

**Univerzita Pardubice**  
**Fakulta ekonomicko-správní**  
**Ústav systémového inženýrství a informatiky**

**Možnosti uplatnění principu elektronického obchodu**  
**v rámci veřejné správy**

**Milan Mládek**

**Diplomová práce**  
**2012**

Univerzita Pardubice  
Fakulta ekonomicko-správní  
Akademický rok: 2011/2012

**ZADÁNÍ DIPLOMOVÉ PRÁCE**  
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Milan MLÁDEK**  
Osobní číslo: **E10242**  
Studijní program: **N6209 Systémové inženýrství a informatika**  
Studijní obor: **Informatika ve veřejné správě**  
Název tématu: **Možnosti uplatnění principu elektronického obchodu  
v rámci veřejné správy**  
Zadávající katedra: **Ústav systémového inženýrství a informatiky**

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

Cílem diplomové práce je analýza možností využití principu elektronického obchodování na portálech veřejné správy a návrh modelu pro vybraný subjekt veřejné správy.

Rozsah grafických prací:  
Rozsah pracovní zprávy: **cca 55 stran**  
Forma zpracování diplomové práce: **tištěná/elektronická**  
Seznam odborné literatury:


**PELC, Vladimír.** *Místní poplatky: Úplné znění zákona o místních poplatcích s poznámkami.* Praha: Linde, 2008. 303 s. ISBN 80-7201-691-4.

**OCHRANA, František.** *Veřejné služby - jejich poskytování, zadávání a hodnocení: teorie a metodika poptávkového způsobu poskytování a zadávání veřejných služeb na úrovni municipalit.* Praha: Ekopress, 2007. 167 s. ISBN 978-80-86929-31-6.

**WALTERS, Robert, et al.** *Mistrovství v Microsoft SQL Server 2008.* Praha: Computer Press, 2009. 864 s. ISBN 978-80-251-2329-4.

**NAGEL, Christian, et al.** *C 2008: Programuje profesionálně.* Praha: Computer Press, 2009. 1081 s. ISBN 978-80-251-2401-7.

Vedoucí diplomové práce:

  
**Ing. Stanislava Šimonová, Ph.D.**


Ústav systémového inženýrství a informatiky

Datum zadání diplomové práce:

**3. října 2011**

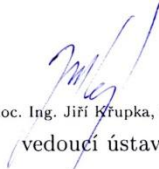
Termín odevzdání diplomové práce:

**30. dubna 2012**

  
doc. Ing. Renáta Myšková, Ph.D.

děkanka

L.S.

  
doc. Ing. Jiří Křupka, Ph.D.

vedoucí ústavu

V Pardubicích dne 3. října 2011

### **PODĚKOVÁNÍ:**

Zde bych velice rád poděkoval Ing. Stanislavě Šimonové Ph.D. za čas, který mi při tvorbě práce věnovala, cenné připomínky a odborné rady, kterými přispěla k vypracování této práce.

## **PROHLÁŠENÍ**

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 23. 4. 2012

Bc. Milan Mládek

## **ANOTACE**

Práce je věnována problematice elektronického obchodování ve veřejné správě a návrhu modelu elektronického obchodu s vybranými službami veřejné správy. V teoretické části práce jsou rozebrány přístupy pro návrh aplikací.

## **KLÍČOVÁ SLOVA**

elektronický obchod, veřejná správa,

## **TITLE**

Possibilities of implementation of e-commerce in public administration

## **ANNOTATION**

This project is dedicated to issue of e-commerce in public administration and to design the model of e-shop with the chosen services of public administration. Theoretical part analyses the ways of application design.

## **KEYWORDS**

e-shop, public administration

# OBSAH

Úvod.....	12
1. Princip elektronického obchodování.....	13
2. Analytické a modelovací nástroje pro vývoj aplikace .....	16
2.1 Strukturovaný přístup.....	16
2.1.1 Datové modelování strukturovaným přístupem.....	17
2.1.2 Koncept tří architektur.....	17
2.2 Objektově orientovaný přístup.....	23
2.2.1 Návrh aplikace objektově orientovaným přístupem .....	23
2.2.1.1 Use Case .....	23
2.2.1.2 Sekvenční diagram .....	25
2.2.1.3 Class diagram.....	25
3. Návrh postupu.....	27
4. Určení služeb ve veřejné správě.....	29
5. Výběr vhodných služeb – aspekty vhodnosti a proveditelnosti.....	33
6. Vývoj prototypu pro vybrané služby.....	35
6.1 Model systému elektronického obchodování s produkty a službami příspěvkových organizací obce.....	35
6.1.1 Charakteristika problému .....	35
6.1.2 Požadavky na systém.....	37
6.1.3 Analýza systému .....	38
6.1.3.1 Use Case .....	38
6.1.3.2 Scénáře.....	44
6.1.3.3 Sekvenční diagramy .....	47
6.1.3.4 Class diagram.....	50
6.1.4 Implementace.....	51
6.1.4.1 Použitá technologie .....	52

6.1.4.2	Obsah aplikace .....	52
6.1.4.3	Rozhraní.....	52
6.1.4.4	Správa uživatelů .....	54
6.1.4.5	Zabezpečení aplikace.....	55
6.1.4.6	Modul příspěvkové organizace typu DDM .....	57
6.1.4.7	Databáze .....	58
6.2	Model systému elektronického obchodování s místními poplatky .....	61
6.2.1	Problematika místních poplatků .....	61
6.2.2	Popis jednotlivých místních poplatků .....	61
6.2.3	Současná situace .....	63
6.2.4	Požadavky na systém.....	64
6.2.5	Analýza systému .....	65
6.2.5.1	Use Case .....	65
6.2.5.2	Scénáře.....	68
6.2.5.3	Sekvenční diagramy .....	71
6.2.5.4	Class diagram.....	74
Závěr .....		76
Použitá literatura.....		77



## SEZNAM OBRÁZKŮ

Obrázek 1: Příklad E-R diagramu [zdroj: vlastní] .....	20
Obrázek 2: Příklad RMD [zdroj: vlastní] .....	22
Obrázek 3: Use Case – Systém z pohledu zákazníka [zdroj: vlastní] .....	39
Obrázek 4: Use Case – Systém z pohledu administrátora [zdroj: vlastní] .....	40
Obrázek 5: Use Case – Systém z pohledu PO kino divadlo [zdroj: vlastní] .....	41
Obrázek 6: Use Case – Systém z pohledu PO typů muzeum lázně [zdroj: vlastní] .....	41
Obrázek 7: Use Case – Systém z pohledu PO kulturní dům [zdroj: vlastní] .....	42
Obrázek 8: Use Case – Systém z pohledu PO DDM [zdroj: vlastní] .....	43
Obrázek 9: Use Case – Systém z pohledu PO typu jídelna [zdroj: vlastní] .....	43
Obrázek 10: Use Case – Systém z pohledu zaměstnanců PO [zdroj: vlastní] .....	44
Obrázek 11: Sekvenční diagram – Přihlášení [zdroj: vlastní] .....	47
Obrázek 12: Sekvenční diagram – Změna hesla [zdroj: vlastní] .....	47
Obrázek 13: Sekvenční diagram – Přidat účet zaměstnance [zdroj: vlastní] .....	48
Obrázek 14: Sekvenční diagram – Přidat zájmový kroužek [zdroj: vlastní] .....	48
Obrázek 15: Sekvenční diagram – Editace odstranění zájmového kroužku [zdroj: vlastní] .....	49
Obrázek 16: Sekvenční diagram – Zaplacení zájmového kroužku [zdroj: vlastní] .....	49
Obrázek 17: Class diagram – Visual Studio 2010 [zdroj: vlastní] .....	50
Obrázek 18: Ukázka aplikace [zdroj: vlastní] .....	54
Obrázek 19: Kontrola formulářů [zdroj: vlastní] .....	57
Obrázek 20: Ukázka aplikace – nákup služeb [zdroj: vlastní] .....	58
Obrázek 21: Databáze aplikace [zdroj: vlastní] .....	59
Obrázek 22: Use Case – Systém z pohledu zákazníka [zdroj: vlastní] .....	66
Obrázek 23: Use Case – Systém z pohledu administrátora [zdroj: vlastní] .....	66
Obrázek 24: Use Case – Systém z pohledu zaměstnance úřadu [zdroj: vlastní] .....	67
Obrázek 25: Use Case – Systém vedoucího úřadu [zdroj: vlastní] .....	68
Obrázek 26: Sekvenční diagram – Registrace poplatkové povinnosti [zdroj: vlastní] .....	71
Obrázek 27: Sekvenční diagram – Zaplacení poplatkové povinnosti [zdroj: vlastní] .....	71
Obrázek 28: Sekvenční diagram – Vyhledání plátce [zdroj: vlastní] .....	72
Obrázek 29: Sekvenční diagram – Editace registrovaného plátce [zdroj: vlastní] .....	72
Obrázek 30: Sekvenční diagram – Kontrola platby [zdroj: vlastní] .....	73
Obrázek 31: Sekvenční diagram – Editace kategorií plateb poplatků [zdroj: vlastní] .....	73
Obrázek 32: Class diagram [zdroj: vlastní] .....	74

## SEZNAM TABULEK

Tabulka 1: MIN-MAX notace [zdroj: vlastní] .....	19
Tabulka 2: Návrh postupu [zdroj: vlastní] .....	27
Tabulka 3: Scénář – Přihlásit [zdroj: vlastní] .....	45
Tabulka 4: Scénář – Změnit heslo [zdroj: vlastní].....	45
Tabulka 5: Scénář – Přihlásit [zdroj: vlastní] .....	45
Tabulka 6: Scénář – Editace odstranění zájmového kroužku [zdroj: vlastní] .....	46
Tabulka 7: Scénář – Zaplacení zájmového kroužku [zdroj: vlastní] .....	46
Tabulka 8: Scénář – Přidat účet zaměstnance [zdroj: vlastní].....	46
Tabulka 9: Scénář – Registrace poplatkové povinnosti za odpad/chov psa [zdroj: vlastní] ....	68
Tabulka 10: Scénář – Zaplacení poplatkové povinnosti za odpad/chov psa [zdroj: vlastní] ...	69
Tabulka 11: Scénář – Editovat registrovaného plátce [zdroj: vlastní].....	69
Tabulka 12: Scénář – Vyhledat plátce [zdroj: vlastní].....	69
Tabulka 13: Scénář – Kontrola platby [zdroj: vlastní].....	70
Tabulka 14: Scénář – Editace kategorie sazeb poplatků [zdroj: vlastní] .....	70

## SEZNAM ZKRATEK

ASP	Active Server Pages
ČR	Česká republika
CSS	Cascading Style Sheets
ČSÚ	Český statistický úřad
DDM	Dům dětí a mládeže
HTML	HyperText Markup Language
IS	Informační systém
JSP	JavaServer Pages
OECD	Organization for Economic Cooperation Development
PO	Příspěvková organizace
Sb.	Sbírka zákonů
SPZ	Státní poznávací značka
UML	Unified Modeling Language
WTO	World Trade Organization

## ÚVOD

Podle statistiky ČSÚ nastal v posledních letech evidentní rozmach elektronického obchodování, tzv. e-shoppingu, u lidí všech věkových kategorií (vyjma kategorie lidí ve věku 65+, ve které je podíl elektronicky nakupujících stále zanedbatelný). Ze statistik také vyplývá, že v roce 2010 mělo přístup k internetu již necelých 62 % lidí. Z dat je patrné, že statistický růst uživatelů internetu a jejich počítačové gramotnosti v České republice za posledních několik let, má pozitivní vliv na elektronické obchodování se soukromým sektorem. Otázkou však zůstává, za jakých podmínek a s jakými možnostmi je možné využít praktiky a zvyklosti elektronického obchodování, se kterými se setkáváme během elektronického obchodování se soukromým sektorem, v sektoru veřejném, tedy například na portálech veřejné správy. [1]

Výše položenou otázkou se zabývá má diplomová práce. Práce je tedy zamyšlením nad možnostmi uplatnění elektronického obchodování v libovolné instituci veřejné správy (obecní a krajské úřady), průzkumem současné legislativy týkající se vybraných možností a především návrhem konkrétního modelu systému elektronického obchodu, který by později mohl fungovat na vybraných portálech veřejné správy.

Už v úvodu je nutné si připustit, že takový systém se bude od klasických elektronických obchodů nejspíše značně lišit, a to jednak z důvodu platné legislativy, která bude daný systém kvůli ochraně osobních údajů omezovat, a také díky velké diferenci produktů a služeb, jež by mohly být v prostředí takového systému nabízeny.

Cílem práce je analýza možností využití principu elektronického obchodování na portálech veřejné správy a návrh modelu pro vybraný subjekt veřejné správy.

# 1. PRINCIP ELEKTRONICKÉHO OBCHODOVÁNÍ

Elektronické obchodování je obecně bráno jako podnikání za pomoci elektronických prostředků. Tato činnost nezahrnuje pouze samotný prodej produktů (hmotných i nehmotných) a služeb, ale i další činnosti související s obchodováním, a to například reklamu, uzavírání smluv, podporu prodeje a další.

Některé jiné definice (OECD) považují elektronické obchodování za provádění jakýchkoliv elektronických transakcí, které jsou založeny na přenosu dat. Další definice (WTO) zahrnuje pod elektronické obchodování nejen hmotné výrobky, které byly prodány a zaplacený přes internet, ale i nehmotné, jež jsou doručovány v digitální podobě prostřednictvím internetu. [2]

Elektronické obchody lze podle účastníků obchodování rozčlenit do několika kategorií. Mezi základní kategorie pak patří [3]:

- B2B („business to business“) – Jedná se obchodování mezi podniky a obchodníky navzájem. Z finančního pohledu jde o nejvýznamnější formu elektronického obchodování, přestože bývá v mnoha případech veřejnosti skryta. V této formě elektronického obchodování se transakce často uskutečňují ve specializovaných sítích, do kterých mají přístup pouze obchodní partneři.
- B2C („business to customer“) – Obchodní vztah mezi obchodními společnostmi a koncovými zákazníky. Jde tedy o klasické spotřebitelské smlouvy s koncovými zákazníky. Tato forma elektronického obchodování je nejrozšířenější.
- C2C („customer to customer“) – Vzájemný obchod mezi spotřebiteli. Typickým příkladem této formy je elektronická aukce.
- C2B („customer to business“) – Tato forma patří mezi nejméně rozšířené. Jde o situaci, kdy zákazník oslovuje obchodníky, přičemž sám definuje poptávaný produkt a další parametry dané transakce (např. maximální cenu).
- B2A („business to authority“), C2A („customer to business“) – Tyto formy vyjadřují vztah k veřejné správě. Příkladem těchto vztahů může být podávání některých dokumentů či formulářů (daňová přiznání) orgánům veřejné správy. V literatuře se často místo označení A (authority) používá označení G (government). Tyto formy však často bývají označovány jako eGovernment.

- A2C („authority to business“) – V tomto případě poskytuje veřejná správa služby občanům. Příkladem může být publikování informací na webových portálech veřejné správy.

Z výše uvedeného rozdělení vyplývá, že zamýšlený elektronický obchod týkající se této práce může být typu B2A, C2A, A2C nebo případně B2C, a to za následujících okolností:

- A2C v případě, že bude nalezena služba (případně produkt), kterou veřejná správa nabízí občanům, přičemž za tuto službu nebude bráno poskytování informací.
- B2A nebo C2A v případě, kdy nalezená služba nebude službou v pravém slova smyslu, ale například poplatkem nebo elektronicky zpracovaným formulářem poskytovaným veřejné správě.
- B2C v případě, kdy nalezená služba či produkt bude nabízena organizací, která sice není orgánem veřejné správy, ale je s veřejnou správou v bližším vztahu.

Uvažujeme-li klasický elektronický obchod formy B2C, pak je tento obchod výhradně vždy webovou aplikací, která je složená z několika hlavních komponent. Tyto komponenty (celky) mají u obecně u většiny elektronických obchodů obdobné funkce, strukturu i zařazení na samotných stránkách. Některé z těchto komponent mají funkcí virtuálně napodobovat zvyklosti a možnosti klasických kamenných obchodů. Mezi základní části internetového obchodu, které obsahuje většina aplikací těchto obchodů, můžeme zařadit následující:

*Úvodní strana* má především prezentační funkci daného elektronického obchodu, a proto by měla působit přehledně. Součástí úvodní strany bývají velmi často informační tabule týkající se novinek v daném obchodu a jeho systému. Dále se zde často také nalézá výpis produktů a služeb, které jsou nejprodávanější nebo těch, které jsou například ve slevě. Tato strana je rozhraním do všech ostatních částí obchodu. [3]

Další základní částí, která je většinou součástí systémů elektronických obchodů, je část pro *vyhledávání*, která umožňuje vyhledávat jednotlivé produkty a služby, jež jsou daným obchodem prodávány. Tato složka je většinou umístěna na více místech. Formulář pro rychlé hledání je často umístěn v hlavičce daného systému tak, aby byl přístupný ze všech částí aplikace. Formuláře pro vyhledávání na základě zvolených parametrů se pak často nacházejí na své vlastní části nebo jsou součástí katalogu produktů a služeb. [3]

Výpis prodávaných produktů a služeb neboli *katalog produktů* je přístupný z úvodní strany. Tyto katalogy bývají logicky tříděné podle produktů a často tak tvoří hierarchickou strukturu. [3]

Na stránce *detailu produktu* se nalézají konkrétní informace a parametry jednotlivých produktů. V mnoha internetových obchodech bývá součástí této stránky i veřejná diskuze k danému produktu. [3]

Jak již bylo výše uvedeno, některé funkce elektronických obchodů simulují klasické kamenné obchody. Typickým příkladem takové funkce je *virtuální košík*, který má totožnou funkci jako košík v kamenném obchodu. Tedy dočasně uchovat vybrané zboží v pomyslném koši, aby je bylo možné například zaplatit společně. S vybraným zbožím v košíku je až do uzavření transakce možné jakkoliv pohybovat je vyměnit za jiné apod.). [3]

Závěrečnou fází celé transakce je zaplacení vybraného zboží na *elektronické pokladně*. Tyto pokladny ve většině případů nabízejí různé možnosti platby, typy dodávky zboží a poskytují další informace k dané transakci. [3]

Samostatnou částí systémů elektronických obchodů bývá *uživatelské rozhraní pro jednotlivé uživatele*. Aby se však uživatelé dostali na své stránky v rámci daného systému, tak se musí do daného systému zaregistrovat. V některých případech je dokonce registrace v daném systému nutná k samotnému nákupu zboží. Uživatelské účty pak ale mohou sloužit dále k reklamním účelům, procházení historie nákupů nebo například k různým výhodám pro zaregistrované zákazníky. [3]

Portály elektronických obchodů by měly dále poskytovat nápovědu k systému, reklamační řád, kontakty na vlastníka daného obchodu a další důležité informace.

Na výše popsaných základních částech je postavena většina elektronických obchodů. Vzhledem k tomu, že možnost elektronického obchodování využívá stále více lidí a popsané funkce jsou velmi intuitivní, tak se nabízí možnost využít tyto funkce v obdobném systému, ve kterém by však byly nabízeny a případně prodávány služby veřejné správy.

## **2. ANALYTICKÉ A MODELOVACÍ NÁSTROJE PRO VÝVOJ APLIKACE**

Modelovací nástroje hrají při tvorbě velkých, ale i menších softwarových produktů, významnou roli. U větších softwarových projektů by z mnoha důvodů bylo prakticky nemožné bez modelu daných systémů tyto projekty úspěšně dokončit, jelikož právě modelování má za svůj cíl vytýčení „nebezpečných“ situací, které mohou při používání vytvářeného softwaru nastat. Simulace a předcházení nebezpečných situací není však jediným smyslem modelování. Nemálo důležitou vlastností modelování je i snadná prezentace myšlenek, které jsou spjaté s funkčností vyvíjeného softwaru. Tato vlastnost je důležitá pro vývojové týmy, které na základě modelu mohou lépe vzájemně komunikovat postup během analýzy, implementace a rozdělování jednotlivých pracovních činností. Účelem modelování však není pouze komunikace ve vývojových týmech, ale i komunikace mezi vývojovým týmem a zákazníky, což může předejít mnoha rizikům. Snadná interpretovatelnost a srozumitelnost modelu vytvořeného modelovacími nástroji umožňuje také přesné definování cílů a funkcí týkajících se vytvářeného softwaru. Modelované systémy jsou běžně popsány bohatou textovou dokumentací a sadou diagramů určených pro popis aplikace. Za výhodu lze i považovat i to, že některé současné vývojové nástroje umožňují na základě modelu vygenerovat části kódu samotné aplikace.

Na datové modelování aplikací se lze dívat dvěma pohledy. Tím prvním je strukturovaný přístup, který slouží k vizuálnímu znázornění struktury aplikace a databáze. V tomto přístupu je navrhována struktura dat a funkce, které s těmito daty pracují. Jako druhý přístup lze označit objektově orientovaný, který na rozdíl od strukturovaného přístupu bere systém jako soubor objektů, jež spolu komunikují.

### **2.1 Strukturovaný přístup**

Jak již bylo výše řečeno, ve strukturovaném přístupu je navrhována struktura dat a funkcí, které s nimi operují. Počátky tohoto přístupu sahají až do 70. let a přestože je v současné době podstatně více používán přístup objektově orientovaný, je tento klasický přístup stále velmi oblíbený, a to zejména z důvodu snadného návrhu relační databáze, které jsou v dnešní době běžně využívané. Využívané prostředky k datovému a funkčnímu modelování tohoto přístupu jsou tyto:

- ERD (Entity Relationship Diagram),



- DFD (Data Flow Diagram),
- FSD (Function Structure Diagram),
- STD (State Transition Diagram),
- DD (Data Dictionary),
- Structure Chart.

E-R diagramy umožňují modelovat datovou strukturu aplikace a bude jim více prostoru věnováno níže v podkapitole o datovém modelování. Pro popis funkcí systému je používán Data Flow Diagram, který má hierarchickou strukturu a je složen ze tří typů prvků (datový sklad, proces, terminátor). Statistický pohled na strukturu systému a subsystémů poskytuje Function Structure Diagram, jehož základními prvky jsou funkce a vazby. Chování systému v závislosti na jeho stavech a událostech, které tyto stavy mění, umožňuje namodelovat ST diagram. U stavů jsou definovány podmínky, za kterých přecházejí do stavu jiného, čímž lze vcelku jednoduše popsat logiku celého systému nebo jeho částí. Datový slovník (Data Dictionary) slouží k formálnímu popisu dat. Hierarchickou strukturu modulů, ze kterých je systém složen, je možné namodelovat pomocí Structure Chart. V tomto diagramu jsou moduly na pozici listů grafu vždy volané svým předkem. [4]

### **2.1.1 Datové modelování strukturovaným přístupem**

Před tím, než zde bude popsána samotná logika modelování struktury systému pomocí E-R diagramu, je vhodné nastínit, z jakého důvodu je modelování těmito prostředky stále oblíbené. Strukturovaný přístup datového modelování, tedy konkrétně modelování E-R diagramem, je součástí jednoduchého postupu pro návrh relační databáze. Celý postup návrhu informačního systému je běžně nazýván jako tzv. koncept tří architektur, které rozdělují postup návrhu IS (zejména relační databáze) do tří úrovní podle stupně abstrakce pohledu na daný systém.

### **2.1.2 Koncept tří architektur**

Jak již bylo výše zmíněno, koncept tří architektur rozděluje postup návrhu IS do tří částí v závislosti na abstrakci pohledu na daný systém. V následujících podkapitolách budou popsány jednotlivé části návrhu IS podle tohoto konceptu. V některých literaturách je popsána i čtvrtá úroveň datového modelování – sémantická, která stojí na počátku datového modelování a její úlohou je přesná definice požadavků na navrhovaný IS. [5]

## Konceptuální úroveň

V této úrovni je pohled na navrhovaný systém nejvíce abstraktní. Výstupem z této úrovně je E-R diagram s bohatým popisem jeho prvků. Základními činnostmi pro správný výstup z této úrovně je definování entit a jejich atributů, vztahů mezi entitami a tzv. integritních omezení.

Entitou nazýváme objekt reálného světa, který je jednoznačně rozlišitelný a může existovat nezávisle na ostatních objektech. Jednoznačná rozlišitelnost entit je určena tzv. klíčovým atributem, tedy vlastností, kterou může být daný objekt jednoznačně identifikován (např. klíčovým atributem entity „zaměstnanec“ může být rodné číslo nebo číslo občanského průkazu). [5]

Entity nejsou však definovány pouze jedním klíčovým atributem, ale mohou mít další atributy, které jsou pro navrhovaný systém podstatné. Obecně lze veškeré vlastnosti entit rozdělit do dvou základních skupin, a to na:

- totální atributy,
- parciální atributy.

Hodnoty totálních atributů se na rozdíl od hodnot parciálních atributů musí vyskytovat v každém objektu dané entity. Krom klíčových atributů to však mohou být i jiné vlastnosti. U entity „zaměstnanec“ by totálním atributem mohlo být například jméno a příjmení zaměstnance. Z uvedené definice totálních atributů je patrné, že parciální atributy nemusí mít hodnotu v každém objektu dané entity. Příkladem parciálního atributu entity „zaměstnanec“ by pak mohla být například informace o SPZ vozu zaměstnance, jelikož není jisté, zda každý zaměstnanec vlastní automobil. [6]

Dalším často uváděným možným dělením atributů je rozdělení na atributy základní a odvoditelné. Základní atributy jsou pak přímo dané a jejich hodnoty nelze z ničeho usuzovat (například jméno a příjmení). Odvoditelným atributem je takový atribut, jehož hodnotu lze odvodit z jiného atribut. Příkladem odvoditelného atributu může být atribut pohlaví, který je odvoditelný z atributu rodného čísla.

Klíčové atributy hrají ve vlastnostech každé entity důležitou roli. Jelikož musí být hodnoty klíčových atributů dané entity vždy známy, jedná se o atribut totální. Klíčový atribut, který jednoznačně definuje daný objekt, je nejčastěji nazýván jako tzv. primární klíč. V mnoha případech se klíčový atribut neskládá pouze z jedné určující vlastnosti, ale z několika vlastností zároveň. V těchto případech je však nutné navrhnout jednotlivé části klíčového

atributu tak, aby tyto části nebyly samostatně možným klíčovým atributem. Chybným příkladem návrhu klíčového atributu by mohla být například dvojice atributů rodné číslo a číslo občanského průkazu, jelikož oba mohou plnit samostatně roli klíčového atributu.

Klíčové atributy entit jsou obecně rozděleny na několik základních typů, a to:

- alternativní klíč,
- cizí klíč,
- kandidátní klíč
- sekundární klíč.

Z těchto typů hrají při návrhu aplikace největší roli primární a cizí klíče. Cizí klíč je primárním klíčem v jiné entitě, která je již definována.

Definice vztahů (včetně jejich atributů) mezi entitami je další důležitou součástí návrhu na konceptuální úrovni. Vztah, který existuje pouze mezi dvěma entitami, nazýváme jako binární vztah. Naopak vztah mezi větším množstvím entit je obvykle nazýván jako vztah polyární.

Poslední klíčovou činností při návrhu IS na konceptuální úrovni je definice tzv. integritních omezení. Integritní omezení se týkají vztahu mezi entitami a popisují tyto vztahy trochu blíže na základě čtyř vlastností, kterými jsou:

- kardinalita,
- parcialita,
- exkluzivita,
- externí identifikace.

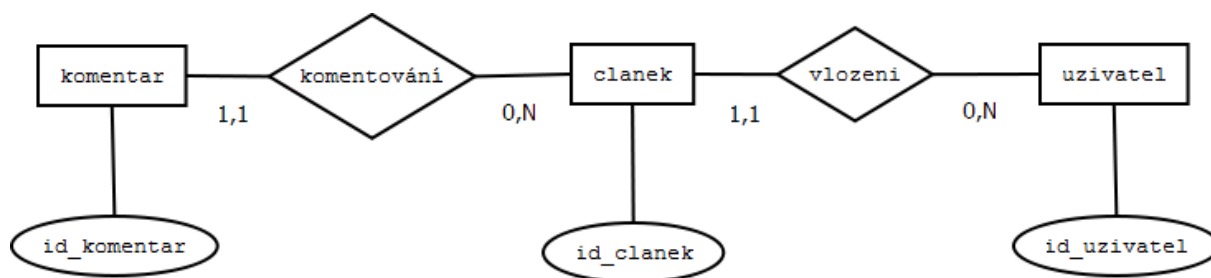
Nejčastěji vyskytujícími se vlastnostmi při definování integritních omezení vztahů jsou parcialita a kardinalita. V E-R diagramu, který bude uveden na příkladu níže, jsou tyto dvě vlastnosti nejčastěji uváděny pomocí tzv. MIN-MAX notace. Princip vyjádření těchto dvou vlastností MIN-MAX notací zobrazuje následující tabulka.

*Tabulka 1: MIN-MAX notace [zdroj: vlastní]*

Hodnota	0	1	N
<b>Parcialita (MIN)</b>	Nepovinné členství ve vztahu	Povinné členství ve vztahu	
<b>Kardinalita (MAX)</b>		Ve vztahu se entita vyskytuje nejvýše jednou	Ve vztahu se entita může vyskytovat vícekrát.

V úvodu této kapitoly bylo uvedeno, že výstupem z této úrovně je E-R diagram s potřebnými komentáři. Jako příklad situace namodelované tímto diagramem může být například vztahy entit „uživatel“, „článek“ a „komentář“, které tvoří základ jednoduchého redakčního systému a platí tedy následující pravidla:

- uživatel může vložit libovolný počet článků,
- článek musí být vložen právě jedním uživatelem,
- článek může být komentován libovolným počtem komentářů,
- komentář musí komentovat právě jeden článek.



Obrázek 1: Příklad E-R diagramu [zdroj: vlastní]

### Technologická úroveň

Na této úrovni návrhu IS musí již být uvažováno o databázové technologii, ve které bude daný model implementován. V této práci bude využita v současné době nejběžněji používaná databázová technologie, která je nazývána jako relační. V technologickém modelu návrhu IS, na jehož vstupu je okomentovaný E-R diagram, budou entity a vztahy z konceptuálního návrhu transformovány do tzv. relací. Po vytvoření relací jsou tyto relace očištěny o tzv. anomálie procesem, který je označován pojmem normalizace.

Transformace entit do relací probíhá velice jednoduše. Každá entita utvoří samostatnou relaci a každému jejímu atributu bude přiřazena doména relace. Primární i ostatní klíče, které jsou mezi vlastnostmi transformované entity, plynule přechází do relace, přičemž jejich úloha i označení zůstávají totožné. [4]

Vztahy mezi entitami musí být na této úrovni také transformovány do relací, a to na základě přesně definovaných pravidel. Tato pravidla zohledňují integritní omezení (parcialita, kardinalita) definovaná na konceptuální úrovni následovně:

Vztah typu 1:1, kde je tedy objekt jedné entity ve vztahu právě s jedním objektem entity druhého typu. Zde mohou nastat tři situace v závislosti na povinnosti účastníků tohoto vztahu:

- a) V případě povinného členství ve vztahu u obou účastníků definujeme jednu relaci, která bude obsahovat původní vztahové atributy a primární klíče obou entit vstupujících do tohoto vztahu.
- b) Při povinném členství pouze objektu pouze jedné entity ve vztahu definujeme dvě relace, které vycházejí z entit vstupujících do vztahu. Primární klíč nezávislé entity je v relaci přidán do relace původní entity, která je v daném vztahu závislá.
- c) Při nepovinném členství obou entit v původním vztahu definujeme tři relace. Jednu vztahovou, která bude obsahovat primární klíče obou entit ze vztahu, a zbylé dvě přímo odvozené z entit vystupujících ve vztahu.

U typu vztahu 1:N, ve kterém může být jeden objekt jedné entity ve vztahu s několika objekty druhé entity, mohou nastat dvě situace:

- a) U povinného členství obou entit ve vztahu definujeme dvě relace analogicky s předešlým případem (1:1 - b).
- b) U nepovinného členství entit ve vztahu definujeme tři relace, a to opět analogicky s předchozím případem (1:1 - c).

Vztahy typu M:N, kdy více objektů jedné entity mohou být ve vztahu s více objekty entity druhého typu, definují v případě povinného i nepovinného členství vždy tři relace. Dvě přímo odvozené z entit vstupujících do vztahu a jednu vztahovou, jež musí obsahovat primární klíče obou entit ve vztahu a případně další atributy daného vztahu.

Po výše popsané transformaci entit do relací je nutné daný model očistit od anomálií, které mohly při vzniku modelu vzniknout. Normalizace, proces, pomocí kterého k tomuto očištění dochází, definuje několik tzv. normálních forem, které jsou pravidly pro očištění tohoto modelu. Při návrhu na této úrovni je účelem dané relace postupně očistit pomocí co nejvyšší možné normální formy, přičemž se vždy musí začínat první normální formou. Je však nutné zdůraznit, že se v praxi často mohou vyskytovat případy, ve kterých nebude nutné některé popisované anomálie očišťovat. Jako příklad může být uvedeno zbytečné rozdělování celé adresy uživatele na jednotlivé adresní části podle 1NF v systému, ve kterém bude údaj o adrese pouze informační. Pravidla jednotlivých nominálních forem vypadají pak takto: [7]

**1NF** – Tuto normální formu splňují relace, jejíž popisné atributy nejsou vícehodnotové a nelze je tedy dále rozdělit.

**2NF** – Druhou normální formu splňují relace patřící do 1NF a zároveň všechny atributy jsou plně závislé na více atributovém primárním klíči.

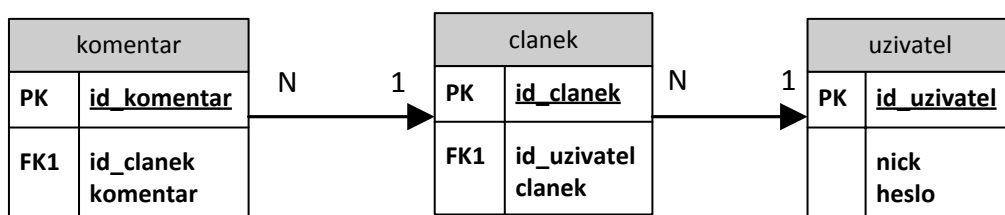
**3NF** – Třetí normální forma je splněna, pokud je relace v 3NF a zároveň každý popisný atribut je plně nezávislý a nezávisí tedy na žádném atributu, který by byl závislý na primárním klíči.

**BCNF** – Boyce-Coddova normální formu (BCNF) splňuje relace, kde každá funkční závislost je vyjádřena kandidátním klíčem. Relace kromě této podmínky musí splňovat i 3NF.

**4NF** – splňuje taková relace, pokud je v BCNF a zároveň všechny hodnoty jejího primárního klíče jsou vzájemně souvislé. Tato normální forma se tedy týká více atributových primárních klíčů.

**5NF** – platí pro relaci, která je v 4NF. Druhá podmínka se týká primární klíčů složených z třech a více atributů. Mezi atributy tohoto klíče musí být párové cyklické závislosti.

Pro ilustraci bude provedena transformace entit a vztahů z předchozího příkladu jednoduchého redakčního systému do relačního modelu. Z E-R diagramu vyplývá, že entity komentář – článek a článek - uživatel jsou vždy vztahu ve vztahu 1:N, přičemž oba vztahy jsou z jedné strany povinné. Z pravidel pro transformaci vztahů vyplývá, že oba vztahy budou transformovány do dvou entit a primární klíče nezávislých entit budou vloženy do relací závislých entit. Takto vzniklý relační model pak bude vypadat následovně:



Obrázek 2: Příklad RMD [zdroj: vlastní]

### Implementační úroveň

Implementační úroveň v sobě zahrnuje samotné naprogramování a uložení vytvořených relací z technologické úrovně do konkrétního relačního databázového systému. Databázový systém je volený na základě mnoha požadavků. Těmito požadavky může být například finanční náročnost, potřebný výkon databázového systému, způsob uložení a přístupu k datům, technologická úroveň databázového systému nebo například programovací jazyk aplikace, která bude mít k databázovému systému přístup. Mezi běžně používané relační

databázové systémy v současnosti patří MS SQL Server, Oracle, MS Access, MySQL, SQL Lite a další.

## **2.2 Objektově orientovaný přístup**

Důvodem k zavedení této metody přístupu pro návrh systému byla zejména existence a velká obliba využívání objektově orientovaných programovacích jazyků. Základem této metody je pojem objekt. Jedná se o prostředek pochopení reálného světa a jeho snadného modelování (v oblasti programovacích jazyků je objekt abstraktní datovou strukturou). Každý objekt je pak charakterizován svým jedinečným označením a charakterizován svými vlastnostmi a činnostmi (metodami), které může provádět. Objekty s totožnými vlastnostmi a metodami jsou sdružovány do tzv. tříd. [8]

V současnosti nejběžněji používaným objektovým modelovacím jazykem je UML (Unified Modeling Language), který má široké možnosti použití, a to nejen při tvorbě návrhu softwarových aplikací, ale například i pro popis business procesů či systémových funkcí. Tento jazyk byl roku 1997 sdružením OMG (Object Management Group) přijat a stal se tak prvním objektově orientovaným standardem pro vizuální modelování, přičemž je do současné doby neustále ve vývoji. [9]

K hlavním výhodám UML v rámci modelování aplikací patří jeho nezávislost na použitém programovacím jazyce, a tak jej lze použít pro aplikace programované např. v jazyce Java, C#, C++, Visual Basic a dalších. Objekty v systému popisuje jak staticky, tak i jejich vzájemné interakce a chování v jejich celém životním cyklu. [9]

### **2.2.1 Návrh aplikace objektově orientovaným přístupem**

V následujících podkapitolách bude popsán klasický postup návrhu aplikace objektově orientovaným přístupem pomocí modelovacího jazyka UML.

#### **2.2.1.1 Use Case**

Prvním krokem k modelování IS je tvorba tzv. Use Case diagramu, ve kterém lze specifikovat a detailně popsat veškeré funkce modelovaného systému v závislosti na uživatelských rolích. Use Case diagram je tvořen několika prvky, a to:

- use case (případ užití),
- hranice systému,
- aktér,

- vazbami mezi případy užití.

Prvek znázorňující případ užití symbolizuje jednu funkci popisovaného systému. V návaznosti na příklad z podkapitoly strukturovaného přístupu může jít např. o funkci „Přidat článek“.

Hranicemi systému lze rozlišit prvky patřící do modelovaného systému a prvky patřící mimo něj. Obvykle do modelovaného systému patří veškeré případy užití. Za hranicemi systému se pak nachází pouze aktéři.

Aktéři reprezentují uživatelské role, jež mají oprávnění využít s nimi spjaté případy užití. Už během modelování Use Case diagramem lze tedy namodelovat oprávnění jednotlivých „typů“ uživatelů IS k jednotlivým funkcím systému.

Některé případy užití jsou spolu v určitých vazbách, které je nutné specifikovat. Základním typem vazeb mezi případy užití jsou následující dvě:

- Includes – tato vazba symbolizuje situaci, ve které volaný případ užití zahrnuje jiný definovaný případ užití. Jedná se například o situaci, kdy k případu užití „Přidat článek“ je vždy vyžadováno přihlášení do systému, tedy například případ užití „Přihlášení“.
- Extends – vazba extends, neboli rozšíření, představuje situaci, kdy volaný případ užití volá jiný případ užití, ale pouze za určitých podmínek. Jako příklad může posloužit klasická situace, kdy je pro přihlášení uživatele do systému nejprve nutná jeho registrace.

Use Case diagram poskytuje jednoduchou možnost pohledu na celkovou funkcionalitu systému. Je nutné však zdůraznit, že pomocí tohoto diagramu nemusí být pouze funkcionalita softwarového systému, ale systémů obecně (např. ekonomické, vnitropodnikové, sociální a další).

Detailní popis jednotlivých případů užití poskytují tzv. scénáře. Scénář případu užití bývá zapsán nejčastěji formou tabulky a popisuje jednotlivé kroky aktérů a popisovaného systému, který na kroky aktéra odpovídá. Tabulka jednoho scénáře by měla, kromě popisu jednotlivých kroků, obsahovat název rozebíraného případu užití; aktéry, kteří komunikují s popisovaným systémem; popis; vstupy; výstupy a také událost (event), který daný případ užití spouští.

Při popisu jednotlivých kroků však může v mnoha případech nastat situace, ve které popisovaný krok náhle ukončí případ užití nebo jej nelze za určitých podmínek splnit. K popisu této situace, kdy hlavní scénář nemůže být z důvodů nesplnění daných podmínek



dokončen, slouží tzv. alternativní scénáře, které by v případě jejich existence měly být také charakterizovány. [9]

### **2.2.1.2 Sekvenční diagram**

Výstupem z předchozí části je Use Case diagram a scénáře detailně popisující všechny případy užití. Další částí v modelování IS je přiřazení jednotlivých kroků popsaných scénářů k objektům, které umí tyto kroky (funkce) vykonat. Toto přiřazení je možné namodelovat a zobrazit pomocí sekvenčních diagramů, pomocí nichž lze vyjádřit v závislosti na čase posloupnost všech kroků scénáře a zároveň přiřadit funkce tvořící tyto kroky k instancím objektů, které tyto funkce umí vykonat. V tomto modelu jsou tedy definovány jednotlivé typy objektů a jejich funkce (metody).

Aby na sebe mohly jednotlivé kroky plynule navazovat, jsou metody objektů propojeny tzv. zprávami. Těmito zprávami pak může objekt požádat o vykonání dalších metod, které však nemá ve své definici. Instance volaného objektu po převzetí zprávy tedy provede požadovanou metodu. [9]

### **2.2.1.3 Class diagram**

Poslední částí modelu aplikace je class diagram neboli diagram tříd. Základním smyslem sekvenčního diagramu byla identifikace objektů a jejich metod. V diagramu tříd jsou tyto objekty sdruženy do tříd, které definují jejich atributy a nalezené metody. Class diagram však neposkytuje pouze soupis definovaných tříd, ale také vazby, které mezi třídami vznikají. Základními vazbami mezi třídami mohou být tyto – dědičnost, kompozice a asociace.

Dědičnost je asi nejběžnějším typem vazby a její smysl tkví v možnosti „zdědit“ veškeré vlastnosti a metody jiné třídy. Instance třídy, která je potomkem jiné třídy, má, kromě sebou definovaných atributů a metod, možnost využít atributy či zavolat metody popsané v definici svého předka. Z praktického hlediska spočívá výhoda této vazby v možnosti definovat společné metody několika různých tříd pouze na jednom místě. Také je nutné zdůraznit, že princip dědičnosti funguje i víceúrovňově. Praktickým příkladem dědičnosti by mohla být abstraktní třída (jako předek), v jejíž definici by byla metoda dohlížející na kontrolu vyplnění formulářového prvku, a dalších několik tříd (potomků), které by zpracovávaly zcela odlišné formuláře a přitom mohly využívat metodu definovanou u svého předka pro kontrolu svých vlastních formulářových prvků.

Princip kompozice mezi třídami představuje závislost instancí jedné třídy na instanci jiné třídy. Tato vazba pracuje se situacemi, ve kterých nemá instance jedné třídy smysl bez existence instance jiné třídy, na které je závislá.

Asociační vazba umožňuje spolupráci instancí různých tříd. Z pravidla se pak jedná o instance tříd, které jsou mezi sebou v úzkém vztahu.

Pro úplný popis vazeb mezi třídami je nutné specifikovat nejen jejich typ, ale i jejich množstevní vztahy. Pro popis těchto vztahů se používá obdobná symbolika jako v relačním modelu dat ve strukturovaném přístupu (jedná se tedy např. o symboly 1, 1 .. n a další). [9]

### 3. NÁVRH POSTUPU

Problematika možností elektronického obchodování ve veřejné správě je velmi rozsáhlá, neboť musí zahrnovat několik obsáhlých kroků, a to v první řadě analýzu služeb veřejné správy. Dále důkladný výběr služeb, které jsou vhodné k elektronickému prodeji. Vybrané služby bude poté nutné pečlivě zanalyzovat a zahrnout do modelu elektronického obchodu. Závěrem by pak měla být alespoň částečná implementace vzniklého modelu.

Pro zpracování této práce navrhuji postup (viz Tabulka 2).

*Tabulka 2: Návrh postupu [zdroj: vlastní]*

Fáze	Kroky (Forma, nástroje)
Určení služeb ve VS	Analýza služeb poskytovaných veřejnou správou (konzultace na úřadech, verifikace)
Výběr služeb	<ul style="list-style-type: none"><li>• Výběr vhodných služeb na základě analýzy (konzultace na úřadech, verifikace)</li><li>• Zdůvodnění výběru</li></ul>
Vývoj prototypu pro vybranou službu	<ul style="list-style-type: none"><li>• Charakteristika problému</li><li>• Požadavky na systém</li><li>• Analýza systému</li><li>• Návrh databáze</li></ul>

Analýza služeb ve veřejné správě bude zpracována zejména na základě průzkumu na portálech veřejné správy, které nabízejí kompletní výčet činností, kterými se úřady veřejné správy zabývají. Vzhledem k množství a obsáhlosti této části bude však nutné redukovat výčet zpracovaných úřadů a jejich činností, a to na základě konzultací, doporučení a subjektivního pohledu.

Určení služeb vhodných k elektronickému obchodování bude založeno na předchozí analýze, přičemž budou vybrány nejdříve všechny služby, které by byly potencionálně vhodné k elektronickému obchodování. Z důkladnější analýzy těchto služeb bude na základě konzultací vybráno několik služeb, na jejichž základě bude namodelován systém.

Vybrané služby bude však nejdříve nutné podrobit pečlivé textové analýze, která bude obsahovat detailní formace k daným službám, jejich současnému a dále možnému zpracování.

Namodelování systému elektronického obchodování s vybranými službami bude obsahovat klasický objektový UML postup obsahující Use Case diagramy, scénáře, sekvenční diagramy, ze kterých vzniknou třídy, jež budou dále zaznamenány v class diagramech.

Implementace vzniklých modelů bude obsahovat zejména návrh relační databáze, která je v dnešní době stále nejpoužívanější. Tato část bude podle obsáhlosti namodelovaného systému obsahovat také alespoň částečné naprogramování a zprovoznění základních funkcionalit ve vybraném programovacím jazyce.

## 4. URČENÍ SLUŽEB VE VEŘEJNÉ SPRÁVĚ

Z důvodu velkého množství institucí veřejné správy budou v této kapitole shrnuty pouze základní činnosti těch institucí, které mají přímý kontakt s občany v lokalitách obcí a krajů. Zdůvodněním tohoto kroku může být to, že větší přímý kontakt občanů s vybranými institucemi přináší větší prostor pro hledání právě těch služeb, které by z hlediska využitelnosti mohly být nabízeny systémem elektronického obchodu. Jako instituce, které mají běžně přímý kontakt s občany, byly vybrány finanční úřady, úřady práce, živnostenské úřady, stavební úřady, krajské a obecní úřady. Na následujících řádcích budou shrnuty základní činnosti těchto institucí.

### Finanční úřady

Finanční úřady jsou územní organizační jednotky, které mají na starost zejména správu daní v dané lokalitě. Seznam činností finančních úřadů zahrnuje například:

- evidenci a příjem daní,
- provádění daňové kontroly,
- příjem podání přiznání k dani dědické, z nemovitostí, z přidané hodnoty, z příjmu fyzických osob,
- ukládání a vymáhání pokut,
- vykonávají správu daní, odvodů a záloh na tyto příjmy.

Kompletní seznam činností je k nalezení například na webových stránkách <http://www.portal.gov.cz/>. [10], [11]

### Úřady práce

Úřady práce se z hlediska rozdělení působnosti dělí v základu na tyto:

- kontaktní pracoviště,
- krajské pobočky,
- státní sociální podpora,
- Úřad práce ČR - generální ředitelství.

Jelikož Generální ředitelství je pověřeno především vedením krajských poboček a dále například přijímáním opatření pro rozvoj trhu práce nebo evidencí agentur práce, tak je možné rovnou usoudit, že této struktury se případný model elektronického obchodu nebude týkat.

Kontaktní pracoviště pracovních úřadů jsou běžně dostupné občanům ve větších obcích (například pro region) Pardubice to jsou Holice, Přelouč a Pardubice). Mezi nejběžnější činnosti těchto pracovišť patří například:

- přijímají hlášení od zaměstnavatelů o volných pracovních místech,
- přijímají hlášení od zaměstnavatelů o zaměstnání osob se zdravotním postižením,
- nabízejí volná pracovní místa,
- přijímají žádosti o podporu v nezaměstnanosti,
- poskytují podporu při rekvalifikaci,
- poskytují informace o volných pracovních místech,
- vedou evidenci zdravotně postižených.

Krajské pobočky úřadů práce jsou v každém krajském městě a jako jejich hlavní činnosti jsou stanoveny zejména provádění výplat pohřebného, porodného, příspěvků na dítě, příspěvků na bydlení, rodičovského příspěvku, sociálního příplatku a také za činnosti spjaté s pěstounskou péčí.

Seznam činností poboček státní sociální podpory je velmi podobný jako seznam činností krajských poboček. Tyto pobočky přijímají žádosti o výplatu pohřebného, porodného, stejně tak přijímají žádosti na výplatu spjaté s pěstounskou péčí apod. Mimo těchto činností se ovšem také zabývají vydáváním rozhodnutí o vydání a přiznání dávek státní sociální podpory. [12]

### **Živnostenské úřady**

Na živnostenských úřadech je možné ohlašování živnosti fyzickými a právnickými osobami. Dále také vydávání koncesních listin fyzickým a právnickým osobám a správou živnostenským rejstříkem (zapisované údaje, výpisy, základní informace). Většina těchto činností je poskytována za stanovené poplatky (například správní poplatek za vydání živnostenského listu je 1000,-). Tyto poplatky musí být uhrazeny před podáním ohlášení živnosti. [13]

### **Stavební úřady**

Tyto úřady spadají zařazením pod městské úřady a řídí se především stavebním zákonem 183/2006 Sb. Mezi hlavní činnosti těchto institucí patří:

- vydávají stavební povolení, územní rozhodnutí, kolaudační souhlas, souhlas ke zkušebnímu provozu, souhlas s ohlášenou stavbou,

- vyzývají k odstranění nedostatků v projektové dokumentaci, k odstranění závad na stavbě,
- vedou stavební řízení,
- přijímají žádosti a povolují odstranění stavby,
- nařizují odstranění, vyklizení, provedení nezbytných úprav staveb,
- oznamují stavební povolení nebo úpravu jeho platnosti,
- ukládají pokuty za přestupky vůči stavebnímu zákonu,
- provádějí stavební dohled.

Za některé z těchto činností jsou vybírány stanovené poplatky. Jedná se například o poplatky za územní řízení (1000,-), stavební povolení (300,- až 3000,- podle náročnosti stavby). [14]

### **Krajské úřady**

Tyto úřady vedou dohled nad obcemi v celém kraji. V rámci kontaktu s občany řeší tyto instituce například přijímání oznámení o shromáždění nebo veřejné sbírky, přijímají žádosti o udělení státního občanství ČR, vydávají informace o evidenci obyvatel nebo vydává povolení chovu zvířete v zajetí. [15]

### **Obecní a městské úřady**

Obecní a městské úřady mají na rozdíl od úřadů krajských větší zastoupení služeb a zejména poplatků a mají tedy s občany více přímého kontaktu. Krom správy obce se tak zabývají například: [16]

- výběrem poplatků spjaté komunálním odpadem,
- výběrem poplatků za psy,
- výběrem poplatků za užívání veřejného prostranství,
- vydáváním vyhlášek o chovu zvířat na území obce, čistotě a pořádku v obci, nakládání s odpady,
- pronájemem hrobových míst,
- přijímáním poplatků za znečištění ovzduší, pronájem hrobového místa,
- pronájemem bytových a nebytových prostor,
- zakládáním a kontrolou příspěvkových organizací obce (divadla, kina, mateřské školky, jídelny apod.).

Jako závěr této kapitoly lze napsat, že podle stručného předchozího šetření je možné ve veřejné správě vybrat služby, které by mohly být prodávány prostřednictvím

elektronického obchodování. Většinou však nejde o služby známé z běžného soukromého prostředí, ale spíše o poplatky za zprostředkování služeb nebo poplatků obecně. Je také nutné brát ohled na to, co lze brát jako službu a co čistě jako poplatek (například poplatek za svoz komunálního odpadu nebo poskytnutí stavebního povolení a poplatek za psa), přičemž myšlený systém by mohl využít principy známé z elektronického obchodování na platbu jak služeb, tak poplatků. Dále je důležité zdůraznit, že bude potřeba vybrat ty služby, jejichž elektronický „prodej“ by splňoval s nimi související legislativu a především ty, které má vůbec smysl prodávat. Smyslem prodávání je myšlena především návaznost na poskytnutí samotné služby a také srovnání výhodnosti elektronického nákupu s běžnou platbou v budově dané instituce.



## 5. VÝBĚR VHODNÝCH SLUŽEB – ASPEKTY VHODNOSTI A PROVEDITELNOSTI

Tato kapitola vysvětluje, z jakých důvodů byly vybrány konkrétní případy služeb vhodných k elektronickému prodeji, na jejichž základě byly v následujících kapitolách vytvořeny modely aplikací. Vybrané služby jsem stanovil především díky dlouhodobým pravidelným konzultacím, a to se zaměstnanci veřejné správy (zejména obecních úřadů a úřadů městských obvodů v Pardubicích).

Z uvedeného seznamu základních služeb vybraných úřadů jsou v této kapitole vybrány služby, které se zdají být pro elektronický obchod nejzajímavější. Z výběru byly vyloučeny ty služby a poplatky, jejichž elektronický „prodej“ již probíhá na portálech samotných úřadů nebo v projektu Czech POINT. Podstatným kritériem je také splňování současné legislativy, tak, aby mohly být naplněny alespoň základní funkce vyvíjeného systému.

Czech POINT je projekt, jehož cílem je vytvoření služby pro komunikaci se státem prostřednictvím jednoho místa. Díky tomuto projektu je možné získávat a ověřovat data z elektronických informačních systémů veřejné správy. Mezi poskytované služby tímto projektem patří poskytování výpisů z některých rejstříků (obchodní, živnostenský, trestů, insolvenční), dále např. výpis z Katastru nemovitostí, výpis z bodového hodnocení řidiče nebo například vydání ověřeného výstupu ze seznamu kvalifikovaných dodavatelů. Zbylé služby jsou k nalezení na oficiálních webových stránkách tohoto projektu (<http://www.czechpoint.cz>). [17]

Do budoucna je cílem tohoto projektu stav, ve kterém bude možné skrz Czech POINT učinit jakákoliv podání úřadům veřejné správy. Dále také získávat veškeré výpisy, opisy, údaje z centrálních evidencí a registrů, převádět dokumenty do elektronické podoby, ověřovat listiny, podpisy apod. Z tohoto důvodu je zřejmé, že velké množství potenciálních možností pro elektronický prodej služeb veřejné správy odpadá.

Existuje však stále několik služeb, u kterých by úvaha o jejich elektronickém prodeji stála za zvážení. Jedná se zejména o služby ve vztahu občan - obec. Tedy ty, jež jsou zprostředkovávány obecními úřady. Konkrétně jde o možnost platby místních poplatků v obcích. Při platbě těchto poplatků by systém navíc mohl disponovat částí pro samotnou registraci pro následné zaplacení, což by v některých případech bylo i nutné (například registrace psa a poté již opakované platby). Tyto poplatky se sice v současné době dají

zaplatit čistě elektronicky, ale systém pro elektronickou registraci a platbu ve stylu elektronického obchodu chybí.

Obec dále zřizuje množství příspěvkových organizací, přičemž se službami některých těchto organizací se již běžně v komerční sféře elektronicky obchoduje (kina, divadla). U mnohých těchto organizací, které jsou však zřizovány obcemi, však elektronický obchod s jejich službami často neexistuje, a to zejména v menších obcích. Nejzajímavější možností pro elektronický obchod se službami vybraných typů příspěvkových organizací obcí představuje centralizovaný systém elektronického obchodu, ve kterém by byly k dostání veškeré služby těchto organizací na jednom místě. Na tento systém by pak mohlo být odkazováno například z oficiálních stránek obcí. Na rozdíl od elektronického obchodu s místními poplatky hovoří pro návrh a implementaci obchodu se službami příspěvkových organizací ještě jedna věc, a tou je nekolidování se současnou legislativou. Navíc by tento systém mohl u některých služeb jednoduše přebrat mnohé principy z komerční sféry (například nákup a rezervace lístků do kina).

Ze předchozích kapitol vyplývá, že velký prostor pro uplatnění elektronického obchodování se ve veřejné správě stále nachází. Bohužel však mnohé služby jsou či v budoucnosti budou elektronicky prodávány v rámci projektu Czech POINT. Další možnosti pro uplatnění těchto principů ovšem nabízí obecní úřady, a to konkrétně zamýšlený systém pro platbu místních poplatků. Zajímavou možností je také centralizovaný systém pro prodej služeb některých příspěvkových organizací zřizovaných obcemi. Návrhem těchto dvou systémů se budu zabývat v následujících kapitolách.

## 6. VÝVOJ PROTOTYPU PRO VYBRANÉ SLUŽBY

Tato kapitola směřuje ke konkrétnímu návrhu elektronického prodeje vybraných služeb, přičemž tento návrh bude vždy rozdělen do několika oblastí (bližší analýza problematiky, kladené požadavky, analýza a návrh systému pomocí UML, návrh databáze).

### 6.1 Model systému elektronického obchodování s produkty a službami příspěvkových organizací obce

Následující podkapitoly nabízí analýzu, modelování a implementaci prototypu elektronického obchodu se službami příspěvkových organizací obcí.

#### 6.1.1 Charakteristika problému

Příspěvkové organizace jsou obecně podle zákona 250/2000 Sb. zakládány svým zřizovatelem, kterým jsou územní samosprávné celky (obce, kraje). Na rozdíl od organizačních složek územních samosprávných celků jsou příspěvkové organizace účetními jednotkami a z důvodu rozsahu, struktury a složitosti mají samostatnou právní subjektivitu. Příspěvkové organizace jsou zpravidla neziskové. Mohou však nabývat peněžních prostředků nejen od svého zřizovatele, ale i z vlastní činnosti nebo peněžních darů. Nabyté peněžní prostředky mohou využívat pouze na své stanovené účely. [18]

Příspěvkové organizace obcí jsou zejména:

- divadla a kina,
- mateřské a základní školy,
- domy dětí a mládeže,
- jídelny,
- kulturní domy,
- muzea,
- lázně,
- domy s pečovatelskou službou,
- různá specializovaná zdravotnická zařízení (rehabilitační péče, dětská péče apod.)
- a jiné.

Z výše uvedených příkladů příspěvkových organizací obcí vyplývá, že některé z nich se zabývají totožnými činnostmi jako některé společnosti z podnikatelského prostředí. Jedná

se především o kina, divadla a jídelny. K těmto organizacím můžeme přičíst dále kulturní domy a muzea, jelikož nabízejí podobný typ služeb, a proto je možné zahrnout prodej jejich služeb do modelového systému. I přestože jídelny nabízejí zcela odlišné služby než předchozí organizace, lze je do navrhovaného modelu zavést také. Naproti tomu rozdílná situace je u domů s pečovatelskou službou, mateřských a základních škol. U domů s pečovatelskou službou by sice teoreticky bylo možné „kupovat“ nájemné skrz elektronický obchod, ale využití takového systému by bylo nulové, a to zejména kvůli dlouhodobosti uzavíraných transakcí a také menší gramotnosti starších obyvatel v oblasti internetu a elektronického obchodování. Základní školy nenabízejí (s výjimkou školních jídelen a školních kroužků) žádné služby, se kterými by bylo možné obchodovat, takže do modelu systému nebudou zahrnuty.

V případě kin a divadel se v praxi setkáváme s možnostmi rezervace či zakoupení lístků online. V příspěvkových organizacích stejného typu by tedy neměl být problém implementovat podobné možnosti k zakoupení lístků nebo alespoň k jejich rezervaci.

Veřejné jídelny, jež jsou příspěvkovými organizacemi obce, by taktéž mohly poskytovat možnosti k zakoupení vybraného jídla elektronickou cestou. Otázkou však zůstává, jakou zvolit zpětnou vazbu pro pracovníky jídelen pro kontrolu elektronicky zakoupeného jídla konkrétním občanem. Převládající klasická ústní identifikace jménem a příjmením by byla lehce zneužitelná například při častých objednávkách jednoho člověka. Proto by byla minimálně nutná identifikace občanským průkazem, ale ta by na druhou stranu mohla být pro některé občany nepříjemná. Další možností by mohl být systémem vygenerovaný krátký klíč, kterým by byla identifikována konkrétní objednávka. Posledním problémem, která se týká objednávky jídla v jídelnách, je základní otázka, zda prodávat jídlo jednotlivě nebo souhrnně na určité období. Řešením je pouze rozlišení konkrétních jídelen podle jejich potřeb a zvyklostí. U školních jídelen se jídlo platí často paušálně, zatímco veřejné jídelny pracují s jednotlivými objednávkami stravy. Systém by tedy musel umožňovat objednávky obou typů v závislosti na konkrétních jídelnách.

Na představení v kulturních domech by mohl být použit obdobný model fungování jako u divadel a kin. Kulturní domy obcí však často poskytují i možnost zapůjčení sálů na určitý termín za určitý poplatek. Zarezervování sálu by též šlo vyřešit formou elektronického obchodu. Požadavkem na systém by pak byla zejména kontrola volných termínů, na které by sál mohl být zarezervován.

Na muzea, která nabízí za poplatek vstup do svých prostor a prohlídku expozic, je také možné aplikovat podobný model jako na kulturní domy, divadla apod. Systém by však v případě muzeí musel počítat s často rozdílnými cenami pro jednotlivé skupiny osob (cena pro dospělé, cena pro děti atd.). Tyto cenové rozdíly musí být tedy v systému obsaženy.

Dalšími službami, se kterými by se dalo elektronicky obchodovat, jsou zájmové kroužky domů dětí a mládeže. Ve většině případů mají tyto organizace seznam svých zájmových kroužků a jejich cen na svých vlastních portálech, ale samotná platba je většinou uskutečňována osobně nebo převodem na účet. V myšleném modelu by tedy krom výčtu zájmových kroužků byla i možnost jejich zaplacení na jméno dítěte. Systém by také měl automaticky hlídat kapacitu daných kroužků, která by byla pevně stanovena samotným domem dětí a mládeže.

Posledními organizacemi, které má smysl obsáhnout v systému elektronického obchodování, mohou být městské lázně. V případě lázní musí být stejně jako u jídelen vyřešena otázka problému paušálních vstupů na určité období a jednotlivých vstupů v závislosti na konkrétních lázních a jejich nabízených službách.

### **6.1.2 Požadavky na systém**

Obdobně jako u modelu s místními poplatky je nutné, aby celá webová aplikace byla dobře zabezpečena. Databáze bude krom osobních informací obsahovat i informace o jednotlivých transakcích, které by mohly být velmi lehce zneužity.

Z předchozí analýzy také vyplývá, že celá aplikace bude složena z několika velmi rozdílných systémů. Jmenovitě se bude jednat o prvky systémů klasických elektronických obchodů, rezervačních systémů a informačních systémů. Velkým požadavkem na samotnou aplikaci by tedy měla být jistá vizuální i funkční jednodušnost celého rozhraní, tak, aby uživatel měl při jeho užívání dojem jednotného systému, byť bude celá databázová vrstva poměrně rozsáhlá a různorodá. I přestože by rozhraní mělo působit jednotně, bude systém funkčně u jednotlivých typů poskytovaných (prodávaných) služeb velmi rozdílný, a proto by měla aplikace obsahovat přesnou nápovědu s popsáním jednotlivými kroky realizace nákupu (nebo rezervace) těchto služeb.

Podstatnou součástí implementace by také měla být dostupná informovanost uživatelů o akcích, které buď samostatně provedli (informace o předchozích transakcích, přihlášení do systému) nebo o akcích, které mají vliv na celý systém (změny cen služeb, přidání nové služby apod.). V prvním případě by se tak jednalo samozřejmě o informace soukromé, tedy

zobrazitelné pouze na pomyslném „profilu“ uživatele po jeho přihlášení do systému a nikdo jiný by k nim neměl mít přístup. V druhém případě by se jednalo o informace veřejné, které by se zobrazovaly na informační tabuli úvodní stránky systému nebo až po přihlášení do systému jakýmkoliv uživatelem.

### **6.1.3 Analýza systému**

V následujících podkapitolách je pomocí UML diagramů a RMD modelu navržen systém elektronického obchodování se službami příspěvkových organizací obcí. Z výše uvedené analýzy vyplývá, že tento systém se bude primárně zabývat službami příspěvkových organizací typů divadlo, kino, kulturní dům, dům dětí a mládeže, jídelna, muzeum a lázně.

#### **6.1.3.1 Use Case**

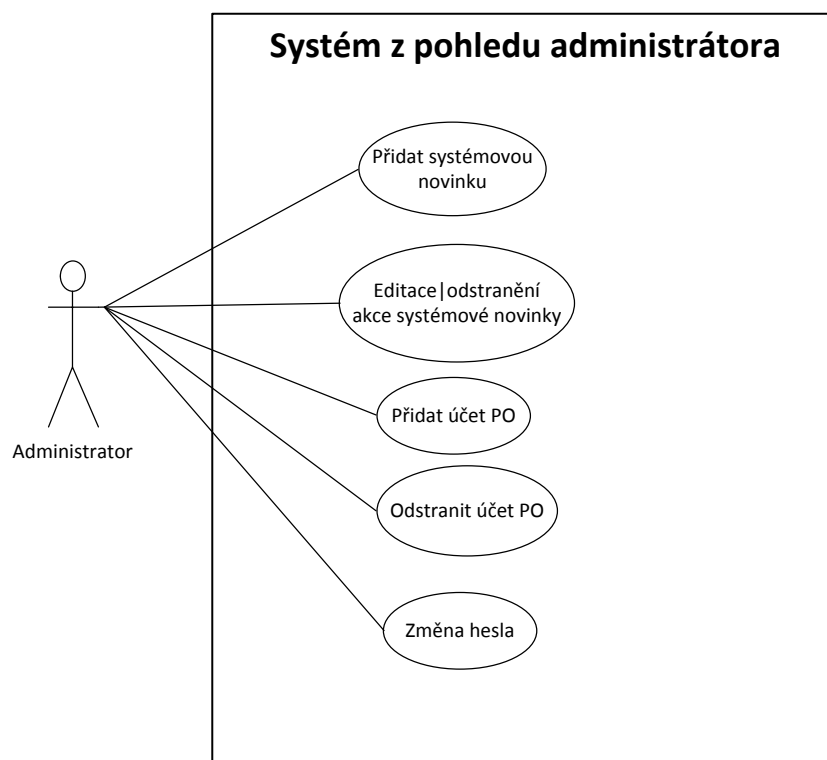
Vzhledem k větší obsáhlosti problému budou UML Use Case diagramy rozděleny logicky podle rozlišení aktérů nebo jejich činností. Základní rozdělení počítá s aktéry typu běžného uživatele, administrátora, správce příspěvkové organizace a jejich zaměstnanců.

Aktér, který symbolizuje běžného uživatele, má v popisu činností především nákupy samotných služeb příspěvkových organizací. Dalšími činnostmi je nutná registrace a možnost zpětného nahlédnutí k provedeným nákupům či rezervacím.



Obrázek 3: Use Case – System z pohledu zákazníka [zdroj: vlastní]

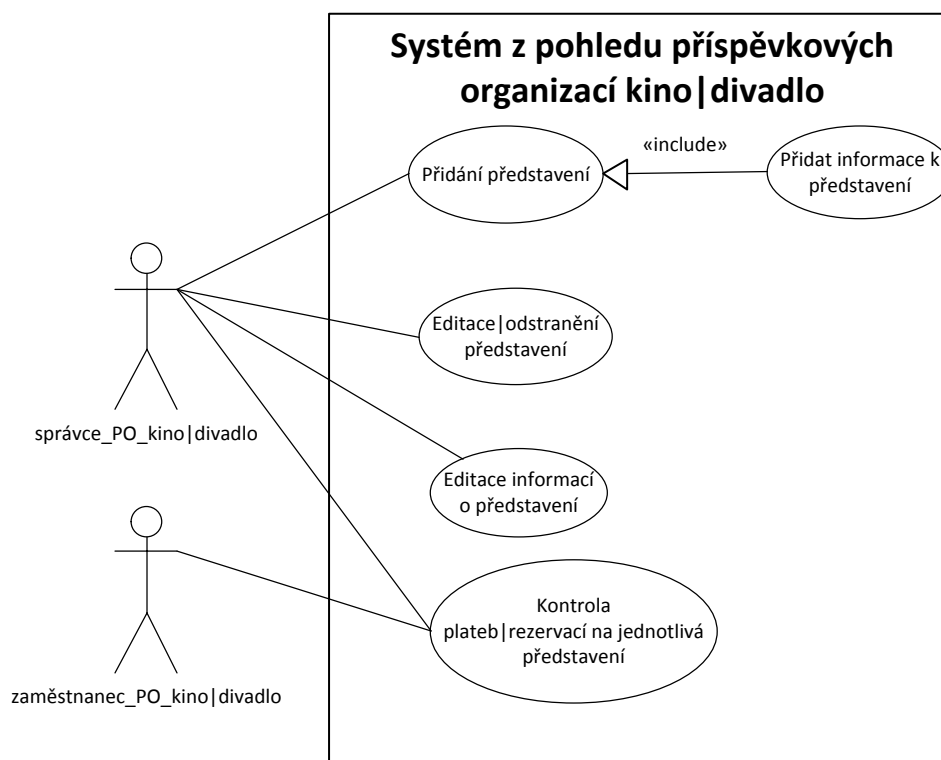
Administrátor celého systému zařizuje především tvorbu a odstraňování účtů příspěvkových organizací. Také má možnost přidávat, editovat a mazat systémové novinky, kterými může informovat všechny uživatele o aktualitách týkajících se systému.



Obrázek 4: Use Case – Systém z pohledu administrátora [zdroj: vlastní]

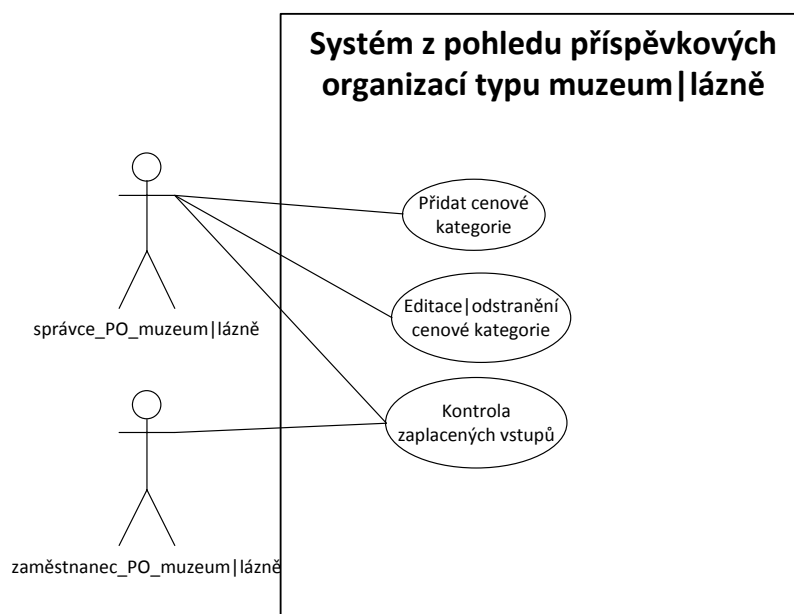
Soubor činností a aktérů z pohledu příspěvkových organizací typu divadlo a kino je velmi podobný, a proto je jej možné v rámci Use Case modelu sjednotit. Základním aktérem je správce příspěvkové organizace, kterým může být někdo z jeho vedení nebo odpovědná osoba z IT oddělení této organizace. Tato osoba má na starosti především tvorbu a úpravy programu a informací k jednotlivým položkám tohoto programu. Editace a odstranění položky z programu bude podmíněno kontrolou, která povolí odstranění položky pouze v případě, kdy na tuto položku není již rezervováno nebo zapláceno ani místo veřejným uživatelem. Aktér zaměstnanec příspěvkové organizace má na starosti přepážkovou kontrolu platby či rezervace na představení.





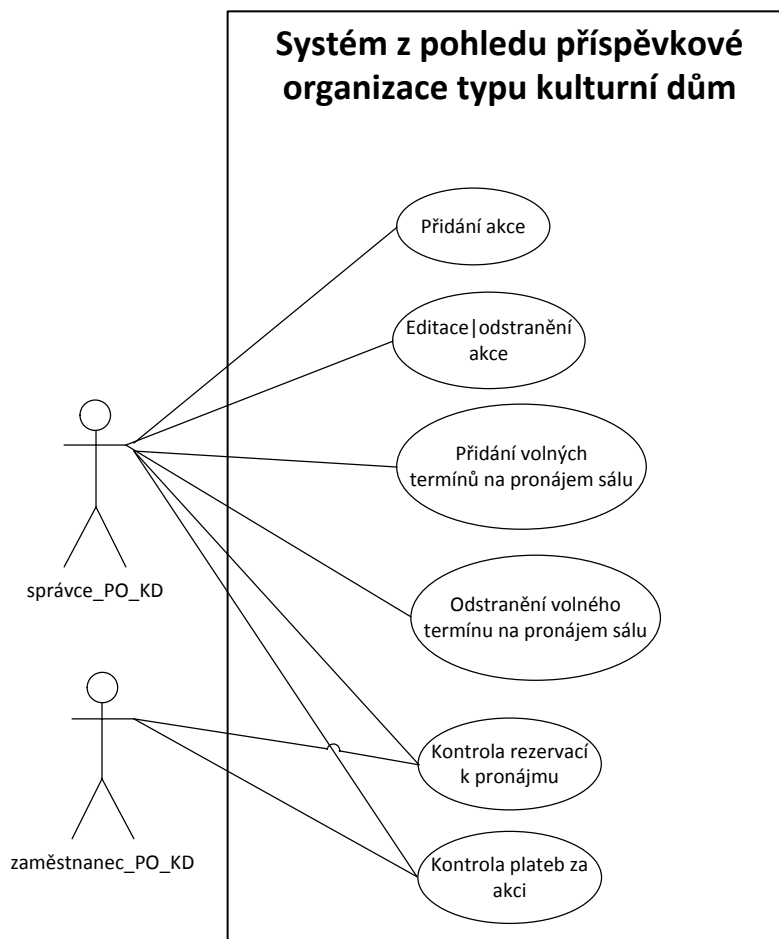
Obrázek 5: Use Case – Systém z pohledu PO kino|divadlo [zdroj: vlastní]

Muzeum a lázně v rámci systému nabízí vždy jednu službu za různou cenu podle různých skupin lidí (děti, studenti, dospělí) a počtu vstupů, a proto je hlavní činností správce těchto organizací v tomto systému především tvorba a úprava těchto kategorií. Zaměstnanec na přepážce pak analogicky s dalšími typy příspěvkových organizací zpracovává kontrolu platby.



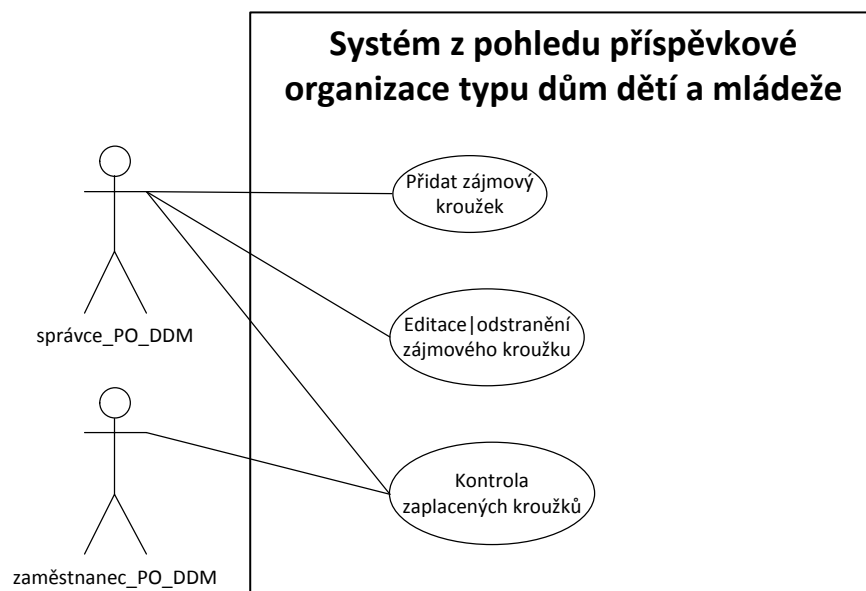
Obrázek 6: Use Case – Systém z pohledu PO typů muzeum|lázně [zdroj: vlastní]

Z analýzy problematiky kulturních domů vyplývá, že tyto organizace pořádají akce, které budou řešeny analogicky jako například divadelní představení. Kulturní domy však také často poskytují prostory kulturního domu, a proto bude správci přidána možnost přidávat a upravovat volné termíny k pronájmu prostor kulturního domu. Ostatní zaměstnanci mají stejně jako v předchozích případech na starosti kontrolu plateb.



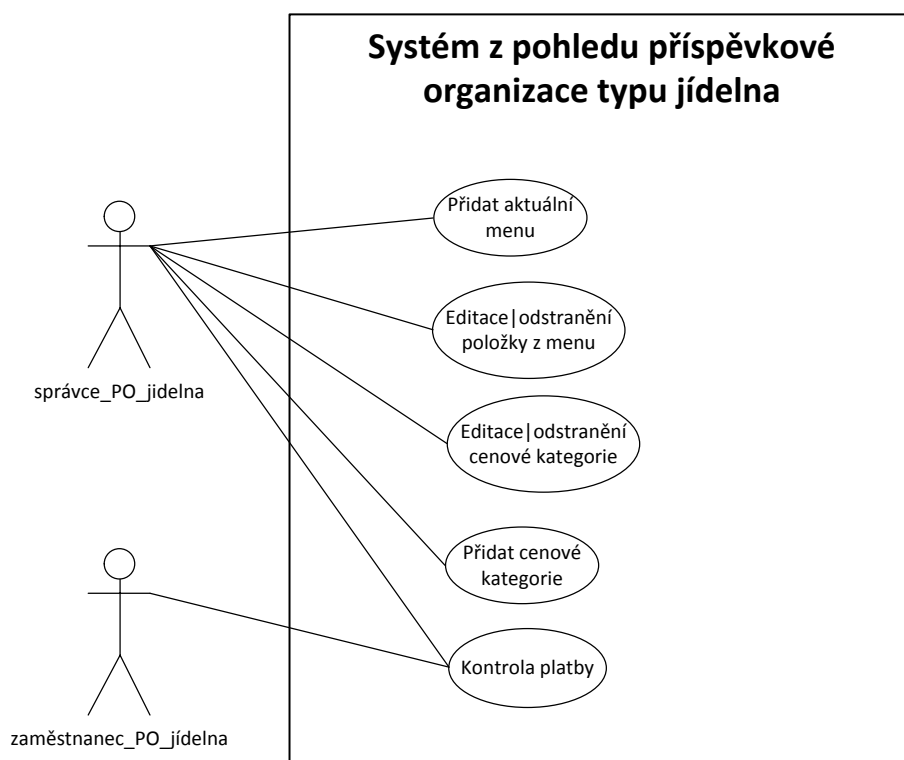
Obrázek 7: Use Case – Systém z pohledu PO kulturní dům  
[zdroj: vlastní]

Správci příspěvkových organizací typů dům dětí a mládeže budou mít možnost přidávat, editovat a odstraňovat zájmové kroužky a informace o nich (cena, popis, kapacita). Ostatní zaměstnanci kontrolují platby.



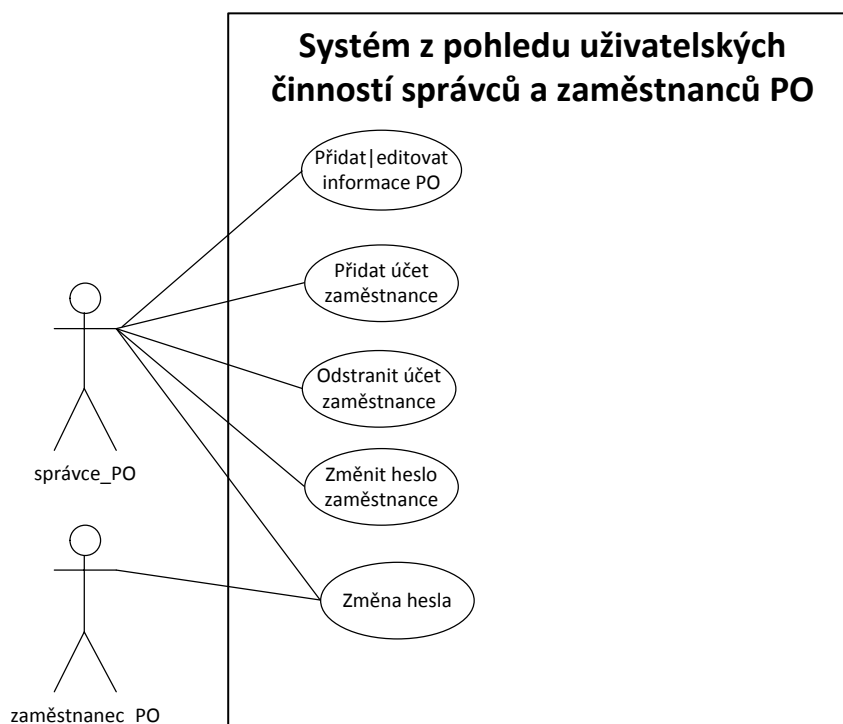
Obrázek 8: Use Case – Systém z pohledu PO DDM [zdroj: vlastní]

Jídelny by se rozdělením jejich služeb daly popsat velmi podobně jako lázně, které jsou popsány výše. Rozdílem je možnost přidávání aktuálního menu a editace či odstranění jeho jednotlivých položek. Tato činnost bude příslušet správci. Další zaměstnanci kontrolují platby jako ve všech předchozích případech.



Obrázek 9: Use Case – Systém z pohledu PO typu jídelna [zdroj: vlastní]

Pro větší přehlednost byly předchozí diagramy, ve kterých vystupovaly pohledy na činnosti konkrétních typů příspěvkových organizací, zbaveny uživatelské politiky systému. Ta je zobrazena na Use Case diagramu níže. Účty zaměstnanců dané příspěvkové organizace vytváří a odstraňuje její správce v systému. Ten má také možnost těmto zaměstnancům změnit heslo (z důvodů ztráty hesla a bezpečnosti) a dále editovat základní informace dané příspěvkové organizace. Všichni uživatelé pak mají možnost samostatné změny hesla.



Obrázek 10: Use Case – Systém z pohledu zaměstnanců PO [zdroj: vlastní]

Z důvodu větší přehlednosti není v žádném z předcházejících Use Case diagramů činnost „Přihlásit“, která je se všemi výše popsány činnostmi kromě činnosti „Registrace“ ve vazbě „include“. Přihlášení všech uživatelů do systému je tedy nutné pro všechny popsání činnosti.

### 6.1.3.2 Scénáře

Jelikož počet případů užití, které byly popsány v diagramech výše, je poměrně vysoký, tak budou v této části popsány scénáře těch případů užití, které jsou jedinečné nebo těch, jež by mohly být použity analogicky s dalšími, přičemž scénáře budou zaměřeny především na příspěvkové organizace typu kino a divadlo.

Tabulka 3: Scénář – Přihlásit [zdroj: vlastní]

<b>Use Case</b>	<b>Přihlásit</b>	
<b>Aktér</b>	Uživatel, administrátor, správce_PO (všichni dále ve scénáři jako uživatel)	
<b>Popis</b>	Cílem této operace je přihlášení uživatele do systému.	
<b>Vstupy</b>	Seznam uživatelů	
<b>Výstupy</b>	Uživatel je přihlášen v systému.	
<b>Hlavní scénář</b>		
<b>Krok</b>	<b>Role</b>	<b>Akce</b>
1.1	Uživatel	Klikne na odkaz „Přihlásit“.
1.2	Systém	Zobrazí formulář pro přihlášení uživatelským jménem a heslem.
1.3	Uživatel	Vyplní formulář přihlašovacími údaji a odešle jej stiskem tlačítka „Přihlásit“.
1.4	Systém	Ověří přihlašovací údaje.
1.6	Systém	Přihlásí uživatele a zobrazí úvodní stránku.
<b>Alternativní scénáře</b>		
1.6.1	Systém	Když údaje nesouhlasí s daty daného uživatele uloženými v databázi, zobrazí formulář pro přihlášení s varováním špatného přihlášení.

Tabulka 4: Scénář – Změnit heslo [zdroj: vlastní]

<b>Use Case</b>	<b>Změnit heslo</b>	
<b>Aktér</b>	Uživatel, administrátor, správce_PO (všichni dále ve scénáři jako uživatel)	
<b>Popis</b>	Cílem této operace je změna hesla uživatele.	
<b>Vstupy</b>	Seznam uživatelů	
<b>Výstupy</b>	Uživatel si změnil heslo.	
<b>Hlavní scénář</b>		
<b>Krok</b>	<b>Role</b>	<b>Akce</b>
1.1	Uživatel	Klikne na odkaz „Změnit heslo“ v nabídce nastavení svého profilu.
1.2	Systém	Zobrazí formulářové pole pro změnu hesla, které má tři textové vstupy.
1.3	Uživatel	Vyplní textový vstup starým heslem a zbylé dva vstupy heslem novým.
1.4	Systém	Zpracuje zaslání hesla. V případě, že je vše v pořádku je změněno heslo a zobrazena obrazovka s profilem uživatele.
<b>Alternativní scénáře</b>		
1.4.1	Systém	Zpracuje zaslání hesla. V případě, že staré heslo nesouhlasí nebo dva formulářové vstupy pro nové heslo nejsou totožné, tak zobrazí stránku pro změnu hesla s vypsanou chybou „Staré heslo nesouhlasí nebo vstupy pro nové heslo nejsou totožné!“.

Tabulka 5: Scénář – Přihlásit [zdroj: vlastní]

<b>Use Case</b>	<b>Přidat zájmový kroužek</b>	
<b>Aktér</b>	Správce_PO_DDM (dále Správce PO)	
<b>Popis</b>	Cílem této operace přidání zájmového kroužku do nabídky DDM	
<b>Vstupy</b>	Seznam uživatelů,	
<b>Výstupy</b>	Zájmový kroužek byl přidán.	
<b>Hlavní scénář</b>		
<b>Krok</b>	<b>Role</b>	<b>Akce</b>
1.1	Správce PO	Klikne na odkaz v menu „Přidat zájmový kroužek“.
1.2	Systém	Zobrazí formulář pro přidání zájmového kroužku.
1.3	Správce PO	Vyplní formulář a odešle jej.
1.4	Systém	Zkontroluje hodnoty v odeslaném formuláři a přidá zájmový kroužek.
<b>Alternativní scénáře</b>		
1.4.1	Systém	V případě, že odeslaný formulář neprojde kontrolou, tak jej vrátí k opětovnému vyplnění.

Tabulka 6: Scénář – Editace|odstranění zájmového kroužku [zdroj: vlastní]

<b>Use Case</b>	<b>Editace odstranění zájmového kroužku z nabídky DDM</b>	
<b>Aktér</b>	Správce_PO_DDM (dále Správce PO)	
<b>Popis</b>	Cílem této operace je editace nebo odstranění zájmového kroužku z nabídky DDM.	
<b>Vstupy</b>	Seznam uživatelů, seznam zájmových kroužků	
<b>Výstupy</b>	Existující zájmový kroužek byl editován nebo odebrán.	
<b>Hlavní scénář</b>		
<b>Krok</b>	<b>Role</b>	<b>Akce</b>
1.1.	Správce PO	Klikne na odkaz odkazující na nabízené zájmové kroužky.
1.2.	System	Zobrazí výpis nabízených zájmových kroužků.
1.3.	Správce PO	Vybere zájmový kroužek, který chce a klikne na ikonku editace.
1.4.	System	Zobrazí detail vybraného zájmového kroužku ve formulářových prvcích.
1.5.	Správce PO	Edituje připravené formulářové prvky.
1.6.	System	Zkontroluje nově vložené údaje a v případě splnění kritérií data uloží. (alt. sc. „B“)
<b>Alternativní scénáře</b>		
1.3.1A	Správce PO	Vybere zájmový kroužek, který chce a klikne na ikonku odstranění.
1.4.1A	System	Zobrazí detail vybraného zájmového kroužku a opětovně se zeptá na odstranění.
1.5.1A	Správce PO	Potvrdí odstranění
1.6.1A	System	Odstraní vybranou položku a vrátí uživatele na úvodní stránku.
1.6.1B	System	V případě nesplnění kritérií vrátí na obrazovku pro editaci údajů s vypsanou chybou.

Tabulka 7: Scénář – Zaplacení zájmového kroužku [zdroj: vlastní]

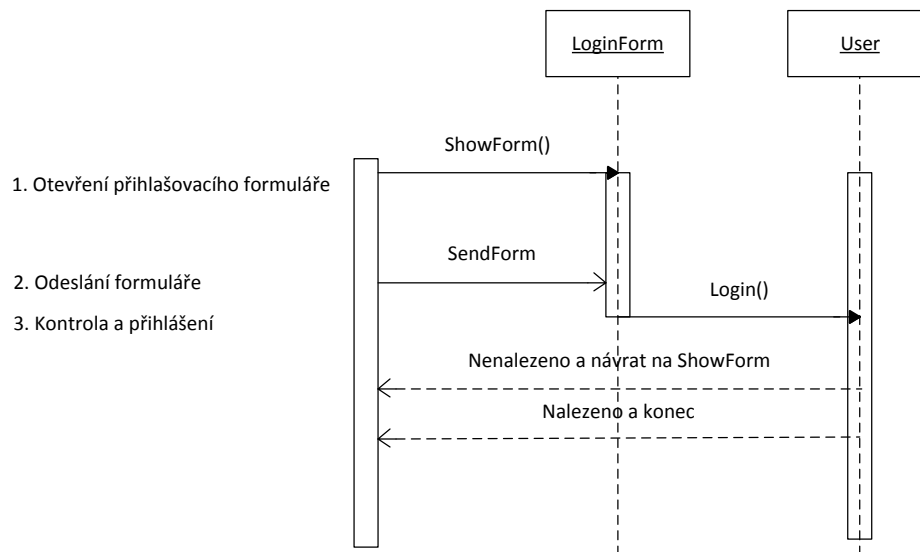
<b>Use Case</b>	<b>Zaplacení zájmového kroužku z nabídky DDM</b>	
<b>Aktér</b>	Uživatel	
<b>Popis</b>	Cílem této operace je přihláška a zaplacení na zájmový kroužek z nabídky DDM.	
<b>Vstupy</b>	Seznam uživatelů	
<b>Výstupy</b>	Uživatel se přihlásil a zaplatil zájmový kroužek. Správce DDM zaevidoval přihlášku.	
<b>Hlavní scénář</b>		
<b>Krok</b>	<b>Role</b>	<b>Akce</b>
1.1.	Uživatel	Klikne na odkaz odkazující na nabízené zájmové kroužky ve vybrané organizaci
1.2.	System	Zobrazí výpis nabízených zájmových kroužků.
1.3.	Uživatel	Vybere zájmový kroužek, který chce a klikne na ikonku nákupu.
1.4.	System	Zobrazí detail vybraného zájmového kroužku a formulářových prvky pro přihlášení.
1.5.	Uživatel	Edituje připravené formulářové prvky.
1.6.	System	Zkontroluje nově vložené údaje a v případě splnění kritérií data uloží.
<b>Alternativní scénáře</b>		
1.6.1	System	Vrátí chybně vyplněný formulář k znovu editování.

Tabulka 8: Scénář – Přidat účet zaměstnance [zdroj: vlastní]

<b>Use Case</b>	<b>Přidat účet zaměstnance</b>	
<b>Aktér</b>	Správce PO (obecně všech)	
<b>Popis</b>	Cílem této operace je přidání účtu zaměstnance.	
<b>Vstupy</b>	Seznam uživatelů	
<b>Výstupy</b>	Existující představení bylo editováno nebo odebráno.	
<b>Hlavní scénář</b>		
<b>Krok</b>	<b>Role</b>	<b>Akce</b>
1.1	Správce PO	Klikne na odkaz v menu „Přidat účet“
1.2	System	Zobrazí stránku s formulářovým polem pro přidání zaměstnance
1.3	Správce PO	Vyplní formulářové prvky (jméno, příjmení, přihlašovací jméno, oprávnění)
1.4	System	System přidá zaměstnance a k jeho účtu vygeneruje heslo.
1.5	System	Zobrazí stránku se všemi přidávanými údaji včetně hesla.
<b>Alternativní scénáře</b>		

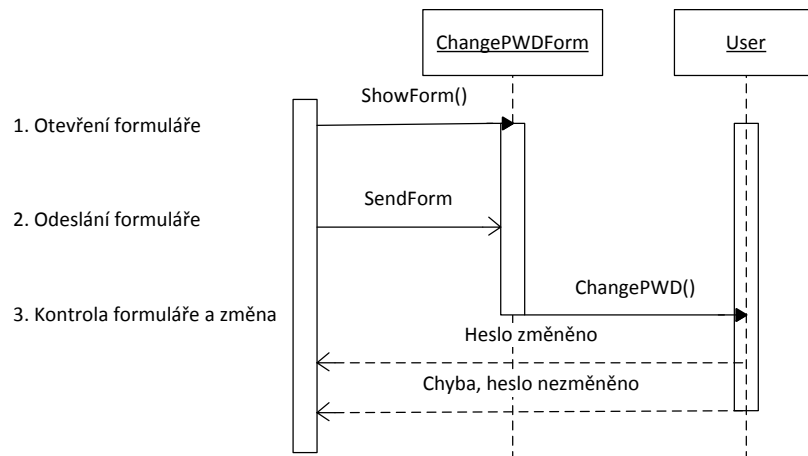
### 6.1.3.3 Sekvenční diagramy

#### Přihlásit



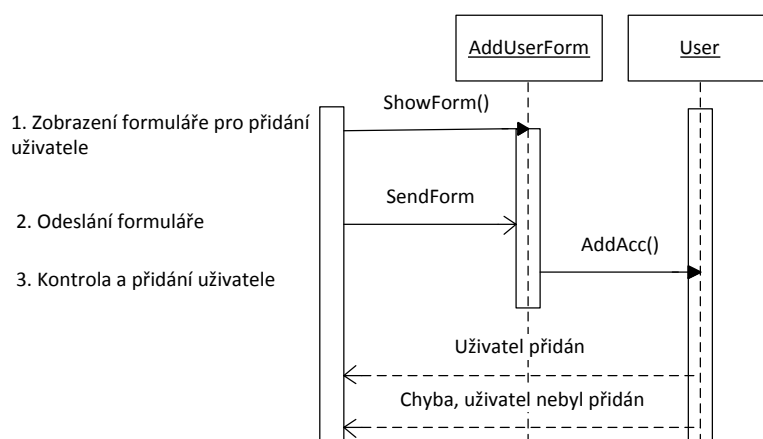
Obrázek 11: Sekvenční diagram – Přihlášení [zdroj: vlastní]

#### Změnit heslo



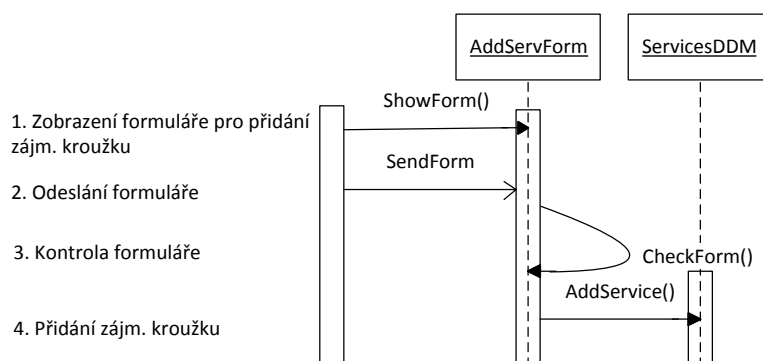
Obrázek 12: Sekvenční diagram – Změna hesla [zdroj: vlastní]

## Přidat účet zaměstnance



Obrázek 13: Sekvenční diagram – Přidat účet zaměstnance [zdroj: vlastní]

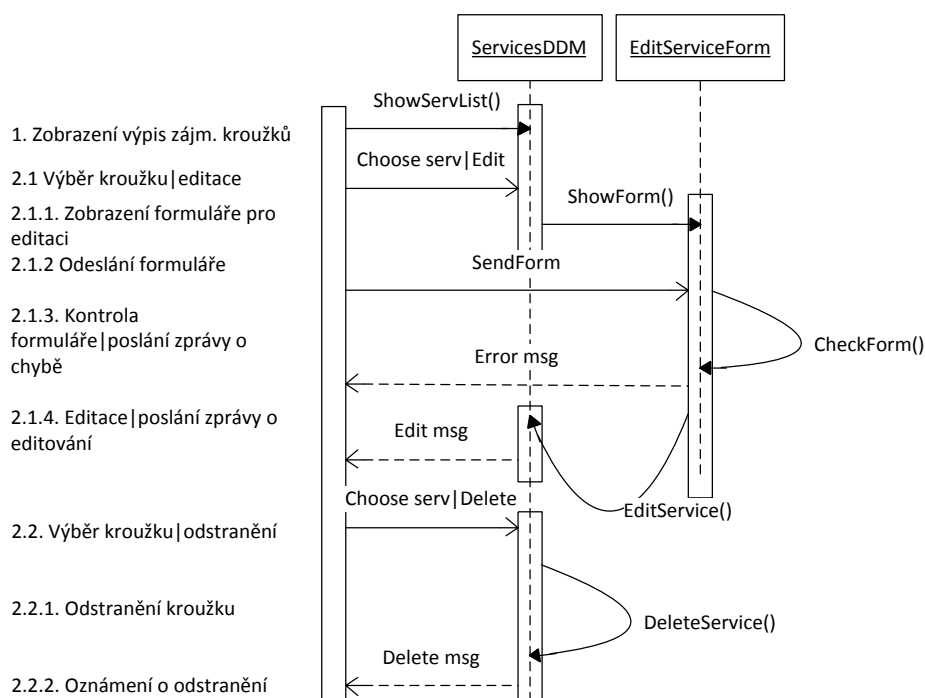
## Přidat zájmový kroužek



Obrázek 14: Sekvenční diagram – Přidat zájmový kroužek [zdroj: vlastní]

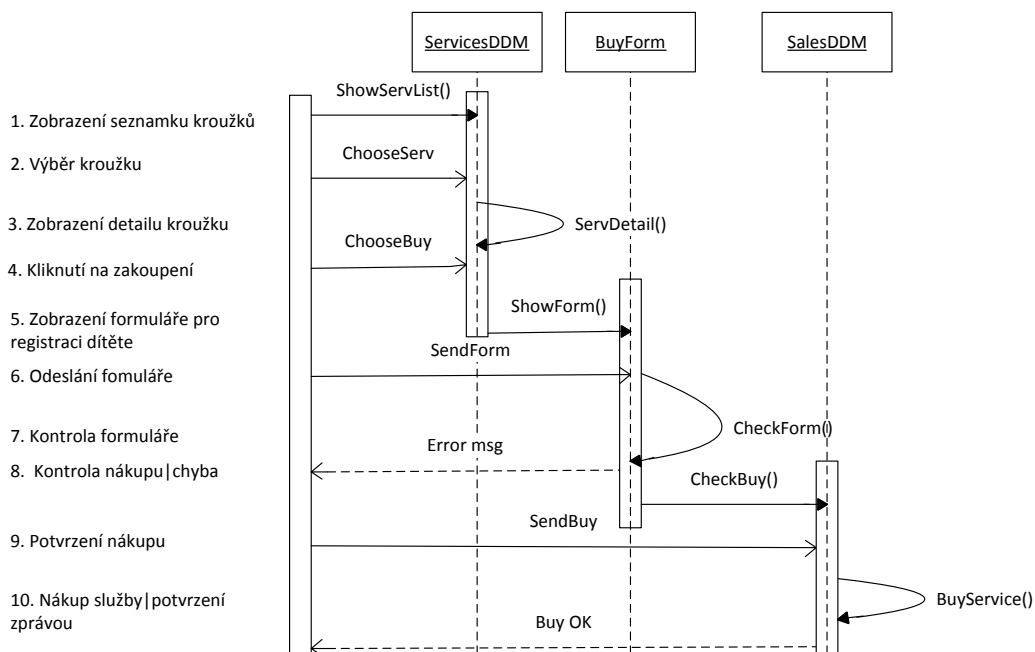


## Editace | odstranění zájmové kroužku



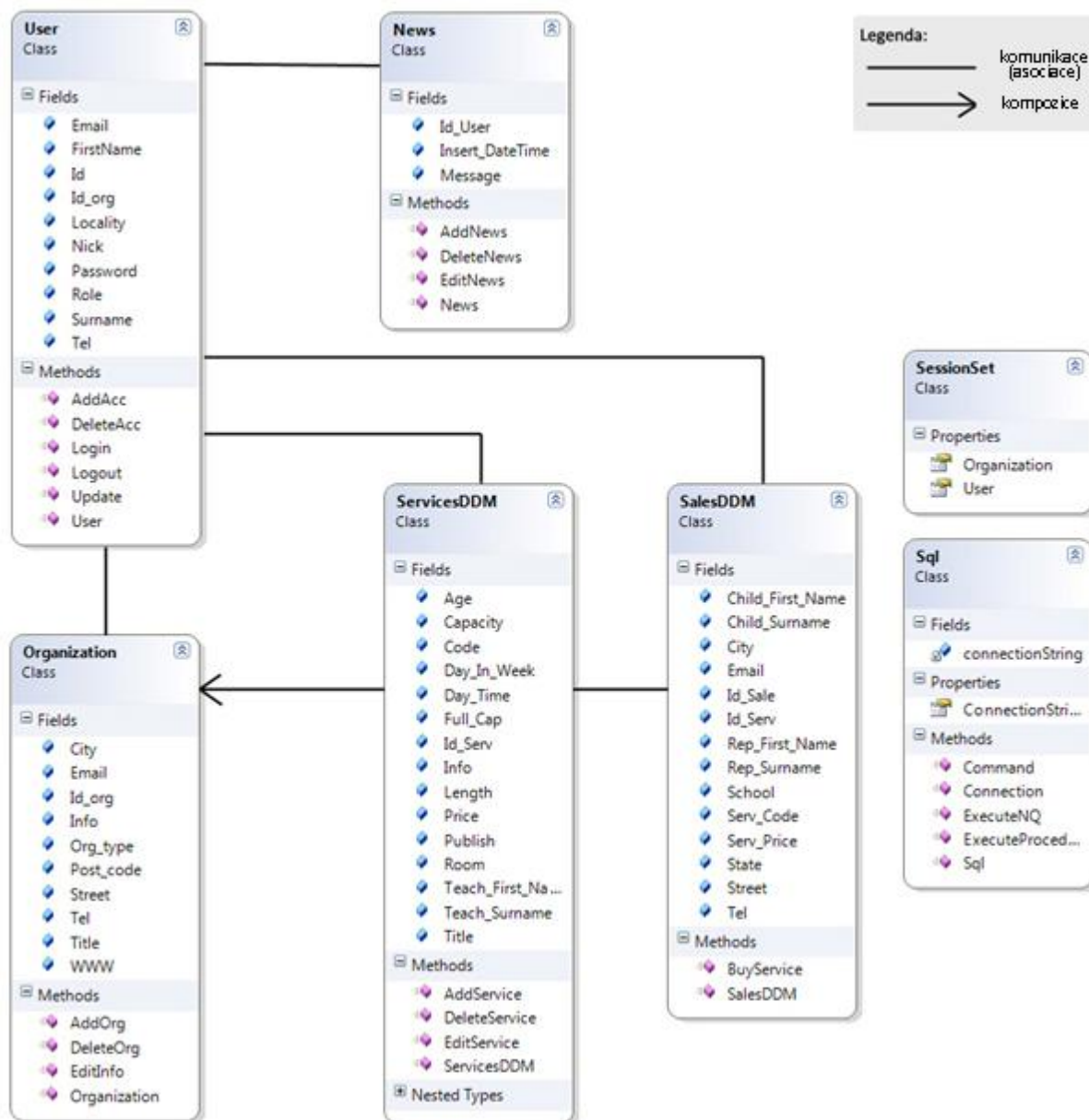
Obrázek 15: Sekvenční diagram – Editace|odstranění zájmového kroužku [zdroj: vlastní]

## Zaplacení zájmového kroužku



Obrázek 16: Sekvenční diagram – Zaplacení zájmového kroužku [zdroj: vlastní]

### 6.1.3.4 Class diagram



Obrázek 17: Class diagram – Visual Studio 2010 [zdroj: vlastní]

Výše zobrazený class diagram (kromě vazeb, které byly doplněny) byl vygenerován přímo z kódu v prostředí Visual Studia 2010 a znázorňuje všechny vytvořené třídy.

Třída `User` definuje vlastnosti uživatelů a všechny metody související s uživatelskými účty (přidání|editace|odstranění účtu a přihlašování do systému). Tato třída popisuje všechny uživatelské role, protože ty se datově liší pouze v hodnotě jedné proměnné – uživatelské roli. Třída `Organization` popisuje základní vlastnosti a metody (přidání|editace|odstranění organizace) přístupné všem typům organizací, které se rovněž mezi sebou liší datově pouze

v hodně tě jedné proměnné – typem organizace. V případném rozšíření o další typy organizací by tato třída mohla být *rodičem* tříd jednotlivých typů organizací.

Třída `Organization` je v asociačním vztahu se třídou `ServicesDDM`, která už představuje služby příspěvkové organizace typu „Dům dětí a mládeže“. Vlastnostmi této třídy jsou vlastnosti jednotlivých zájmových kroužků (cena, kapacita, vhodný věk, den a čas konání, jméno učitele apod.). Záznam z této třídy musí být vždy ve vztahu s jedním záznamem ze třídy `Organization`. V případě rozšíření aplikace o další typy příspěvkových organizací by třídy služeb těchto organizací byly vždy ve stejném vztahu k třídě `Organization`.

Samotné prodeje služeb DDM (v případě DDM jde o registrace na zájmové kroužky) jsou definovány třídou `SalesDDM`, jejíž vlastnosti popisují registrovaný objekt na zájmový kroužek, jde tedy například o jméno a příjmení dítěte; školu, kterou navštěvuje; jeho bydliště; kontakty na zastupující osobu a hlavně odkaz na id služby (kroužku), na který je registrován. Byť je tato třída v těsném vztahu se třídou definující služby DDM, tak její záznamy mohou existovat samostatně. Konkrétně vznikají pouze z existujícího objektu služby, ale mohou být přístupné i po smazání služby (uživateli musí být přístupné detaily starších nákupů, i když dané služby již neexistují).

Třída `News` definuje vlastnosti a metody pro práci se systémovými zprávami a také zprávami organizací. Třídy `SessionSet` a `Sql` jsou pomocné třídy pro práci s databází a sessions.

Metody, které v sekvenčních diagramech popisovaly události typu „ukaz seznam“ nebo „ukaz detail“ nebo „zkontroluj formulář“ jsou v implementaci schované přímo pod *template* soubory, a proto je nebylo nutné definovat jako metody tříd. Obdobným způsobem jsou zpracovány formulářové prvky, které jsou buď tvořeny přímo kódem HTML nebo pomocí ASP .NET komponent.

#### **6.1.4 Implementace**

Součástí práce je i implementace prototypu elektronického obchodu se službami příspěvkových organizací. V této podkapitole bude popsána funkčnost a struktura tohoto systému.

### 6.1.4.1 Použitá technologie

Mezi běžně používané technologie k tvorbě webových aplikací patří v současné době skriptovací jazyk PHP, ASP.NET a JSP (Java Server Pages), která je založená na programovacím jazyce Java. Pro tvorbu této aplikace byla použita technologie ASP.NET, který je součástí .NET Framework od společnosti Microsoft. Tato součást umožňuje tvorbu dynamických webových aplikací a je v mnohých ohledech podobná ostatním technologiím, přičemž nejvíce konkuruje technologii JSP. Nejvíce používané programovací jazyky využívané v této technologii jsou Visual Basic NET a C#. Tato technologie však umožňuje použít například i další programovací jazyky včetně C++. Výhodou této technologie oproti skriptovacím jazykům je kompilace napsaného kódu, který z tohoto důvodu rychleji zpracován. Velkou výhodou je i plné využití objektově orientovaného modelu na straně serveru. Díky tomu se s veškerými komponentami v kódu pracuje jako s objekty, což přináší jednodušší možnosti návrhu aplikací, čitelnost kódu nebo jeho možné znovu využití.

Samotná aplikace je psána programovacím jazykem C# a využívá databázový systém MS SQL Server 2008. Jako prostředí pro tvorbu aplikace bylo využito Visual Studio 10 Ultimate Edition, které kromě psaní a *debugování* aplikací různých druhů (například Windows Forms Application, ASP.NET, Silverlight Application, SQL Database Projects aj.) nabízí i jejich snadné namodelování. Dále byl využit značkovací jazyk XHTML 1.0 a CSS3 pro stylování jednotlivých tagů.

### 6.1.4.2 Obsah aplikace

Základem aplikace je implementovaný model tříd, který byl zobrazen v předchozí kapitole. Součástí prototypu je pak:

- kompletní *frontend* a *backend* rozhraní,
- správa a funkčnost všech uživatelských rolí definovaných v modelu,
- část (modul) umožňující práci s příspěvkovou organizací typu DDM,
- zabezpečení aplikace na straně serveru i na straně klienta,
- zjednodušující třídu pro práci s databází.

### 6.1.4.3 Rozhraní

Rozhraní celé webové aplikace je rozděleno do dvou částí, na tzv. *frontend* a *backend* část. Jako *frontend* část bývá obecně označována ta část rozhraní, ke které mají přístup i systému neznalé osoby (běžní uživatelé), kteří systém využívají pro svoji potřebu. Nutností

v této části je především pečlivé zabezpečení. V případě této aplikace jsou touto částí rozhraní ty části, ke kterým mají přístup nepřihlášení uživatelé a dále běžní uživatelé. *Backend* částí se naopak rozumí ta část, která slouží administrátorům pro správu systému. V této aplikaci je více *backend* rozhraní v závislosti na roli uživatele. Konkrétně mají odlišné možnosti přístupu administrátoři celého systému, správci jednotlivých organizací a jejich zaměstnanci.

Celá aplikace je řízena na základě URL parametrů a uživatelských rolí, které jsou přenášeny společně s instancí přihlášeného uživatele. URL adresou jsou přenášeny tyto parametry:

- parametr *section* sloužící pro orientaci ve větších částech aplikace (administrace, jednotlivé typy organizací),
- parametr *action*, který udává konkrétní akci v dané sekci,
- a parametr *id*, který v některých případech označuje konkrétní objekt (organizaci, službu apod.).

Dynamické části rozhraní jsou v celé aplikaci dvě, a to postranní levé menu a obsah. Zatímco postranní levé menu mění svůj obsah pouze v závislosti na roli přihlášeného uživatele, tak obsah je dynamicky měněn i na základě URL parametrů. Pro dynamické načítání obsahu je využívána třída `Control`, která umožňuje pomocí metody `LoadControl(string path)` načítat tzv. *web user control* soubory (soubory s příponou `.ascx`), jež mohou sloužit jako *template* soubory (tedy části rozhraní). V těchto souborech jsou pak na základě událostí volány metody modelem definovaných tříd.

Část kódu, která je zobrazena níže, náleží do souboru, ve kterém je dynamicky rozhodováno o zobrazeném obsahu. Pomocí kolekce `Params` jsou zjištěny URL parametry, podle kterých je posláno na základě podmínek načten správný `.ascx` soubor. K načtení souboru slouží procedura `Add(Control child)` z kolekce `Controls`.

```
protected void Page_Load(object sender, EventArgs e)
{
    action = Request.Params["action"];
    section = Request.Params["section"];
    Control content = LoadControl("~/Templates/Common/Homepage.ascx");
    User user = SessionSet.User;

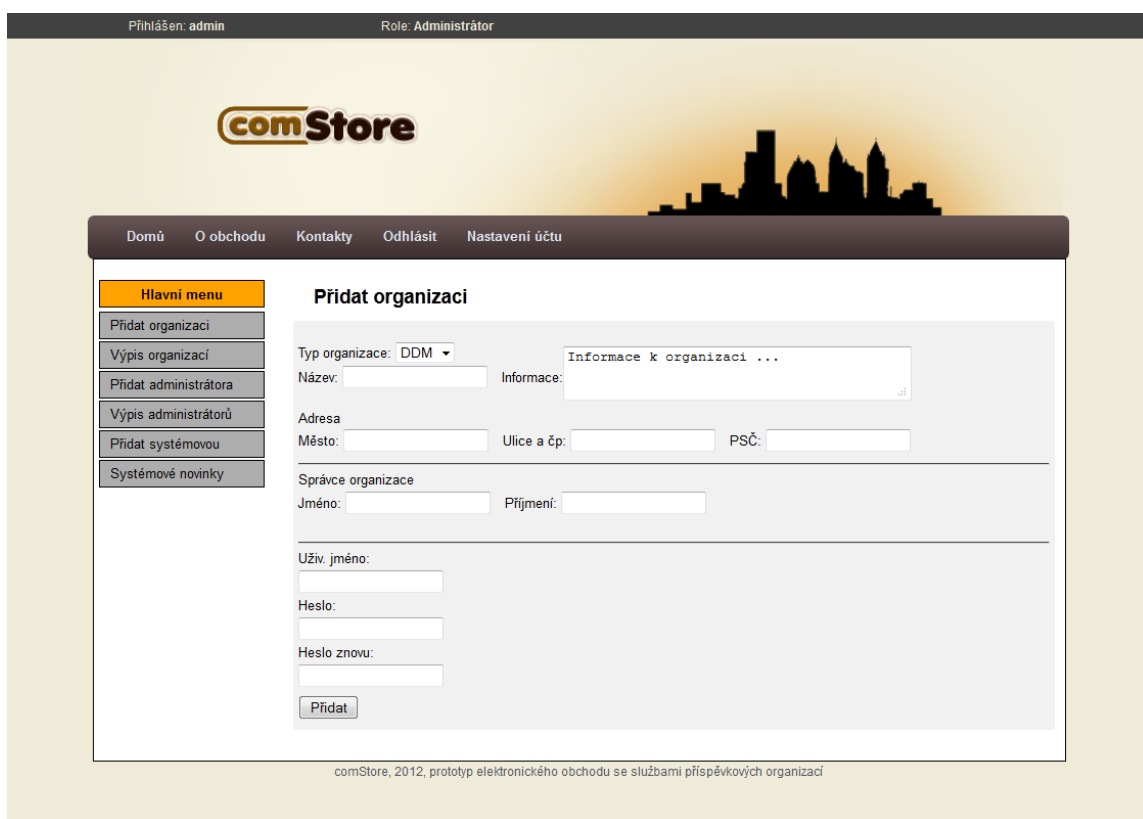
    if (user == null)
    {
        if (user.Role == 'A')
        {
            if (section == "administration")
            {
                switch (action)
                {
```

```

        case "addorg":
            content = LoadControl("~/Templates/Administration/AddOrg.ascx");
            break;
        case "showorg":
            content = LoadControl("~/Templates/Administration/ShowOrg.ascx");
            break;
        case "addadmin":
            content = LoadControl("~/Templates/Administration/AddAdmin.ascx");
            break;
        default:
            content = LoadControl("~/Templates/ErrorPage.ascx");
            break;
    }
}
}
}
Controls.Add(content);
}
}
}

```

Následující obrázek je pak ukázkou samotné aplikace.



Obrázek 18: Ukázka aplikace [zdroj: vlastní]

#### 6.1.4.4 Správa uživatelů

Akce spojené s uživateli jsou definovány ve třídě `Users`. Tato třída obsahuje množství vlastností a metod spojující všechny uživatele. Instance této třídy vzniká v případě úspěšného přihlášení do systému a je dále přenášena pomocí `sessions`. Díky tomu jsou informace o přihlášeném uživateli v programovém kódu neustále k dispozici. Instance třídy `Users` zaniká odhlášením ze systému nebo po uplynutí `session`, kterou je objekt přenášen. Část kódu zpracovávající přihlášení do systému je níže. V této části je nejprve volána statická metoda

Login(`string` nick, `string` password) třídy `Users`. Tato metoda v případě úspěšného přihlášení vrací instanci uživatele a v případě neúspěšného přihlášení `null` hodnotu. Jestliže existuje instance uživatele, pak je tato instance předána do session. V případě, že je navíc tento uživatel v roli správce organizace nebo zaměstnance, pak je vytvořena a předána do session i instance organizace spjaté s tímto uživatelem. Nakonec je stránka přesměrována na úvodní stranu obchodu.

```
user = User.Login(login_nick.Value, login_password.Value);

if (user == null)
{
    error_msg.InnerText = "Chyba při přihlášení";
}
else
{
    SessionSet.User = user;
    if (user.Role == 'S' | user.Role == 'K')
    {
        Organization org = new Organization(user.Id, true);
        SessionSet.Organization = org;
    }

    Response.Redirect("~/index.aspx");
}
```

#### 6.1.4.5 Zabezpečení aplikace

Celá aplikace je zabezpečena na dvou úrovních. První úroveň je zabezpečení především proti neoprávněným vstupům do URL adresy. Druhou úrovní je zabezpečení vstupů z formulářových prvků.

Zabezpečení proti neoprávněným vstupům do URL adresy je nutností pro veškeré webové aplikace. V této aplikaci by bez zabezpečení na této úrovni bylo například možné se jednoduchou změnou parametrů sekce či akce dostat do uživateli nepřislušných částí, ve kterých by mohlo docházet například ke smazání jednotlivých objektů nebo k jejich úpravám. Základní část tohoto zabezpečení je již definována ve výše zmíněné funkci, která rozhoduje o otevírání konkrétního obsahu na základě parametrů z URL adresy. V této funkci jsou striktně definovaná povolení ke vstupům pro jednotlivé uživatelské role. Další části tohoto zabezpečení jsou definované pro konkrétní případy v jednotlivých funkcích systému. Jedná se například o situace, ve kterých sice typ uživatelské role má přístup k dané funkci a části webu, ale daný uživatel musí být prověřen, zda neoperuje s daty, jež mu nejsou svěřeny. Příkladem této situace by mohlo být smazání služby organizace správcem jiné organizace.

Pro zabezpečení vstupů z formulářů byl na straně klienta využit framework *jQuery* objektově orientovaného skriptovacího jazyka *JavaScript*. Framework *jQuery* v mnohých

případech velmi zjednodušuje práci a je v současné době často používán nejen pro kontrolu formulářů, ale i například pro implementaci technologie *Ajax*, dynamické změny vlastností HTML elementů na základě událostí, manipulaci s objekty nebo designovým úpravám za pomoci různých efektů.

```
function isNumberValidate(form, act_stat) {
    a = true;
    str = $(form).val();

    var id = $(form).attr("id");

    if (isNaN(str)) {
        a = false;
        $(form).addClass("errInput");
        $(form).after("<span id=\"" + id + "_hint\" class=\"hint\">Vlozte cislo! </span>");
    }

    if (act_stat == false) {
        a = false;
    }
    return a;
}

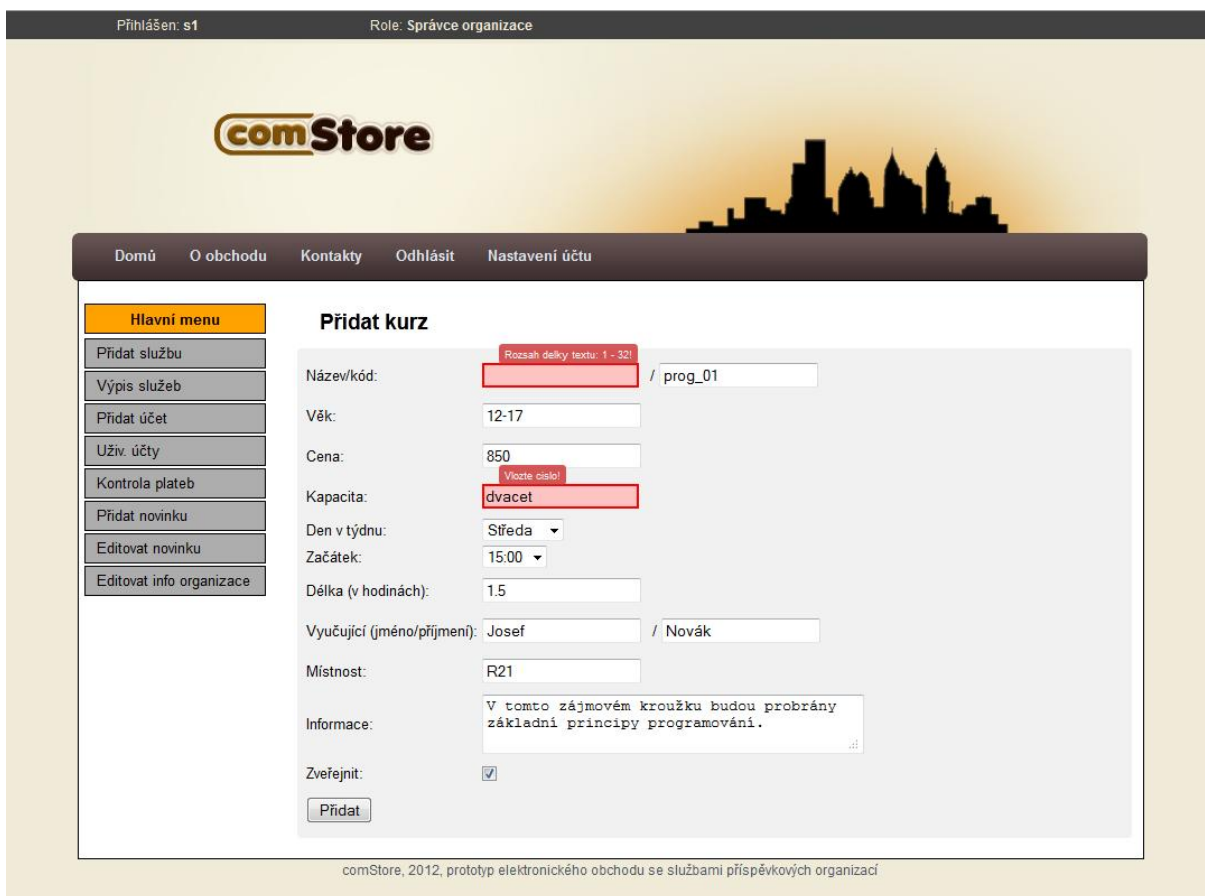
//-----
$("#price, #capacity, #length").val
(
    function () {
        value = $(this).val();
        c = b;
        b = lengthValidate(this, 0, 32, c);

        if (lengthValidate(this, 0, 32, true) == true) {
            b = isNumberValidate(this, c)
        }

        return value;
    }
);
```

Výše uvedené dvě části kódu jsou části ucelené validační funkce. Funkce `isNumberValidate(form, act_state)` ověřuje, zda uživatel vepsal číselnou hodnotu do formuláře, přičemž v závislosti na úspěchu vrací `true/false` hodnotu. V druhé části je tato funkce volána pro konkrétní formulářové prvky. Formulář je odeslán ve chvíli, kdy jsou ověřeny všechny jeho formulářové prvky.





Obrázek 19: Kontrola formulářů [zdroj: vlastní]

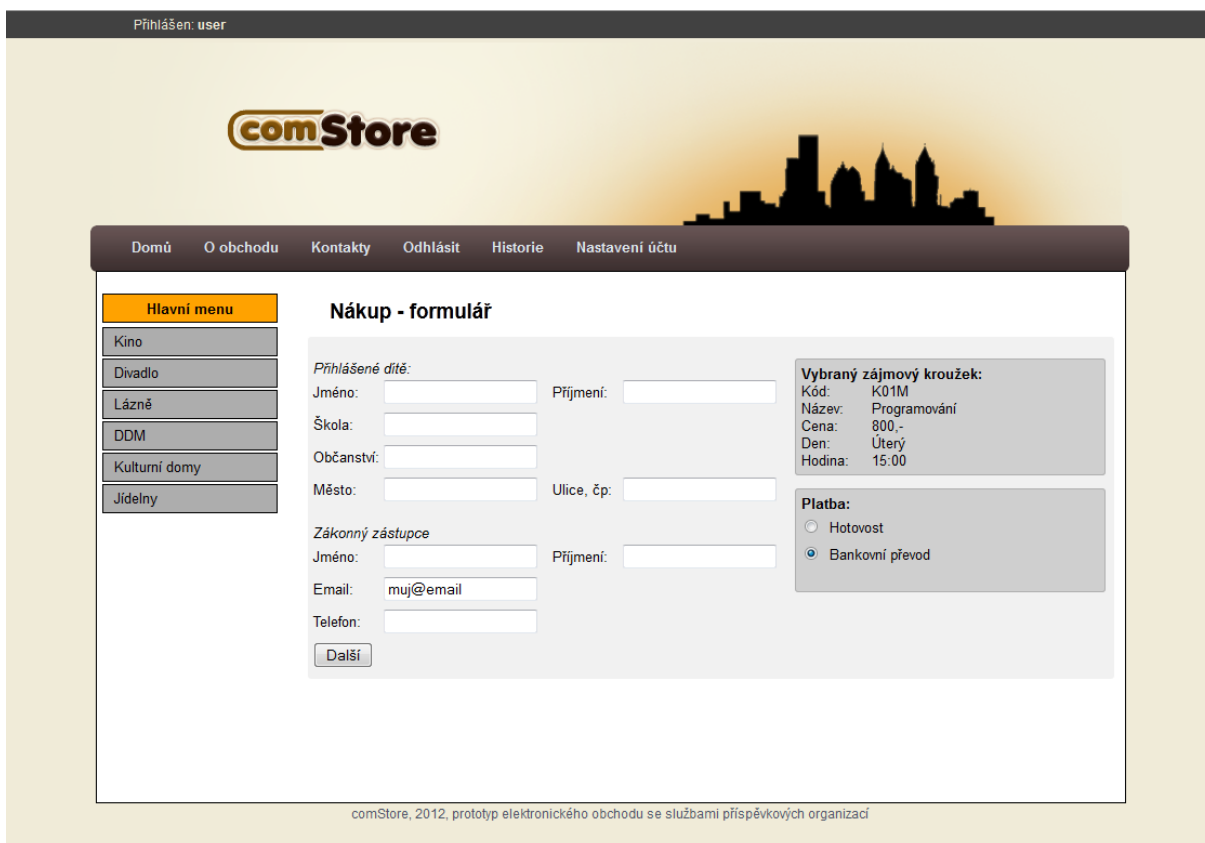
Obrázek výše zachycuje situaci nesprávně vyplněných formulářových prvků.

#### 6.1.4.6 Modul příspěvkové organizace typu DDM

Jak již bylo uvedeno, prototyp vytvořené aplikace obsahuje i funkční část pro práci s příspěvkovou organizací typu Dům dětí a mládeže. Funkčnost je zajištěna jak na straně nakupujícího uživatele, tak i v *backend* rozhraní pro správce a zaměstnance tohoto typu organizace.

Implementačně se obecně o funkčnost jednoho typu příspěvkové organizace starají dvě třídy, které rozlišují služby a platby za služby. Jak již bylo uvedeno v popisu diagramu třídy, v případě DDM jde o třídy `ServicesDDM` a `SalesDDM`, které obsahují metody pro přidávání, editaci, odstranění zájmových kroužků a druhá obstarává metodou `BuyService` samotný nákup zákazníkem.

Z pohledu funkčnosti je nepřihlášenému uživateli dovoleno nahlížet na detaily a informace k jednotlivým příspěvkovým organizacím a jejich nabízeným službám. Pro nákup nabízených služeb je nucen se uživatel přihlásit.



Obrázek 20: Ukázka aplikace – nákup služeb [zdroj: vlastní]

Backend rozhraní zahrnující správu jednotlivých organizací je obsahově rozděleno podle role přihlášeného uživatele. Zatímco uživatelské roli Zaměstnanec je dovoleno pouze kontrolovat platby a nahlížet na výpis všech nabízených služeb organizace, se kterou je spjatý, tak uživatel s právy správce organizace jsou umožněny krom těchto dvou činností i editace těchto služeb, editace informací k dané organizaci, správa zaměstnanců (i správců) organizace a jejich účtů, přidávání nových služeb a systémových novinek k dané organizaci.

#### 6.1.4.7 Databáze

Jako databázový systém byl použit systém MS SQL Server 2008, z něhož byl po definici relačních tabulek a vztahů mezi nimi vyjmut následující diagram, na kterém je možné spatřit detaily navržené databáze.



Obrázek 21: Databáze aplikace [zdroj: vlastní]

Pro zjednodušení práce s databází byla vytvořena třída `Sql`, ve které byly definovány metody pro a vlastnosti spojení s databází a provedení SQL dotazů. Metody v této třídě jsou definovány jako statické a lze je tedy okamžitě volat z kterékoliv části aplikace.

Krom klasických SQL dotazů byly v některých případech volány nadefinované SQL procedury (stored procedures), jež umožňují pomocí rozšířených možností SQL jazyka řešit náročnější úlohy. Díky tomu, že jsou tyto procedury součástí databáze, je jejich nespornou výhodou i rychlost vykonávání příkazů. Níže je uveden příklad využití SQL procedury pro přidání organizace.

```
ALTER PROCEDURE dbo.AddOrg
```

```
(
  @org_type nvarchar(16),@title nvarchar(32),@info text,@city nvarchar(32),
  @street nvarchar(32),@post_code nvarchar(12),@nick nvarchar(32),
  @password0 nvarchar(32),@password1 nvarchar(32),@first_name nvarchar(32),
  @surname nvarchar(32),@msg int OUTPUT
)
```

```
AS
```

```
DECLARE @last_id INT;
DECLARE @count INT;
SET @count = (SELECT COUNT(nick) FROM users WHERE nick = @nick);

IF @count != 0
  BEGIN
    SET @msg = 1;
  END
ELSE IF @password0 != @password1
  BEGIN
    SET @msg = 2;
  END
ELSE
```

```

        BEGIN
            SET @msg = 3;
            INSERT INTO organizations (org_type, title, city, street, post_code, info)
VALUES (@org_type, @title, @city, @street, @post_code, @info);
            SET @last_id = SCOPE_IDENTITY();
            INSERT INTO users (nick, password, first_name, surname, role, id_org) VALUES
(@nick, @password0, @first_name, @surname, 'S', @last_id);
        END

```

Parametry této SQL procedury jsou údaje přidávané příspěvkové organizace a jejího správce. Na základě přihlašovacích údajů přidávaného správce je nejdříve zjišťováno, zda již neexistuje uživatel se stejným uživatelským jménem. Poté probíhá kontrola vložených hesel k účtu tohoto správce a teprve v případě, že jsou obě předchozí podmínky kladně splněny, je přidávána příspěvková organizace a její správce. K zjištění id čerstvě přidané organizace je využita funkce `SCOPE_IDENTITY()`, jež vrací id posledního přidaného záznamu v kontextu aktuálního *scope* (tedy např. v jedné proceduře, nikoliv v kontextu celého připojení). Jelikož tato procedura vrací zprávu o stavu jejího zpracování, je nutné tuto zprávu zachytit. K tomu slouží naprogramovaná metoda vlastní třídy `Sql`, jejíž definice je níže.

```

public static int ExecuteProcedure(string procedure, SqlParameter[] parameters, string outputpar =
null)
{
    int returnvalue = 0;

    SqlCommand exec = new SqlCommand("dbo." + procedure + "");
    exec.CommandType = CommandType.StoredProcedure;
    exec.Parameters.AddRange(parameters);

    if (outputpar != null)
    {
        int outputindex = exec.Parameters.IndexOf(outputpar);
        SqlParameter output = exec.Parameters[outputindex];
        output.Direction = ParameterDirection.Output;
        exec.Parameters[outputindex] = output;
    }

    exec.Connection = Connection();
    exec.Connection.Open();

    exec.ExecuteNonQuery();

    if (outputpar != null)
    {
        returnvalue = Int32.Parse(exec.Parameters[outputpar].Value.ToString());
    }

    exec.Connection.Close();

    return returnvalue;
}

```

## **6.2 Model systému elektronického obchodování s místními poplatky**

V rámci vybraných služeb bude v následujících podkapitolách charakterizována problematika místních poplatků. Na základě úvahy o možné platbě místních poplatků skrz systém podobný elektronickému obchodu, bude navržen model tohoto systému.

### **6.2.1 Problematika místních poplatků**

Místní poplatky jsou poplatky, které obce mohou požadovat za poskytované služby. Některé poplatky mají spíše povahu zdanění nebo jsou určeny k regulaci některých aktivit. Legislativně jsou místní poplatky zohledněny v zákoně o místních poplatcích (565/1990 Sb.), ve kterém jsou specifikovány veškeré pravomoci obcí pro vybírání poplatků. Tento zákon umožňuje vybírat obcím následující poplatky:

- a) poplatek ze psů,
- b) poplatek za lázeňský nebo rekreační pobyt,
- c) poplatek za užívání veřejného prostranství,
- d) poplatek ze vstupného,
- e) poplatek z ubytovací kapacity,
- f) poplatek za povolení k vjezdu s motorovým vozidlem do vybraných míst a částí měst,
- g) poplatek za provoz systému shromažďování, sběru, přepravy, třídění, využívání a odstraňování komunálních odpadů,
- h) poplatek za zhodnocení stavebního pozemku možností jeho připojení na stavbu vodovodu nebo odstraňování kanalizace.

Výše zmíněný zákon přesně specifikuje situace, za které si obec může nárokovat platbu, uvádí maximální možnou výši jednotlivých poplatků a také stanovuje možné sankce za jejich nezaplacení. Upřesnění podmínek a výběru poplatků pak stanovují samotné obce ve svých vyhláškách. [19], [20]

### **6.2.2 Popis jednotlivých místních poplatků**

Jednotlivé druhy poplatků jsou velmi různorodé. V jejich zákonném stanovení lze nalézt velké množství výjimek. Ne všechny jsou proto vhodné k platbě skrz myšlený elektronický portál. V následujících odstavcích budou charakterizovány jednotlivé poplatky.

Poplatek za psa je placen jeho držitelem, kterým může být fyzická i právnická osoba. Poplatek se vztahuje na všechna psí plemena a jsou od něj osvobozeny např. osoby nevidomé,

osoby s těžkým zdravotním postižením, osoby provozující útulek nebo výcvik psů apod. Sazba může činit až 1500,- na jeden rok a jednoho psa. Nižší sazby (do 200 Kč/rok) platí osoby, které doloží, že jsou držiteli starobního, invalidního nebo vdoveckého důchodu. Tento poplatek patří mezi nejobvyklejší. [19], [20]

Poplatek za lázeňský nebo rekreační pobyt platí osoby, které jsou přechodně ubytovány v lázeňských místech nebo v lokalitách turistického ruchu. Poplatek vybírá a odvádí obci ubytovatel, který má povinnost vést evidenční knihu o ubytovaných fyzických osobách. Poplatek nemusí platit osoby, které prokážou jiný účel pobytu a dále osoby mladší 18 let, starší 80 let a také např. osoby s těžkým zdravotním postižením. Sazba poplatku je započítávána každým započatým dnem pobytu v dané lokalitě. Obec si však může stanovit dlouhodobé paušální částky. [19], [20]

Poplatky za užívání veřejného prostranství jsou další formou místních poplatků. Tento poplatek se týká fyzických i právnických osob, které např. za účelem výkopů, reklamy, parkování, pořádání sportovních a kulturních akcí zasahují do veřejného prostranství. Platbu nemusí platit organizátoři charitativních akcí nebo akce s veřejně prospěšnými účely. Sazba se pohybuje do 10 Kč na jeden metr čtvereční za jeden den a je v některých případech vybírána zaměstnanci obecního úřadu osobně na místě konání (například pouličních trhů). Tento poplatek má především regulační účel. [19], [20]

Poplatek ze vstupného může být v obcích vybírán z pořádaných kulturních, sportovních, prodejních a reklamních akcí. Sazba poplatku je snížena o daň z přidané hodnoty a může být stanovena až do výše 20 % z vybraného vstupného. Poplatek je placen právnickou nebo fyzickou osobou, která akci pořádá. Z platby tohoto poplatku jsou vyloučeny charitativní a veřejně prospěšné akce. [19], [20]

Poplatek z ubytovací kapacity může být vybírán v obcích v zařízeních, která slouží pro přechodné ubytování za úplatu. Obdobně jako u poplatku za lázeňský a rekreační pobyt platí tento poplatek ubytovatel, který má povinnost vést evidenční knihu. Sazba může činit až 6 Kč za každé využití lůžko za jeden den. Poplatek nemusí platit zdravotnická a lázeňská zařízení a dále zařízení se sociálním a charitativním účelem. [19], [20]

Poplatek za povolení k vjezdu s motorovým vozidlem do vybraných míst a částí měst platí fyzické nebo právnické osoby, kterým byl povolen vjezd motorovým vozidlem do vybraných částí obce. Tento poplatek se netýká osob (včetně osob blízkých – manželé, děti), které mají v těchto vybraných částech obce trvalý pobyt nebo zde vlastní nemovitost.

Sazba je placena za vydání povolení k vjezdu a může být stanovena až do výše 20 Kč na jeden den. [19], [20]

Poplatek za provoz systému shromažďování, sběru, přepravy, třídění, využívání a odstraňování komunálních odpadů je placen fyzickou osobou, která má trvalý pobyt v dané obci a dále fyzická osoba, která má v dané obci ve svém vlastnictví stavbu určenou k rekreaci (není v ní evidován trvalý pobyt žádné fyzické osoby). Sazba z tohoto poplatku je tvořena dvěma částmi. První je pevná a činí 250 Kč na osobu za jeden kalendářní rok a druhá část je stanovena na základě skutečných nákladů obce na sběr a svoz netříděného odpadu a může činit rovněž až 250 Kč na osobu ročně. V případě změny trvalého bydliště je uhrazena částka odpovídající počtu kalendářních měsíců pobytu. [19], [20]

Poplatek za zhodnocení stavebního pozemku možností jeho připojení na stavbu vodovodu nebo odstraňování kanalizace, kterou vybudovala obec, platí vlastníci pozemků v dané obci. Sazba musí být maximálně tvořena rozdílem hodnoty pozemku bez možnosti na připojení k obecní kanalizační soustavě a s možností k připojení k obecní kanalizační soustavě. Výše sazby je pak dána obcí v obecně závazné vyhlášce. [19], [20]

### **6.2.3 Současná situace**

Ze zákona o místních poplatcích vyplývá, že si každá obec může vybírat výše popsané poplatky se svými vlastními pravidly, které si stanoví ve svých vyhláškách. Omezení, kterými se obce musí řídit, jsou pevně dané zákonem. Tato omezení se týkají zejména maximální výše sazby poplatku pro určité kategorie plátců (v případě, že jsou nějaké kategorie stanoveny) a také o výjimky fyzických a právnických osob, které za určitých předpokladů nemusí jednotlivé poplatky platit.

V současné době je často již možné učinit podání nejen ústně či písemně, ale i elektronickou cestou prostřednictvím datové schránky (v tomto případě je nutné zprávu opatřit uznatelným elektronickým podpisem). Osoby splňující zákonem popsané výjimky musí vždy na obecním úřadě, ke kterému jsou přihlášeny, prokazatelně doložit, že splňují dané podmínky. Za takové doložení je bráno předložení konkrétních ověřitelných dokumentů, a proto je mnohdy nutné, aby tyto osoby byly osobně přítomny při platbě nebo při uplatnění nároků na výjimky.

Platby lze uskutečňovat osobně na obecních úřadech, složenkou nebo pomocí elektronického bankovníctví. Některé úřady pomáhají občanům u některých poplatků např. zasíláním složenky.

U větších měst mají místní poplatky na starost jednotlivé úřady městských obvodů. Obecní úřady a úřady městských obvodů spolupracují s jinými orgány veřejné správy pro důkladnější kontrolu vybíraných poplatků. Příkladem může být evidence přistěhovaných občanů a s tím související kontrola podání přihlášky pro platbu za odvoz komunálního odpadu.

Technicky jsou databáze plátců poplatků uloženy na lokálních serverech obecních úřadů. K těmto databázím mají přístup pouze pověřeni zaměstnanci těchto úřadů, a to prostřednictvím rozličných softwarových produktů, které umožňují zpracování dat souvisejících s místními poplatky. Přístup k těmto databázím je velmi omezený, a to zejména z toho důvodu, že obsahují citlivé informace o plátcích (rodná čísla, čísla občanského průkazu apod.), která podléhají zákonu o ochraně osobních údajů (zákon č. 101/2000 Sb.).

#### **6.2.4 Požadavky na systém**

Základní požadavky na myšlený systém budou totožné jako v případě modelu pro prodej služeb příspěvkových organizací. Tedy bezpečnost přístupu k datům, informovanost uživatelů, jednodušnost celého systému a jeho rozhraní. Bezpečnost přístupu k datům se týká zejména důsledného zneprístupnění informací o ostatních uživateli. Informovanost uživatelů je myšlena obdobně jako u předchozího modelu, a to možnost zveřejňování změn v systému (např. změna sazeb a kategorií sazeb u poplatků). Druhou složkou informovanosti uživatelů jsou historická data jejich vlastních nákupů, v případě tohoto modelu minulých plateb.

Nezbytnou součástí analýzy tohoto systému je výběr poplatků, které se pro elektronický „prodej“ hodí nejvíce. Z charakteristiky jednotlivých poplatků vyplývá, že by výběr a platba některých skrz „elektronický obchod“ byla velmi náročná, a to zejména z důvodu špatné návaznosti na pozdější případné kontroly ze strany veřejných orgánů. Jiné poplatky (např. poplatek za lázeňský nebo rekreační pobyt) jsou navíc využitelné pouze v některých obcích. Jejich využitelnost by tedy v takovém systému byla velmi malá.

Většina obcí však v zásadě vždy využívá poplatek za psa, poplatek za svoz odpadu a poplatek za užívání veřejného prostranství. Všechny tyto poplatky bývají kontrolovány samotným obecním úřadem bez účasti jiných orgánů veřejné správy. Nevýhodou poplatku za užívání veřejného prostranství pro možnost platby skrz systém elektronického obchodu je časté rozlišování kategorií tohoto poplatku podle katastrálních území. Tato skutečnost by v systému znamenala nutnost identifikace katastrálního území na základě adresy. Zbylé



dva poplatky by však nevyžadovaly větší nároky na myšlený systém, a tak je možné tyto poplatky zahrnout do systému.

Požadavkem systému u obou poplatků je nutnost jejich dynamické správy, tak aby bylo možné např. měnit sazbu nebo kategorie plátců.

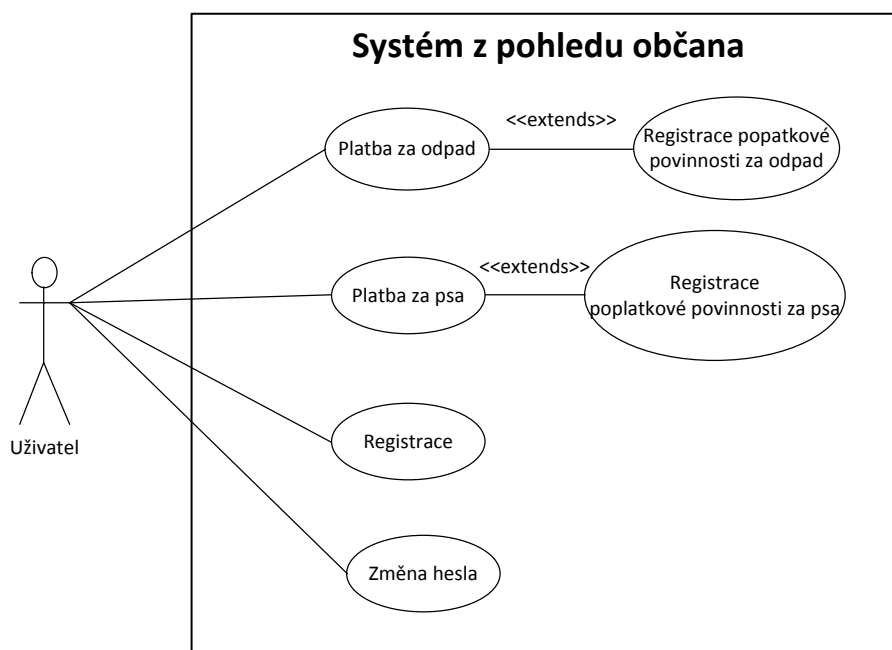
### **6.2.5 Analýza systému**

Na základě předchozí analýzy bude v následujících podkapitolách naznačen návrh modelu myšleného systému, který by umožňoval platbu poplatků za psa a za odpad.

#### **6.2.5.1 Use Case**

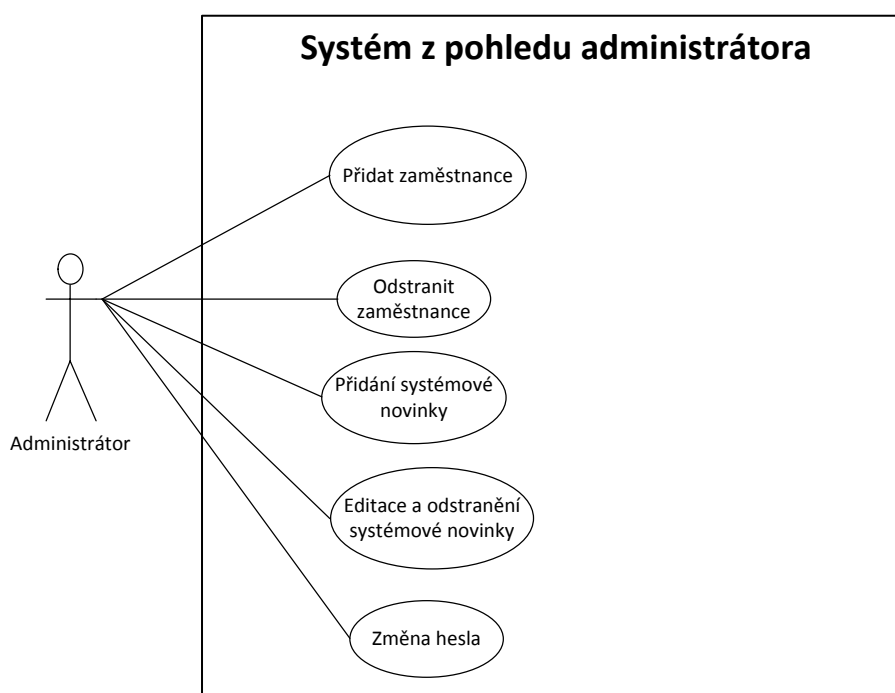
Stejně jako v případě prvního modelu týkajícího se příspěvkových organizací budou Use Case diagramy rozděleny podle uživatelských rolí.

Uživatel symbolizující „běžného“ občana má povinnost být v systému registrován, chce-li využít možnosti platby skrz webovou aplikaci. Po úspěšné registraci a nalogování do systému má posléze možnost zaplatit poplatek za odpad nebo za psa. K platbě poplatku za psa je však nutné, aby byl tento pes uživatelem zaregistrován. Obdobná situace nastává u platby poplatku za sběr odpadu, kde musí být uživatelem registrováno každé platební místo. Tyto situace naznačuje v následujícím Use Case diagramu vazba *extends*. Poslední možností je změna hesla k účtu pro přihlášení do systému. Nepřihlášenému uživateli není povolena žádná akce kromě možnosti registrování do systému, nahlédnutí do obecných informací k systému a možnosti číst systémové novinky. Všem aktivitám (kromě aktivity Registrace) ve všech následujících Use Case diagramech předchází aktivita „Přihlášení do systému“, která se všemi ostatními aktivitami ve vazbě typu *includes*.



Obrázek 22: Use Case – System z pohledu zákazníka [zdroj: vlastní]

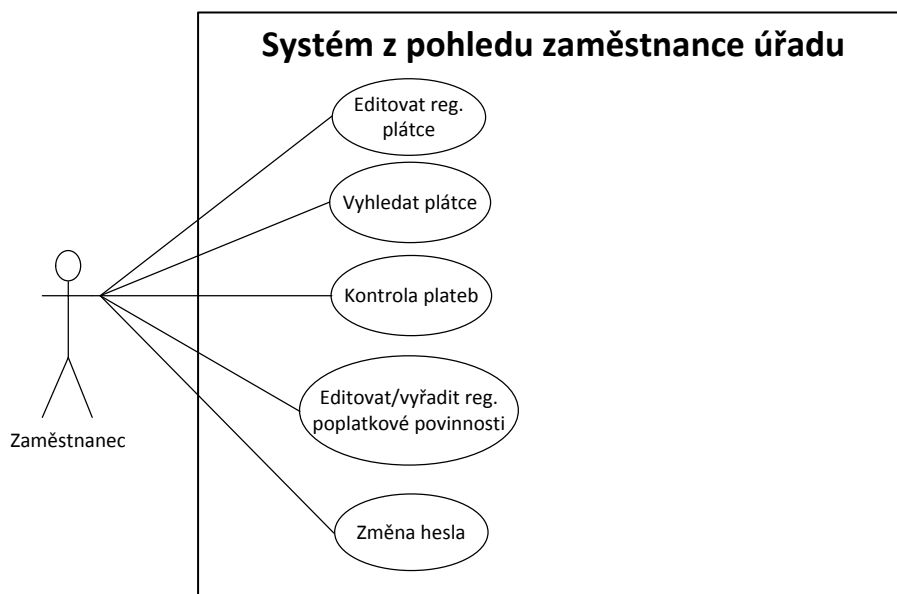
Administrátor celého systému má na starost pouze správu zaměstnanců úřadu a dalších administrátorů přidáváním a odstraňováním jejich účtů. Zároveň může přidávat, editovat a odstraňovat systémové novinky (novinky týkající se samotné funkčnosti systému).



Obrázek 23: Use Case – System z pohledu administrátora [zdroj: vlastní]

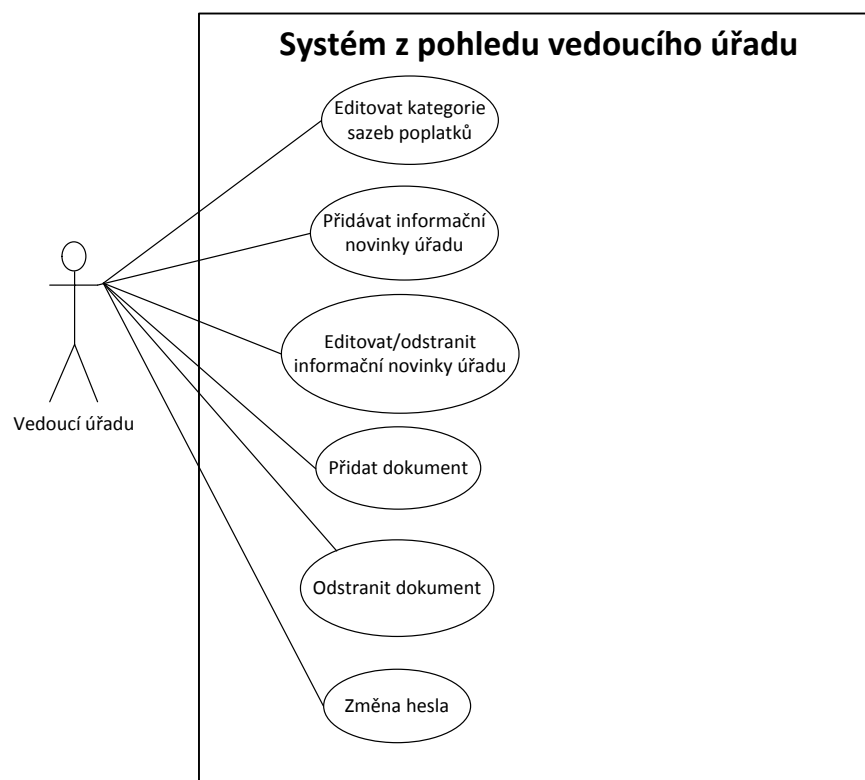
Řadový zaměstnanec úřadu má především možnost editovat informace zaregistrovaných plátců (např. při změně bydliště) a kontrolovat zaplacení aktuálních plateb. Jeho role mu také

dovoluje vyhledat a zjišťovat veškeré údaje o registrovaných občanech a editovat (případně vyřadit) registrované poplatkové povinnosti.



Obrázek 24: Use Case – Systém z pohledu zaměstnance úřadu [zdroj: vlastní]

Úloha vedoucího úřadu v modelovaném systému zastává především informační roli na úřadu. V této roli je tedy možné informovat občany novinkami týkající se správy úřadu a poplatků a také zveřejňováním důležitých dokumentů (například vyhlášky týkající se poplatků). V případě změny sazby některé kategorie poplatků má možnost tuto změnu dynamicky zavést do systému.



Obrázek 25: Use Case – Systém vedoucího úřadu [zdroj: vlastní]

### 6.2.5.2 Scénáře

Scénáře některých činností tohoto modelu jsou naprosto shodné se scénáři činností z modelu systému pro obchodování se službami příspěvkových organizací. Jedná se zejména o činnosti spjaté s přihlášením do systému, změnou hesla a přidáváním zaměstnanců. Tyto činnosti zde z těchto důvodů nebudou popsány.

Tabulka 9: Scénář – Registrace poplatkové povinnosti za odpad/chov psa [zdroj: vlastní]

<b>Use Case</b>	<b>Registrace poplatkové povinnosti za odpad/chov psa</b>	
<b>Aktér</b>	Uživatel	
<b>Popis</b>	Cílem této operace je zaregistrování poplatkové povinnosti do systému	
<b>Vstupy</b>	Seznam uživatelů	
<b>Výstupy</b>	Poplatková povinnost byla zaregistrována	
<b>Hlavní scénář</b>		
<b>Krok</b>	<b>Role</b>	<b>Akce</b>
1.1.	Uživatel	Klikne na odkaz pro registraci poplatkové povinnosti poplatku podle své potřeby
1.2.	System	Zobrazí Registrační formulář
1.3.	Uživatel	Vyplní registrační formulář a odešle jej.
1.4.	System	Zkontroluje data odeslaná reg. formulářem.
1.5.	System	V případě správnosti odeslaných dat zaeviduje poplatkovou povinnost.
<b>Alternativní scénáře</b>		
1.4.1	System	V případě špatného vyplnění formuláře vrátí uživatele na stránku formuláře a informuje ho o chybě.

Tabulka 10: Scénář – Zaplacení poplatkové povinnosti za odpad/chov psa [zdroj: vlastní]

<b>Use Case</b>	<b>Zaplacení poplatkové povinnosti za odpad/chov psa</b>	
<b>Aktér</b>	Uživatel	
<b>Popis</b>	Cílem této operace je zaplacení poplatkové povinnosti.	
<b>Vstupy</b>	Seznam uživatelů	
<b>Výstupy</b>	Poplatková povinnost byla zaplacená	
<b>Hlavní scénář</b>		
<b>Krok</b>	<b>Role</b>	<b>Akce</b>
1.1.	Uživatel	Klikne na odkaz pro platbu poplatkové povinnosti poplatku podle své potřeby
1.2.	Systém	Zobrazí poplatkové povinnosti k možnému zaplacení.
1.3.	Uživatel	Vybere poplatkovou povinnost, kterou chce uhradit
1.4.	Systém	Zobrazí detail vybrané poplatkové povinnosti
1.5.	Uživatel	Potvrdí platbu a zaplatí ji prostřednictvím elektronického bankovníctví.
<b>Alternativní scénáře</b>		

Tabulka 11: Scénář – Editovat registrovaného plátce [zdroj: vlastní]

<b>Use Case</b>	<b>Editovat registrovaného plátce</b>	
<b>Aktér</b>	Zaměstnanec úřadu	
<b>Popis</b>	Cílem této operace je editace informací reg. vybraného plátce.	
<b>Vstupy</b>	Seznam uživatelů	
<b>Výstupy</b>	Informace o konkrétním plátcovi byly editovány.	
<b>Hlavní scénář</b>		
<b>Krok</b>	<b>Role</b>	<b>Akce</b>
1.1.	Zaměstnanec	Vyhledá konkrétního plátce na základě činnosti „Vyhledat plátce“
1.2.	Systém	Zobrazí detail plátce s citovatelnými formuláři.
1.3.	Zaměstnanec	Edituje v zobrazených formulářích požadované změny a odešle formulář.
1.4.	Systém	Zkontroluje data odeslaná reg. formulářem.
1.5.	Systém	V případě správnosti odeslaných dat zaeviduje poplatkovou povinnost.
<b>Alternativní scénáře</b>		
1.4.1	Systém	V případě špatného vyplnění formuláře vrátí zaměstnance na stránku formuláře a informuje ho o chybě.

Tabulka 12: Scénář – Vyhledat plátce [zdroj: vlastní]

<b>Use Case</b>	<b>Vyhledat plátce</b>	
<b>Aktér</b>	Zaměstnanec úřadu	
<b>Popis</b>	Cílem této operace je vyhledání konkrétního plátce zaregistrovaného v systému a vypsat detailní informace o něm.	
<b>Vstupy</b>	Seznam uživatelů	
<b>Výstupy</b>	Byly zobrazeny detailní informace hledaného plátce.	
<b>Hlavní scénář</b>		
<b>Krok</b>	<b>Role</b>	<b>Akce</b>
1.1.	Zaměstnanec	Klikne na odkaz směřující na formuláře pro vyhledávání občanů v systému.
1.2.	Systém	Zobrazí formulář pro vyhledávání.
1.3.	Zaměstnanec	Vypíše známé informace do formuláře a odešle jej.
1.4.	Systém	Vyhledá všechny relevantní osoby odpovídající parametrům odeslaných formulářem.
1.5.	Zaměstnanec	Z nabídky vyhledaných plátců vybere ten záznam, který požadoval.
<b>Alternativní scénáře</b>		
1.5.1	Zaměstnanec	V případě, že systém nenašel hledanou osobu, opakuje vyhledávání.

Tabulka 13: Scénář – Kontrola platby [zdroj: vlastní]

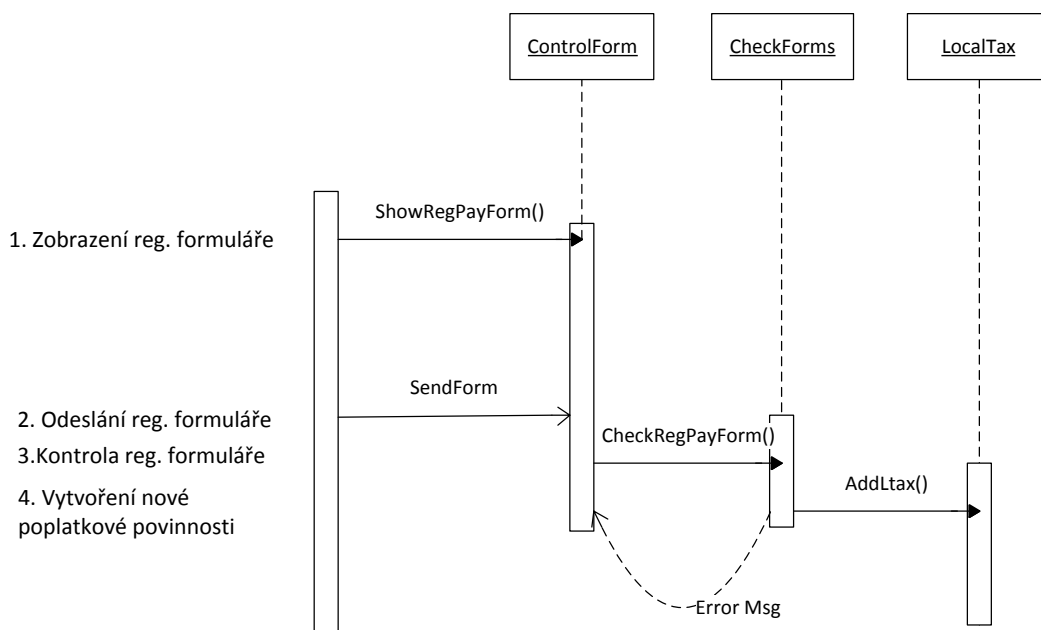
<b>Use Case</b>	<b>Kontrola platby</b>	
<b>Aktér</b>	Zaměstnanec úřadu	
<b>Popis</b>	Cílem této operace je potvrzení zaplacení poplatku.	
<b>Vstupy</b>	Seznam uživatelů, seznam příchozích plateb	
<b>Výstupy</b>	Platba za poplatek byla potvrzena.	
<b>Hlavní scénář</b>		
<b>Krok</b>	<b>Role</b>	<b>Akce</b>
1.1.	Zaměstnanec	Klikne na odkaz směřující na výpis aktuálně zaplacených poplatků.
1.2.	Systém	Zobrazí výpis nezkontrolovaných plateb.
1.3.	Zaměstnanec	Klikne na detail vybrané platby.
1.4.	Systém	Zobrazí detail vybrané platby a jeho plátce.
1.5.	Zaměstnanec	Zkontroluje, zda platba na úřad opravdu došla a potvrdí platbu.
<b>Alternativní scénáře</b>		

Tabulka 14: Scénář – Editace kategorie sazeb poplatků [zdroj: vlastní]

<b>Use Case</b>	<b>Editace kategorie sazeb poplatků</b>	
<b>Aktér</b>	Vedoucí úřadu (dále jako „Vedoucí“)	
<b>Popis</b>	Cílem této operace je změnit vybranou sazbu poplatku nebo jeho kategorie v systému.	
<b>Vstupy</b>	Seznam uživatelů, seznam kategorií sazeb místních poplatků	
<b>Výstupy</b>	Změna sazby poplatku byla provedena.	
<b>Hlavní scénář</b>		
<b>Krok</b>	<b>Role</b>	<b>Akce</b>
1.1.	Vedoucí	Klikne na odkaz směřující na změnu sazeb poplatků.
1.2.	Systém	Zobrazí výpis poplatků a jejich kategorií.
1.3.	Vedoucí	Vybere poplatek, jehož sazbu lze změnit.
1.4.	Systém	Zobrazí detail poplatku s formulářem pro změnu jeho sazby.
1.5.	Vedoucí	Vepíše novou sazbu a potvrdí.
1.6.	Systém	Zobrazí kontrolně navrhovanou změnu sazby.
1.7.	Vedoucí	Potvrdí změnu.
1.8.	Systém	Zapíše změnu sazby do databáze.
<b>Alternativní scénáře</b>		
1.7.1	Vedoucí	Nepotvrdí změnu.
1.7.2	Systém	Zobrazí opětovně původní formulářové pole vybraného poplatku pro změnu jeho sazby.

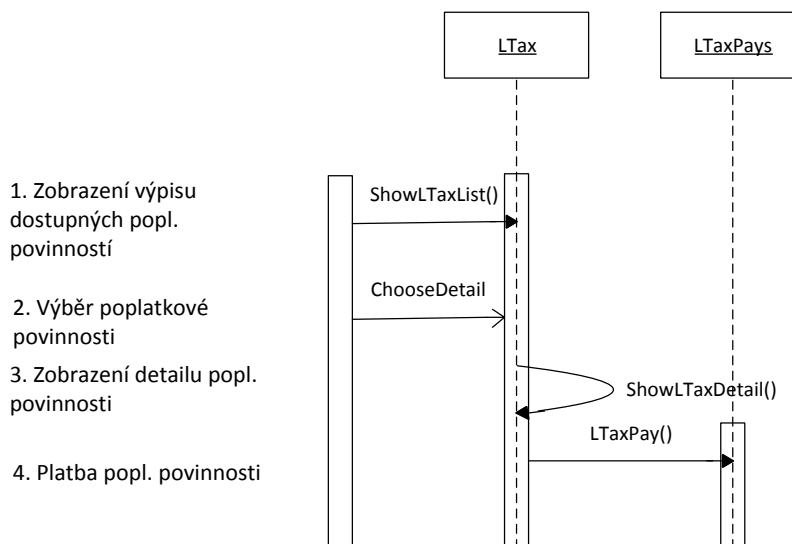
### 6.2.5.3 Sekvenční diagramy

#### Registrace poplatkové povinnosti



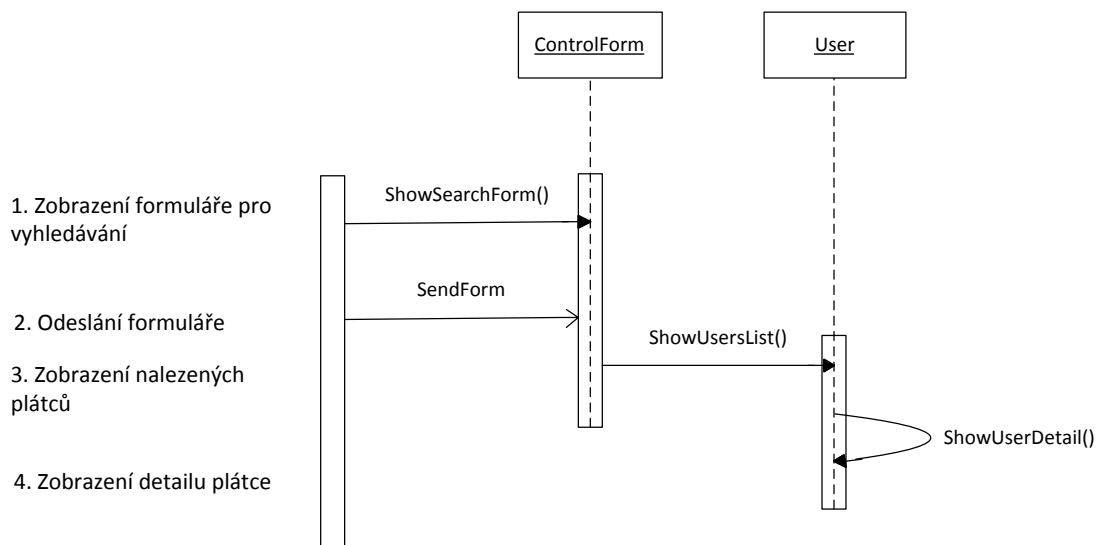
Obrázek 26: Sekvenční diagram – Registrace poplatkové povinnosti [zdroj: vlastní]

#### Zaplacení poplatkové povinnosti



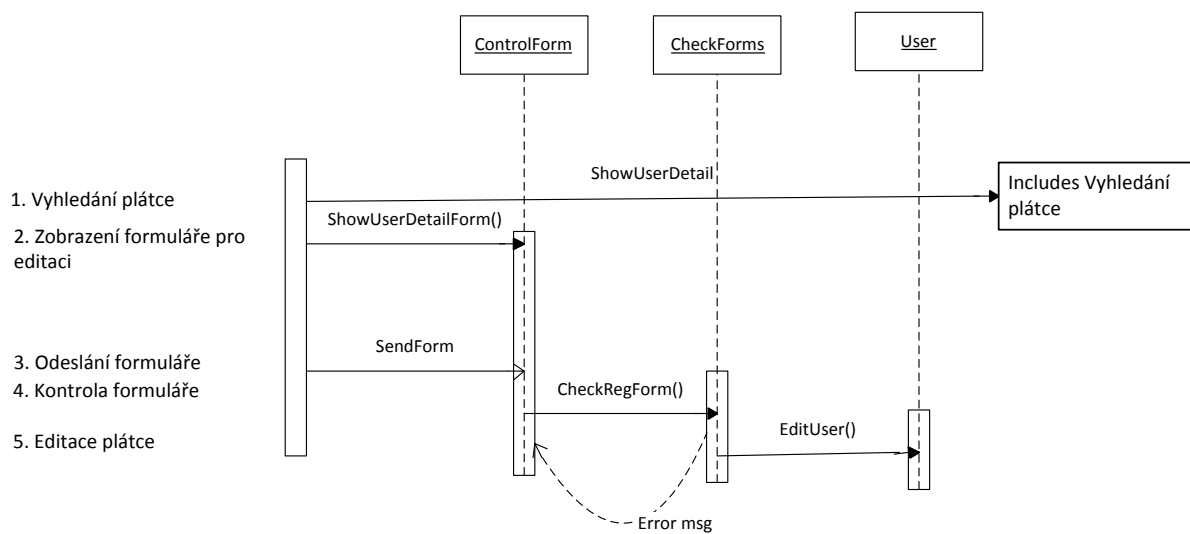
Obrázek 27: Sekvenční diagram – Zaplacení poplatkové povinnosti [zdroj: vlastní]

## Vyhledání plátce



Obrázek 28: Sekvenční diagram – Vyhledání plátce [zdroj: vlastní]

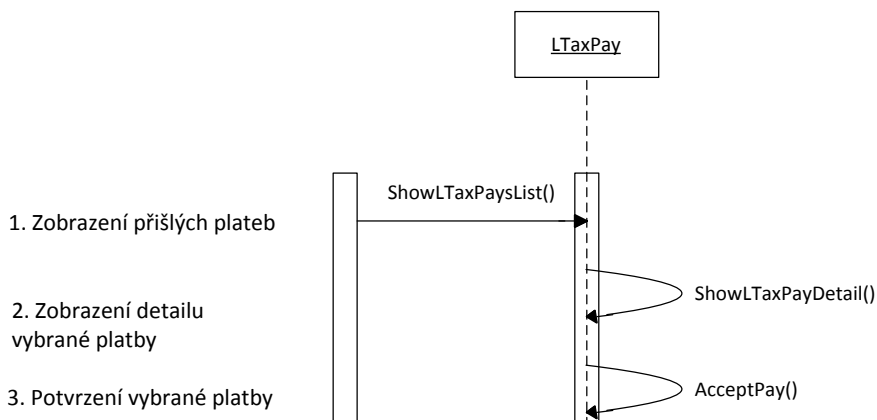
## Editace registrovaného plátce



Obrázek 29: Sekvenční diagram – Editace registrovaného plátce [zdroj: vlastní]

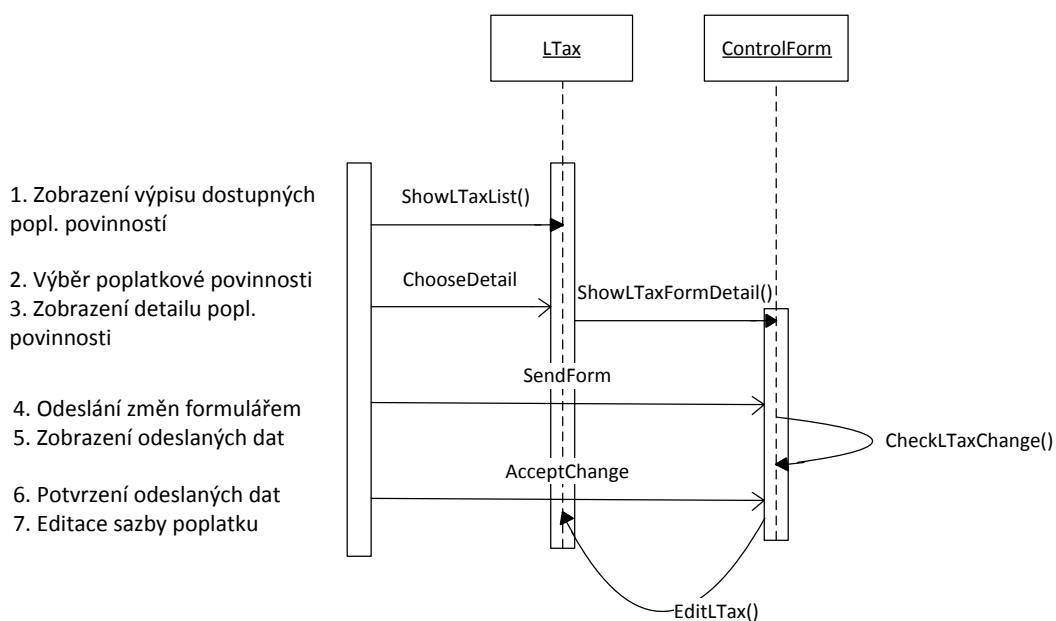


## Kontrola platby



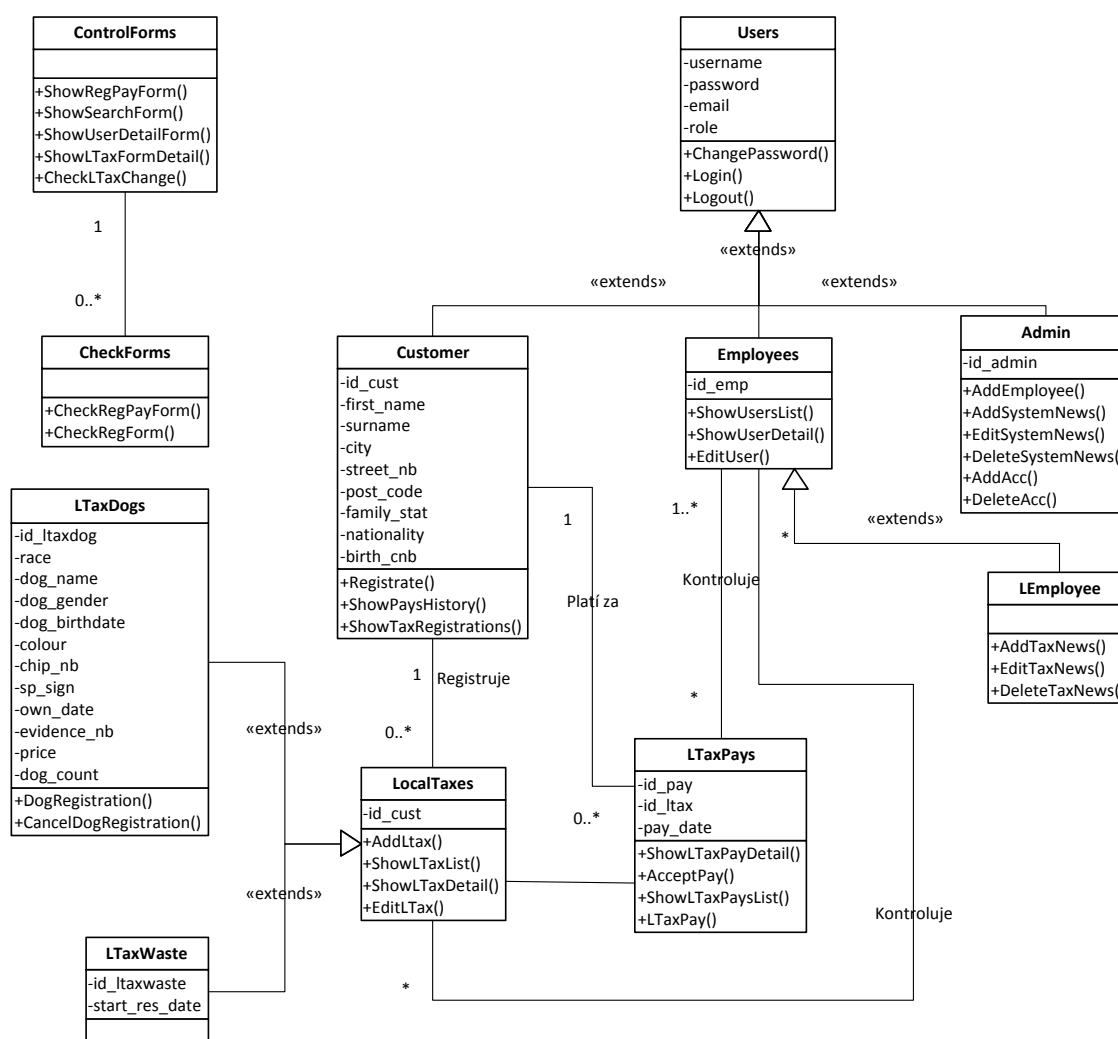
Obrázek 30: Sekvenční diagram – Kontrola platby [zdroj: vlastní]

## Editace kategorie plateb poplatků



Obrázek 31: Sekvenční diagram – Editace kategorií plateb poplatků [zdroj: vlastní]

## 6.2.5.4 Class diagram



Obrázek 32: Class diagram [zdroj: vlastní]

Výše navržený diagram tříd lze rozdělit na několik logicky rozdělených částí. Prvním celkem jsou třídy spjaté s uživatelskými rolemi. Třídy všech uživatelských rolí jsou potomky třídy *Users*, která zahrnuje základní vlastnosti metody pro přístup k uživatelským účtům. Třída *Customer* popisuje uživatelskou roli zákazníka a obsahuje metody umožňující registraci, zobrazení výpisu plateb poplatků a registrovaných poplatkových povinností. Třída *Employees* definuje vlastnosti a funkce všech zaměstnanců úřadu. Mezi které patří zobrazení seznamu a detailu registrovaných plátců a také jejich editaci. Od této třídy je zděděna třída *LEmployee*, která popisuje vedoucího úřadu a v její definici se nachází funkce pro přidávání|editaci|odstranění systémových zpráv týkajících se úřadu. Uživatelskou roli administrátora definuje třída *Admin*, jejíž metody umožňují přidávání|editaci|odstranění systémových zpráv týkajících se systému a také metod pro správu uživatelů (zaměstnanců úřadu).

Za druhý velký celek by se dala považovat část diagramu zahrnující třídy definující poplatkové povinnosti a jejich uskutečněné platby. Třída *Local Taxes* definuje poplatkové povinnosti a předává své vlastnosti do dalších dvou děděných tříd zatupujících jednotlivé typy poplatkových povinností (třída *LTaxDogs* – poplatková povinnost za psa, třída *LTaxWaste* – poplatková povinnost za odpad). Tyto třídy definují především vznik a zánik poplatkových povinností, zobrazení jejich seznamu a detailů a také úpravy jejich kategorií. Třída *LTaxPays* pak reprezentuje samotné platby registrovaných poplatkových povinností.

Posledním celkem by mohly být systémové třídy definující práci s formulářovými prvky a jejich kontrolu, práci s databází a dalšími ryze implementačními prvky. Tyto třídy ve většině případů komunikují se všemi ostatními třídami.

Z databázového hlediska lze zamýšlet dva přístupy systému, a to:

- napojení se na existující databázi obecního úřadu,
- tvorba nové (druhé) databáze.

Problémem prvního přístupu je to, že k těmto databázím je omezený přístup a není tedy známa jejich struktura. V případě možného přístupu k těmto databázím by vzniknul další problém, a to velká rozdílnost jednotlivých systémů (tedy i databází), které využívají jednotlivé obce.

Druhým zamýšleným přístupem by mohla být tvorba další databáze, která by byla samostatně spojena s webovou aplikací elektronického obchodu. Jediným problémem této možnosti by byla mírně větší administrativní náročnost ze strany obecního úřadu, který by musel kontrolovat odlišné dvě databáze s poplatníky. I přes tento problém se jeví druhý pohled jako dostupnější pro zamýšlený systém, byť ideální případem by bylo napojení na stávající databáze.

## ZÁVĚR

Cílem diplomové práce byla analýza možností využití principu elektronického obchodování na portálech veřejné správy a návrh modelu pro vybraný subjekt veřejné správy.

Nejprve bylo nutné zmapovat služby veřejné správy, které by byly vhodné k poskytování prostřednictvím systému elektronického obchodování, a teprve na základě analýzy těchto služeb byl takový systém navržen.

První vybraná oblast vhodná k možnému uplatnění elektronického obchodování v rámci veřejné správy se dotýkala služeb některých příspěvkových organizací obcí. Druhou vybranou oblastí byla zvolena možnost uplatnit některé principy elektronického obchodu pro systém operující s některými místními poplatky.

Analýza zvolených služeb probíhala zejména ústními konzultacemi se zaměstnanci příslušných úřadů městských obvodů v Pardubicích a dále také především z legislativy týkající se vybraných oblastí. K návrhu modelu této aplikace byl využit modelovací jazyk UML.

Vzhledem k tomu, že model elektronického obchodu se službami příspěvkových organizací není v žádném rozporu se současnou legislativou, byl vytvořen prototyp naprogramovaný programovacím jazykem ASP. NET (C#), který zahrnuje veškerou správu uživatelů tohoto systému a také modul pro prodej služeb příspěvkových organizací typu „dům dětí a mládeže“. Implementace tohoto prototypu navazuje na vytvořený model a je jí věnována jedna kapitola, ve které nechybí obrázky z aplikace, ani úryvky kódů vybraných programových částí. Celá aplikace je k práci přiložena na CD.

Přínos diplomové práce je:

- nalezení dalších možností uplatnění principů elektronického obchodování pro služby veřejné správy,
- zjištění reálných možností uplatnění daného nástroje (opakované konzultace a verifikace na úřadech VS),
- vytvořený model systému je vhodným nástrojem zejména pro obce (zatímco např. Czech POINT poskytuje služby občanům a firmám).

## POUŽITÁ LITERATURA

- [1] *Český statistický úřad* [online]. 2011 [cit. 2012-03-20]. Dostupné z: [http://www.czso.cz/csu/redakce.nsf/i/domacnosti\\_a\\_jednotlivci](http://www.czso.cz/csu/redakce.nsf/i/domacnosti_a_jednotlivci)
- [2] *BusinessInfo.cz: Oficiální portál pro podnikání a export* [online]. 7.6.2010 [cit. 2012-04-15]. Dostupné z: <http://www.businessinfo.cz/cz/clanek/orientace-v-pravnich-ukonech/elektronicky-obchod-opu/1000818/7013/>
- [3] VACULÍK, Josef, Radim KERŤ, Zdeněk HAVRAN, Aleš BERKA a Michal KUBĚNKA. *Marketingové řízení*. 2005. vyd. Pardubice: Univerzita Pardubice, 2005. ISBN 80-7194-765-2.
- [4] *Strukturovaný přístup k analýze a návrhu IS* [online]. 2012 [cit. 2012-04-15]. Dostupné z: <http://home1.vsb.cz/~dan11/aps/Strukturovany%20pristup%20k%20navrhu%20systemu.pdf>
- [5] ŘEPA, Václav. VÝVOJOVÉ TRENDY METODIK VÝVOJE INFORMAČNÍCH SYSTÉMŮ - VÝZVA BPR. [online]. č. 1 [cit. 2012-2-19]. Dostupné z: <http://nb.vse.cz/~repa/veda/EurOpen99%20Paper.pdf>
- [6] BRUCKNER, Tomáš, Jiří VOŘÍŠEK a Alena BUCHALCEVOVÁ. *Tvorba informačních systémů: Principy, metodiky, architektury*. Praha: GRADA Publishing, a.s., 2012. ISBN 978-80-247-4153-6.
- [7] *Databázové a prezentační systémy* [online]. 2007 [cit. 2012-02-20]. Dostupné z: [https://it.fd.cvut.cz/k614daps/prednasky/dps\\_02\\_teorie.htm](https://it.fd.cvut.cz/k614daps/prednasky/dps_02_teorie.htm)
- [8] *Navrhování systémů řízení* [online]. 2006 [cit. 2012-04-15]. Dostupné z: <http://www.vns.wz.cz/4.php>
- [9] NEUSTADT, Illa a Arlow JIM. *UML 2 a unifikovaný proces vývoje aplikací*. Praha: Computer Press, 2007-03-23. ISBN 9788025115039.
- [10] *Finanční úřady v regionech* [online]. 2011 [cit. 2012-04-16]. Dostupné z: <http://www.financi-urady.cz/>
- [11] *Statnisprava.cz* [online]. 2011 [cit. 2012-04-16]. Dostupné z: <http://www.statnisprava.cz/rstsp/ciselniky.nsf/i/d0027>
- [12] *Statnisprava.cz* [online]. 2011 [cit. 2012-04-16]. Dostupné z: <http://www.statnisprava.cz/rstsp/ciselniky.nsf/i/d0237>
- [13] *Statnisprava.cz* [online]. 2011 [cit. 2012-04-16]. Dostupné z: <http://www.statnisprava.cz/rstsp/ciselniky.nsf/i/d0056>

- [14] *Statnisprava.cz* [online]. 2011 [cit. 2012-04-16]. Dostupné z: <http://www.statnisprava.cz/rstsp/ciselniky.nsf/i/d0061>
- [15] *Statnisprava.cz* [online]. 2011 [cit. 2012-04-16]. Dostupné z: <http://www.statnisprava.cz/rstsp/ciselniky.nsf/i/d0045>
- [16] *Statnisprava.cz* [online]. 2011 [cit. 2012-04-16]. Dostupné z: <http://www.statnisprava.cz/rstsp/ciselniky.nsf/i/d0055>
- [17] *Czech POINT* [online]. 2011 [cit. 2012-04-16]. Dostupné z: <http://www.czechpoint.cz/web/>
- [18] Česká republika. ZÁKON ze dne 7. července 2000, o rozpočtových pravidlech územních rozpočtů. In: *250/2000 Sb.* 2000. Dostupné z: [http://cds.mfcr.cz/cps/rde/xchg/cds/xsl/182\\_4026.html](http://cds.mfcr.cz/cps/rde/xchg/cds/xsl/182_4026.html)
- [19] Česká republika. ZÁKON České národní rady ze dne 13. prosince 1990 o místních poplatcích, o rozpočtových pravidlech územních rozpočtů. In: *565/1990 Sb.* 1990. Dostupné z: [http://cds.mfcr.cz/cps/rde/xchg/cds/xsl/182\\_2314.html](http://cds.mfcr.cz/cps/rde/xchg/cds/xsl/182_2314.html)
- [20] PELC, Vladimír. Místní poplatky : Úplné znění zákona o místních poplatcích s poznámkami . Vyd. 3. [s.l.]. Praha : Linde, 2008. 303 s. ISBN 80-7201-691-4.
- [21] OCHRANA, František. Veřejné služby - jejich poskytování, zadávání a hodnocení : teorie a metodika poptávkového způsobu poskytování a zadávání veřejných služeb na úrovni municipalit. Vyd. 1. [s.l.]. Praha : Ekopress, 2007. 167 s. ISBN 978-80-86929-31-6.
- [22] WALTERS, Robert, et al. Mistrovství v Microsoft SQL Server 2008. Vyd. 1. [s.l.]. Praha : Computer Press, 2009. 864 s. ISBN 978-80-251-2329-4.
- [23] NAGEL, Christian, et al. C# 2008 : Programuje profesionálně. Vyd. 1. [s.l.]. Praha : Computer Press, 2009. 1081 s. ISBN 978-80-251-2401-7.
- [24] SHARP, John. *Microsoft Visual C# 2008: krok za krokem*. Brno: Computer Press, 2008. ISBN 9788025120279.