zaměstnavatel roznásobí cenu pohonné hmoty a množství skutečně spotřebované pohonné hmoty. Cenu pohonné hmoty musí zaměstnanec prokázat dokladem o nákupu, ze kterého patrně vyplývá, že byly pohonné hmoty nakoupeny za účelem pracovní cesty (tzn. je prokazatelná souvislost). Vlastní-li zaměstnanec více dokladů o nákupu pohonné hmoty, které jsou prokazatelně souvislé s pracovní cestou, vypočítá se cena pohonné hmoty za účelem určení náhrady aritmetickým průměrem z takto prokázaných cen. Pokud však zaměstnanec nevlastní žádné doklady o nákupu pohonné hmoty a nemůže tedy žádným hodnověrným způsobem cenu pohonné hmoty prokázat, využije zaměstnavatel za účelem určení výše náhrady průměrnou cenu konkrétní pohonné hmoty určenou v § 189 Zákoníku práce. (Andraščíková a kol., 2015)

Použije-li zaměstnanec pro přepravu na místo výkonu pracovní cesty silniční motorové vozidlo, vypočítá zaměstnavatel spotřebu pohonné hmoty konkrétního vozidla z údajů o spotřebě, které jsou uvedeny v technickém průkazu vozidla, které je zaměstnanec povinen předložit. Pokud technický průkaz vozidla takové údaje neobsahuje, náleží zaměstnanci náhrada výdajů za pohonné hmoty jen v případě, že spotřebu pohonné hmoty prokáže technickým průkazem vozidla se shodnými parametry (tj. typ a objem válců). Pro účel určení spotřeby pohonné hmoty poté zaměstnavatel použije údaj o spotřebě pro kombinovaný provoz podle norem Evropské unie. (Andraščíková a kol., 2015)

Dle § 159 Zákoníku práce náleží zaměstnanci náhrada jízdních výdajů při použití místní hromadné dopravy v souladu s danými podmínkami pracovní cesty v prokázané výši. Pokud zaměstnanec při své pracovní cestě využije hromadnou dopravu obce, ve které je sjednáno místo výkonu práce, zaměstnavatel tuto náhradu proplácí ve výši odpovídající ceně jízdného, které bylo platné v době konání pracovní cesty, aniž by zaměstnanec byl povinen jízdní výdaje prokazovat jízdními doklady. Pokud však zaměstnavatel zajistí svému zaměstnanci použití místní hromadné dopravy takovým způsobem, na který zaměstnanec nemusí finančně přispívat, odpadá zaměstnanci nárok na tuto náhradu jízdních výdajů. (Andraščíková a kol., 2015)

3.2.3 Náhrada výdajů za ubytování

Zaměstnanec vyslaný na pracovní cestu má nárok na náhradu výdajů za ubytování, které byly vynaloženy v souladu s podmínkami pracovní cesty, a to až do výše, kterou zaměstnavateli prokáže. Pokud je zaměstnanec na návštěvě u člena rodiny, má nárok na náhradu prokázaných výdajů za ubytování pouze v případě, že si je musel zaměstnanec zachovat. (Andraščíková a kol., 2015)

Zaměstnavatel není povinen hradit výdaje na ubytování v případě, že dojde k předem dohodnutému přerušení pracovní cesty z důvodů na straně zaměstnance, i když byl zaměstnanec po tuto dobu nucený výdaje za ubytování hradit. (Andraščíková a kol., 2015)

3.2.4 Náhrada výdajů za stravování (stravné)

V následujících bodech je uvedeno, v jaké výši náleží zaměstnanci stravné za každý kalendářní den jeho pracovní cesty:

1. je-li pracovní cesta v rozmezí 5 až 12 hodin – **58 Kč**,
2. je-li pracovní cesta v rozmezí 12 hodin až 18 hodin – **88 Kč**,
3. je-li pracovní cesta vyšší než 18 hodin – **138 Kč**.

Tyto sazby za stravné náleží zaměstnanci v případě, že zaměstnavatel neurčí před počátkem pracovní cesty jinou výši stravného, která by se se stávajícími sazbami neshodovala. Stravovací sazby se mění v závislosti na vývoji cen a jsou ošetřovány § 189 Zákoníku práce. (Andraščíková a kol., 2015)

Pokud zaměstnanec během své pracovní cesty obdržel jídlo (tj. snídaneň, oběd či večeře), za které nemusel vynaložit žádné náklady (tzv. „bezplatné jídlo“), snižuje se stravné o každé bezplatné jídlo až do výše:

1. **70 % stravného**, je-li pracovní cesta v rozmezí 5 až 12 hodin,
2. **35 % stravného**, je-li pracovní cesta v rozmezí 12 hodin až 18 hodin,
3. **25 % stravného**, je-li pracovní cesta delší než 18 hodin.

Tyto procentuální hodnoty snížení připadají zaměstnanci v případě, že zaměstnavatel před počátkem pracovní cesty nestanoví nižší hodnotu ponížení než jak je uvedeno. (Andraščíková a kol., 2015)

Od odděleného posuzování doby trvání pracovní cesty je upuštěno v případě, že pracovní cesta spadá svojí délkou do 2 kalendářních dnů a že je to pro zaměstnance výhodnější. (Andraščíková a kol., 2015)

Po dobu návštěvy člena rodiny nebo po dobu dohodnutého přerušení pracovní cesty z důvodů na straně zaměstnance stravné zaměstnanci nepřísluší. Doba rozhodná pro právo na stravné před návštěvou člena rodiny nebo dohodnutým přerušením pracovní cesty končí ukončením výkonu práce nebo jiným předem dohodnutým způsobem, a po návštěvě člena rodiny nebo přerušení pracovní cesty z důvodů na straně zaměstnance začíná současně se začátkem výkonu práce, nebo jiným předem dohodnutým způsobem. (Andraščíková a kol., 2015)

V případě, že je zaměstnanec na návštěvě člena rodiny či došlo k dohodnutému přerušení pracovní cesty skrze důvody ze strany zaměstnance, nepřísluší mu za tuto dobu žádné stravné. V tomto případě doba, po kterou trvá právo na stravné, končí ukončením výkonu práce (nebo jinak dohodnutým způsobem) a začíná současně se začátkem výkonu práce (nebo jinak dohodnutým způsobem). (Andraščíková a kol., 2015)

Pokud je zaměstnanec zaměstnavatelem vyslán na pracovní cestu do místa svého bydliště, které není stejné jako místo výkonu práce či pravidelné pracoviště, náleží zaměstnanci stravné pouze za cestu do místa bydliště a zpět, a za dobu výkonu práce v místě bydliště. (Andraščíková a kol., 2015)

3.2.5 Náhrada nutných vedlejších výdajů

Náhrada nutných vedlejších výdajů, které vzniknou v přímé souvislosti s pracovní cestou, náleží zaměstnanci ve výši, kterou prokáže zaměstnavateli. Pokud zaměstnanec není schopen výši svých nutných vedlejších výdajů prokázat, náleží mu náhrada, která odpovídá ceně produktů a služeb, jež byly pořízeny v době a místě konání pracovní cesty. (Andraščíková a kol., 2015)

3.3 Cestovní náhrady při zahraniční pracovní cestě

Kromě tuzemské pracovní cesty může být zaměstnanec svým zaměstnavatelem vyslán i na zahraniční pracovní cestu, na jejíž zpracování nahlíží Zákoník práce trochu odlišně. Z toho důvodu bude následující podkapitola věnována právě problematice cestovních náhrad při zahraniční pracovní cestě.

3.3.1 Druhy cestovních náhrad

S ohledem na podmínky, které jsou stanoveny v § 163, je zaměstnavatel povinen poskytnout zaměstnanci stravné a náhrady zejména za tyto položky:

- jízdní výdaje,
- jízdní výdaje k návštěvě člena rodiny,
- výdaje za ubytování,

- stravovací výdaje v cizí měně (tzv. "zahraniční stravné"),
- nutné vedlejší výdaje. (Andraščíková a kol., 2015)

Zaměstnavatel však může dle svého uvážení při zahraniční pracovní cestě proplatit zaměstnanci i další cestovní výdaje. (Andraščíková a kol., 2015)

3.3.2 Náhrada jízdních výdajů

Zaměstnanec má nárok na náhradu jízdních výdajů ve výši a při splnění podmínek, které jsou stanoveny v § 157 až 160 s tím rozdílem, že náhrada za spotřebované pohonné hmoty v příslušné cizí měně a průkazné ceně se zaměstnanci účtuje pouze za kilometry, které byly ujety mimo území České republiky. Pokud nastane situace, že zaměstnanec nebude vlastnit doklad o nákupu pohonných hmot v zahraničí (tj. mimo území ČR), může zaměstnavatel poskytnout náhradu za spotřebované pohonné hmoty v příslušné zahraniční měně pouze na základě zaměstnancova prohlášení o skutečně zaplacené ceně pohonných hmot. (Andraščíková a kol., 2015)

Náhrada jízdních výdajů k návštěvě člena rodiny

Pokud je zahraniční pracovní cesta delší než 1 měsíc, a pokud byla návštěva člena rodiny sjednána či určena zaměstnavatelem před počátkem zahraniční pracovní cesty, má zaměstnanec nárok na náhradu jízdních výdajů při návštěvě člena rodiny do místa jeho bydliště a zpět (nebo do jiného předem domluveného místa pobytu daného člena rodiny), a to nejvýše v částce, která odpovídá jízdním výdajům do místa výkonu práce či pravidelného pracoviště, anebo do místa bydliště zaměstnance na území ČR. Limitem je pak ta částka, která je z pohledu zaměstnance nejvýhodnější. (Andraščíková a kol., 2015)

3.3.3 Náhrada výdajů za ubytování

Splnil-li zaměstnanec podmínky zahraniční cesty uvedené v § 162 Zákoníku práce, náleží mu náhrada výdajů za ubytování, na než vyložil své finanční prostředky. (Andraščíková a kol., 2015)

3.3.4 Náhrada výdajů za stravování (zahraniční stravné)

Zaměstnanec má při zahraniční pracovní cestě nárok na stravné v cizí měně a za podmínek stanovených § 170. Pokud zaměstnavatel stanoví nebo dohodne před vysláním zaměstnance na zahraniční pracovní cestu základní sazbu týkající se stravování v zahraničí, musí takto stanovená sazba činit v měnových jednotkách nejméně 75 % (a u členů posádek vnitrozemských plavidel nejméně 50 %) základní sazby zahraničního stravného, které je stanoveno pro příslušný stát v § 189 Zákoníku práce. (Andraščíková a kol., 2015)

Zaměstnavatel může zaměstnanci určit výši základní sazby také podle prováděcího právního předpisu, který byl vydán podle § 189 Zákoníku práce. Výše zahraničního stravného se určuje ze základní sazby zahraničního stravného, které

bylo stanoveno či určeno pro stát, kde zaměstnanec stráví nejvíce času (v rámci své zahraniční pracovní cesty). (Andraščíková a kol., 2015)

Jestliže doba strávená mimo území ČR trvá déle než 18 hodin (v rámci jednoho kalendářního dne), zaměstnanec má nárok na zahraniční stravné ve výši základní sazby podle § 189 Zákoníku práce. Spadá-li tato doba svojí délkou do intervalu 12 až 18 hodin, má zaměstnanec nárok na stravné ve výši dvou třetin sazby zahraničního stravného, a je-li doba strávená mimo území ČR kratší než 12 hodin, ale alespoň 1 hodinu dlouhá, má zaměstnanec nárok na stravné podle § 163 nebo § 176. Pokud je doba strávená mimo území České republiky kratší než 1 hodinu, nárok na stravné zaměstnanci odpadá. (Andraščíková a kol., 2015)

Doby, které trvají nejméně 1 hodinu při více zahraničních pracovních cestách v jednom kalendářním dni, se pro účely stanovení výše zahraničního stravného sčítají. Doby, za které nevzniká nárok na poskytnutí zahraničního stravného, se přičítají k době rozhodné pro poskytnutí stravného (§ 163). (Andraščíková a kol., 2015)

Pokud zaměstnanec během své zahraniční pracovní cesty obdržel nějaké bezplatné jídlo (snídaně, oběd či večeře), má nárok na zahraniční stravné, které bude vycházet ze základní sazby ponížené za každé bezplatné jídlo až o hodnotu:

1. **70 % zahraničního stravného**, jedná-li se o zahraniční stravné ve výši jedné třetiny základní sazby,
2. **35 % zahraničního stravného**, jedná-li se o zahraniční stravné ve výši dvou třetin základní sazby,
3. **25 % zahraničního stravného**, jedná-li se o zahraniční stravné ve výši celé základní sazby.

Pokud zaměstnanec nesjedná či neurčí nižší hodnotu snížení zahraničního stravného před počátkem zahraniční pracovní cesty, má zaměstnanec nárok na zahraniční stravné ponížené o nejvyšší hodnotu stanovenou v bodě č. 1. (Andraščíková a kol., 2015)

Při návštěvě člena rodiny či při dohodnutém přerušení zahraniční pracovní cesty ze strany zaměstnance odpadá zaměstnanci nárok na zahraniční stravné. Doba rozhodující o nároku na zahraniční stravné končí ukončením výkonu práce (či jiným předem dohodnutým způsobem) a začíná současně se začátkem výkonu práce (či jiným předem dohodnutým způsobem). (Andraščíková a kol., 2015)

Pokud je zaměstnanec vyslán na zahraniční pracovní cestu do místa svého bydliště, má nárok na stravné a zahraniční stravné pouze za dobu, kterou stráví na cestě do bydliště a zpět, dále za cestu k výkonu práce a zpět a za dobu výkonu práce v tomto místě. (Andraščíková a kol., 2015)

3.3.5 Náhrada nutných vedlejších výdajů

Nutné vedlejší výdaje jsou v případě zahraniční pracovní cesty zohledněny stejným paragrafem, jako v případě tuzemské pracovní cesty, a to § 164 Zákoníku práce. Z paragrafu vyplývá jak výše náhrady, tak případné podmínky a omezení.

3.4 Informační systémy

Aplikace, jejíž návrh a implementace jsou předmětem této diplomové práce, se dá označit za informační systém, proto bude tato podkapitola věnována právě problematice informačních systémů, jejich základní terminologii, typům architektur informačních systémů a návrhu informačního systému včetně popisu existujících implementačních technologií.

3.4.1 Základní pojmy

Pro lepší pochopení základních pojmů je potřeba začít od počátku, čili od definice samotných informací. Za **informace** lze podle Rábové (2008) označit data, kterým je z hlediska uživatele přiřazen nějaký konkrétní význam, a která jsou schopna uspokojit informační potřebu svého příjemce. Informace tak vzniká z dat a vzniká až v okamžiku, kdy jsou daná data využita. (Rábová, 2008)

Pojem **systém** lze chápat různými způsoby, nejčastěji je však zmiňována definice, kdy je systém komplexem prvků, které se nacházejí ve vzájemné interakci. (Rábová, 2008) Další definice označuje systém jako účelově vymezenou množinu prvků (či objektů) a množinu vazeb mezi těmito prvky, které jako celek vykazují určité chování. (Rábová, 2008)

Samotný **informační systém (IS)** lze pak tedy označit za „komplex informací, lidí, použitých informačních technologií, organizace práce, řízení chodu systému a konečně technických prostředků, sloužících ke sběru, přenosu, uchování a dalšímu zpracování dat za účelem tvorby a prezentace informací“. (Rábová, 2008)

V souvislosti s informačními systémy je vhodné objasnit taktéž pojem **informační technologie (IT)**. Ty jsou podle Rábové (2008) chápány jako technické, programové či metodické prostředky, které se využívají k pořízení, uchování, zpracování, prezentaci a přenosu dat.

V praxi se uživatel často setkává také s pojmem **IS/ICT** (nebo IS/IT), což se dá zjednodušeně pochopit tak, že zatímco část označující informační systém (IS) potřebu informací vyvolává, část nesoucí označení informační a komunikační technologie (ICT) tuto potřebu uspokojuje. (Rábová, 2008)

3.4.2 Architektura informačního systému

S informačními systémy úzce souvisí také pojem **architektura informačního systému**, což je celková koncepce informačního systému zahrnující budoucí podobu a vlastnosti systému, jeho jednotlivé součásti a vazby mezi těmito součástmi, včetně vazeb na okolí podniku a stávající strukturu organizace. (Rybička, Čačková, 2009)

Architektura IS je pro vývoj informačního systému velice důležitá, a to zejména z těchto hledisek:

- **Stabilní rámec řešení IS/ICT.** Architektura tvoří relativně stabilní rámec, do kterého jsou v rámci vývoje IS postupně začleňovány jednotlivé

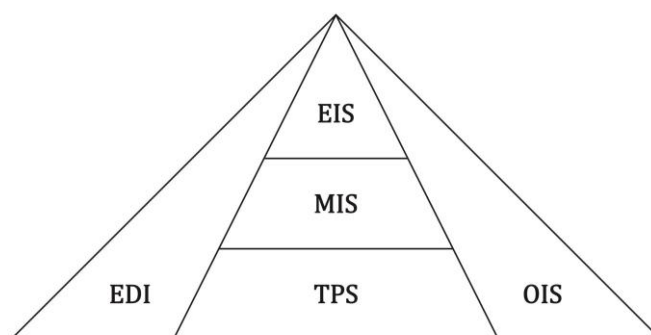
komponenty podle předem připraveného plánu a s již předem definovanými vazbami na ostatní součásti IS/ICT. (Bruckner a kol., 2012)

- **Komunikační prostředek.** Architektura je významný komunikační prostředek pro všechny subjekty zainteresované v řešení IS/ICT (jsou jimi např. vedení podniku, byznys analytici a vývojáři). Architektura umožňuje investorům, uživatelům a řešitelům lépe porozumět tomu, které aplikace, data a rozhraní budou implementovány v daném čase. (Bruckner a kol., 2012)
- **Stabilita vývoje IS/ICT.** Pokud je architektura navržena takovým způsobem, aby dokázala předvídat předpokládané změny, zajišťuje stabilitu vývoje IS/ICT i při rychlém vývoji ICT. Pokud je některá komponenta systému nevyhovující, musí být umožněno ji jednoduše nahradit za jinou, vyhovující. (Bruckner a kol., 2012)
- **Zohlednění hlavních požadavků na vlastnosti IS/ICT.** Architektura zajišťuje již v počátcích řešení IS/ICT zohlednění hlavních požadavků na vlastnosti IS/ICT, ze kterých pak lze odvíjet konkrétní specifikace ICT projektů. (Bruckner a kol., 2012)
- **Ekonomické hledisko.** Architektura IS umožňuje redukovat náklady na chybně zadané projekty případně náklady na rekonstrukci celého IS/ICT. (Bruckner a kol., 2012)

3.4.3 Globální architektura IS

Základním schématem znázorňujícím hrubou podobu celého informačního systému je tzv. **globální architektura**. Aby byla tato hrubá podoba co nejjednodušší a srozumitelná, je abstrahována od veškerých detailů. Samotná architektura má hierarchickou strukturu a dělí se do tří vertikálních částí, které odpovídají členění managementu v podniku (tj. nižší, střední a vrcholový management). Kromě vertikálních částí má architektura také horizontální části, které odpovídají jednotlivým podnikovým útvarům a funkcím (účetnictví, výroba, marketing, sklad, atd.). (Rybička, Čačková, 2009)

Globální architektura je standartně vyobrazována jako pyramida, která je rozdělena podle tří úrovní řízení v podniku (viz obrázek níže).



Obr. 1 Dílčí architektura informačního systému

Zdroj: Rybička, Čačková (2009)

Bloky globální architektury

- **TPS (Transaction Processing System).** Jedná se o operativní část informačního systému, kdy jsou operace závislé na charakteru podniku. Tento blok je zároveň také nejvíce specifický. Jeho úkolem je zajišťovat pořízení a aktualizaci dat, udržování evidence, poskytování základních přehledů o činnosti podniku. V tomto bloku se tvoří data pro zbývající vrstvy podnikového informačního systému. (Rábová, 2008)

Typy TPS:

- *CIM* (Computer Integrated Manufacturing, obsahují výrobní a zakázkovou linii),
 - *RIS* (Reservation IS),
 - *GIS* (Geographic IS),
 - *MRP* (Material Resource Planning),
 - *ERP* (Enterprise Resource Planning). (Rábová, 2008)
- **MIS (Management Information System).** Tento blok je určen pro řízení podniku na taktické úrovni. Struktura tohoto systému je více standardizovaná, a proto je podobný i pro podniky různých typů. (Rábová, 2008)
- Procesy se dělí do tří linií:
- *obchodně-logistická* (nákup, prodej, sklady, atd.),
 - *finančně-účetní* (majetek, mzdy, pokladna, účetnictví, atd.),
 - *průřezová* (řízení lidských zdrojů, marketing, legislativa, atd.).
- **EIS (Executive Information System).** Blok informačního systému, který je určen pro řízení podniku na strategické úrovni. Jeho hlavní náplní je získávání dat z ostatních úrovní IS a také z externích zdrojů. Tato data poté kombinuje a vytváří časové řady, prognózy, vazby či trendy. Zaměřuje se spíše na delší časová období a to jak do minulosti, tak do budoucnosti. Základem tohoto bloku je uložení dat v n-dimenzionální databázi, dimenze pak mohou být následující:
 - *ekonomické* (zisk, obrat),

- *časové* (měsíc, čtrtletí, rok),
- *zvolený pohled* (typy zákazníků, závody, teritoria, atd.). (Rábová, 2008)
- **OIS (Office Information System).** Blok informačního systému, který se orientuje zejména na podporu kancelářských a týmových prací. Hlavní funkcí systému je tvorba, modifikace a přenášení dat administrativního charakteru v psané a grafické podobě. Tento blok IS má za následek zvýšení produktivity a výkonnosti pracovníků administrativy. (Rábová, 2008) Nejznámějším zástupcem tohoto typu IS je pravděpodobně kancelářský balík Microsoft Office či LibreOffice.
- **EDI (Electronic Data Interchange).** Tento blok IS slouží k výměně dat za pomoci elektronických prostředků. Používá se jako nástroj pro provádění elektronického obchodu. Blok se skládá ze tří základních modulů:
 - *propojení aplikací* (import/export do účetnictví, aktivní brána do informačního systému),
 - *konvertor* (zde probíhá převod dat vlastního formátu do dat partnera),
 - *komunikační modul* (internet, modem, atd.). (Rábová, 2008)

3.4.4 Dílčí architektury IS

Zatímco globální architektura IS zaznamenávala hrubou podobu informačního systému odproštěnou od detailních informací, u **dílčích architektur IS** je tomu opačně. Z toho důvodu by se dílčí architektura IS dala ve zkratce označit za detailní pohled na budoucí podobu IS z hlediska software, hardware, technologického zpracování, okolí systému, apod. Dílčí architektury jsou tedy již konkretizovanými návrhy IS/ICT.

Typy dílčích architektur IS

- **Procesní dílčí architektura.** Ta se zaměřuje na externí podněty (události), které představují vazby podniku na okolí. Nástrojem procesní dílčí architektury je pak kontextový diagram (Rábová, 2008).

Účelem kontextového diagramu je reprezentovat hranice systému, vymezovat okolí systému, zdroje a místa určení dat nebo-li terminátory. (Bruckner a kol., 2012)
- **Funkční dílčí architektura.** Svým zpracováním navazuje na předchozí procesní architekturu. Funkční architektura zobrazuje návrh hierarchického rozpadu funkcí a služeb, které jsou od budovaného IS požadovány. Nástrojem funkční architektury je diagram datových toků a slovní popis funkcí nebo-li minispecifikace. (Rábová, 2008)

Dle Brucknera a kolektivu (2012) lze diagram datových toků (Data Flow Diagram, DFD) chápat jako bezprostřední rozklad kontextového diagramu na nižší úroveň. DFD diagram obsahuje základní funkce systému a vztahy mezi nimi vyjádřené prostřednictvím datových toků a zásobníků. (Bruckner a kol., 2012)

- **Datová dílčí architektura.** Slouží k návrhu datové základny informačního systému. Nástrojem datové architektury je pak diagram datových entit (Entity Relationship Diagram, ERD) ve své logické podobě. (Rábová, 2008)
- **Softwarová dílčí architektura.** Určuje, které softwarové komponenty budou v IS použity a jaké budou mezi těmito komponentami vazby. Softwarová architektura je definována moduly a vazby mezi jednotlivými moduly. Tento typ dílčí architektury se dále dělí na:
 - *lineární softwarová architektura* – celková funkčnost je zajištěna sekvenčním uspořádáním základních funkcí, tato architektura je levná, avšak vhodná pro jednoduché funkce,
 - *hierarchická architektura* – je uspořádána do podoby stromového grafu, kdy každá funkce využívá ty, které jí jsou podřízené a ze kterých se tedy i skládá (mohou se zde vyskytovat funkční duplicity),
 - *vrstvená architektura* – prezentuje ji acyklický graf, kdy funkce vyšší vrstvy mohou využívat funkce nižších vrstev, základní funkce mohou být využívány všemi vyššími vrstvami,
 - *síťová architektura* – jedná se o prezentaci obecného orientovaného grafu, čili zde odpadají pravidla podřízenosti/nadřízenosti funkcí, jedná se o vcelku flexibilní a levnou architekturu. (Rábová, 2008)
- **Hardwarová dílčí architektura.** Tato architektura znázorňuje typy, množství a vazby mezi hardwarovými komponentami (převážně počítače, servery, tiskárny či jiná přídatná zařízení). Většinou se jedná o prostý textový popis jednotlivých komponent a jejich parametry, případně umístění v podniku. (Rábová, 2008)
- **Technologická dílčí architektura.** Tento typ architektury rozhoduje o technologickém řešení (resp. jednotlivých funkcí) aplikace. Jedná se o propojení hardwarové, softwarové a datové architektury, kdy je definován způsob zpracování, vnitřní stavba a uživatelské rozhraní jednotlivých aplikací. (Rábová, 2008)

Technologická architektura se dále dělí podle:

- *metody zpracování* (dávkové, interaktivní, řízené událostmi, v reálném čase),
- *uspořádání PC* (centralizované, decentralizované či distribuované),
- *vrstev*. (Rábová, 2008)

3.4.5 Vrstvená architektura IS

Pod pojmem **vrstvená architektura IS** se skrývá vnitřní architektura systému a struktura implementačních vrstev, které umožňují přenášet systém z jednoho prostředí do jiného (např. změna operačního systému či databáze). (Rybička, Čáčková, 2009)

Nejstarším a zároveň také nejjednodušším typem vrstvené architektury je architektura *s jednou vrstvou*, kdy veškerá funkcionálna byla obstarávána

jediným subsystémem (čili celým systémem). Dalším a také později vzniklým typem byla *dvouvrstvá architektura* na bázi klient-server (na straně serveru se vyskytovala databáze, na straně klienta pak samotná aplikace a prezentace dat uživateli). (Rybička, Čáčková, 2009)

Se stále se zvyšujícími nároky na tvorbu integrovaných systémů a různých zařízení pak došlo ke vzniku *třívrstvé architektury*, která se skládá z těchto vrstev:

1. **Prezentační** – Tato vrstva je nejvýše umístěná, v praxi se označuje jako *frontend* a jejím účelem je zajišťovat prezentaci výsledků včetně komunikace s uživatelem přes uživatelské rozhraní. (Rybička, Čáčková, 2009)
2. **Aplikační** – Jedná se o prostřední vrstvu třívrstvé architektury, která obstarává veškeré transformace vstupně-výstupních dat. Provádí se zde nejrůznější operace a výpočty. Vrstva zajišťuje funkčnost aplikací, bezpečnost samotného systému, integritu dat a mnoho dalšího. Bývá označována jako *middleware* (to kvůli jejímu umístění). Pokud je tato vrstva realizována ve webovém prostředí, bývá řešena skripty na straně serveru. (Rybička, Čáčková, 2009)
3. **Datová** – Datová vrstva se vyskytuje na nejnižší úrovni a je reprezentována systémem, který poskytuje společnou datovou základnu různým podsystémům IS. Tato vrstva obstarává ukládání/získávání/editaci dat, kontroluje jedinečnost vkládaných dat, provádí veškeré kontroly před uložením a data dále agreguje. Vrstva bývá označována jako *backend*. (Rybička, Čáčková, 2009)

Podle množství zajišťovaných funkcí se pak rozlišují klientské programy na *tlusté* a *tenké*. Více o jednotlivých typech bude napsáno v dalších částech práce.

Výhody vrstvené architektury

- Výměna či úprava kterékoliv vrstvy bez nežádoucích vlivů na funkčnost aplikace. (Čermák, 2010)
- Možnost provádění změn za chodu (čili bez nutnosti výpadku). (Čermák, 2010)
- Rychlejší vývoj vícevrstvé aplikace, který je zapříčiněn dělbou práce mezi více vývojářů. (Čermák, 2010)

Nevýhody vrstvené architektury

- V případě využití hardware/software od několika různých výrobců je nutno provádět monitoring/údržbu všech těchto komponent včetně jejich upgrade, což může být časově i finančně velice nákladné. (Čermák, 2010)

3.5 Návrh informačních systémů

Podle Rybičky a Čáčkové (2009) prochází informační systém od okamžiku nasazení v podniku až do ukončení používání různými fázemi. Tyto fáze se souhrnně označují za **životní cyklus informačního systému**. Konkrétní fáze jsou pak: úvodní studie, analýza, návrh, implementace, testování a zavádění do provozu.

V této podkapitole však nebude rozebrán podrobně celý životní cyklus, nýbrž jedna konkrétní fáze – návrh – a přístupy, které se při navrhování IS používají.

3.5.1 Strukturovaný přístup

Strukturovaný přístup se vyvíjel v průběhu 70. let a na počátku 80. let. Jeho základní charakteristikou je, že rozděluje problematiku návrhu IS do menších, přesně definovaných činností (jedná se o metodu shora dolů) a přesně definuje návaznosti mezi nimi. Tento fakt je tudíž i hlavní výhodou, jelikož je možno projekt rozdělit na menší části, které se snáze kontrolují a řídí. (Vymětal, 2009)

Druhým hlavním aspektem strukturovaného přístupu dle Vymětala (2009) je, že se na modelovanou realitu lze dívat ze dvou hledisek. Prvním z nich je pohled na procesy a s nimi související datové toky v systému, druhým hlediskem je pak zobrazení datových struktur. (Vymětal, 2009)

Dalším aspektem strukturovaného přístupu je, že při modelování reality jsou využívány různé grafické nástroje. Nejčastěji se používají diagram datových toků (DFD), diagram entit a vztahů (ERD) a diagram přechodů (State Transition Diagram, STD). Pro popis využívaných či měněných informací je třeba používat také datový slovník (Data Dictionary, DD) a specifikaci procesů. (Vymětal, 2009)

Samotná dokumentace, která je v strukturovaném přístupu využívána, se může prezentovat těmito způsoby:

- *graficky* – využívají se sady různých diagramů včetně slovního doprovodu,
- *v samotných oddílech*, které mohou být využívány nezávisle na sobě,
- *pomocí dokumentu s minimem redundantních dat* – výhodou je zavádění změn, které se ve specifikaci projeví pouze jednou. (Vymětal, 2009)

Jednou z výhod strukturovaného přístupu je dostatečná přehlednost a názornost dokumentace, což způsobuje, že lze využívat i méně kvalifikované pracovníky. Vývoj strukturovaných metod a zřetelná definice použitých nástrojů daly možnost vzniku nástrojů softwarového inženýrství a jejich automatizace. Tyto nástroje jsou označovány jako tzv. CASE, nebo-li *Computer Aided Software Engineering*. Mezi nejčastěji používané nástroje tohoto typu se řadí např. Oracle Designer. (Vymětal, 2009)

V průběhu doby však strukturovaný přístup a jeho metody narážely na bariéry, které spočívaly hlavně v přesné definici uvedených posloupností celého projektu, což se jevílo spíše jako nevýhoda než výhoda. Strukturovaný přístup tak vedl k nikdy nekončícím změnám a špatné udržitelnosti dokumentace. (Vymětal, 2009)

3.5.2 Objektově orientovaný přístup

Od poloviny 80. let 20. století se pak vyvíjel druhý přístup – **objektově orientovaný**. Tento přístup je založen, jak sám název napovídá, na objektech. Objekt je pak dle Vymětala (2009) definován jako „*určitý prvek v systému, který vykazuje vlastní chování (reakci na vstupní impulsy – metody) a má určité vlastnosti*

(*atributy*)“. Objekty se stejným či podobným chováním a vlastnostmi pak vytvářejí *třídy objektů*. (Vymětal, 2009)

Charakteristickým aspektem objektově orientovaného přístupu je pak to, že objekty nižší úrovně dědí vlastnosti objektů vyšší úrovně. Tento přístup také umožňuje, aby různými třídami byly propojovány operace stejného jména (každá třída má však přesně definováno, jakým způsobem má být operace provedena). Tento koncept se nazývá *polymorfismus* a je dalším charakteristickým jevem či dokonce i hlavní výhodou objektově orientovaného přístupu. (Vymětal, 2009)

Dalším rysem je pojem *zapouzdření*, což lze dle Vymětala chápat tak, že každá třída může provádět své operace skrytě (pokud tedy není definováno jinak). Pro znázornění provázanosti mezi jednotlivými třídami se pak používá různých druhů *asociací* (vazeb). Objektově orientovaný přístup tak na rozdíl od strukturovaného vychází z identifikace objektů a tříd. (Vymětal, 2009)

Vývoj objektově orientovaného přístupu přispěl k vývoji různých forem zápisu, např. také standardu UML, kterému bude věnována samostatná část práce. Vývoj také přispěl k přípravě softwarových CASE nástrojů. Ale jelikož to jsou nástroje primárně pro strukturovaný přístup, jedná se zde zejména o podporu datových modelů, které mohou být tvořeny buď z diagramů tříd či exportem modelu z již existující databáze. Mezi pravděpodobně nejznámější a nejpoužívanější nástroje objektově orientovaného přístupu patří produkt Rational Rose od firmy Rational (aktuální vlastník firmy je IBM). (Vymětal, 2009)

3.6 UML notace

Tato podkapitola svým charakterem spadá do předchozí podkapitoly, neb se týká objektově-orientovaného přístupu u návrhu informačních systémů. Vzhledem k její obsáhlosti však bude přesunuta na samostatnou úroveň.

3.6.1 Charakteristika UML

Podle Arlowa a Neustadta (2007) je UML¹ „*univerzální jazyk pro vizuální modelování systémů*“. Jedná se o druh notace, která svými grafickými prvky umožňuje popisovat a navrhovat jakékoliv objektově orientované softwarové systémy. (Fowler, 2010)

Využití unifikovaného modelovacího jazyka je však mnohem širší, což také vyplývá z rozšiřovacích mechanismů, které jsou v jazyce zabudovány. (Arlow a Neustadt, 2007)

Grafické modelovací jazyky jako takové se v softwarovém průmyslu vyskytují již řadu let. Hlavním důvodem je to, že programovací jazyky není možno abstrahovat na tak vysokou úroveň, aby to pomáhalo při diskuzích o softwarovém návrhu. (Fowler, 2010)

Je důležité si uvědomit, že jazyk UML neposkytuje žádné přesné metodiky modelování. Nabízí pouze grafickou syntaxi, kterou je možné využít při

¹ UML = Unified Modeling Language nebo-li unifikovaný modelovací jazyk

sestavování modelů. Jazyk UML také není vázaný žádnou konkrétní metodikou nebo životním cyklem, proto ho lze využít ve spojení se všemi ostatními existujícími metodami. UML je však hojně využíváno v metodice *Unified Process*, která může být z tohoto pohledu označována za upřednostňovanou metodu jazyka UML, neboť je pro tento jazyk nejlépe přizpůsobena. (Arlow a Neustadt, 2007)

UML je standard řízený volným uskupením společností nesoucím název Object Management Group (zkráceně OMG). Toto konsorcium vzniklo za účelem tvorby standardů, které by sloužily k podpoře slučitelnosti objektově orientovaných systémů. Jejich nejznámějším produktem je pravděpodobně CORBA (Common Object Request Broker Architecture). (Fowler, 2010)

3.6.2 Struktura a základní stavební bloky jazyka UML

Jazyk UML se skládá ze tří základních částí. Jsou jimi:

1. **stavební bloky** – do této položky se řadí základní stavební bloky – předměty (prvky), relace (vztahy) a diagramy,
2. **společné mechanismy** – sem patří obecné postupy, kterými lze v jazyku UML docílit konkrétních cílů,
3. **architektura** – aneb jak se v jazyku UML dívá na architekturu navrhovaného systému. (Arlow a Neustadt, 2007)

Jak již bylo zmíněno výše, jazyk UML se skládá také ze tří základních stavebních kamenů alias bloků. Tyto bloky jsou:

1. **předměty (things)** nebo také jednotlivé prvky modelu,
2. **vztahy (relationships)** nebo také relace – představují pojítka mezi předměty, vztahy nám ukazují, jakým způsobem spolu předměty souvisejí,
3. **diagramy (diagrams)** – znázorňují pohledy na modely UML jazyka, které ukazují CO bude navrhovaný systém dělat (analytické diagramy, analysis-level diagram) a JAK toho docílí (návrhové diagramy nebo-li design-level diagrams). (Arlow a Neustadt, 2007)

Jednotlivé stavební bloky budou blíže přiblíženy v následujících podkapitolách.

3.6.3 Předměty (things)

Předměty jazyka UML se dělí následujícím způsobem:

- **strukturní předměty (structural things)** jsou podstatná jména vytvářeného modelu UML, patří sem především: třídy, rozhraní, spolupráce, případy užití, aktivní třídy, komponenty a uzly,
- **předměty chování (behaviour things)** jsou slovesa vytvářeného modelu UML (například stav, interakce, atd.),
- **předměty seskupení (grouping things)** nebo-li balíčky, které se používají k seskupování prvků, jež spolu významově souvisejí, do jednotek,

- **předměty poznámek (annotational things)** se používají proto, aby bylo možné k modelu připojit jakoukoliv informaci, která byla sestavena jen pro tento jeden účel (tento typ předmětu se dá přirovnat ke zvýraznění textu zvýrazňovačem). (Arlow a Neustadt, 2007)

3.6.4 Relace (relationships)

Relace nebo-li vazby zachycují v modelu jazyka UML důležité vztahy mezi dvěma nebo více předměty. Vztahují se převážně k strukturním předmětům a předmětům seskupení. Existují následující typy relací:

Závislost (Dependency)

Závislost se používá v případě, kdy změna jednoho předmětu ovlivní význam předmětu na něm závislém. V jazyku UML je závislost znázorňována přerušovanou čarou se šipkou značící směr od ovlivňujícího předmětu k závislému. (Arlow a Neustadt, 2007)

Asociace (Association)

Asociace popisuje množinu spojení mezi objekty. V jazyku UML se značí (ne)orientovanou plnou čarou. (Arlow a Neustadt, 2007)

Agregace (Aggregation)

Agregace se používá, když je cílový prvek součástí prvku zdrojového. V případě, že dojde k zániku prvku zdrojového, cílový prvek ale nepřestává existovat. V jazyku UML je označena prázdným kosočtvercem na konci plné čáry. (Arlow a Neustadt, 2007)

Kompozice (Composition)

Kompozice je silnější druh agregační vazby. Na rozdíl od agregační vazby ale spolu se zánikem zdrojového prvku zaniká i prvek cílový. (Arlow a Neustadt, 2007) Vhodným příkladem pro názorné vysvětlení kompoziční vazby může být vztah mezi automobilem a motorem – v případě, že zanikne automobil, zaniká spolu s ním i motor. V jazyku UML se kompozice značí plným kosočtvercem na konci plné čáry.

Ochranná nádoba (Containment)

Vazba ochranná nádoba se používá v případě, že je cílový prvek obsažen v prvku zdrojovém. V jazyku UML je tato vazba znázorněna křížkem v kolečku na konci plné čáry. (Arlow a Neustadt, 2007)

Zobecnění (Generalization)

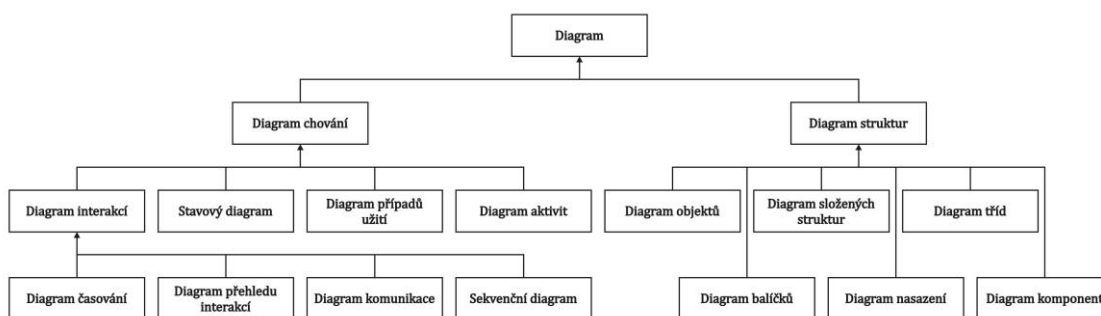
Vazba zobecnění se použije v případě, že jeden prvek je konkretizací jiného prvku, který se tak dá nahradit obecnějším prvkem. (Arlow a Neustadt, 2007) Příkladem může být zobecnění zaměstnanec, kdy jeho konkretizací jsou pak jednotlivá zaměstnání – pokladník, účetník, informatik, atd. V jazyku UML je zobecnění znázorněno prázdným trojúhelníkem na konci plné čáry.

Realizace (Realization)

Jedná se o asociační vazbu mezi dvěma klasifikátory, kdy jeden z nich určuje akci, jejíž vykonání zaručuje druhý z klasifikátorů. V jazyku UML se značí trojúhelníkem na konci přerušované čáry. (Arlow a Neustadt, 2007)

3.6.5 Diagramy (diagrams)

Již bylo zmíněno, že diagramy jsou pouze pohledy na model, ne samotný model. Celkem existuje 13 druhů diagramů, které lze vidět na obrázku níže. Jednotlivé diagramy pak budou podrobněji popsány pod obrázkem.



Obr. 2 Diagramy UML

Zdroj: Fowler, 2010

Diagram tříd (class diagram)

Úkolem diagramu tříd je popis typů objektů v systému a vztahů mezi nimi. Diagram tříd zahrnuje vlastnosti (atributy), operace třídy a také omezení, která se týkají způsobu, jakým jsou objekty spojovány. (Fowler, 2010)

Kdy je vhodné jej použít: Diagramy tříd jsou jakýmsi základním prvkem celého UML, tudíž je velká pravděpodobnost častého používání tohoto typu diagramu. Je to hlavně z toho důvodu, že diagramy tříd mají bohaté výrazové prostředky. (Fowler, 2010)

Sekvenční diagram (sequence diagram)

Sekvenční diagram nejčastěji zachycuje chování nějakého konkrétního scénáře. Ukazuje vzorové objekty a zprávy, které jsou v rámci konkrétního případu užití předávány. (Fowler, 2010)

Kdy je vhodné jej použít: Sekvenční diagram by se měl používat právě ve chvíli, kdy je potřeba podívat se na chování několika objektů v rámci jednoho případu užití. (Fowler, 2010)

Diagram objektů (object diagram)

Diagram objektů snímá objekty systému v určitém časovém okamžiku. Místo tříd zobrazuje instance, z toho důvodu může být nazýván také diagramem instancí. (Fowler, 2010)

Kdy je vhodné jej použít: Tento typ diagramu je užitečný pro zobrazení příkladů objektů a jejich vzájemného propojení. Mohou tedy snáze objasnit situace, ve kterých je použití diagramu tříd příliš těžkopádné. (Fowler, 2010)

Diagram balíčků (package diagram)

Balíček je jakási sdružující konstrukce, která umožňuje vzít jakýkoliv balíček UML a seskupit jeho prvky do společných jednotek na vyšší úrovni. Diagram balíčků pak tedy zobrazuje jednotlivé balíčky a vztahy mezi nimi. (Fowler, 2010)

Kdy je vhodné jej použít: Nejčastěji se tento typ diagramu používá u rozsáhlých systémů, neboť umožňují vytvořit lepší představu o závislostech mezi hlavními prvky systému. (Fowler, 2010)

Diagram nasazení (deployment diagram)

Diagram nasazení zobrazuje fyzické rozvržení systému a odhaluje, který software běží na kterém hardwaru. Výhoda diagramu je v jeho jednoduchosti. (Fowler, 2010)

Kdy je vhodné jej použít: Právě v případě, kdy je potřeba přehledně znázornit, kde (hardware) bude co (software) nasazeno. (Fowler, 2010)

Diagram případů užití (use case diagram)

Případ užití je metoda, která slouží k zachycení funkčních požadavků na systém. Popisuje interakce mezi uživatelem systému a samotným systémem. Diagram případů užití je pak jakousi grafickou tabulkou obsahu samotných případů užití. Zobrazuje hranice systému a interakce s okolním světem. (Fowler, 2010)

Kdy je vhodné jej použít: V případě, že je potřeba lépe porozumět funkčním požadavkům na systém. (Fowler, 2010)

Stavový diagram (state diagram)

Stavový diagram se používá pro popis chování systému. V rámci objektově orientovaného přístupu je stavový diagram vytvářen pro jednu třídu a zobrazuje chování jednoho objektu v rámci jeho životního cyklu. (Fowler, 2010)

Kdy je vhodné jej použít: Hodí se při popisu objektu napříč několika případy užití. Naopak je není vhodné využívat při popisu chování, které zahrnuje řadu spolupracujících objektů. (Fowler, 2010)

Diagram aktivit (activity diagram)

Diagram aktivit se používá k popisu procedurální logiky, business procesů a work flow (toku práce). (Fowler, 2010)

Kdy je vhodné jej použít: Diagram aktivit je dobré použít v případě, že je potřeba namodelovat procesy či oběh dokumentů (work flow). (Fowler, 2010)

Diagram komunikace (communication diagram)

Diagram komunikace zdůrazňuje datové vazby mezi jednotlivými účastníky interakce. Diagram dovoluje volné rozmístění účastníků včetně zakreslení vazeb zobrazujících propojení účastníků. Pro zobrazení pořadí zpráv používá číslování. (Fowler, 2010)

Kdy je vhodné jej použít: Použití diagramu komunikace závisí na preferencích návrhářů, ve většině případů se totiž místo tohoto typu diagramu používá sekvenční diagram. (Fowler, 2010)

Diagram komponent (component diagram)

Diagram komponent slouží ke znázornění komponent (logických či fyzických) použitých v systému. Tento typ diagramu je nápomocný zejména ve chvíli, kdy je použitý vývoj založený na komponentách. (Pavus, 2006)

Kdy je vhodné jej použít: Diagram komponent je potřeba použít, pokud se navrhovaný systém rozděluje do komponent a je-li potřeba zobrazit vzájemné vztahy prostřednictvím rozhraní. (Fowler, 2010)

Diagram spolupráce (colaboration diagram)

Diagram spolupráce je téměř stejný jako sekvenční diagram. Jedná se o téměř izomorfní diagramy, tudíž je lze převádět z jednoho tvaru do druhého (často se tak děje i automatizovaně). (Pavus, 2005)

Kdy je vhodné jej použít: Použití je vhodné v případě, kdy je potřeba zdůraznit strukturální aspekty spolupráce (Pavus, 2005)

Diagram interakcí

Diagram interakcí kombinuje diagram aktivit se sekvenčním diagramem. Zároveň popisuje, jakým způsobem spolu skupiny objektů spolupracují v rámci nějakého chování. (Fowler, 2010)

Kdy je vhodné je použít: U tohoto diagramu nelze přímo říci, kdy jej použít, neboť kombinuje dva odlišné styly diagramů. Tudíž je vhodnější, aby uživatel místo této kombinace použil buď diagram aktivit nebo sekvenční diagram. (Fowler, 2010)

Diagramy časování

Diagram časování je určen k zobrazení interakcí, kde je hlavním záměrem zobrazení podrobností o čase. Tento typ diagramu se dá označit za alternativní formu sekvenčního diagramu zobrazujícího změny stavu životní linie (lifeline) v čase. (Pavus, 2005)

Kdy je vhodné je použít: Diagramy časování jsou vhodné v případě, že je potřeba zobrazit časová omezení mezi změnami stavu u různých objektů. Tyto diagramy jsou používány převážně inženýry, kteří se zabývají hardwarem. (Fowler, 2010)

3.7 Implementační technologie informačních systémů

Aplikace navrhovaná a vyvíjená v rámci diplomové práce je založena na vícevrstvé architektuře. Z tohoto důvodu bude následující kapitola věnována právě softwarovým klientům a implementačním technologiím, které lze k jejich vývoji využít.

3.7.1 Softwaroví klienti

Softwaroví klienti se váží k vícevrstvé architektuře a lze je rozdělit na dva typy: *tlusté* a *tenké*. Některé zdroje uvádějí i tzv. *chytrého* klienta. V této podkapitole budou rozebrány všechny typy včetně sumarizace výhod a nevýhod.

Tlustý/hrubý klient (thick client)

Tlustý klient se ve většině případech váže jak k vrstvě prezentační, tak k vrstvě aplikační. Připojuje se přímo k databázovému (či jinému) serveru. Tlustý klient

často přes síť stahuje velký objem dat, tato data zpracuje a výsledek pak přenese zpět na server. Instalace a celkové nastavení tlustého klienta je časově a finančně náročnější než u klienta tenkého, protože je potřeba jej nainstalovat na každý stroj, na kterém má být tento klient spuštěn. (Čermák, 2010)

Existují dva způsoby, jakými lze klienta na stroj nasadit: *push* a *pull*. Metodou *push* bude zajištěno cílené „tlačení“ klienta na stroj, metodou *pull* si pak každý uživatel musí klienta obstarat (stáhnout ze zdroje) sám. Ať už bude použit jakýkoliv způsob přenosu klienta, vždy se bude uživatel setkávat s různými problémy a ošetřováním chybových stavů (např. přerušení přenosu klienta, nedostatečná šířka pásma či podpora klientů, atd.) (Čermák, 2010)

Výhody:

- + ze strany klienta je požadován výkonný hardware stroje,
- + ze strany serveru není vyžadován výkonný hardware stroje,
- + relativně snadný a rychlý vývoj klienta,
- + přívětivé uživatelské rozhraní a možnost práce offline
- + snadná podpora uživatelů,
- + snadná přenositelnost klienta,
- + rychlá odezva klienta. (Čermák, 2010)

Nevýhody:

- vysoké nároky na šířku přenosového pásma,
- obtížné nasazení klienta na stroj,
- obtížné updatování klienta,
- obtížná správa klienta,
- aplikace v systému zanechává stopy (vlastní program, data, registry, atd.). (Čermák, 2010)

Tenký klient (thin client)

Tenký klient se váže k vrstvě prezentační, se kterou komunikuje skrze HTTP protokol, jenž se stará o zobrazování dat. Proto bývá tenkým klientem nejčastěji webový prohlížeč. Tenký klient je zbaven veškeré rozhodovací logiky (kromě validace dat zadávaných do webového formuláře). Z těchto skutečností vyplývá, že nároky na instalaci a veškeré doprovodné kroky (konfigurace, podpora, atd.) jsou v případě tenkého klienta minimální – uživateli stačí, aby měl nainstalovaný jakýkoliv webový prohlížeč. Odpadá tedy problém potřeby nejnovější verze klienta, neboť taková verze klienta je uživatelům k dispozici v okamžiku jeho nasazení na servery. (Čermák, 2010)

Komunikace mezi klientem a serverem je sice složitější, ale ve většině případech není potřeba přenášet velký objem dat. Ke zpracování dat dochází přímo na straně serveru, kdy je na klienta přenášen pouze výsledek. Nevýhodou tenkého klienta by mohla být ta skutečnost, že vývojář dopředu nikdy neví, kolik uživatelů

se k jeho aplikaci připojí. U tlustého klienta je zřejmé, kolik klientů bylo již na stroje nasazeno a tudíž je snazší odhadnout, jak výkonný HW bude potřeba. (Čermák, 2010)

Výhody:

- + ze strany klienta není vyžadován výkonný hardware stroje,
- + nízké nároky na šířku přenosového pásma,
- + snadné nasazení/update/správa/podpora uživatelů,
- + klient běží v každém webovém prohlížeči,
- + minimální stopy v systému (cookies, temporary files, history),
- + klient nabízí snadnou podporu uživatelů,
- + snadná přenositelnost klienta,
- + rychlá odezva klienta. (Čermák, 2010)

Nevýhody:

- ze strany serveru je vyžadován výkonný hardware stroje,
- obtížný vývoj,
- strohé uživatelské rozhraní,
- nemožnost pracovat offline,
- nižší odezva aplikace. (Čermák, 2010)

Chytrý klient (SMART client)

Chytrý klient je ve své podstatě vhodnou kombinací výhod tenkého a tlustého klienta a potlačením jejich nevýhod. Je mu umožněna práce offline, z toho důvodu obsahuje určitou logiku, aby mohl udržovat data, která v okamžiku navázání spojení synchronizuje. Využívá převážně lokální systémové zdroje (např. procesor, paměť RAM či diskové úložiště), ale v případě potřeby může využívat i externí zařízení. Z tohoto důvodu je schopen využívat taktéž nainstalované aplikace, např. Microsoft Office. (Čermák, 2010)

Chytrý klient je založen na existenci např. frameworku .NET, JRE (Java Runtime Environment) či pluginu v prohlížeči – všechny tyto prostředky si při svém spuštění (a dostupné síti) ověří, a pokud existuje novější verze, klient ji stáhne a spustí. V případě, že se novější verze aplikace objeví při práci uživatele s aplikací, je na ni uživatel upozorněn. Bezesporně největšími výhodami chytrého klienta je tedy jeho možnost práce offline a automatické stahování aktuálních verzí aplikace v okamžiku navázání spojení se serverem. (Čermák, 2010)

3.7.2 Programovací jazyky

K implementaci navrhovaných aplikací a informačních systémů je zapotřebí využít některého z programovacích jazyků. V této podkapitole se objeví nejznámější (a pravděpodobně také nejpoužívanější) programovací jazyky a jejich stručný

popis. Výhody a nevýhody jsou příliš subjektivního charakteru, tudíž do této kapitoly zahrnutý nebudou.

Programovací jazyk je dle Vojíře (2011) jakýsi komunikační nástroj mezi programátorem, který v daném jazyce formuluje algoritmy (nebo-li postupy řešení daných problémů), a mezi strojem (počítačem), který vyvíjený program interpretuje za pomoci svých technických prostředků. Jedná se tedy o soubor jakýchsi formálních pravidel pro zápis algoritmů. (Vojíř, 2011)

C a C++

Jazyk C a C++ jsou si velice podobné, nicméně je třeba mít na paměti, že v případě C++ se nejedná o pouhé rozšíření jazyka C. V jazyce C se vyskytují konstrukce, které v jazyce C++ nejsou povoleny. Vedle toho se v obou jazycích mohou vyskytovat stejné konstrukce, které ale v každém z nich mají jiný význam. (Virus, 2011)

Jazyk C byl navržen a implementován na počátku 70. let 20. století Denisem Ritchiem, který ho použil při implementaci operačního systému Unix. Jazyk se stal velice brzy oblíbeným a začal se používat k nejrůznějším účelům. V roce 1990 byl jazyk C standardizován na mezinárodní úrovni. (Virus, 2011)

Co se týče jazyka C++, jedním z prvních kroků k němu bylo tzv. C s třídami. V roce 1985 pak vznikla verze, která se již dala považovat za objektově orientovaný programovací jazyk. Tato verze je dodnes známá pod názvem C++. Jazyk C++ byl úspěšný zejména z důvodu kompatibility C a C++, která programátorům zaručovala, že mohli použít již existující kódy napsané v jazyce C. (Virus, 2011)

C#

C# je objektově orientovaný programovací jazyk vyvinutý společností Microsoft, jehož cílem je vhodně kombinovat výpočetní schopnosti jazyka C++ s programovou jednoduchostí jazyka Visual Basic. C# je založen na jazyce C++ a obsahuje funkce podobné těm, kterými disponuje Java. (Rouse, 2007)

C# je navržen tak, aby byl schopen spolupracovat s platformou .NET společnosti Microsoft. Cílem Microsoftu je usnadnit výměnu informací a služeb přes webové rozhraní a umožnit vývojářům vytvářet vysoce přenositelné aplikace. C# zjednodušuje programování skrze využití XML (Extensible Markup Language) a SOAP (Simple Object Access Protocol), které umožňují přístup k objektům či metodám bez nutnosti sepsání dodatečného kusu kódu pro každý krok. A jelikož programátoři mohou využívat existující kód bez nutnosti vzniku duplicit, očekává se od jazyka C#, že umožní rychlejší a levnější způsob umístění produktů a služeb na trh. (Rouse, 2007)

Společnost Microsoft spolupracuje s ECMA (mezinárodní útvar pro standardy) na standardizaci jazyka C#. Mezinárodní organizace pro normalizaci (ISO) se domnívá, že uznání jazyka C# jako standardu by povzbudilo ostatní společnosti k vývoji jejich vlastních verzí jazyka. Společnosti, které již používají C#, jsou: Apex Software, Bunka Orient, Component Source, devSoft, FarPoint Technologies, LEAD Technologies, ProtoView a Seagate Software. (Rouse, 2007)

Java

Programovací jazyk Java byl vyvíjen od roku 1991 firmou Sun Microsystems na principech programovacích jazyků C a C++ původně pro běžná elektronická zařízení (pračky, mikrovlnné trouby, apod.), která jsou ovládána mikroprocesorem. Původní název jazyka byl Oak, nicméně se zjistilo, že programovací jazyk s tímto názvem již existoval, proto vývojová skupina hledala nový název a po návštěvě firemního bufetu zvítězil návrh Java (volně přeloženo jako „kafe“). (Herout, 2010)

Projekt zpočátku nebyl úspěšný a tak se firma v roce 1993 rozhodla využít Javu pro programování aplikací služby WWW. Na konferenci v roce 1995 již bylo účastníkům jasné, že bude Java hrát významnou roli při programování webových aplikací. Mnozí z nich si však již tehdy uvědomili, že Java bude mít významné opodstatnění i jako klasický programovací jazyk. (Herout, 2010)

JavaScript

JavaScript je programovací jazyk nejčastěji využívaný při webovém vývoji. Původně byl vyvinut společností Netscape jako způsob pro vkládání dynamických a interaktivních elementů do webových stránek. Ačkoliv bývá JavaScript často spojován s Javou, jeho syntaxe je více podobná jazyku C a je založena na ECMAScriptu, což je skriptovací jazyk vyvinutý společností Sun Microsystems. (Christensson, 2014)

JavaScript je klientsky založený skriptovací jazyk, což znamená, že zdrojový kód je zpracován přímo webovým prohlížečem na klientské straně. To také znamená, že funkce JavaScriptu mohou běžet i poté, co je webová stránka načtena, bez nutnosti komunikace se serverem. Například, JavaScriptová funkce může zkontrolovat formulář ještě předtím než je odeslán, aby se ujistila, že jsou vyplněna všechna vyžadovaná pole. JavaScriptový kód tak může zobrazit chybové hlášení ještě předtím, než jsou informace přeneseny na server. (Christensson, 2014)

Jako skriptovací jazyky na straně serveru (např. PHP a ASP) může být JavaScript vložen kamkoliv na webovou stránku v rámci jejího HTML kódu. JavaScript může být k webové stránce připojen také jako samostatný .js soubor. (Christensson, 2014)

Perl

Perl je skriptovací programovací jazyk, který má podobnou syntaxi jako jazyk C a který zahrnuje nespočet oblíbených UNIXových funkcí (jako je např. sed, awk či tr). Perl se řadí mezi intepretované jazyky, tudíž může být libovolně kompilován ještě před spuštěním do kódu jazyka C či do bytekódu. Když je kód zkompilován, program psaný v Perlu je téměř (ale ne úplně) rychlý jako plně předkompilovaný program v jazyku C. (Rouse, 2005)

Perl je označován za dobrou volbu pro vývoj programů typu CGI (Common Gateway Interface) a to z toho důvodu, že obsahuje funkce vhodné pro dobrou manipulaci s textovým obsahem (ačkoliv také dokáže zpracovat binární soubory). Jazyk byl vyvinut Larry Wallem. (Rouse, 2005)

Obecně platí, že pro nového programátora je snazší a rychlejší psaní programů v jazyce Perl než ve více strukturovaných jazycích C a C++. Ale i přes tento fakt mohou být programy v jazyce Perl velice propracované. (Rouse, 2005)

PHP

PHP (hypertextový preprocesor) je open-sourcový, webově skriptovací jazyk používaný na straně serveru, který lze vložit do HTML kódu v rámci webových stránek. PHP je kompatibilní téměř se všemi hlavními webovými servery (především těmi Apache) a umožňuje vývojářům vkládat fragmenty kódu do HTML stránek – jedná se o kód, který je interpretovaný tak, jak jsou stránky předkládány uživatelům. PHP je často označován za tzv. spojovací jazyk, neboť usnadňuje propojení vyvíjených webových stránek s databázemi na straně serveru. (Converse, Park, 2004)

Python

Python je interpretovaný, objektově orientovaný programovací jazyk podobný Perlu, který si získal velkou oblibu hlavně díky své čisté syntaxi a čitelnosti. O Pythonu se říká, že je relativně snadný na naučení se a přenosný, což znamená, že může být interpretován na velkém množství operačních systémů (včetně systémů na bázi Unixu, Mac OS, atd.). (Grey, Hamblin, 2005)

Python byl vytvořen Guido van Rossumem, jehož oblíbená komediální skupina té doby nesla název Monty Python's Flying Circus (po této skupině je pojmenován i samotný programovací jazyk). Zdrojový kód je volně dostupný a otevřený všem modifikacím či znovupoužitím. Python má významný počet svých uživatelů. (Grey, Hamblin, 2005)

Ruby

Ruby je dynamický programovací jazyk s komplexní, ale expresivní gramatikou a hlavní knihovnou tříd s bohatým a silným API (= aplikační programové rozhraní). Ruby čerpá inspiraci z programovacích jazyků Lisp, Smalltalk a Perl, ale používá gramatiku, která je snadno pochopitelná pro programátory pracujícími s jazyky C a Java. (Rouse, 2010)

Ruby je čistě objektově orientovaný jazyk, ale je vhodný také pro procedurální a funkcionální programovací styly. Zahrnuje silné metaprogramovací schopnosti a může být využitý pro vytváření specifických doménových jazyků nebo DSL. (Rouse, 2010)

3.8 Shrnutí kapitoly

V této kapitole byla nastíněna problematika pracovních cest z hlediska Zákoníku práce, byly vymezeny druhy cestovních náhrad, stanoveny podmínky a nastíněny situace, za kterých má zaměstnanec nárok na proplacení svých nákladů a také určeny sazby pro přepočtení jednotlivých nákladových složek. Následně bylo třeba alespoň stručně objasnit problematiku návrhu a tvorby informačních systémů. V případě návrhu IS bylo dále nutné ponořit se do problematiky grafické notace

UML, která s touto fází souvisí. Pro možnosti tvorby IS pak bylo potřeba zmínit typy implementačních přístupů a nejčastěji používaných technologií.

4 Analytická část

V této části diplomové práce budou představena současná dostupná řešení problematiky pracovních cest, popsán podnik, pro který je praktická část realizována a vymezeny požadavky na navrhovanou aplikaci. Součástí kapitoly bude i objasnění celého procesu tvorby a schvalování pracovní cesty s popisem konkrétního cestovního příkazu, který je pro tento proces podstatnou položkou.

4.1 Dostupná řešení

Předtím, než bylo možné identifikovat požadavky podniku, bylo nutno zanalyzovat existující dostupná řešení stejné problematiky. Z průzkumu tak bylo zjištěno, že aplikací na správu pracovních cest jako takových je relativně málo, mnohem více existuje různých modulů napojitelných na stávající firemní systémy a stejně tak existuje nepřehledné množství různých online kalkulaček či pluginů, které jsou součástí velkých ekonomických systémů. Vybraní zástupci těchto existujících řešení budou popsáni níže.

4.1.1 plusPortal

plusPortal (či také Kompas 2) je produkt akciové společnosti PC Help, která se zaměřuje na vývoj a implementaci softwarových řešení v podnikové a veřejné sféře. Cílem společnosti je nabízet zákazníkům kvalitní a odborná řešení, jistotu, stabilitu, zázemí a vstřícnost v duchu hesla: „S naší pomocí se v IT neztratíte“. (plusPortal, 2015)

Samotný plusPortal je pak popisován jako nástroj podporující řízení lidských zdrojů a firemních procesů, webová aplikace dostupná odkudkoliv, software schopný rozvoje firmy, ve které je nasazený, a systém, který lze jednoduše napojit na stávající podnikový systém. plusPortal nabízí následující moduly: Personalistika, Vzdělávání zaměstnanců, Volno, Pracovní cesty, Drobné výdaje, Odměny a Stravenky. (plusPortal, 2015)

plusPortal slibuje při zakoupení jejich produktu následující:

- jednoduché a rychlé zaškolení,
- jednoduché ovládání celého modulu,
- veškeré náklady pod kontrolou
- automatická upozornění v podobě mailových notifikací,
- spolupráce s účetními a ERP systémy. (plusPortal, 2015)

Propojení plusPortalu a stávajícího systému je zajištěno tzv. datovou pumpou, což je jakýsi univerzální propojovací můstek schopný čtení a zápisu z různých databází či tabulkových procesorů. (plusPortal, 2015)

4.1.2 HELIOS Green

HELIOS Green je produkt akciové společnosti Asseco Solutions. Jedná se o největšího producenta podnikových IS na česko-slovenském trhu. Softwarové aplikace této společnosti jsou nabízeny také na dalších trzích v rámci střední Evropy. Společnost se zabývá také implementací a podporou specializovaných systémů všech velikostí. (HELIOS, 2015)

Produkt HELIOS Green je moderní ERP systém pro středně velké až velké firmy. Poskytuje podklady pro strategické rozhodování. Slibuje snadné přizpůsobení požadavkům organizace a také pokrytí všech oborových řešení. (HELIOS, 2015)

Systém se dělí na tyto části: Řízení společnosti, Finance a ekonomika, Obchod a marketing, Lidské zdroje, Provozní a podpůrné agendy, Výroba, Logistika a Sklady. Modul pracovních cest se pak nachází v části Provozní a podpůrné agendy, které dokáží podpořit informační systém. (HELIOS, 2015)

Modul pracovních cest nabízí:

- snazší poskytování náhrad v souladu s platnými zákony, vnitropodnikovými směrnici a dalšími požadavky organizace,
- porovnání plánu pracovní cesty a jejího skutečného průběhu,
- podpora výpočtu náhrad a vypořádání pracovní cesty v různých měnách,
- evidence uskutečněných jízd dopravních prostředků, informace o cílové destinaci a účelu cest, počty projetých služebních a privátních kilometrů, čerpání pohonných hmot, nákladů na údržbu a další,
- samostatná evidence všech dopravních prostředků včetně jejich oprav. (HELIOS, 2015)

4.1.3 Informační systém Vema

Akciová společnost Vema je dodavatelem komplexních řešení v oblasti informačních systémů pro řízení lidských zdrojů, ekonomiky či logistiky jak ve státních, tak v privátních organizacích České a Slovenské republiky. (Vema, 2015)

Informační systém Vema nabízí řešení pro podporu řízení lidských zdrojů, ekonomiky či logistiky. Tento systém je využíván již téměř 7000 organizacemi. Systém se skládá z těchto produktů: Mzdy, Personální systém, Komplexní řízení lidských zdrojů, Ekonomický systém a Ostatní produkty – do této kategorie spadá i problematika pracovních cest s názvem Plánování a evidence pracovních cest. (Vema, 2015)

Tento modul pak nabízí:

- plánování a evidenci pracovních cest s využitím libovolného dopravního prostředku od zadání cestovního příkazu, přes vyplacení zálohy, po následné zúčtování celé cesty,
- automatizace propočtů všech nákladových složek,

- aktivní práce se spolucestujícími, což vede k usnadnění zadávání pracovních cest, kterých se účastní více osob,
- evidence všech potřebných technických údajů u jednotlivých vozidel včetně plateb pojištění i daní a knihy jízd,
- výpočet náhrad včetně zúčtování vyplacené zálohy,
- automatické záznamy v knize jízd. (Vema, 2015)

4.1.4 Cesty ver.3.0 Perm

Společnost s ručením omezeným Kvasar se zaměřuje na oblast vývoje speciálního aplikačního software. Na základě rozvoje společnosti a trendů vývoje v oblasti IT došlo k rozšíření jejího oboru podnikání také na zpracování dat a poradenské služby se zaměřením na oblast personálních agend a výpočtů mezd. (Kvasar, 2015)

Cesty ver.3.0 Perm je softwarový produkt poskytující silný nástroj na řešení problémů s evidencí pracovních cest, vyplácením náhrad při pracovních cestách, s vedením knih jízd, plánováním cest (s ohledem na dostupnost vozidel) a sledováním firemního autoparku včetně všech nákladů. (Kvasar, 2015)

Co aplikace nabízí:

- evidenci firemního autoparku včetně dokumentace,
- evidenci pracovníků (řidičů) včetně jejich soukromých vozidel,
- evidenci uskutečněných pracovních cest s možností kombinování dopravy,
- automatický výpočet náhrad včetně stravného i při zahraničních cestách,
- tvorbu cestovního výkazu s možností vygenerování PDF souboru,
- záznamy o provozu vozidla, kniha nákupů PHM, servisní kniha,
- evidenci pracovních výkazů pracovníků jako možný podklad pro fakturaci práce,
- evidenci (zahraničních) zákazníků (podniků, firem, živnostníků),
- ekonomické vyhodnocení provozu vozidla,
- plánování cest včetně rezervace vozidel a tvorby cestovního příkazu,
- možnost definování pravidelně se opakujících cest,
- uživatelsky rozšiřitelnou databázi tras, vzdáleností, hraničních přechodů,
- možnost definování trasových okruhů,
- část **Autodoprava** poskytuje evidenci dopravy (např. mezinárodní kamionová doprava) včetně ekonomického zhodnocení,
- možnost chráněného přístupu do aplikace pomocí osobních přístupových kódů. (Kvasar, 2015)

4.1.5 Wheel CRM & DMS

Společnost s ručením omezeným TCL DigiTrade je mezinárodní softwarová společnost se sídlem v České republice, která se zabývá tvorbou specializovaných softwarových řešení v nejvyšší možné kvalitě. (TCL DigiTrade, 2015)

Wheel CRM & DMS je komplexní informační systém, který řeší činnosti v organizaci v oblasti dokumentově orientovaných informací. Jeho součástí je modul Cestovní příkazy, který je určen k plánování a následnému vyúčtování pracovních cest. Tento modul nabízí přehledné vytváření a zpracování evidence pracovních cest a jejich následných vyúčtování. Modul zahrnuje také schvalovací proces i účetní dokumentaci včetně tiskových výstupů přes Microsoft Word. (TCL DigiTrade, 2015)

Hlavní výhody modulu:

- přehledné uspořádání záznamů podle různých kritérií,
- výběr dat z nastavených číselníků,
- propojení s knihou jízd,
- tisk dokumentů přes Microsoft Office. (TCL DigiTrade, 2015)

4.1.6 Ekonomický systém POHODA

STORMWARE je česká softwarová společnost, která se zabývá výrobou softwarových produktů pro operační systém Microsoft Windows. (POHODA, 2015)

Program POHODA je komplexním účetním a ekonomickým softwarem určeným pro malé, střední i větší podniky z řad fyzických či právnických osob. Systém je oborově neutrální a je tak vhodný téměř pro všechny podnikatelské subjekty. Jádrem systému je propracovaný adresář a moduly jako: Banka, Pokladna, Majetek, atd. (POHODA, 2015)

Program POHODA nabízí přehled o uskutečněných pracovních cestách a nákladech či výdajích souvisejících jak s pracovními cestami, tak s provozem veškerých vozidel. Program taktéž umožňuje sestavit knihu jízd pro všechna vozidla (jak firemní, tak soukromá) používaná pro účely služebních cest. Napomáhá také s tuzemskými a zahraničními cestovními příkazy. (POHODA, 2015) POHODA automaticky napomáhá těmto úkonům:

- evidenci záloh na pracovní cestu v různých měnách,
- zpracování jednotlivých etap pracovní cesty,
- evidenci všech výdajů s možností nastavení typu výdajů,
- automatickému přepočtu výše stravného na základě údajů o etapách pracovní cesty,
- generování cestovních náhrad při použití vozidla,
- automatickému přepočtu přeplatku/doplatku na záloze,
- zaúčtování cestovních příkazů. (POHODA, 2015)

4.2 Stručná charakteristika podniku

Podnik, pro který je praktická část diplomové práce realizována, je celosvětovou společností zaměřující se na výrobu spotřebního materiálu s přidanou hodnotou. V České republice tento podnik spadá do automobilového průmyslu. Společnost se svou velikostí zařazuje do středně velkého podniku, neboť má méně než 1000 zaměstnanců. Společnost si však nepřála být v práci jmenována, tudíž nelze zmínit více informací. Všechny podstatné informace však budou popsány v dalších částech kapitoly.

4.3 Informace o pracovní cestě

Před samotným návrhem a implementací aplikace pro správu pracovních cest bylo nutné zjistit, jakým způsobem funguje stávající proces tvorby a schvalování pracovní cesty. Z komunikace s firmou bylo zjištěno, že současný stav zahrnuje cestovní příkaz ve formátu .xls (formát tabulkového editoru Excel) odpovídající potřebám firmy, které každý zaměstnanec vyplňuje dle svého nejlepšího vědomí a svědomí. Tento příkaz je následně vytištěn a zaměstnanec musí fyzicky navštívit další tři zaměstnance firmy (vedoucí pracovník oddělení, pracovník pokladny a pracovník ekonomického úseku), aby byla pracovní cesta úspěšně zaúčtována, proplacena a následně uzavřena.

4.3.1 Cestovní příkaz

Pro potřeby diplomové práce a úspěšné splnění požadavků firma poskytla cestovní příkaz v papírové podobě. Je třeba konstatovat, že cestovní příkaz sám o sobě není pevně daný a firma si jeho podobu může přizpůsobit dle svých potřeb, pokud však s ohledem na příslušný zákon Zákoníku práce zachová položky, které jsou potřebné pro účetnictví a ekonomické zpracování. Konkrétní příkaz se skládá z několika částí, které budou popsány nyní.

První částí příkazu je *hlavička* nesoucí údaje o samotném zaměstnanci, který má být vyslán na pracovní cestu. V hlavičce tak lze najít údaje o jménu, příjmení či titulech, bydlišti, identifikačním čísle zaměstnance, telefonním čísle či pracovní doby zaměstnance.

Další částí jsou *základní údaje o pracovní cestě*. Zde musí zaměstnanec vyplňovat údaje o počátku pracovní cesty (konkrétní místo počátku, datum a čas), dále místo jednání (město a stát), účel cesty a konec pracovní cesty (zpravidla místo konce cesty a datum příjezdu zpět). Zaměstnanec dále vyplňuje případného spolucestujícího (zde se firma spokojí se jménem a příjmením konkrétního zaměstnance), určený dopravní prostředek (s ohledem na stanoviska získaná z příslušného zákona Zákoníku práce) a povolenou zálohu. V této části se také nacházejí údaje, které již vyplňuje vedoucí pracovník oddělení a pracovník pokladny, konkrétně se jedná o údaje o výši povolené zálohy, datum vyplacení zálohy, číslo pokladního dokladu a podpisy jednotlivých zainteresovaných pracovníků.

Poté následuje část s názvem *vyúčtování pracovní cesty*, která se dá dále rozdělit na dvě samostatné položky. První položkou je zpráva o výsledku pracovní cesty, kterou zaměstnanec podává svému vedoucímu v okamžiku návratu z pracovní cesty. V této části je nutno vyplnit datum podání zprávy a podpis schvalujícího vedoucího pracovníka oddělení. Poté je již možno přistoupit k vyplnění samotného vyúčtování pracovní cesty, což je nejobsáhlejší část celého procesu – zaměstnanec musí vyplnit nespočet údajů s ohledem na příslušný zákon Zákonníku práce (jedná se o nejrůznější datумы, časové údaje, místa odjezdu/příjezdu, jednotlivé nákladové položky, atd.). V této části je třeba, aby zaměstnanec doložil veškeré doklady, které během své cesty získal (tzn. účty, jízdenky, atd.). Do této části cestovního příkazu vstupuje také zaměstnanec ekonomického úseku, který kontroluje jednotlivé nákladové položky právě s ohledem na příslušný zákon Zákonníku práce a také vedoucí zaměstnanec oddělení, který provede potřebná schválení.

Poslední částí cestovního příkazu je *výdajový-příjmový pokladní doklad*, do kterého doplňuje údaje pracovník pokladny. Jedná se o celkovou rekapitulaci nákladů za pracovní cestu, tzn. údaje o výši vyplacené zálohy, zda se jedná o doplatek/přeplatek s ohledem na vykázané náklady zaměstnance, a dále potřebné podpisy (pracovník ekonomického úseku, příjemce finanční částky, pracovník pokladny a vedoucí pracovník oddělení).

4.3.2 Proces pracovní cesty

Pro větší názornost bude celý proces ještě jednou zjednodušeně popsán v následujících bodech:

1. Zaměstnanec do cestovního příkazu vypíše údaje o své osobě a také základní informace o pracovní cestě, tzn. počátek/konec cesty, místo jednání a účel cesty. Zaměstnanec také vyplní výši předpokládané zálohy. Poté údaje o pracovní cestě předá vedoucímu pracovníkovi oddělení ke schválení/zamítnutí.
2. Vedoucí pracovník oddělení zkontroluje informace vyplněné zaměstnancem a údaje buď schválí nebo zamítné.
3. V případě schválení údajů zaměstnanec navštíví pokladnu, kde mu pracovník pokladny vyplatí schválenou výši zálohy. V případě zamítnutí údajů sdělí vedoucí pracovník oddělení své námitky zaměstnanci a zaměstnanec tyto námitky zapracuje do cestovního příkazu.
4. Poté zaměstnanec vycestuje na pracovní cestu a po návratu z ní sepíše zprávu o výsledku pracovní cesty, kterou odešle ke schválení vedoucímu pracovníkovi oddělení.
5. Vedoucí pracovník oddělení zprávu o výsledku pracovní cesty potvrdí a zaměstnanec může začít vyplňovat do cestovního příkazu detailní vyúčtování pracovní cesty, tzn. veškeré nákladové položky, použité dopravní prostředky, datумы cesty, odjezdy/příjezdy, apod. V této fázi je zaměstnanec

povinen přiložit veškeré doklady nabyté během pracovní cesty. Po vyplnění všech potřebných údajů je toto vyúčtování předáno vedoucímu pracovníkovi oddělení, který údaje zkontroluje a schválí.

6. Po schválení vyúčtování pracovní cesty vedoucím zaměstnancem oddělení je toto vyúčtování předáno pracovníkovi ekonomického úseku, který provede kontrolu všech nákladových položek a přepočítá (upraví) je podle příslušného zákona Zákoníku práce. Na základě tohoto přepočtu jsou vyčísleny skutečné náklady na pracovní cestu. Tyto náklady jsou poté porovnány s vyplacenou zálohou a je zjištěn buď doplatek či přeplatek.
7. Zaměstnanec navštíví pokladnu, kde mu na základě upraveného vyúčtování pracovník pokladny peníze buď doplatí a nebo naopak od zaměstnance získá přeplatek na záloze. Následně je pracovníkem pokladny vyplněn pokladní doklad, doplněny veškeré potřebné údaje (tzn. údaje o záloze, přeplatku/doplatku, potřebné podpisy) a pracovní cesta je uzavřena.

4.4 Analýza požadavků

Na základě výše zmíněných poznatků o procesu tvorby a schvalování pracovní cesty bylo možné dohodnout s firmou požadavky na realizovanou aplikaci. Požadavky jako takové se dělí na *funkční* (jedná se o funkcionalitu aplikace) a *nefunkční* (jedná se převážně o požadavky na software či hardware).

4.4.1 Funkční požadavky

- Pro práci v aplikaci bude vyžadováno přihlášení, kdy budou použity přihlašovací údaje (uživatelské jméno a heslo).
- Aplikace bude zobrazovat formulářové prvky či samotné formuláře na základě práv přihlášených uživatelů.
- Aplikace bude umožňovat vytváření nových pracovních cest, jejich ukládání do databáze a následnou editaci.
- Aplikace bude pracovní cesty na základě stavu, ve kterém se nachází, podsouvat jednotlivým účastníkům schvalovacího procesu.
- Aplikace bude upozorňovat jednotlivé účastníky schvalovacího procesu za pomoci mailových notifikací.
- Aplikace bude kontrolovat duplicitní záznamy v databázi a vykonávat příslušné akce pro zamezení jejich výskytu.
- Aplikace bude kontrolovat správnost vkládaných údajů, aby se co nejvíce omezil dodatečný lidský zásah (např. manuální úprava již uloženého záznamu v databázi).
- Aplikace bude přepočítávat náklady dle příslušného zákona Zákoníku práce.
- Aplikace bude umožňovat automatické plnění cestovního příkazu a jeho převod do formátu PDF.

- Aplikace bude co nejvíce uživatelsky přívětivá.

4.4.2 Nefunkční požadavky

Co se týče nefunkčních požadavků, zde podnik neměl až tak vysoké nároky. Nejdůležitější body jsou popsány níže.

- Aplikace bude spustitelná na platformě Windows.
- Aplikace bude implementována v jazyce C#, který je použit taktéž ve stávajícím firemním systému.
- Pro vývoj aplikace budou používány přednostně nástroje poskytované firmou Microsoft.
- Aplikace bude desktopového charakteru s oddělenou databází, která se bude nacházet na vzdáleném serveru.
- Aplikace bude využívat databázového systému MySQL, který je ve firmě nyní používán.

4.4.3 Požadavky na účastníky procesu

Pro lepší přehlednost mohou být funkční požadavky rozděleny ještě způsobem, který jasně ukáže, jaké kompetence budou mít jednotliví účastníci procesu. Systémové požadavky nebudou v této podkapitole zahrnovány.

Tab. 1 Požadavky na účastníky procesu pracovní cesty

	Zaměstnanec vyslaný na pracovní cestu	Vedoucí pracovník oddělení	Pracovník ekonomického úseku	Pracovník pokladny
Stejně	<ul style="list-style-type: none"> • Přihlášení do systému skrze přihlašovací údaje • Zobrazení správy pracovních cest • Zobrazení formulářů a konkrétních formulářových prvků na základě práv přihlášeného uživatele 			
Rozdílné	<ul style="list-style-type: none"> • Tvorba/editace pracovní cesty • Navržení výše zálohy • Tvorba zprávy o výsledku pracovní cesty • Doplnění údajů do vyúčtování pracovní cesty • Příložení potřebných dokladů v podobě příloh 	<ul style="list-style-type: none"> • Zobrazení příloh a jejich kontrola • Schválení/zamítnutí základních údajů pracovní cesty • Schválení/zamítnutí/úprava výše zálohy • Schválení/zamítnutí zprávy o výsledku pracovní cesty 	<ul style="list-style-type: none"> • Zobrazení příloh a jejich kontrola • Přepočít/kontrola nákladových složek vyúčtování pracovní cesty 	<ul style="list-style-type: none"> • Vyplacení zálohy • Vytvoření pokladního dokladu • Finanční vypořádání se zaměstnancem po skončení pracovní cesty

4.5 Srhnutí kapitoly

V analytické části byla představena pravděpodobně nejznámější dostupná řešení zabývající se problematikou pracovních cest. Bylo zjištěno, že produktů daného zaměření, se na trhu vyskytuje relativně velké množství, většinou se však jedná o moduly velkých firemních systémů. Samostatných aplikací bylo velice málo. Dále bylo zjištěno, že se na trhu vyskytuje také velké množství různých kalkulátorů, které však z obsahového hlediska nejsou plně dostačující.

V této kapitole byla také stručně představena charakteristika společnosti, pro kterou je praktická část diplomové práce realizována. Komunikací se společností byly stanoveny důležité poznatky pro implementaci (došlo např. k objasnění současného stavu v podniku a k objasnění procesu tvorby a schvalování pracovní cesty), a taktéž požadavky na aplikaci a jednotlivé uživatele.

5 Vlastní řešení

Studiem teoretických východisek práce a požadavků firmy na výslednou aplikaci bylo stanoveno, že aplikace bude mít vícevrstvou architekturu a bude se jednat o řešení ve formě tlustého klienta (desktopová aplikace s databází na vzdáleném serveru). Na základě těchto zjištění byl stanoven postup, který se skládá z návrhu aplikace v podobě diagramů a samotné implementace aplikace v požadovaném jazyce. Obě části budou řešeny právě v této kapitole.

5.1 Návrh aplikace pro správu pracovních cest

Předtím, než bylo možné aplikaci implementovat, bylo potřeba vytvořit co nejdetailnější návrh. V předchozích kapitolách práce bylo zmíněno, že pro návrh aplikace s vícevrstvou архитектурou se jako nejvhodnější jeví grafická notace UML. Bezespornou výhodou této notace je fakt, že problémům a jejich řešením nastíněných skrze jednotlivé diagramy, porozumí i uživatelé s nižší počítačovou gramotností a taktéž osoby, které nejsou v IT sféře až tak zainteresované. Důvodem je hlavně fakt, že UML není metodika, ale soubor grafických prvků, které lze přizpůsobit potřebám a možnostem každého uživatele.

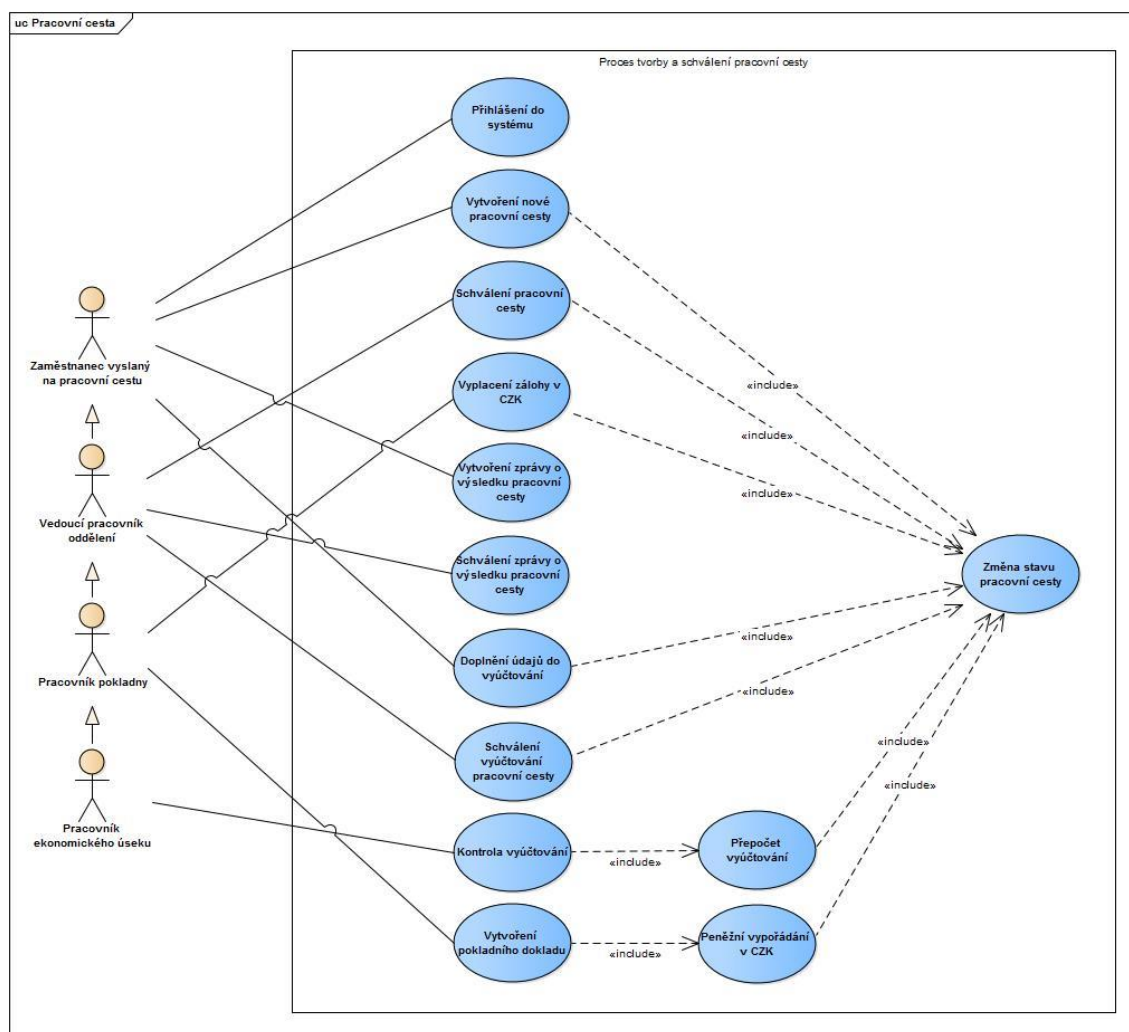
Pro tvorbu jednotlivých diagramů byl vybrán software Enterprise Architect firmy Sparx Systems. Dle Sparx Systems (2015) je Enterprise Architect výjimečný nástroj disponující high-end možnostmi a bohatou sadou funkcí, které napomáhají spravovat informace a celkově inovovat v dnešním složitém a náročném prostředí. Enterprise Architect nabízí modelovací funkce pro celý životní cyklus obchodních a IT systémů, softwarového/systemového inženýrství a real-time vývoje. Enterprise Architect je multi-uživatelský, grafický nástroj navržený pro potřebu tvorby robustních a udržitelných systémů. (Sparx Systems, 2015)

5.1.1 Diagram případů užití

Diagram případů užití přehledně znázorňuje všechny uživatele, kteří budou s vyvíjenou aplikací pracovat. Jak již bylo zmíněno v analytické části práce, do procesu tvorby a schvalování pracovní cesty vstupují celkem čtyři účastníci. Pro zopakování se jedná o tyto aktory:

- zaměstnanec vyslaný na pracovní cestu,
- vedoucí pracovník oddělení,
- pracovník ekonomického úseku,
- pracovník pokladny.

Každý z nich pak na základě požadavků obsluhuje konkrétní operace nebo-li případy užití. V této fázi lze vycházet také z přehledné tabulky požadavků na jednotlivé účastníky pracovní cesty. Na obrázku níže lze vidět namodelovaný diagram případů užití, který odpovídá domluveným požadavkům.



Obr. 3 Diagram případů užití pro proces tvorby a schválení pracovní cesty

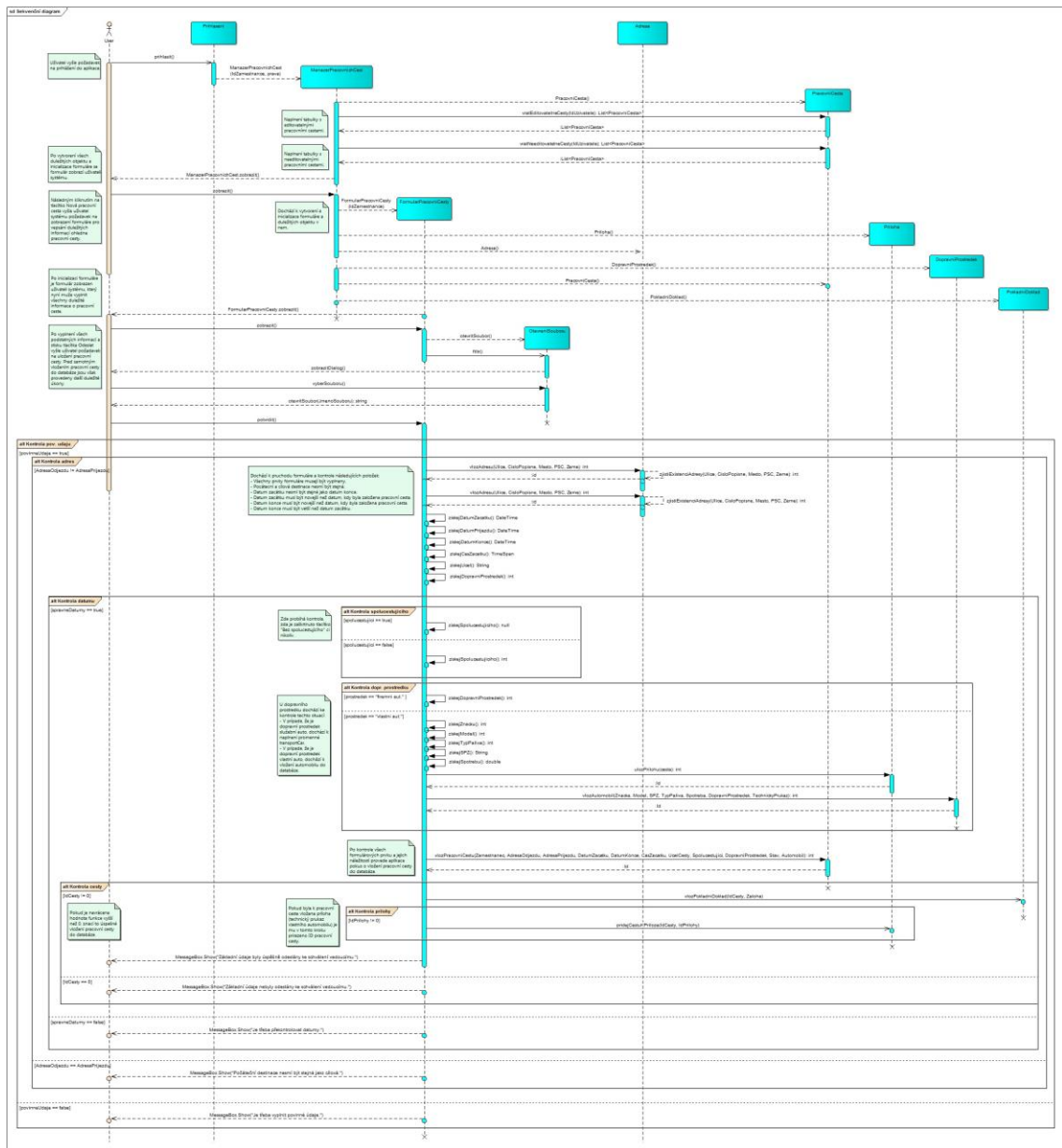
Každý ze čtyř aktorů se pro příslušné systémové akce musí nejprve přihlásit do systému, jedná se tak o společný případ užití. Na základě přístupových práv pak může vykonávat konkrétní činnosti.

Z obrázku je patrné, že o zaměstnanci vyslaném na pracovní cestu lze tvrdit, že jeho práva jsou nejnižší úrovně. V jeho kompetenci se nachází „pouze“ vytvoření nové pracovní cesty, vytvoření zprávy o výsledku pracovní cesty a doplnění údajů do vyúčtování. Žádné schvalovací procesy se v jeho pravomocích nenacházejí.

Nejvíce schvalovacích kompetencí má vedoucí pracovník oddělení, který schvaluje jak nově vytvořenou pracovní cestu (respektive základní údaje vyskytující se v hlavičce cestovního příkazu včetně výše požadované zálohy), tak zprávu o výsledku pracovní cesty a i samotné detailní vyúčtování pracovní cesty.

Pracovník pokladny je pak důležitý zejména z toho finančního hlediska, neboť je jeho hlavní náplní vyplacení záloh na pracovní cestu, tvorba a evidence pokladních dokladů, a také finanční vyrovnání se se zaměstnancem po kontrole a přepočtu veškerých nákladových složek.

zabývající se založením nové pracovní cesty, respektive vyplněním základních údajů.



Obr. 5 Sekvenční diagram tvorby nové pracovní cesty

Pro názornost bude sekvenční diagram popsán v bodech:

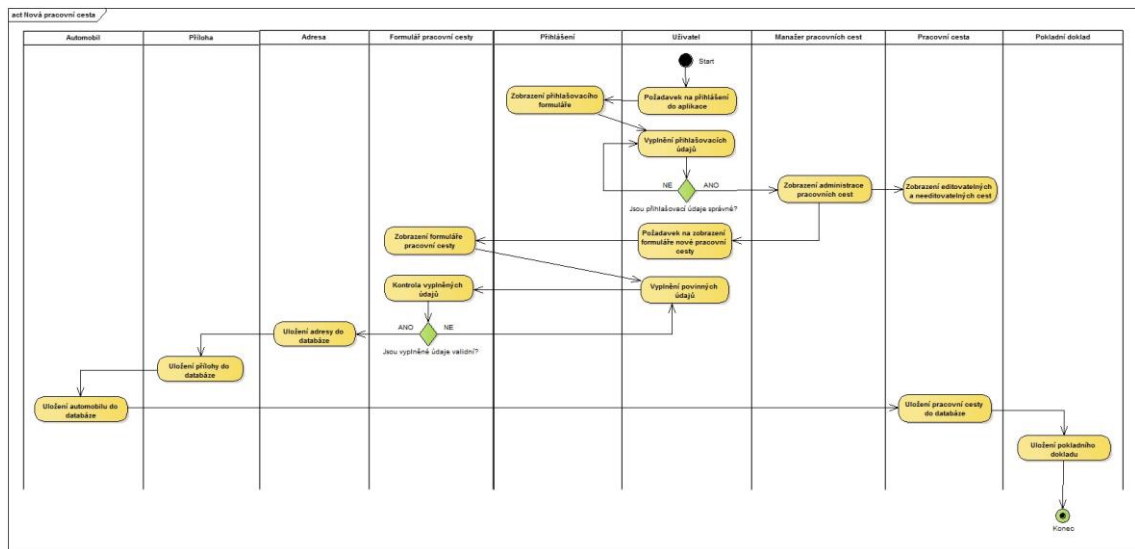
1. Uživatel spustí aplikaci pro správu pracovních cest, čímž se zobrazí přihlašovací formulář. Po zadání přihlašovacích údajů je uživatel buď přihlášen do aplikace, anebo upozorněn na chybně zadané údaje a vyzván k jejich opakovanému zadání.

2. Po přihlášení do aplikace je vyslán požadavek na zobrazení formuláře se správou pracovních cest. V tomto kroku je požadovaný formulář naplněn (ne)editovatelnými pracovními cestami a následně zobrazen uživateli.
3. Při stisku tlačítka *Nová pracovní cesta* uživatel vyšle požadavek na zobrazení formuláře pro tvorbu nové pracovní cesty. Dochází tak k inicializaci formuláře (tzn. vytvoření všech formulářových prvků a důležitých objektů) a formulář je zobrazen uživateli.
4. V dalším kroku uživatel vyplní pravdivě všechny povinné informace ohledně pracovní cesty a následně tuto cestu odešle ke zpracování stiskem tlačítka *Odeslat ke schválení*. Po stisku tlačítka je vykonáno několik samostatných úkonů, jedná se o:
 - 4.1. kontrolu vyplnění všech povinných polí,
 - 4.2. kontrolu validity jednotlivých zadaných údajů (údaje musejí být ve správném datovém formátu a taktéž dávat smysl, striktně jsou tak kontrolovány veškeré datумы, které musejí vyhovovat jak potřebám cestovního příkazu, tak umožňovat co nejvyšší automatizaci procesu),
 - 4.3. kontrolu existence objektů v databázi a jejich následné uložení do databáze (adresy a použitý vlastní dopravní prostředek jsou ukládány do databáze, tudíž musejí být kontrolovány duplicity a v případě, že se objekty v databázi již nacházejí, aplikace tuto skutečnost musí zpracovat a podniknout další potřebné kroky),
 - 4.4. uložení pracovní cesty do databáze (po náležitě kontrole všech formulářových prvků je vyslán požadavek na uložení cesty do databáze).

Veškeré chybné kroky budou v aplikaci ošetřeny dialogovými okny s chybovými hláškami, které budou uživatele směřovat k dalším potřebným krokům. Jakmile bude formulář validní, aplikace uloží pracovní cestu se všemi náležitostmi do databáze.

5.1.4 Diagram aktivit

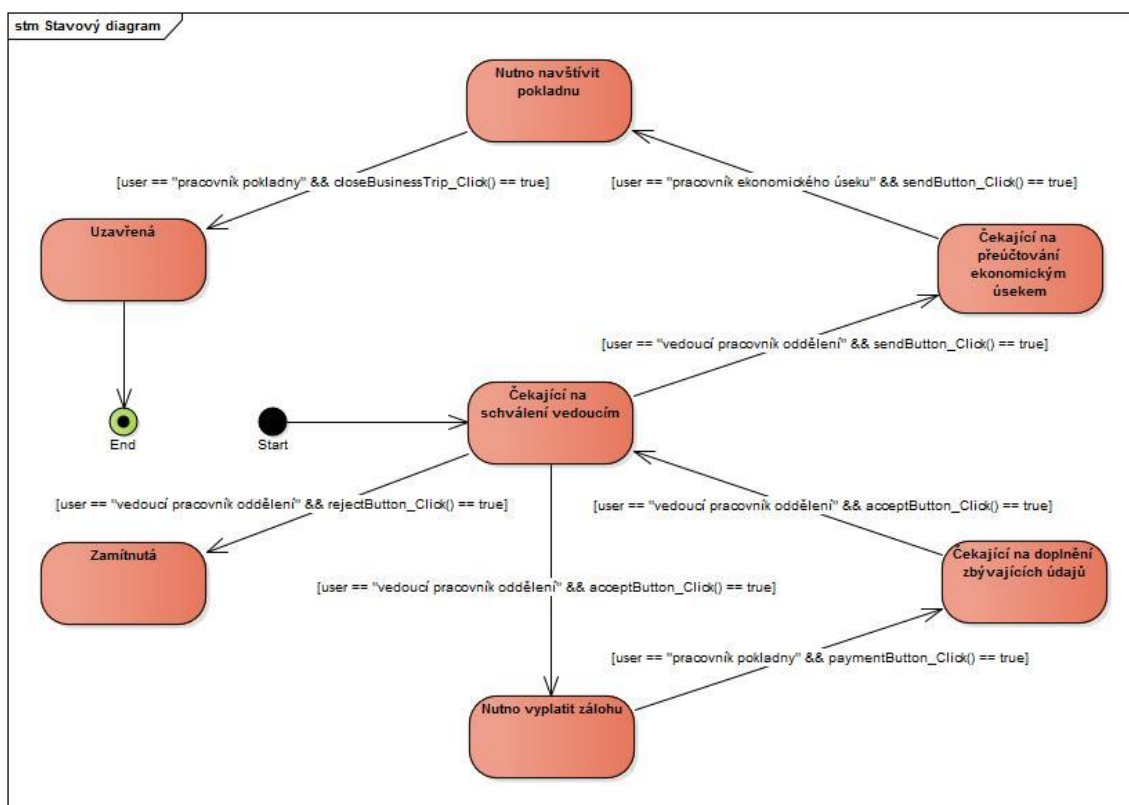
Diagram aktivit vychází ze sekvenčního diagramu a ukazuje ho z trochu jiného pohledu. Z toho důvodu není potřeba jej více rozebírat, neboť se jedná víceméně o to stejné jako v případě sekvenčního diagramu. Vyšší rozlišení diagramu bude opět k nahlédnutí v kapitole Přílohy.



Obr. 6 Diagram aktivit pro založení nové pracovní cesty

5.1.5 Stavový diagram

Důležitou položkou při návrhu aplikace se stal tzv. stavový diagram. Ten přehledně zobrazuje, jakým způsobem se jednotlivé stavy pracovní cesty mění při procesu její tvorby a schvalování.



Obr. 7 Stavový diagram procesu tvorby a schvalování pracovní cesty

Stavy, kterých může pracovní cesta během procesu nabývat:

- *Čekající na schválení vedoucím (VED)* – Tohoto stavu může pracovní cesta nabývat hned několikrát, neboť je vedoucí pracovník oddělení ústředním schvalovacím článkem celého procesu. Stav je cestě přiřazen kliknutím na příslušná tlačítka formuláře a vyvoláním požadavku na změnu.
- *Zamítnutá (X)* – Jedná se o případ zamítnutí pracovní cesty vedoucím pracovníkem oddělení. Změna stavu je provedena odesláním požadavku přes kliknutí na tlačítko znázorňující zamítnutí pracovní cesty.
- *Nutno vyplatit zálohu (POK)* – Do tohoto stavu se pracovní cesta dostane jakmile vedoucí pracovník oddělení schválí základní údaje a tedy i požadovanou zálohu. Zaměstnanec vyslaný na pracovní cestu pak musí navštívit pokladnu, kde mu pracovník pokladny vyplatí schválenou částku.
- *Čekající na doplnění zbývajících údajů (EDIT)* – Tento stav je cestě přiřazen po vyplacení schválené výše zálohy pracovníkem pokladny. Ten fyzickou částku předá zaměstnanci a potvrdí vyplacení zálohy v systému, čímž je vyslán požadavek na změnu stavu.
- *Čekající na přeúčtování ekonomickým úsekem (EKU)* – Jakmile zaměstnanec vyplní veškeré potřebné detaily vyúčtování (a přiloží potřebné doklady),

stiskem příslušného tlačítka vyvolá požadavek na změnu stavu u pracovní cesty. Ta se tak bude nacházet ve fázi čekání na přeúčtování.

- *Nutno navštívit pokladnu (POK)* – Pro zjednodušení implementace aplikace byl této fázi schvalovacího procesu přiřazen vlastní stav (i když by šlo navázat na již existující stav *Nutno vyplatit zálohu*). Tento stav je pracovní cestě přiřazen po kontrole a případném přepočtu údajů ekonomickým úsekem, kdy příslušný pracovník opět stiskem tlačítka vyvolá požadavek na změnu stavu.
- *Uzavřená (✓)* – Jakmile jsou nákladové položky překontrolovány pracovníkem ekonomického úseku, pracovník musí navštívit pokladnu, kde se s ním pracovník pokladny podle upraveného vyúčtování finančně vyrovná. Po tomto kroku pracovník pokladny cestu uzavře stiskem příslušného tlačítka, čímž se vyvolá požadavek na změnu stavu, a ta se tak stane dokončenou a kompletně uzavřenou.

5.2 Implementace aplikace pro správu pracovních cest

Po vyřešení návrhu aplikace lze přistoupit k její samotné realizaci nebo-li implementaci. V rámci komunikace a spolupráce s firmou, pro kterou byla aplikace vyvíjena, bylo navrženo řešení v podobě menšího informačního systému či aplikace, která by umožňovala komplexní správu vytvářených pracovních cest.

Z požadavků vyplývajících ze strany firmy bylo zjištěno, že nevhodnějším řešením by byla aplikace na bázi klient-server (čili vícevrstvá architektura), kdy jednotliví klienti (aplikace) budou nainstalováni na zaměstnaneckých počítačích, samotná databáze pak poběží na jednom ze vzdálených firemních serverů. Pro implementaci tak bylo potřebné použít některý z objektově orientovaných jazyků. Dle požadavků firmy byl vybrán programovací jazyk C# a vývojové prostředí Microsoft Visual Studio. Pro usnadnění práce s databází typu MySQL pak byl využit framework LINQ a platforma .NET Framework.

LINQ je programovací model zavádějící dotazy do všech jazyků platformy Microsoft .NET. Aby bylo LINQ úplně podporováno, jsou vyžadována určitá rozšíření jazyka, kterého vývojář používá. Tato rozšíření však zvyšují efektivitu vývojářů a poskytují kratší, jasnější a smysluplnější syntaxi pro manipulaci s daty. (Pialorsi, Russo, 2009)

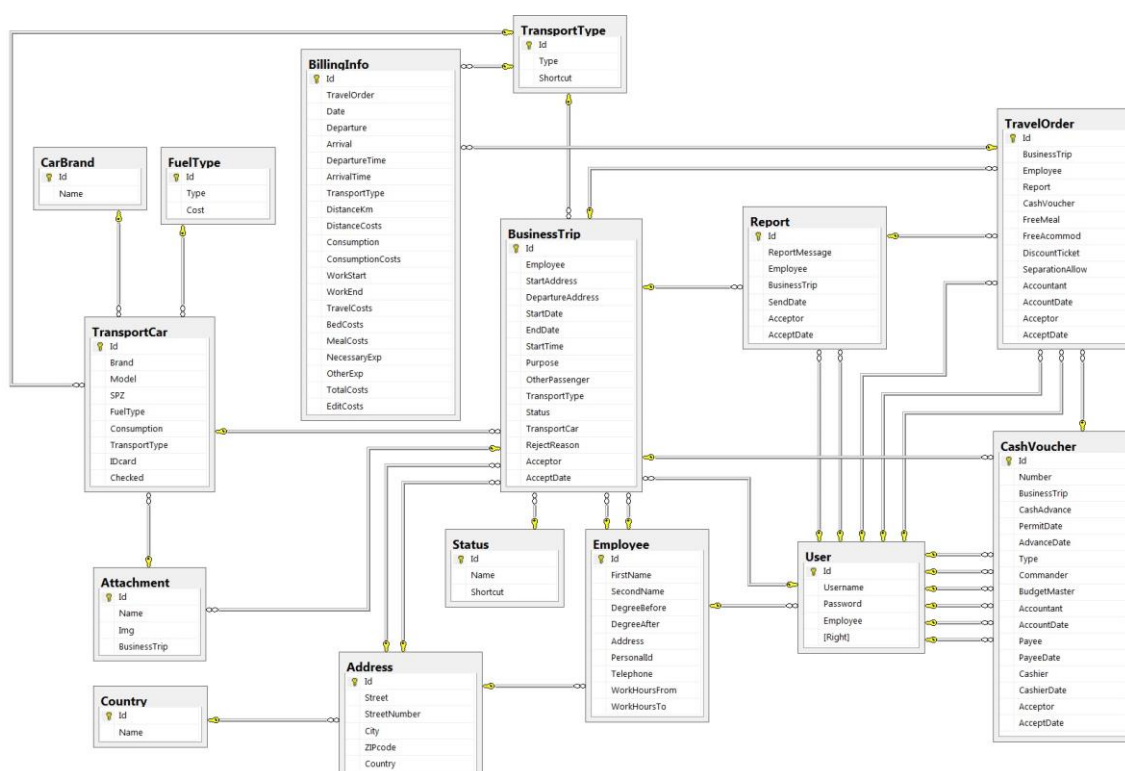
.NET Framework je konzistentní programovací model společnosti Microsoft, který je určen pro vyvíjení aplikací, které se vyznačují působivým uživatelským rozhraním, bezpečnou a bezproblémovou komunikací, a taktéž schopností modelování řady obchodních procesů. (Microsoft, 2015)

Řešení mělo být uživatelsky co nejvíce přívětivé, a proto bylo při implementaci dbáno na využití standardních formulářových tříd a prvků – tyto prvky jsou pro samotného uživatele určitě příjemné, neboť nekladou vysoké nároky na počítačovou gramotnost (jako je tomu například u konzolových aplikací). Design aplikace tak bude vesměs klasický s ohledem na možnou úpravu v blízké budoucnosti skrze Windows Presentation Foundation (WPF).

5.2.1 Návrh databáze

Aplikace by dle požadavků měla komunikovat s databází MySQL umístěnou na vzdáleném podnikovém serveru. MySQL je open-sourcový relační databázový systém založený na strukturovaném dotazovacím jazyku SQL. (Eng, McCormack, 2013)

Celá databáze byla navržena a spravována v nástroji Microsoft SQL Server Management Studio. Navržená databáze vychází hlavně z UML diagramů, konkrétně z diagramu tříd, a stejně jako tento diagram i databáze samotná obsahuje 15 tabulek. Jednotlivé tabulky jsou vytvářeny s ohledem na fyzický formulář cestovního příkazu, aby dokázaly věrně zachytit jeho současnou podobu a usnadnit tak celý proces tvorby a schvalování pracovní cesty.



Obr. 8 Model databáze

Tabulky, které se nacházejí v modelu databáze:

- **Address** – Tabulka obsahuje všechny důležité adresy. Ať už se jedná o adresu potřebnou pro vytvoření pracovní cesty či bydliště zaměstnance. *Atributy:* Id, Street, StreetNumber, ZIPcode, Country
- **Attachment** – Tato tabulka shromažďuje veškeré obrazové přílohy, může se tedy jednat buď o přiložený technický průkaz při použití vlastního automobilu či o doklady, které zaměstnanec nabyl při své pracovní cestě. *Atributy:* Id, Name, Img, BusinessTrip

- **BillingInfo** – Tato tabulka obsahuje detailní vyúčtování pracovní cesty. Je řešena tak, že pro každý řádek fyzického cestovního příkazu je ukládán jeden záznam právě do této tabulky. Pokud tedy uživatel do formuláře vyúčtování pracovní cesty vyplní 12 řádků s údaji, bude tato tabulka obsahovat 12 záznamů s odkazem na ID konkrétní pracovní cesty. Jedná se tak o nejrozsáhlejší tabulku celé databáze. *Atributy:* Id, TravelOrder, Date, Departure, Arrival, DepartureTime, ArrivalTime, TransportType, DistanceKm, DistanceCosts, Consumption, ConsumptionCosts, WorkStart, WorkEnd, TravelCosts, BedCosts, MealCosts, NecessaryExp, OtherExp, TotalCosts, EditCosts
- **BusinessTrip** – Jedná se o tabulku obsahující hlavičku cestovního příkazu. Jsou v ní tedy uloženy základní údaje o pracovní cestě včetně odkazu na ID zaměstnance vyslaného na pracovní cestu a vedoucího zaměstnance oddělení, který pracovní cestu schvaluje. *Atributy:* Id, Employee, StartAddress, StartDate, EndDate, StartTime, Purpose, OtherPassenger, TransportType, Status, TransportCar, RejectReason, Acceptor, AcceptDate
- **CarBrand** – Pro možnost využití vlastního automobilu jakožto dopravního prostředku během pracovní cesty bylo potřeba vytvořit také tabulku uchovávající dostupné značky automobilů. *Atributy:* Id, Name
- **CashVoucher** – Tabulka obsahující údaje o pokladních dokladech, které jsou součástí cestovního příkazu. Tabulka obsahuje relativně velké množství atributů, neboť je pro potřeby pokladního dokladu nutné uchovávat velké množství údajů. *Atributy:* Id, Number, BusinessTrip, CashAdvance, PermitDate, AdvanceDate, Type, Commander, BudgetMaster, Accountant, AccountDate, Payee, PayeeDate, Cashier, CashierDate, Acceptor, AcceptDate
- **Country** – Tato tabulka uchovává všechny státy, které souvisejí s jednotlivými adresami. Momentálně je tabulka omezena pouze na evropské státy. *Atributy:* Id, Name
- **Employee** – Podstatnou částí databáze je tabulka obsahující údaje o jednotlivých zaměstnancích. Zde konkrétně jsou evidovány jak osobní údaje zaměstnance, tak jeho pracovní údaje. *Atributy:* Id, FirstName, SecondName, DegreeBefore, DegreeAfter, Address, PersonalId, Telephone, WorkHoursFrom, WorkHoursTo
- **FuelType** – Pro potřeby výpočtu nákladových složek cestovního příkazu bylo nutno evidovat taktéž palivové typy u dopravních prostředků a jejich ceny za litr. Typ paliva je potřeba evidovat i v případě, že zaměstnanec při své pracovní cestě využije vlastní vozidlo. *Atributy:* Id, Type, Cost
- **Report** – Tato tabulka ukládá údaje o zprávě, kterou zaměstnanec sepisuje po návratu z pracovní cesty. Jedná se o zprávu o výsledku pracovní cesty. *Atributy:* Id, ReportMessage, Employee, BusinessTrip, SendDate, Acceptor, AcceptDate

- **Status** – Vzhledem k tomu, že pracovní cesta během svého schvalovacího procesu prochází různými stavy, je potřeba tyto stavy taktéž evidovat, což vykonává tabulka Status. *Atributy:* Id, Name, Shortcut
- **TransportCar** – V případě využití vlastního či firemního automobilu je potřeba tyto dopravní prostředky evidovat. O to se stará právě tabulka TransportCar. *Atributy:* Id, Brand, Model, SPZ, FuelType, Consumption, TransportType, IDcard, Checked
- **TransportType** – Pro korektní vyplnění cestovního příkazu je také důležitá evidence jednotlivých typů dopravních prostředků. To provádí tabulka TransportType. *Atributy:* Id, Type, Shortcut
- **TravelOrder** – Tato tabulka propojuje záznamy z více tabulek dohromady. Jedná se o takovou sumarizaci veškerých kroků (či údajů), které účastníci procesu tvorby a schvalování pracovní cesty během tohoto procesu provedou. Lze tedy konstatovat, že údaje v této tabulce jsou opravdu jakýmsi cestovním příkazem. *Atributy:* Id, BusinessTrip, Employee, Report, CashVoucher, FreeMeal, FreeAcommod, DiscountTicket, SeparationAllow, Accountant, AccountDate, Acceptor, AcceptDate
- **User** – Tabulka uživatelů, kteří mohou obsluhovat aplikaci pro správu pracovních cest. Každý uživatelský účet je spjat s jedním zaměstnancem a má svá specifická práva. *Atributy:* Id, Username, Password, Employee, Right

Vztahy (relace) v databázi:

- Jedna konkrétní adresa (Address) může být evidována u více zaměstnanců, jedná se o příklad, kdy jsou zaměstnanci příbuzenského či partnerského vztahu, spolubydlíci, anebo jen žijí ve stejném domě. Každá adresa může být součástí pouze jedné země. Adresa je také součástí pracovní cesty a může být součástí více pracovních cest zároveň.
- Příloha (Attachment) může být součástí tvorby nové pracovní cesty (v podobě technického průkazu při použití vlastního automobilu), v takovém případě může být pouze jedna příloha u jednoho evidovaného automobilu. Příloha s největší pravděpodobností bude využita také při doložení dokladů nabytých během pracovní cesty, v tomto případě může být jedna příloha spojována pouze s jednou pracovní cestou.
- Vyúčtování pracovní cesty (BillingInfo) je vázáno vždy k jednomu konkrétnímu cestovnímu příkazu. Každý tento jednotlivý řádek vyúčtování je pak vázán k jednomu typu dopravního prostředku.
- Pracovní cesta (BusinessTrip) obsahuje údaje o dopravním prostředku (jedná se o dopravní prostředek, který je uveden při zakládání pracovní cesty jako primární, tudíž může pracovní cesta obsahovat pouze jeden dopravní prostředek), z toho stejného důvodu se váže také k tabulce automobil, kdy může každá pracovní cesta mít pouze jeden (nebo žádný) automobil. Pracovní cesta dále eviduje svůj stav, který se sice v jednotlivých fázích procesu mění,

ale v jednu chvíli může nabývat pouze jednoho stavu. Každá pracovní cesta má dále jednoho schvalujícího zaměstnance a jednoho spolucestujícího (ten však není povinný). Pracovní cesta je vázána také k jednomu uživateli, což je zaměstnanec vyslaný na pracovní cestu. Každá pracovní cesta může mít žádnou nebo více příloh. Jedna konkrétní pracovní cesta je vázána taktéž k jednomu konkrétnímu pokladnímu dokladu a k jedné zprávě o výsledku pracovní cesty. Pracovní cesta obsahuje také dvě adresy, tyto dvě adresy nesmějí být stejné.

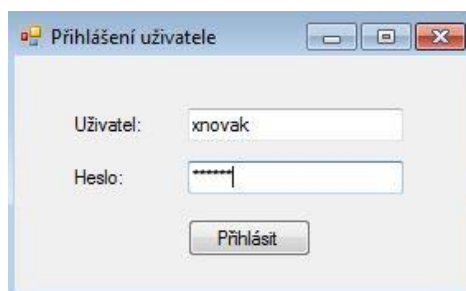
- Značka automobilu (CarBrand) se váže k tabulce automobil a může být použita u libovolného množství automobilů (v databázi se tak může vyskytovat více automobilů např. značky Škoda).
- Pokladní doklad (CashVoucher) je specifický tím, že obsahuje velké množství uživatelů, kdy se každý uživatel může v jednom dokladu vyskytovat vícekrát. Doklad je taktéž vázán k jedné konkrétní pracovní cestě a je vytvářen pro jeden konkrétní cestovní příkaz.
- Země (Country) se relačně váže pouze k tabulce adresa, kdy se může jedna konkrétní země vyskytovat u více adres zároveň.
- Každý zaměstnanec (Employee) má v tabulce evidovanou jednu konkrétní adresu (bydliště), tentýž zaměstnanec může být součástí více pracovních cest zároveň (ať už jako spolucestující či jako schvalující vedoucí pracovník oddělení) a zároveň je každý zaměstnanec vázán k jednomu uživateli.
- Co se týče typu paliva (FuelType), tak to se váže relačně pouze k automobilu. Každý typ paliva může být součástí více automobilů zároveň.
- Zpráva o výsledku pracovní cesty (Report) zahrnuje jednoho konkrétního uživatele (zaměstnanec vyslaný na pracovní cestu) a také schvalujícího uživatele – tyto dvě osoby mohou být stejné. Zpráva se váže k jedné konkrétní pracovní cestě.
- Stav (Status) pracovní cesty je součástí všech pracovních cest v databázi. Každý stav se tedy může vyskytovat v tabulce pracovních cest vícekrát.
- Automobil (TransportCar) se váže jak ke značce (může nabývat pouze jedné konkrétní značky), tak k typu paliva a k dopravnímu prostředku (stejně případy jako se značkou). Dále se může vyskytovat u více pracovních cest v databázi současně a je u něj evidována jedna konkrétní příloha v podobě technického průkazu.
- Dopravní prostředek (TransportType) souvisí s pracovní cestou, s detailem vyúčtování pracovní cesty a s automobilem. Pro všechny tyto případy platí, že se dopravní prostředek může vyskytovat ve více případech v databázi současně.
- Cestovní příkaz (TravelOrder) se dá chápat jako jakési zastřešení všech zbývajících tabulek v databázi. Obsahuje tři uživatele, přičemž uživatelé mohou být stejní (jedná se o zaměstnance vyslaného na pracovní cestu,

schvalujícího vedoucího pracovníka a pracovníka pokladny) a mohou se vyskytovat i v jiných cestovních příkazech. Dále je vázán na tabulku se zprávami o výsledcích pracovních cest, kdy každý cestovní příkaz obsahuje právě jednu takovou zprávu. Cestovní příkaz je vázán také k jednomu konkrétnímu pokladnímu dokladu a k jedné pracovní cestě. Jako propojovací prvek je vázán taktéž k tabulce s vyúčtováním pracovní cesty, kdy může obsahovat alespoň jeden nebo více takových záznamů.

- Uživatel (User) je napojen na jednoho konkrétního zaměstnance, pracovní cestu (ale může být součástí více pracovních cest zároveň), pokladní doklad (kde je velké množství uživatelů zároveň, ti mohou být různí/stejní, a jeden konkrétní uživatel může být ve více pokladních dokladech zároveň). Uživatel je dále vázán na zprávu o výsledku pracovní cesty, kdy může taktéž figurovat u více zpráv v databázi zároveň. V poslední řadě je součástí cestovního příkazu, což je stejné jako v případě pokladního dokladu.

5.2.2 Přihlášení do aplikace

Po objasnění návrhu databáze je možno přejít k jednotlivým aplikačním formulářům a jejich implementačnímu řešení. Prvním formulářem, který se objeví uživateli po spuštění aplikace, je formulář pro *přihlášení do aplikace*. Zde musí uživatel vyplnit své uživatelské jméno a uživatelské heslo. Na základě těchto údajů systém zkontroluje příslušné záznamy v databázi a pokud se uživatel v databázi opravdu nachází, je vpuštěn do aplikace. Pokud ne, je vyzván ke znovuzadáání údajů. Heslo je v databázi uloženo jako binární kód, který je hešován kryptografickou funkcí SHA1.



Obr. 9 Úvodní přihlášení uživatele do aplikace

5.2.3 Manažer správy pracovních cest

Při přihlašování uživatele do aplikace a procházení uložených záznamů v databázi se kontrolují také uživatelská práva. Na základě těchto práv je provedena identifikace článku procesu tvorby a schvalování pracovní cesty (jeden ze čtyř uživatelů), a je mu zobrazen *manažer správy pracovních cest* s konkrétními formulářovými prvky.

Na obrázku níže je zobrazen formulář, který je určen pro zaměstnance vyslaného na pracovní cestu. Tento formulář je pro všechny účastníky procesu

tvorby a schvalování pracovní cesty téměř stejný. Jediný rozdíl je v tom, že zbylí tři účastníci (tj. vedoucí pracovník oddělení, pracovník pokladny a pracovník ekonomického úseku) mají ve formuláři další tlačítko. Toto tlačítko vykonává různé funkce opět s ohledem na práva jednotlivých uživatelů systému.

Na formuláři lze nalézt sekci s rekapitulací osobních údajů zaměstnance. Další sekcí jsou vlastní pracovní cesty. Ty se dělí na *Editovatelné* (to jsou ty se stavem EDIT) a *Needitovatelné* (to jsou všechny ostatní pracovní cesty). Pracovník tak přehledně vidí, v jakém stavu se jeho pracovní cesta nachází a na základě těchto stavů může vykonávat příslušné akce (třeba navštívit pokladnu kvůli vyplacení zálohy).

The screenshot shows a window titled 'Správce pracovních cest'. It is divided into two main sections: 'Osobní údaje' (Personal data) and 'Moje pracovní cesty' (My business trips).

Osobní údaje:

- 1. Příjmení, jméno, titul: Nováková, Josefína, Ing. Ph.D.
- 2. Bydliště: Bořetická 1, Brno, Česká republika
- 3. Osobní číslo: 1234
- 4. Útvar: -
- 5. Telefon, linka: +420 545 123 456

Moje pracovní cesty:

There are two tabs: 'Needitovatelné' (selected) and 'Editovatelné'. Below is a table of business trips:

Stav	Účel cesty	Začátek	Konec	Důvod zamítnutí
VED ...	Rekvalifikační kurz	7.1.2015	14.1.2015	
VED ...	Konference CDE	23.10.2015	23.10.2015	
VED ...	Pracovní výjezd	23.10.2015	26.10.2015	
VED ...	Konference ABC	2.11.2015	8.11.2015	

Below the table, there are several status codes and their meanings:

- VED - Čekající na schválení vedoucím.
- ZÁL - Nutno vyplatit zálohu. Navštivte pokladnu.
- EDIT - Čekající na doplnění údajů.
- EK - Čekající na přeučtování ekonomickým úsekem.
- POK - Nutno navštívit pokladnu
- X - Zamítnutá pracovní cesta.
- ✓ - Dokončená pracovní cesta.

At the bottom, there is a button labeled 'Nová pracovní cesta'.

Obr. 10 Manažer správy pracovních cest z pohledu zaměstnance

5.2.4 Nová pracovní cesta

Když uživatel klikne na tlačítko *nová pracovní cesta* v předchozím formuláři, objeví se mu formulář pro zadání základních údajů o pracovní cestě, který je zobrazen na obrázku níže.

Formulář obsahuje všechny potřebné formulářové prvky tak, aby věrohodně pokrýval fyzický cestovní příkaz poskytnutý podnikem. Ve formuláři tak lze nalézt sekci pro zadání počátku pracovní cesty (tj. místo, ze kterého zaměstnanec svoji pracovní cestu započne) a místa jednání (nebo-li cílová destinace pracovní cesty). Dále je nutno zadat datum konce pracovní cesty, účel cesty a spolucestujícího. Tento krok disponuje zatržítkem pro případ, že zaměstnanec na pracovní cestu vycestuje sám. Dále je důležité, aby zaměstnanec vyplnil výši zálohy, kterou považuje za adekvátní a určil primární dopravní prostředek.

Co se týče vlastního automobilu, formulář disponuje blokem, který se zobrazí v případě výběru tohoto typu dopravního prostředku. Zaměstnanec pak vyplní povinné údaje o svém automobilu a přiloží obrazový soubor technického průkazu. Po stisku tlačítka *Odeslat ke zpracování* je aplikací zkontrolován celý formulář (vyplnění povinných polí, validace formuláře, atd. – celý tento krok byl popsán již dříve) a údaje z něj jsou uloženy do databáze (včetně případné obrazové přílohy).

The screenshot shows a web form titled "Základní údaje o pracovní cestě" (Basic data about a business trip). The form is organized into several sections:

- Počátek cesty (adresa odjezdu z počáteční destinace)** (Start of the trip (address of departure from the starting destination)):
 - Ulice: Bořetická
 - Město: Brno
 - PSC: 62800
 - Stát: Česká republika
 - Číslo popisné: 1
 - Datum odjezdu: 21. prosince 2015
 - Čas odjezdu: 10:00
- Místo jednání (adresa cílové destinace)** (Meeting place (address of the target destination)):
 - Ulice: Sokolovská
 - Město: Praha 9
 - PSC: 19022
 - Stát: Česká republika
 - Číslo popisné: 217/42
- Datum konce cesty:** 22. prosince 2015
- Účel cesty:** Školení JavaScript/JQuery
- Spolucestující:** Kalina, Miroslav (with a checked box for "Bez spolucestujícího")
- Požadovaná záloha:** 1000 Kč
- Dopravní prostředek:** Auto vlastní
- Přidat vlastní automobil:**
 - Značka: Škoda
 - Model: Octavia
 - SPZ: 2BZ 2345
 - Palivo: Nafta
 - Spotřeba: 3.0
- Technický průkaz:** C:\Users\Gábi\Gábi\Desktop\IMG_8394.jpg (with a "Procházet" button)
- Poznámka:** Vypisujte údaje pravdivě dle technického průkazu, který přiložíte ve formátech jpg/jpeg, png či bmp.

At the bottom right, there is a button labeled "Odeslat ke zpracování" (Send for processing).

Obr. 11 Doplnění základních údajů o pracovní cestě

5.2.5 Kontrola základních údajů

Na obrázku níže lze pak vidět, jak vypadá formulář *nové pracovní cesty* v případě vedoucího zaměstnance oddělení. Takový formulář je pak naplněn údaji z databáze a obsahuje navíc možnost upravení výše zálohy, zobrazení přílohy (v případě vlastního automobilu, na základě přiloženého obrazového souboru technického průkazu vedoucí pracovník oddělení zkontroluje údaje o automobilu) a textové pole pro komentář v případě zamítnutí pracovní cesty. Zde je potřeba, aby vedoucí pracovník oddělení napsal důvody, díky kterým pracovní cestu zamítá (např. špatně uvedené údaje o automobilu nebo celkově změna rozhodnutí o uskutečnění pracovní cesty).

V případě zamítnutí pracovní cesty (stisk tlačítka *Zamítnout základní údaje*) je stav pracovní cesty nastaven na hodnotu Zamítnutá, při schválení pracovní cesty (stisk tlačítka *Schválit základní údaje*) je stav nastaven na Nutno vyplatit zálohu a zaměstnanec tak může navštívit pokladnu a odnést si svoji zálohu.

Základní údaje o pracovní cestě

Počátek cesty (adresa odjezdu z počáteční destinace)

Ulice: Bořetická Číslo popisné: 1
Město: Brno Datum odjezdu: 21. prosince 2015
PSČ: 62800 Čas odjezdu: 10:00:43
Stát: Česká republika

Místo jednání (adresa cílové destinace)

Ulice: Sokolovská Číslo popisné: 217/42
Město: Praha 9 PSČ: 19022
Stát: Česká republika

Datum konce cesty: 22. prosince 2015
Účel cesty: Školení JavaScript/JQuery
Spolucestující: -
Požadovaná záloha: 1000 Kč Upravit zálohu
Dopravní prostředek: Auto vlastní

Vlastní automobil:

Značka: Škoda Model: Octavia SPZ: 2BZ 2345
Palivo: Nafta Spotřeba: 3 Načíst přílohu

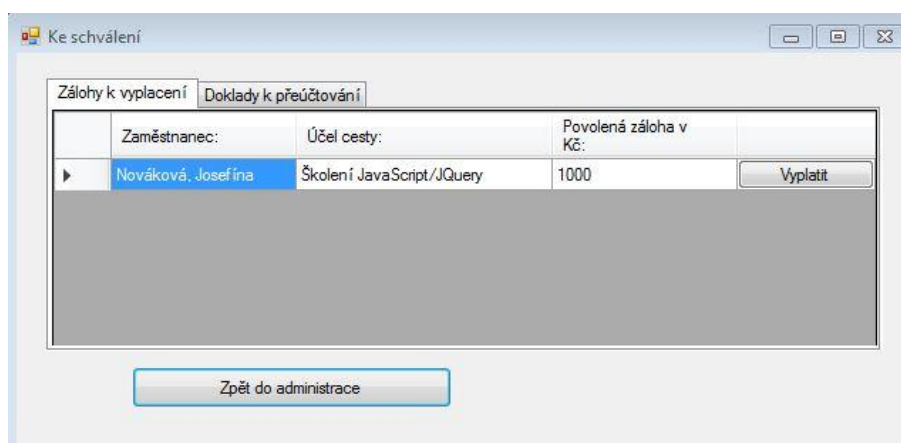
Do tohoto textového pole vepište důvod v případě, že hodláte správnost základních údajů zamítnout.

Zamítnout základní údaje Schválit základní údaje

Obr. 12 Kontrola základních údajů o pracovní cestě

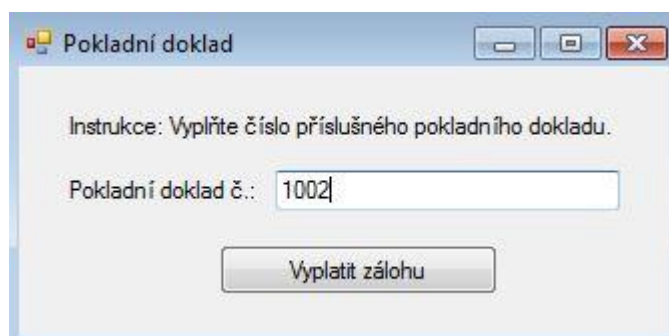
5.2.6 Vyplacení zálohy

Jakmile jsou všechny základní údaje zkontrolovány a stav pracovní cesty změněn, vyplatí pracovník pokladny zaměstnanci zálohu. Z manažera pracovních cest se skrze stisk tlačítka *Ke schválení* vyšle požadavek na zobrazení formuláře obsahujícího záložku *Zálohy k vyplacení* (viz obrázek níže) a *Doklady k přeúčtování*.



Obr. 13 Vyplacení zálohy na pracovní cestu

Skrze stisk tlačítka *Vyplatit* u konkrétní zálohy dojde k vyvolání formuláře pro vyplnění čísla pokladního dokladu.

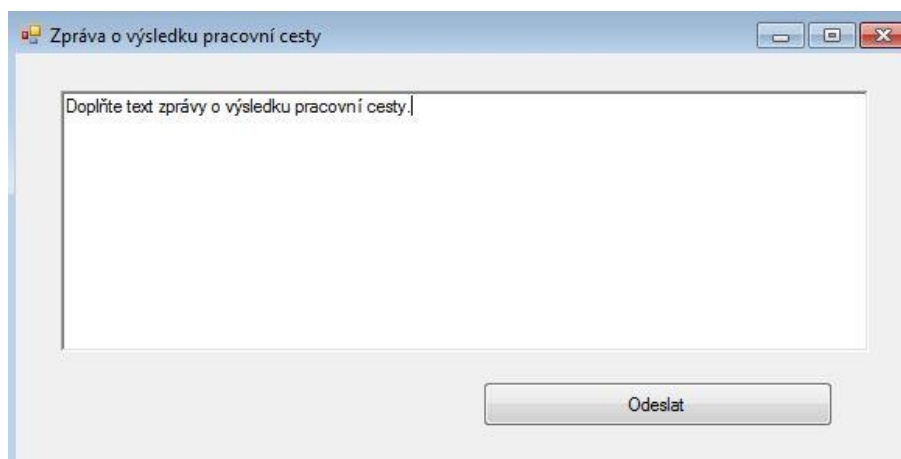


Obr. 14 Vytvoření pokladního dokladu

Do textového pole musí pracovník pokladny vyplnit číslo fyzického pokladního dokladu, na jehož základě bude vyplacena záloha. Jakmile pracovník vyplní povinný údaj, stiskem tlačítka *Vyplatit zálohu* vyšle požadavek na změnu stavu u pracovní cesty, ta se tak bude nacházet ve stavu Čekající na doplnění zbývajících údajů. Poté bude zaměstnanci fyzicky vyplacena záloha a zaměstnanec může vycestovat na pracovní cestu.

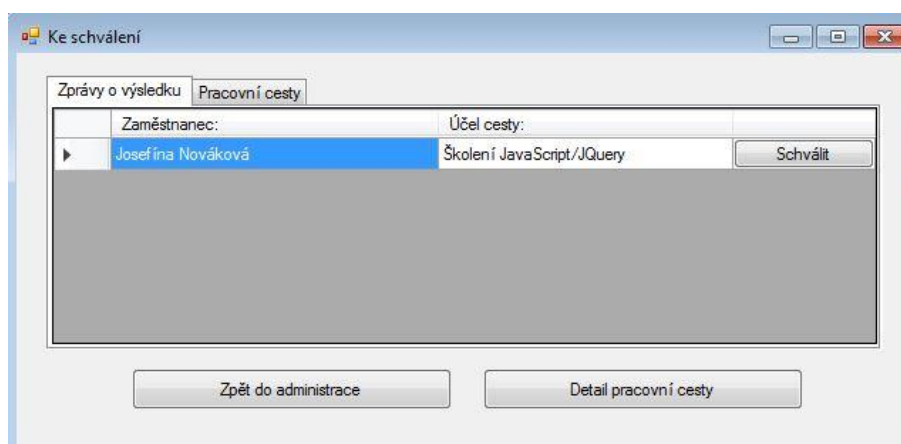
5.2.7 Zpráva o výsledku pracovní cesty

Po návratu z pracovní cesty se zaměstnanec znovu přihlásí do správy pracovních cest a skrze úvodního manažera pracovních cest zvolí tu konkrétní cestu, ze které se vrátil, aby k ní mohl sepsat zprávu o výsledku pracovní cesty. Zobrazí se mu jednoduchý formulář s textovým polem (viz obrázek níže), do kterého vyplní text zprávy a stiskem tlačítka *Odeslat* odešle požadavek na uložení zprávy do databáze. Aplikace pak tento požadavek zpracuje.



Obr. 15 Zpráva o výsledku pracovní cesty

V dalším kroku je zpráva zobrazena vedoucímu pracovníkovi oddělení v sekci *Ke schválení* a ten ji stiskem tlačítka *Schválit* schválí. Pracovní cestě tak bude přiřazen stav *Čeká na doplnění údajů*. Jak lze vidět na obrázku níže, vedoucí pracovník oddělení má v sekci *Ke schválení* taktéž záložku *Pracovní cesty*, přes které probíhá právě kontrola základních údajů (o této fázi bylo pojednáváno již dříve).



Obr. 16 Zprávy o výsledku pracovní cesty

5.2.8 Vyúčtování pracovní cesty

V této fázi je již zaměstnanec oprávněn doplnit k pracovní cestě veškeré potřebné informace. Přes záložku editovatelných cest v manažerovi pracovních cest a stiskem tlačítka *Doplnit* u konkrétní pracovní cesty vyvolá zobrazení formuláře pro zadávání potřebných údajů. Formulář můžete vidět na obrázku níže, kde jsou však již zobrazeny formulářové prvky viditelné vedoucímu pracovníkovi oddělení a pracovníkovi ekonomického úseku. Formulář zaměstnance je téměř stejný s tím rozdílem, že sekce *Použitý vlastní automobil* není viditelná a text na levém tlačítku je *Připojit doklady* (viz obrázek níže).

Instrukce
 Vyplňte všechny údaje úplně a pravdivě. Odjezd a příjezd vyplňte včetně přechodu hranic. Doba odjezdu a příjezdu vyplňte podle jízdního řádu.
 Počet ujetých km uvádějte jen v případě použití vlastního dopravního prostředku. Na závěr přiložte doklady ve formě obrazových souborů.
 DATUMY: Přecontrolujte všechny datумы tak, aby odpovídaly jak rozsahu pracovní cesty (dny mezi počátkem a koncem cesty včetně), tak skutečnosti.
 ČASY: Přecontrolujte všechny časové údaje tak, aby odpovídaly jak dříve zadaným údajům, tak realitě.
 Údaje s hvězdičkou jsou povinné.
 TIP: Položky Ujeté km v Kč, spotřeba PHM v Kč a Celkem se dopočítávají automaticky.

Zaškrtněte v případě, že je Vaše odpověď ANO.

Stravování bylo poskytnuto bezplatně. Ubytování bylo poskytnuto bezplatně. Volná - zlevněná jízdenka. Pobírám odlučné.

	Datum*	Odjezd*	v hod.*	Příjezd*	v hod.*	Použitý dopravní prostředek *	Ujeté km	Ujeté km v Kč	Spotřeba PHM v litrech	Spotřeba PHM v Kč	Počáteční pracovní výkon
	21.12.2015	Bmo	10:00:00	Praha	13:00:00	Auto vlastní	210,0	819,00...	7,0	252,00 Kč	15:00:00
▶	22.12.2015	Praha	08:00:00	Bmo	11:00:00	Auto vlastní	205,0	799,50...	6,0	216,00 Kč	00:00:00

Použitý vlastní automobil:

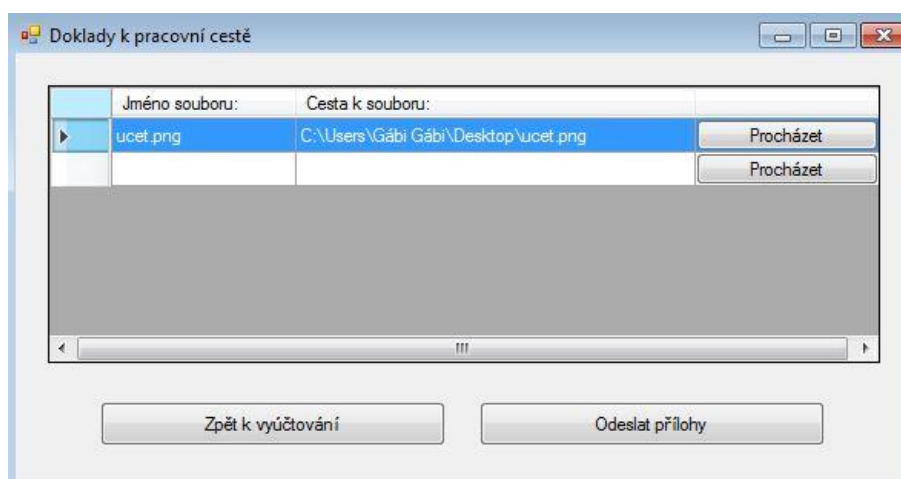
SPZ: 2BZ 2345 Spotřeba: 31 / 100 km Palivo: Nafta Cena: 36 Kč / l

Prohlašuji, že jsem všechny údaje přecontroloval/a.

Obr. 17 Vyúčtování pracovní cesty

Samotný formulář obsahuje dosti podrobné instrukce pro vyplnění, aby odpadla potřeba dodatečného zásahu lidského faktoru v podobě manuálních úprav záznamů v databázi. Tyto instrukce jsou důležité také z toho důvodu, že při vyplňování údajů aplikace průběžně kontroluje, zda jsou údaje validní, a v případě zadání špatných údajů zvýrazňuje chyby (popřípadě zobrazuje chybová hlášení). Kdyby formulář neobsahoval žádné instrukce, zaměstnanec by pravděpodobně nevěděl, kde udělal chybu a nemohl by tak doplnění údajů úspěšně dokončit.

Další důležitou částí je tabulka pro vepisování samotných údajů. Tyto údaje věrně kopírují stávající podobu poskytnutého cestovního příkazu, a na základě zadaných údajů provádí propočty (např. výpočet nákladů za pohonné hmoty, součet celkových nákladů, atd.). Přes tlačítko *Připojit doklady* zaměstnanec vyvolá formulář pro nahrávání dokladů (obrazové soubory), které během své pracovní cesty nabyt (viz obrázek níže).



Obr. 18 Výběr dokladů nabytých během pracovní cesty

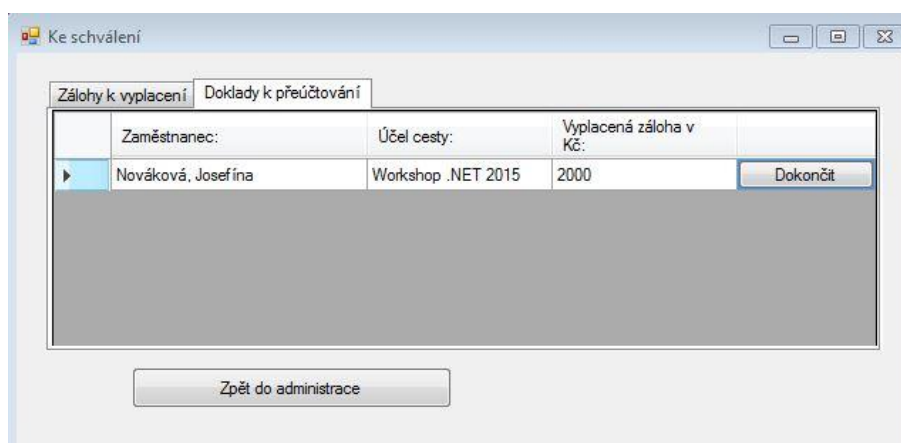
Tyto soubory jsou uloženy do databáze a během tohoto procesu je jim přiřazeno ID pracovní cesty, pro kterou je vyúčtování vytvářeno. Následným stiskem tlačítka *Odeslat ke zpracování* vyšle zaměstnanec požadavek na vložení vyúčtování do databáze. Během této fáze proběhne další potřebná kontrola formuláře a v případě, že je všechno v pořádku, vyúčtování jsou uložena do databáze. Stav pracovní cesty je poté změněn na čekající na schválení vedoucím (VED).

Jakmile je vyúčtování (respektive jednotlivé řádky) vloženo do databáze a pracovní cestě je přiřazen správný stav, je toto vyúčtování přístupné nejprve vedoucímu pracovníkovi oddělení a po jeho kontrole taktéž pracovníkovi ekonomického úseku. Jelikož kontrola údajů vedoucím pracovníkem oddělení je spíše formalita, kdy jde hlavně o změnu stavu pracovní cesty na Čekající na schválení vedoucím, nebude tomuto mezikroku přiřazována vyšší priorita a bude přístupeno ke kontrole pracovníkem ekonomického úseku. Ten svým požadavkem vyvolá formulář (který je zobrazen výše) a zkontroluje správnost všech údajů s ohledem na příslušný zákon Zákonníku práce.

Kontrola pracovníkem ekonomického úseku je nutná, neboť bylo s podnikem dohodnuto, že vzhledem k často se měnícím sazbám v zákonech bude jednodušší provádět kontrolu ručně. Při této kontrole je pracovník ekonomického úseku povinen zobrazit veškeré přiložené doklady a na jejich základě provést případnou úpravu stávajících vypočítaných částek. Jakmile si je pracovník svojí kontrolou jistý, zaškrtně prohlášení a stiskem tlačítka *Odeslat úpravu vyúčtování* vyšle požadavek na úpravu vyúčtování v databázi. Pracovní cestě je pak přiřazen stav Nutno navštívit pokladnu.

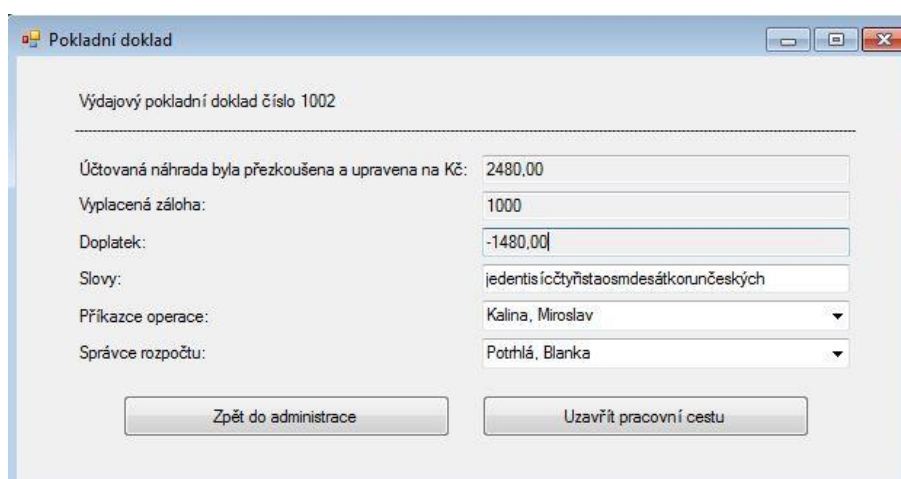
5.2.9 Doplnění pokladního dokladu

Tento krok je závěrečnou fází celého procesu tvorby a schvalování pracovní cesty. Pracovník pokladny se přes manažera pracovních cest a tlačítko *Ke schválení* dostane k seznamu dokladů, které je potřeba přeúčtovat.



Obr. 19 Pokladní doklady k přeúčtování

Výběrem dokladu odpovídajícího konkrétní pracovní cestě a stiskem tlačítka *Dokončit* vyše pracovník pokladny požadavek na zobrazení editačního formuláře (viz obrázek níže).



Obr. 20 Doplnění pokladního dokladu

V tomto formuláři se nacházejí údaje plynoucí z přeúčtování pracovní cesty ekonomickým úsekem. Konkrétně jsou v něm obsaženy údaje o upravené náhradě v Kč, o vyplacené záloze v Kč a údaje o doplatku či přeplatku. Pokud je hodnota upraveného přeúčtování vyšší než vyplacená záloha, vzniká doplatek a pokladna tak musí rozdíl vyplatit zaměstnanci. V opačném případě vzniká přeplatek na záloze a pokladna přijímá finanční částku od zaměstnance.

Ve formuláři je dále potřeba vypsát slovy hodnotu doplatku/přeplatku a vyplnit správce rozpočtu a příkazce operace. Stiskem tlačítka *Uzavřít pracovní cestu* pak pracovník cesty vyše požadavek na změnu stavu u pracovní cesty, která se tak dostane do fáze kompletního dokončení. Stisk tlačítka dále vyvolá

vygenerování cestovního příkazu, který je vyplněn všemi údaji, které byly v rámci celého procesu tvorby a schvalování pracovní cesty uloženy do databáze.

5.3 Testování aplikace

Výsledná aplikace měla být po implementaci nasazena a testována v podniku, nicméně ve chvíli dokončování této diplomové práce byly projednávány dodatečné úpravy a změny v implementaci, které však nejsou nikterak výrazné, aby ovlivnily celou práci. I přesto však aplikace do provozu nasazena nebyla a tudíž nemohla být řádně otestována.

5.4 Shrnutí kapitoly

Praktická část práce se zabývá návrhem diagramů UML a implementací aplikace pro správu pracovních cest dle požadavků podniku, pro který byla realizována. V této kapitole tak bylo ukázáno, jakým způsobem probíhal celý návrh aplikace a že byl pro tuto část použit diagram případů užití, diagram tříd, sekvenční diagram, stavový diagram a diagram aktivit. Na hotových diagramech bylo předvedeno, jakým způsobem lze grafickou notaci UML využít. V další části kapitoly byla krok po kroku vysvětlena funkčnost a implementace celé aplikace včetně náhledů aplikace.

6 Diskuze

Tato diskuzní část diplomové práce se bude zabývat pouze implementací výsledné aplikace pro správu pracovních cest. Je to z toho důvodu, že návrh UML diagramů byl čistě informativního charakteru pro potřeby podniku a implementace.

6.1 Vlastní řešení vs. dostupná řešení

V analytické části práce bylo představeno několik dostupných (ne)komerčních řešení problematiky pracovních cest. Bylo zjištěno, že aplikace, která by sama o sobě dokázala fungovat jako správa pracovních cest, není žádná (nebo alespoň v tuto chvíli tomu tak je). Všichni vybraní zástupci popsaní v analytické části jsou modulového charakteru či součástí velkých ekonomických systémů. Taková řešení je tedy nutné připojit ke stávajícímu firemnímu informačnímu systému, anebo zakoupit celý nový informační systém, který má tuto doménu zabudovanou již v sobě. Aplikace vyvíjená v rámci diplomové práce není modulového charakteru a je tedy schopna samostatného provozu. Jediné, co potřebuje, je napojení na stávající firemní databázi, nikoliv na celý informační systém. Z toho důvodu je poměrně obtížné srovnávat dva charakterově odlišné produkty, ale není to nemožné.

Z průzkumu dostupných řešení vyplynulo jedno – všechny společnosti, které se zabývají ekonomickou problematikou, mají tendenci poskytnout zákazníkovi (uživateli) co nejvyšší uživatelskou přívětivost. Z toho důvodu jsou všechna řešení formulářového charakteru. Jinak tomu není ani v případě aplikace pro správu pracovních cest v této diplomové práci. I přesto, že se jednalo o požadavek podniku lze konstatovat, že i bez tohoto požadavku by aplikace byla formulářového charakteru.

Další, mnohokrát se opakující výhodou u dostupných řešení, byla co nejvyšší automatizace celého procesu tvorby a schvalování pracovní cesty. Lze říci, že tento bod splnila všechna dostupná řešení. Automatizace procesu šetří obsluhujícím zaměstnancům čas a tudíž i finanční náklady celému podniku. A snížení nákladů podniku je hlavní lákadlo, přes které se jednotlivé společnosti snaží získat zákazníky pro své produkty. I tento požadavek aplikace pro správu pracovních cest splňuje. A stejně jako v předchozím bodě i zde lze konstatovat, že aplikace by se snažila o co nejvyšší míru automatizace i bez požadavku ze strany podniku – bez automatizace by se jednalo jen o jakýsi interaktivní formulář, který by tak postrádal smysl.

Neopomenutelnou funkcionalitou byl také přepočítání nákladových složek. Většina dostupných řešení slibovala plně automatizovaný přepočítání nákladových složek, některé byly dokonce napojeny na příslušné státní systémy pro co nejvyšší přesnost. Jiná řešení slibovala pouze částečnou automatizaci přepočtu, lze ji tedy pochopit jako přepočítání s kontrolou. Touto formou disponuje i aplikace pro správu pracovních cest. Během komunikace s podnikem byl totiž původní požadavek na

úplnou automatizaci upraven a domluvilo se, že bude vhodnější vydat se cestou přepočtu s kontrolou.

Samozřejmostí většiny zmíněných dostupných řešení byla evidence vozového parku a napojení na knihu jízd. Zde už se projevuje charakter modulového řešení, který využívá dostupné funkce stávajících účetních a ekonomických softwarů. Jelikož je aplikace pro správu pracovních cest samostatně zcela funkční a tudíž i prozatímne nenapojitelná na stávající systémy v podniku, tento bod neplní. Neobsahuje knihu jízd a neobsahuje ani evidenci vozidel. Vozidla lze skrze aplikaci pouze ukládat do databáze, ale už není možné kontrolovat veškeré potřebné účetní či ekonomické údaje.

Samotná aplikace je unikátní v tom, že je šitá na míru, tudíž splňuje všechny požadavky podniku a odpovídá přesně organizační struktuře a celému procesu tvorby a schvalování pracovní cesty.

6.2 Výhody a nevýhody vlastního řešení

Tato podkapitola navazuje na srovnání s dostupnými řešeními. Je to z toho důvodu, že výhody a nevýhody budou určeny na základě funkcionalit těchto dostupných řešení a podle subjektivního pocitu.

6.2.1 Výhody

Bezesporu největší výhodou systému je jeho provedení na míru podniku, pro který byl vytvářen. Systém přesně kopíruje požadavky domluvené s podnikem a také organizační strukturu podniku, jeho směrnice, apod. K tomu se váže také výhoda v podobě připravenosti aplikace na komunikaci se stávající databází.

Další výhodou je automatizovaný proces tvorby a schvalování pracovních cest, kdy zaměstnanec vyslaný na pracovní cestu nemusí složitě navštěvovat všechny účastníky tohoto procesu, ale bohatě mu postačí komunikovat s aplikací. Ta ho sama upozorní na případné potřebné kroky, neboť jsou k dispozici mailové notifikace, popřípadě přehledný manažer pracovních cest.

Uživatelé pracující v podniku zajisté ocení uživatelskou přívětivost zajištěnou skrze řešení formulářového charakteru. I přesto, že se jednalo o požadavek podniku, je to jedna z výhod aplikace. Ruku v ruce s uživatelskou přívětivostí jde i jednoduše a přehlednost celé aplikace.

Velkou výhodou systému je jeho všudypřítomná kontrola. Systém v každém kroku kontroluje, co uživatel v aplikaci provádí, a v případě chybných kroků uživatele na tuto skutečnost ihned upozorní v podobě chybových hlášení. Tato výhoda se váže k uživatelské přívětivosti, neboť má za úkol uživateli co nejvíce usnadnit práci se samotnou aplikací.

Za výhodou se dá označit také řešení v podobě vícevrstvé architektury. Takto je aplikace schopna komunikovat s databází uloženou na vzdáleném podnikovém serveru, tudíž veškeré změny provedené v databázi uživatel uvidí téměř ihned. V případě ne klient-server řešení by každý stroj musel mít svoji vlastní lokální

databázi, která by jednou za čas uploadovala změny na server, což by znamenalo prodlevy a možné kolize.

Aplikace po dokončení procesu tvorby a schvalování pracovní cesty vytvoří vyplněný cestovní příkaz a exportuje ho do formátu PDF, který je v dnešní době velice populární a dá se tak označit za standardní. Toto řešení je pohodlné a bezesporu výhodné.

Aplikace jako taková je vysoce komplexní, což by se dalo označit za další výhodu. Skrze tuto položku dokáže podniku šetřit jak čas, tak finanční náklady.

6.2.2 Nevýhody

Nevýhodou aplikace je to, co ji odlišuje od ostatních – její stávající aplikační řešení, které způsobuje, že není schopná komunikovat s podnikovými informačními systémy. Z toho důvodu není možné skrze aplikaci vyúčtovat jednotlivé pracovní cesty či je automaticky propojit s pokladními doklady, které jsou uloženy ve stávajícím účetním systému (vše se musí provést ručně).

Z této nevýhody vyplývá například také absence propojení aplikace s knihou jízd, a (jak již bylo zmíněno v dřívější podkapitole) také neschopnost zjistit více informací o ukládaných vozidlech (stav opotřebení, atd.) či stávajícím vozovém parku.

Propočet s kontrolou, který byl domluven během návrhu aplikace, se dá také označit za jistou nevýhodu. Ačkoliv je toto řešení implementačně méně náročné, stále je potřeba přítomnost fyzického pracovníka, který veškeré položky podle jednotlivých zákonů překontroluje a případně upraví.

Jistou nevýhodou je i způsob nahrávání obrazových souborů do databáze. Aktuální stav umožňuje nahrávání již existujících souborů, ale pro zjednodušení by bylo vhodné implementovat také skenovací funkci, kdy by uživatel nemusel dokumenty nejprve skenovat a potom je nahrávat skrze aplikaci na server.

Za nevýhodu by mohla být považována i absence cizí měny, která by byla využitelná hlavně v rámci zahraničních pracovních cest.

6.3 Zhodnocení vlastního řešení a návrh vylepšení

Aplikace v tuto chvíli splňuje téměř všechny požadavky podniku, nicméně i přes drobné nedostatky je již zcela funkční. V další fázi bude podrobena testování funkcionality nasazením do provozu. Na základě testování pak budou prováděny veškeré další úpravy a implementace dodatečných požadavků.

Mezi tyto další úpravy a požadavky bude pravděpodobně patřit eliminace nevýhod. Z vyjmenovaných nedostatků se nejspíše jeví implementace skenovací funkce, cizí měny a plně automatizovaný proces přepočtu nákladových složek. Nejkomplikovanější pak pravděpodobně bude propojení se stávajícími informačními systémy, neboť bude potřeba získat příslušná práva, prozkoumat implementační řešení těchto systémů, porozumět jim a poté navrhnout a implementovat propojení.

I přesto, že budou odstraněny nevýhody aplikace, lze přemýšlet nad dalšími možnými vylepšeními aplikace. Pravděpodobně nejpřínosnějším vylepšením by mohlo být propojení s některým mapovým serverem (Google Maps, Mapy.cz, atd.), aby docházelo k automatickému propočtu trasy v případě použití vlastního či firemního automobilu a jejímu zaznamenání do cestovního příkazu. V takovém případě by pak zaměstnanec pouze zkontroloval, zda trasa odpovídala realitě a případně provedl změny.

Dalším možným vylepšením aplikace by mohlo být propojení s jízdními řády, kdy by aplikace na základě vybraného primárního dopravního prostředku zjistila potřebné spoje a opět je zanesla do cestovního příkazu.

7 Závěr

Diplomová práce se zabývala návrhem a implementací aplikace pro správu pracovních cest a s ní souvisejícími teoretickými východisky.

V kapitole Teoretická východiska práce byla rozebrána problematika pracovních cest z hlediska ekonomického s ohledem na příslušný zákon Zákoníku práce. V kapitole byla dále řešena problematika návrhu informačních systémů. Z toho důvodu bylo nutno představit architektury informačních systémů a přístupy k návrhu informačních systémů spolu s příslušnými grafickými notacemi, kdy na základě vybraného přístupu pak byla konkrétní grafická notace podrobněji objasněna. Stejným způsobem byla rozebrána problematika tvorby informačních systémů, kdy byli popsáni softwaroví klienti, kteří s touto problematikou souvisejí a také nejpoužívanější implementační technologie.

V analytické části práce (respektive v kapitole Analytická část) pak byl proveden průzkum současného trhu a byli popsáni vybraní zástupci dostupných řešení stejné problematiky, která je řešena v práci. Snahou tedy bylo nalézt taková řešení, která se taktéž zabývala problematikou pracovních cest. Poté byl stručně charakterizován podnik, pro který byla realizována praktická část práce a sepsány veškeré požadavky, které podnik na implementovanou aplikaci měl.

V praktické části práce pak došlo k popisu návrhu diagramů v notaci UML, které měly spíše informativní charakter, neboť sloužily primárně pro potřeby podniku. Nicméně samotné diagramy UML jsou vhodnou ukázkou toho, jakým způsobem lze tuto grafickou notaci při návrhu využít. Po objasnění návrhu UML diagramů bylo přistoupeno k popisu návrhu databáze a objasnění funkcionality celé aplikace. Stěžejní tak zůstávaly náhledy aplikace, které vytvářely lepší vizuální představu popisované funkcionality.

Na závěr práce bylo provedeno celkové zhodnocení aplikace v porovnání s dostupnými řešeními. Dále byly definovány výhody a nevýhody aplikace, ze kterých vycházelo finální zamyšlení se nad možným vylepšením aplikace do budoucna. V rámci práce se podařilo splnit veškeré cíle a naplnit téměř všechny zásady pro vypracování (kromě fáze testování, která se nemohla před odevzdáním práce uskutečnit). Navrhovaná a implementovaná aplikace pro správu pracovních cest je v tomto momentě plně funkční, a i přesto, že se nachází ve stavu úprav, by mohla být nasazena do ostrého provozu (s čímž se po implementaci drobných úprav a testování počítá).

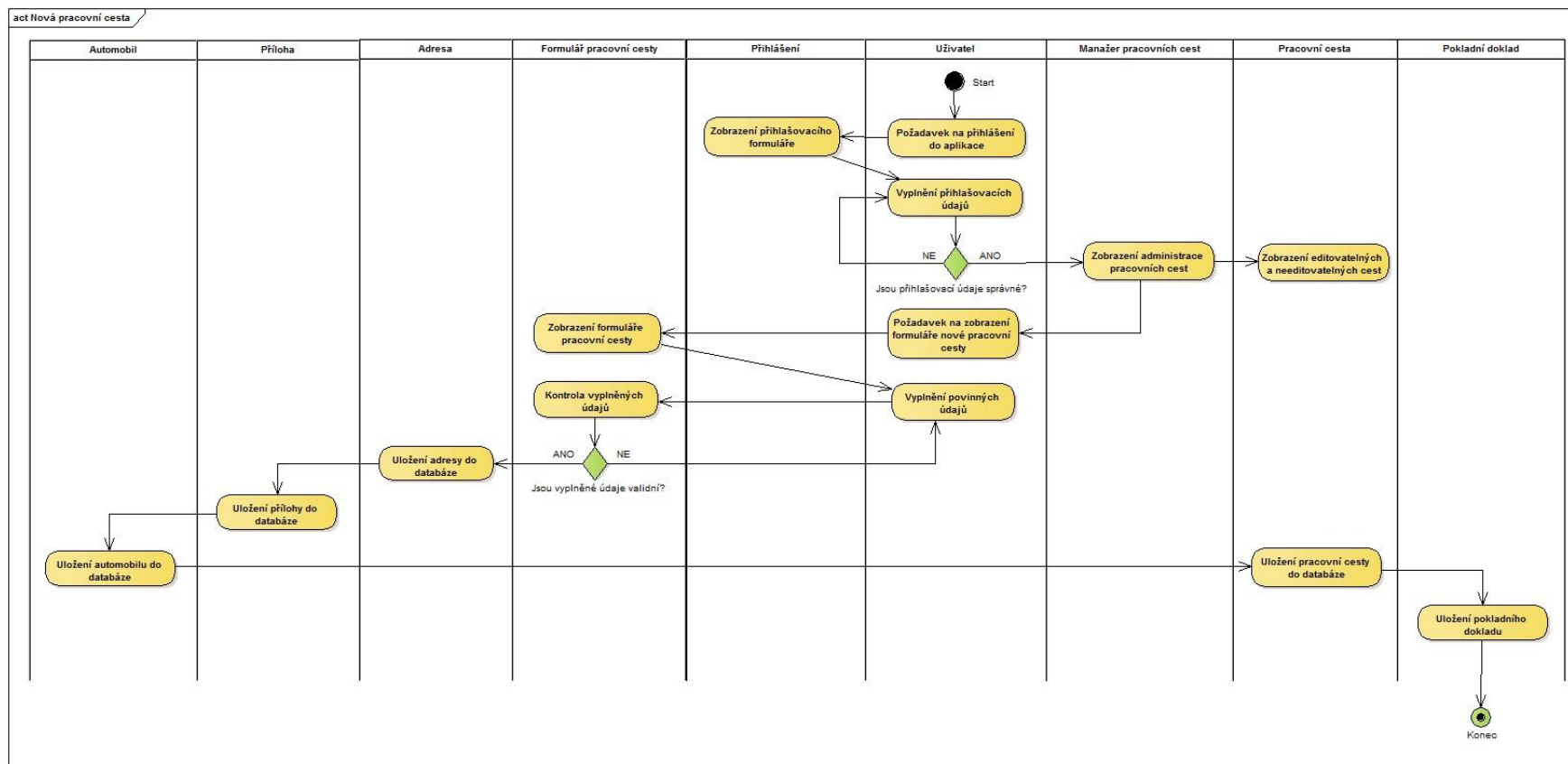
8 Literatura

- ANDRAŠČÍKOVÁ, Mária, Pavla HLOUŠKOVÁ, Eva HOFMANNOVÁ, Pavel KNEBL, Zdeněk SCHMIED, Ladislav TRYLČ a Ludmila TOMANDLOVÁ. Zákoník práce.: Prováděcí nařízení vlády a další související případy. S komentářem k 1. 1. 2015. 9. aktualizované vydání. Olomouc: Anag, 2015. ISBN 978-80-7263-921-2.
- BRUCKNER, Tomáš, Jiří VOŘÍŠEK, Alena BUCHALCEVOVÁ, Iva STANOVSKÁ, Dušan CHLAPEK a Václav ŘEPA. Tvorba informačních systémů: principy, metodiky, architektury. První vydání. Praha: Grada Publishing, 2012, 357 stran. Management v informační společnosti. ISBN 978-80-247-4153-6.
- Cesty: Cesty ver.3.0 Perm. Kvasar, spol. s r. o. [online]. 2015 [cit. 2015-12-08]. Dostupné z: <http://www.kvasar.cz/24732-cesty>
- CHRISTENSSON, Per. JavaScript Definition. TechTerms [online]. 2014 [cit. 2015-12-03]. Dostupné z: <http://techterms.com/definition/javascript>
- CONVERSE, Tim a Joyce PARK. PHP5 and MySQL bible. [3rd ed.]. Indianapolis, IN: Wiley, 2004, xxxvii, 1042 p.
- ČERMÁK, Miroslav. Vícevrstvá architektura: tenký, tlustý a chytrý klient. CleverAndSmart [online]. 2010, 11.02.2012 [cit. 2015-12-05]. Dostupné z: <http://www.cleverandsmart.cz/vicestva-architektura-tenky-tlusty-a-chytry-klient/>
- ČERMÁK, Miroslav. Vícevrstvá architektura: výhody a nevýhody. CleverAndSmart [online]. 2010, 30.11.2010 [cit. 2015-12-05]. Dostupné z: <http://www.cleverandsmart.cz/vicestva-architektura-vyhody-a-nevyhody/>
- ENG, P. a Rob MCCORMACK. MySQL definition. TechTarget: SearchEnterpriseLinux [online]. 2013 [cit. 2015-12-28]. Dostupné z: <http://searchenterpriselinix.techtarget.com/definition/MySQL>
- Enterprise Architect: Ultimate Modeling Power. SparxSystems [online]. 2015 [cit. 2015-12-28]. Dostupné z: <http://www.sparxsystems.com.au/products/ea/index.html>
- FOWLER, Martin. Destilované UML. 1. vyd. Praha: Grada, 2009, 173 s. Knihovna programátora (Grada). ISBN 978-80-247-2062-3.
- GREY, Jeff a Michael HAMBLIN. Python Definition. TechTarget: SearchEnterpriseLinux [online]. 2005 [cit. 2015-12-03]. Dostupné z: <http://searchenterpriselinix.techtarget.com/definition/Python>
- HELIOS Green: Moderní ERP systém pro středně velké a velké firmy. HELIOS - podnikový informační systém, ekonomický a účetní software, systém pro veřejnou správu [online]. 2015 [cit. 2015-12-08]. Dostupné z: <http://www.helios.eu/produkty/helios-green/>

- HEROUT, Pavel. Učebnice jazyka Java. 5., rozš. vyd. České Budějovice: Kopp, 2010, 386 s. ISBN 978-80-7232-398-2.
- Informační systém Vema. Podnikové informační systémy - ERP systémy - Vema a.s. [online]. 2015 [cit. 2015-12-08]. Dostupné z: <http://www.vema.cz/hr-system/>
- Kniha jízd a cestovní příkazy: Cestovní příkazy. POHODA - ekonomický a informační systém [online]. 2015 [cit. 2015-12-08]. Dostupné z: http://www.stormware.cz/prirucka-uctujeme-online/Kniha_jezd/Cestovni_prikazy/
- Microsoft .NET Framework. Microsoft [online]. 2015 [cit. 2015-12-08]. Dostupné z: <https://www.microsoft.com/cs-cz/download/details.aspx?id=17851>
- Modul: Cestovní příkazy. TCL DigiTrade [online]. 2015 [cit. 2015-12-08]. Dostupné z: <http://www.tcl-digitrade.cz/produkty/wheel-crm-dms/moduly/cestovni-prikazy/>
- O nás: Základní charakteristika společnosti. Kvasar, spol. s r. o. [online]. 2015 [cit. 2015-12-08]. Dostupné z: <http://www.kvasar.cz/24735-o-nas>
- O nás: Vše o naší firmě. TCL DigiTrade [online]. 2015 [cit. 2015-12-08]. Dostupné z: <http://www.tcl-digitrade.cz/o-nas/>
- O společnosti. *Podnikové informační systémy - ERP systémy - Vema a.s.* [online]. 2015 [cit. 2015-12-08]. Dostupné z: <http://www.vema.cz/o-spolecnosti/>
- O společnosti. POHODA - ekonomický a informační systém [online]. 2015 [cit. 2015-12-08]. Dostupné z: <http://www.stormware.cz/kontakty/>
- PAVUS, M. Collaboration diagram - diagram spolupráce. OO, UML, analýza, metodologie [online]. 2005 [cit. 2015-12-08]. Dostupné z: <http://mpavus.wz.cz/uml/uml-collaboration-8.php>
- PAVUS, M. Component diagram - diagram komponent. OO, UML, analýza, metodologie [online]. 2006 [cit. 2015-12-28]. Dostupné z: <http://mpavus.wz.cz/uml/uml-s-component-3-3-2.php>
- PAVUS, M. Timing diagram - diagram časování. OO, UML, analýza, metodologie [online]. 2005 [cit. 2015-12-08]. Dostupné z: <http://mpavus.wz.cz/uml/uml-b-timing-3-2-7.php>
- PIALORSI, Paolo a Marco RUSSO. Microsoft LINQ: kompletní průvodce programátora. Vyd. 1. Brno: Computer Press, 2009, 615 s. Programování (Computer Press). ISBN 978-80-251-2735-3.
- Pracovní cesty. plusPortal [online]. 2015 [cit. 2015-12-08]. Dostupné z: <http://www.plusportal.cz/moduly/pracovni-cesty>
- RÁBOVÁ, Ivana. Podnikové informační systémy a technologie jejich vývoje. V Tribun EU vyd. 1. Brno: Tribun EU, 2008, 139 s. ISBN 978-80-7399-599-7.
- ROUSE, Margaret. C# definition. TechTarget: SearchWinDevelopment.com [online]. 2007 [cit. 2015-12-05]. Dostupné z: <http://searchwindevelopment.techtarget.com/definition/C>

- ROUSE, Margaret. Perl definition. TechTarget: SearchWinDevelopment.com [online]. 2005 [cit. 2015-12-03]. Dostupné z: <http://searchenterpriselinux.techtarget.com/definition/Perl>
- ROUSE, Margaret. Ruby. TechTarget: WhatIs [online]. 2010 [cit. 2015-12-03]. Dostupné z: <http://whatis.techtarget.com/definition/Ruby>
- RYBIČKA, Jiří a Petra ČAČKOVÁ. Informatika pro ekonomy. 1. vyd. Praha: Alfa Nakladatelství, 2009, 251 s. Informatika (Alfa Nakladatelství). ISBN 978-80-87197-24-0.
- Společnost Asseco Solutions. HELIOS - podnikový informační systém, ekonomický a účetní software, systém pro veřejnou správu [online]. 2015 [cit. 2015-12-08]. Dostupné z: <http://www.helios.eu/o-nas/>
- VIRIUS, Miroslav. Jazyky C a C++: kompletní průvodce. 2., aktualiz. vyd. Praha: Grada, 2011, 367 s. Knihovna programátora (Grada). ISBN 978-80-247-3917-5.
- VYMĚTAL, Dominik. Informační systémy v podnicích: teorie a praxe projektování. 1. vyd. Praha: Grada, 2009, 142 s. Průvodce (Grada). ISBN 978-80-247-3046-2.

Přílohy



Obr. 23 Diagram aktivit pro založení nové pracovní cesty (vyšší rozlišení)