

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
EKONOMICKÁ FAKULTA

KATEDRA APLIKOVANÉ INFORMATIKY

Analýza, návrh a implementace databázové aplikace pro správu místních poplatků
Analysis, Design and Implementation of the Database Application for the Administration of
Local Taxes

Student: Bc. Petra Knichalová
Vedoucí diplomové práce: Ing. Vítězslav Novák, Ph.D.

Ostrava 2014

VŠB - Technická univerzita Ostrava
Ekonomická fakulta
Katedra aplikované informatiky

Zadání diplomové práce

Student: **Bc. Petra Knichalová**

Studijní program: N6209 Systémové inženýrství a informatika

Studijní obor: 1802T001 Aplikovaná informatika

Téma: **Analýza, návrh a implementace databázové aplikace pro správu místních poplatků**
Analysis, Design and Implementation of the Database Application for the Administration of Local Taxes

Zásady pro vypracování:

1. Úvod
 2. Teoretická východiska návrhu databázových systémů
 3. Analýza současného stavu
 4. Návrh a implementace databázového systému
 5. Závěr
- Seznam použité literatury
Seznam zkratk
Prohlášení o využití výsledků diplomové práce
Seznam příloh
Přílohy

Seznam doporučené odborné literatury:

- RITCHIE, Colin. *Relational database principles*. 2nd ed. London: Cengage Learning EMEA, 2002. ISBN 0-8264-5713-4.
- ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací*. Přeložil Bogdan KISZKA. Brno: Computer Press, 2007. ISBN 978-80-251-3363-7.
- SHEPHERD, Richard. *Access VBA: výukový průvodce*. Přeložil Jakub MUŽÍK. Brno: Computer Press, 2012. ISBN 978-80-251-3686-7.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Vítězslav Novák, Ph.D.**

Datum zadání: 22.11.2013

Datum odevzdání: 25.04.2014

Ing. Petr Růžehna, Ph.D.
vedoucí katedry



prof. Dr. Ing. Dana Dluhošová
děkanka fakulty

Chtěla bych poděkovat Ing. Vítězslavu Novákovi Ph.D. za odbornou pomoc a rady při vedení mé diplomové práce. Děkuji také mé rodině za psychickou i materiální podporu.

V Ostrave dne 25.4.2014

Petra Knichalová

„Prohlašuji, že jsem celou diplomovou práci včetně všech příloh vypracovala samostatně.“

V Ostravě 25. 04. 2014

Petra Knichalová
Bc. Petra Knichalová

Obsah

1 ÚVOD.....	5
2 TEORETICKÁ VÝCHODISKA NÁVRHU DATABÁZOVÝCH SYSTÉMŮ	6
2.1 DATABÁZOVÉ SYSTÉMY A JEJICH VLASTNOSTI.....	6
2.2 RELAČNÍ DATABÁZOVÝ MODEL	7
2.2.1 <i>Systém řízení báze dat</i>	7
2.2.2 <i>SQL</i>	8
2.3 NÁVRH RELAČNÍ DATABÁZE	9
2.3.1 <i>Analýza požadavků</i>	9
2.3.2 <i>Datové modelování</i>	10
2.3.3 <i>Normalizace</i>	13
2.4 DATOVÁ INTEGRITA	15
2.5 DATOVÉ TYPY V MS ACCESS.....	15
2.6 STRUKTURA JAZYKA UML	16
2.6.1 <i>Stavební bloky jazyka UML</i>	17
2.6.2 <i>Obecná mechanika jazyka UML</i>	20
2.6.3 <i>Architektura</i>	21
3 ANALÝZA SOUČASNÉHO STAVU	23
3.1 ZÁKON O MÍSTNÍCH POPLATCÍCH	23
3.2 OBCE PRVNÍHO STUPNĚ	23
3.2.1 <i>Organizační struktura obcí 1. stupně</i>	24
3.2.2 <i>Současný stav databázových aplikací</i>	24
4 NÁVRH A IMPLEMENTACE DATABÁZOVÉHO SYSTÉMU	26
4.1 NÁVRH DATABÁZOVÉ APLIKACE POMOCÍ MODELOVACÍHO JAZYKA UML	26
4.1.1 <i>Zpracování uživatelských požadavků</i>	26
4.1.2 <i>Diagram případu užití</i>	29
4.1.3 <i>Matice RTM</i>	41
4.1.4 <i>Analytický model tříd</i>	42
4.1.5 <i>Realizace případu užití</i>	46
4.2 LOGICKÝ NÁVRH DATABÁZE	50
4.2.1 <i>Tabulky</i>	50
4.2.2 <i>ER diagram</i>	55
4.3 IMPLEMENTACE DATABÁZOVÉ APLIKACE.....	56
4.3.1 <i>Formuláře a sestavy</i>	58

5 ZÁVĚR	63
SEZNAM POUŽITÉ LITERATURY	64
SEZNAM ZKRATEK	66
PROHLÁŠENÍ O VYUŽITÍ VÝSLEDKŮ DIPLOMOVÉ PRÁCE	67

1 Úvod

Mezi specifické pravomoci obcí patří vybírání místních poplatků, které jsou upraveny obecně závaznou vyhláškou. Hlavní funkcí místních poplatků je snaha o zvýšení objemů příjmu do obecního rozpočtu. Do těchto místních poplatků patří poplatek za komunální odpad, poplatek ze psů, poplatek za lázeňský nebo rekreační pobyt, poplatek za užívání veřejného prostranství, poplatek ze vstupného, poplatek z ubytovací kapacity, poplatek za povolení k vjezdu s motorovým vozidlem do vybraných míst a částí měst a poplatek za provozovaný výherní hrací přístroj.

Cílem této diplomové práce je navrhnout databázovou aplikaci, která by pomohla racionalizovat výběr těchto místních poplatků. Návrh bude směřován především na malé obce, kde takové aplikace jsou nekomplexní nebo dokonce zcela chybí. Při návrhu bude kladen důraz na zákon o místních poplatcích 565/1990 Sb, který musí být při jejich výběru dodržován.

Diplomová práce je členěna do kapitol a podkapitol. Druhá kapitola popisuje teoretická východiska návrhu databázových systémů. Zabývá se podrobně konceptuálním a logickým modelem, dále se věnuje jazyku SQL a datovým typům v MS Access. Popisuje strukturu a architekturu jazyka UML. Tyto teoretická východiska jsou poté aplikovány ve čtvrté kapitole při návrhu databázové aplikace.

Třetí kapitola zkoumá aktuální stav databázových aplikací na malých obcích. Popisuje, jaké moduly jednotlivým uživatelům chybějí a také se zabývá nedostatky současných aplikací.

Ve čtvrté kapitole jsou na základě analýzy současného stavu provedeny návrhy na databázovou aplikaci, která bude použitelná pro všechny menší obce zabývající se správou místních poplatků. Pomocí komunikace s pracovníky finančních odborů je provedena analýza požadavků. V jazyku UML je vytvořen diagram případu užití, tříd, sekvence a komunikace. Dále je navržen vhodný relační databázový model.

2 Teoretická východiska návrhu databázových systémů

Cílem této kapitoly je popsat teoretická východiska návrhu databázových systémů. Kapitola definuje databázové systémy a jejich vlastnosti, návrh relačního modelu a jazyk SQL. Dále se zabývá modelovacím jazykem UML. Tato teorie je dále aplikována ve čtvrté kapitole.

2.1 Databázové systémy a jejich vlastnosti

Databáze je uspořádaná množina navzájem propojených dat, která jsou uložena na paměťovém médiu. Pod pojmem databáze si můžeme představit cokoliv. Může se jednat o jednoduchou množinu dat, například telefonní seznam, až po složitou množinu různých nástrojů. [1]

Existuje několik komerčních databázových systémů a samozřejmě i aplikací na nich založených. Mezi nejznámější patří Microsoft Access, Oracle, Microsoft SQL Server, Progress a DB2. Tyto systémy jsou vybudovány na relačním konceptu. Jejich výkonnost se liší s ohledem na výpočetní systém, pro který jsou určeny.

Uživatelé pracují s databází prostřednictvím většího počtu databázových aplikací, které se používají k vytváření a správě databáze a generování informací. Tyto programy mohou být obvyklými dávkovými aplikacemi nebo také online aplikacemi. [2]

Vlastnosti databázových systémů:

- Snaha o odstranění redundantních dat je jednou z vlastností databázových systémů. Teoreticky by měla být redundance dat v databázích zcela eliminována, praktické důvody technického a výkonnostního charakteru však vedou ke kompromisu spočívajícím v existenci velmi omezeného rozsahu redundance. Pokud dojde k aktualizaci redundantně vyskytujícího se údaje, systém zajistí shodnou úpravu všech jeho výskytu a takto zabrání nekonzistenci dat.
- Nezávislost dat aplikačních programů znamená, že změna datových struktur nevyvolá nutnost změny všech programů. Změnit je třeba jen ty programy, které pracují přímo se změněnou částí struktury.
- Sdílení dat představuje fakt, že aplikace, existující nebo budoucí, využívají stejná data.
- Ochrana dat ve smyslu jejich utajení, tj. zabezpečení proti neoprávněnému přístupu je důležitou vlastností každé databáze. Soustředění dat do jednoho místa totiž zvyšuje nebezpečí zneužití nepovolanými osobami. Přístupová práva přiděluje správce databáze k jednotlivým prvkům datové struktury pro jednotlivé uživatele nebo skupiny uživatelů pro každý typ přístupu (výběr, změny hodnot, změny struktury, přidávání či

rušení hodnot údajů). Obvykle rozsah těchto práv má svou hierarchickou strukturu, která klesá po linii: správce databáze – aplikační programátor – uživatel.

- Integrita databáze souvisí s problémem redundantních dat, která zvyšují nebezpečí nekonzistence uložených dat. Integrita může být narušena zadáním nepřipustných hodnot a databázový systém musí umět takové situace ošetřit. Také softwarové či hardwarové chyby mohou narušit integritu databáze. Proto jsou databázové systémy vybaveny prostředky k zálohování a obnově dat.
- Pružnost z hlediska požadavků uživatele je zabezpečena různými prostředky, pomocí nichž mohou uživatelé jednoduchým způsobem vybírat data z databáze podle svých okamžitých potřeb. [3]

2.2 Relační databázový model

Relační model vychází z určitého souboru základních matematických principů odvozených z teorie množin a predikátové logiky. Poprvé byl představen v roce 1970 Dr. E.F.Coddem (IBM). Relační model definuje způsob, jakým je možné data reprezentovat (tedy strukturu dat), způsoby jejich ochrany (neboli integritu dat) a dále operace, které můžeme nad daty provádět (manipulace s daty). [1]

Celkově můžeme shrnout, že relační databázové systémy vykazují následující charakteristické vlastnosti:

- Veškerá data se pomyslně dají reprezentovat v pravidelně uspořádaných strukturách s řádky a sloupci, kterým se říká relace.
- Všechny hodnoty v databázi jsou skalární. To znamená, že v každé konkrétní pozici řádku a sloupce dané relace se nachází právě jedna hodnota.
- Operace v databázi se provádějí vždy nad celou relací a jejich výsledkem je opět jiná celá relace. Tomuto mechanismu se říká uzávěr. [1]

Software pro řízení databáze se nazývá systém řízení báze dat. Jazyk sloužící k ovládní databáze je SQL.

2.2.1 Systém řízení báze dat

Systém řízení báze dat (SŘBD, angl. Database Management System) je: *Programové vybavení (software), které slouží k vymezení struktury databáze, jejímu naplnění daty, aktualizaci dat a k výběrů údajů podle potřeb uživatele prostřednictvím aplikačních programů. Systém řízení báze dat je součástí systémového softwaru.* [3]

Definováním struktury databáze se rozumí popis složení a uspořádání vět a dále charakteru jednotlivých údajů. Struktura všech součástí databáze je uložena v její zvláštní části, která se nazývá slovník dat (Data Dictionary).

Naplnění databáze daty představuje vstup a uložení dat přesně podle struktury uložené ve slovníku dat.

Aktualizace obsahu databáze znamená přidávání, rušení a změny dat uložených v databázi.

Výběr dat je to nejdůležitější pro koncového uživatele, protože potřebuje uložená a aktualizovaná data vybírat podle nejrůznějších kritérií. Vybraná data jsou následně upravena (setříděna, součtována a seskupěna) a prezentována tak, jak si to uživatel přeje.

Systém řízení báze dat musí také zajistit řízení transakcí, podporu komunikace dat v síti, obnovu databáze po poškození, oprávněnost přístupu uživatelů k datům a integritu dat. [3]

Díky existenci systému řízení báze dat je uživatel databáze odstíněn od konkrétních specifik hardwaru a operací s ním. U databázového zpracování dochází k oddělení té složky programového vybavení, která zajišťuje naplňování, úpravu, aktualizace datové struktury a výběrů dat, od vlastních aplikací. Aplikační programy tedy nepracují přímo s uloženými daty, ale zprostředkovaně přes standardní programy, které databázi ovládají. [3]

2.2.2 SQL

SQL (Structured Query Language) je jazyk určený pro práci s relačními databázovými systémy, poskytující gramatiku a syntaxi pro definici a manipulaci s daty. Začala jej vyvíjet společnost IBM (Raymond F. Boyce, Donald C. Messerly a Andrew Richardson) na začátku 70. let 20. století. [6]

SQL patří mezi deklarativní jazyky a to znamená, že říká, co chceme obdržet a ne, jak toho má být dosaženo. Tento jazyk je přenositelný mezi platformami, takže nejsme závislí na jediném poskytovateli. [7]

Příkazy jazyka SQL můžeme rozdělit do 4 skupin:

- **DDL** jsou příkazy, které vytvářejí či upravují strukturu databáze (*CREATE*, *ALTER*, *DROP*).
- **DML** příkazy, které slouží k získávání, ukládání a mazání dat v databázi (*SELECT*, *INSERT*, *UPDATE*, *DELETE*).
- **DCL** příkazy, které slouží ke správě uživatelských rolí a práv (*GRANT*, *REVOKE*).

- **TCL** příkazy pro správu databázových transakcí (*BEGIN, COMMIT, ROLLBACK*). [6]

2.3 Návrh relační databáze

Návrh relační databáze se obvykle skládá ze tří fází, mezi které patří analýza požadavků na systém, datové modelování a normalizace.

2.3.1 Analýza požadavků

Jedná se o shromažďování informací o tom, co by nový systém měl dělat, a to ze všech možných zdrojů: od uživatelů existujícího systému (rozhovory a pozorování), z nejrůznějších dokumentů (formulářů, zpráv, norem, popisů práce, počítačových výstupů, dokumentace existujícího systému, manuálů apod.). [4]

Rozhovor si musí analytik dobře připravit. Respondent musí být předem seznámen s povahou rozhovoru, může obdržet některé otázky k promyšlení nebo může být vyzván k přípravě určitých dokumentů. Termín rozhovoru musí respondentovi vyhovovat, vypracuje se scénář otázek, který se pak může dynamicky vyvíjet v závislosti na reakcích respondenta. [4]

Dotazníky jsou na rozdíl od rozhovoru pasivní, jdou obvykle méně do hloubky, nejsou tak nákladné a časově náročné. Můžeme oslovit obrovské množství respondentů, ale oproti rozhovoru jsou chudší na informace. Dotazník může být sestaven v papírové, elektronické nebo ústní formě. [4]

Skupinový rozhovor se provádí tak, že analytik hovoří se skupinkou lidí. Nevýhodou individuálního rozhovoru i dotazníků je nutnost řešit řadu protikladných, nekonzistentních informací. Proto je nutné následně vyjasňování. [4]

Pozorování patří mezi doplňkové metody, procházející celou fází analýzy. Během pozorování analytik zjišťuje, jak systém funguje, co lidé dělají, podstatu jejich práce. Nevýhodou je, že nezachytíme všechny podstatné skutečnosti. [4]

Analýza písemných materiálů probíhá tak, že se zkoumají veškeré dokumenty (zákony, normy, předpisy, postupy práce), které v organizaci existují a vztahují se k předmětu řešení – statut a organizační řád, podnikatelský plán, popisy práce, směrnice a ostatní interní normy, korespondence včetně interní, studie a dokumentace existujícího systému, formuláře (objednávky, faktury, dodací listy atd.). [4]

Joint application design je metoda, která vznikla u firmy IBM na konci 70. let. Metoda je založená na společném setkání klíčových uživatelů, analytiků a manažerů. Podobá se tedy skupinovému rozhovoru, nicméně je specifická programem a strukturou rolí. Smyslem

je zjistit požadavky na systém od všech rozhodujících zúčastněných. Setkání probíhá na místě mimo pracoviště. [4]

V **prototypování** ukazujeme uživateli co nejdříve vstupní a výstupní formuláře (relace a funkce ho nezajímají). Je protikladem systémového přístupu, ve kterém postupujeme postupně po předem naplánovaných etapách. Jde o to, abychom co nejdříve vytvořili prototyp nového systému pro uživatele, kdy se ořežou vedlejší funkce a rychle se naprogramuje hlavní funkce, aby uživatelé mohli vidět část systému. Postupně se přidávají další funkce a ke konci se integrují vedlejší funkce, které však mohou být dodělány v rychlosti, což může být zdrojem budoucích problémů. Tento přístup lze použít u malých systémů, na rozsáhlé systémy to nelze aplikovat. [4]

2.3.2 Datové modelování

Datový model je soubor prostředků a konceptů, popisujících data na určité úrovni abstrakce s různým pohledem na data v jednotlivých fázích procesu vývoje aplikace. Cílem je navrhnout kvalitní datovou strukturu pro konkrétní aplikaci, kterou bude tato aplikace využívat k uložení dat.

Datové modely dělíme na tři skupiny:

- Konceptuální model je výsledkem datové analýzy a vzniká jako popis dat v databázi bez ohledu na jejich uložení. Představuje formální popis modelované reality. Hlavními úkoly je nalezení entit, vztahů a atributů. Pro popis používá zejména ER model.
- Logický model se vztahuje ke konkrétnímu datovému modelu a používá jeho konstrukční dotazovací a manipulační prostředky.
- Fyzický – zajišťuje fyzické uložení dat. Obvykle řeší konkrétní SŘBD. [12]

1) Konceptuální modelování obsahuje datovou strukturu systému nezávislou na databázové koncepci. Mezi jeho základní konstruktory patří entita, vztah, atribut, doména, klíč, kandidátní klíč, primární klíč, alternativní klíč a cizí klíč.

Modelování entit a vztahů představuje přirozený a zjednodušený pohled na svět. Pod pojmem entita si můžeme představit cokoliv, o čem potřebujeme v systému uchovávat informace. Entity můžeme získat analýzou požadavků. Entita je vlastně prvek reálného světa (např. člověk, stroj, vyučovaný předmět, město), který je popsán svými charakteristikami (vlastnostmi). Ty se většinou považují za atribut (např. jméno, příjmení, stav, plat, hmotnost). V relační databázi jsou atributy tvořeny sloupci. [3]

Mezi entitami určujeme vztahy sloužící ke svázání dat, která spolu souvisejí a jsou umístěny v různých databázových tabulkách. Každý vztah má tři základní vlastnosti stupeň, kardinalitu a volitelnost účasti.

Stupeň vztahu určuje, kolik záznamu jedné tabulky musí být ve vztahu k jednomu záznamu druhé tabulky. [8]

Kardinalita vztahu popisuje počet možných relací pro každou zúčastněnou entitu:

- 1:1 je vztah, kdy záznamu v první tabulce odpovídá právě jeden záznam v druhé databázové tabulce a naopak.
- 1:N je vztah, kdy záznam v jedné tabulce je ve vztahu s několika záznamy v tabulce druhé. Ale 1 záznam v tabulce druhé je ve vztahu pouze s jedním záznamem v tabulce.
- M:N je vztah, který umožňuje několika záznamům z jedné tabulky přiřadit několik záznamů z tabulky druhé. Většinou bývá realizován v relační koncepci kombinací dvou vztahů 1:N a 1:M, které ukazují do pomocné, tzv. vazební tabulky složené z kombinace obou použitých klíčů. Tato část už patří do logického modelu, ale jeví se jako konceptuální model. [8]

Volitelnost vztahu popisuje, zda se relace účastní všech výskytů entity nebo jen některé:

- Povinná - před vložením záznamu do tabulky jedné musí být záznam v tabulce druhé.
- Volitelná - před vložením záznamu do tabulky jedné nemusí být záznam v tabulce druhé [8]

Doména je množina přípustných hodnot pro jeden nebo více atributů. Je určena datovým typem (text, číslo, datum, logický, objekt), délkou (počet znaků) a rozsahem hodnot. [6]

Kandidátní klíč jednoznačně identifikující výskyt dané entity. Kandidátní klíč pro tabulku má tyto vlastnosti:

- jedinečnost,
- neredukovatelnost – žádná vlastní podmnožina kandidátního klíče nezajišťuje jedinečné určení záznamu. [2]

Primární klíč (Primary Key) je kandidátní klíč, který je vybrán, aby jedinečně určoval záznamy v tabulce. Aby pole mohlo být označeno jako primární klíč, mělo by splňovat tyto podmínky:

- Musí jednoznačně identifikovat každý řádek.
- Nesmí obsahovat hodnotu Null (chybějící nebo neznámá hodnota).
- Pole by se nemělo měnit. [13]

Alternativní klíč je kandidátní klíč, který není primárním klíčem. [8]

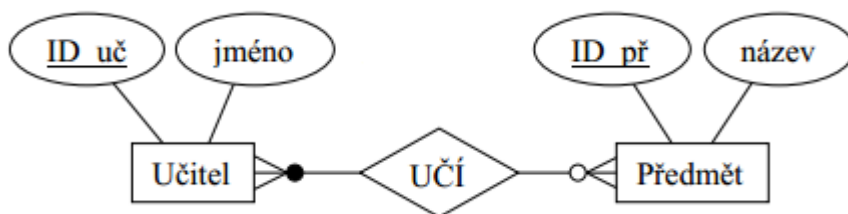
Cizí klíč (Foreign Key) je klíč entity, který je současně primárním klíčem jiné entity. [8]

Model entit a vztahů

Model entit a vztahů, který popisuje data jako entity, atributy a relace (vztahy), zavedl poprvé Peter Pin Shan Chen v roce 1976. Brzy se rozšířil a stal se uznávaným.

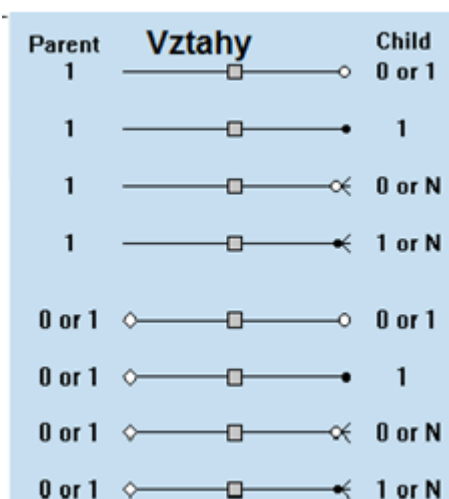
Chenova notace ERD:

- Obdélníky reprezentují množiny entit.
- Elipsy reprezentují atributy.
- Kosočtverce reprezentují množiny vztahů.
- Čáry spojují atributy s množinami entit a množiny entit s množinami vztahů.
- Atributy primárního klíče jsou podtržené. [9]



Obrázek 1: Chenova notace [9]

Protože Chenova notace zabírá mnoho prostoru, začala se používat notace "Crow's Foot". V obdélnících entit mohou být zapsány atributy, entity jsou vzájemně propojeny čarami, jejichž symboly na obou koncích čáry definují kardinalitu vztahu. [7]



Obrázek 2: notace "Crow's Foot" [7]

2) Logický model

Příkladem logického modelu mohou být databáze síťové, hierarchické, relační, objektově relační a objektově orientované. Tato práce se však zabývá relačním modelem.

Relační model sdružuje data do relací, které obsahují záznamy. Z pohledu uživatele je relační databáze tvořena soustavou tabulek, které tvoří základ relační databáze. Tabulka je struktura záznamů s pevně stanovenými sloupci (poli, atributy). Každý sloupec má definován jednoznačný název, typ a doménu. [7]

Relace má tyto vlastnosti:

- Jméno tabulky musí být jedinečné v databázi.
- Každá buňka tabulky obsahuje přesně jednu hodnotu.
- Každý sloupec má jedinečné jméno v tabulce.
- Hodnoty v každém ze sloupců by měly být dále nerozložitelné.
- Všechny hodnoty v jednom sloupci jsou ze stejné domény.
- Každý záznam je jedinečný, nesmí obsahovat duplicitní hodnoty.
- Pořadí sloupců nemá význam.
- Pořadí záznamů nemá význam. [2]

Mapování ER modelu do relačního DB schématu

Jsou-li entity ve vztahu 1:N, přidá se primární klíč na straně 1, jako další atribut, do tabulky na straně N. Takový atribut se pak nazývá cizí klíč. U vztahu 1:1 je situace podobná, jako u 1:N. Přidá se cizí klíč, který zde může tvořit primární klíč. U vztahu M:N se to řeší vložením třetí tabulky. [8]

2.3.3 Normalizace

Normalizace je proces rozkladu velkých tabulek na menší. Tím se snaží zabránit redundanci dat a eliminovat budoucí problémy s vkládáním, rušením a aktualizací dat. Při rozkládání relací by měl platit princip bezztrátové dekompozice. To znamená, že původní relace rozdělujeme takovým způsobem, abychom mohli nové relace jakkoliv zpětně spojit bez ztráty informace. Normalizovaná databáze musí zachovat všechny závislosti původní databáze. [7]

Občas se setkáme s procesem denormalizace a to znamená, že se výkon databáze zvyšuje snížením počtu tabulek i za cenu existence redundantních dat a jiných problémů v databázi.

Normální formy:

- 1.NF – První normální forma
- 2.NF – Druhá normální forma
- 3.NF – Třetí normální forma
- BCNF – Boyce Coddova normální forma
- 4.NF – Čtvrtá normální forma
- 5.NF – Pátá normální forma

Prvním normální forma je splněna, pokud v tabulce nejsou žádné opakující se záznamy, jsou definovány všechny klíčové atributy, všechny atributy závisejí na primárním klíči a sloupce nejde dále dělit.[8]

Tabulka splňuje 2NF, když je v 1NF a každý neklíčový atribut je plně závislý na primárním klíči, a to na celém klíči a nejen na některé z podmnožin. Druhá normalizační forma se týká tabulek, jejichž primární klíč tvoří dva anebo více atributů. Tabulka v 1NF, která je tvořena jedním primárním klíčem je automaticky ve 2NF. [10]

Tabulka splňuje 3NF, pokud již je v 1NF a 2NF a žádný z jejich atributu není tranzitivně závislý na primárním klíči. Tranzitivní závislost je závislost mezi minimálně dvěma atributy a klíčem, kde jeden atribut je funkčně závislý na klíči a druhý atribut je funkčně závislý na prvním atributu. Tedy relace je v 3NF, pokud je ve 2NF a všechny neklíčové atributy jsou navzájem nezávislé. [10]

Boyce Coddova normální forma se pokládá za variaci třetí normální formy. Byla publikována v 70 letech. Je vymezena stejnými pravidly jako 3. NF forma. Pokud je relace v BCNF, je i v 3NF. Naopak to neplatí. Relace se nachází v BCNF, jestliže pro každou netriviální závislost $X \rightarrow Y$ platí, že X je nadmnožinou nějakého klíče relace R . Aby byla porušena BCNF musí být splněny tři podmínky:

- Relace musí mít více kandidátních klíčů.
- Minimálně 2 kandidátní klíče musí být složené z více atributu.
- Některé složené kandidátní klíče musí mít společný atribut. [10]

Relace je ve čtvrté normální formě, pokud je v Boyce Coddově normální formě. Navíc musí platit, že všechny vícehodnotové závislosti jsou zároveň funkčními závislostmi z kandidátních klíčů.

Relace je v páté normální formě, pokud je ve čtvrté a není možné do ní přidat další atribut (skupinu atributu) tak, aby se vlivem skrytých závislostí rozpadla na několik dílčích relací. [8]

2.4 Datová integrita

Integrita dat označuje platnost, konzistentnost a přesnost dat v databázi. Je jedním z nejdůležitějších aspektů procesu logického návrhu databáze. Při jejím porušení hrozí nebezpečí vzniků chyb, které se těžko odstraňují. [10]

Můžeme definovat čtyři typy integrity, které se implementují v průběhu procesu návrhu databáze. Jsou založeny na různých aspektech struktury databáze a jsou označeny podle oblasti, kde se objevují:

- Integrita na úrovni tabulky bývá označovaná také jako integrita entity. Říká, že primární klíč tabulky musí obsahovat v každém řádku jedinečnou hodnotu, která nesmí být null.
- Integrita na úrovni pole (doménová integrita) zajišťuje, že hodnoty v každém poli jsou platné (tedy struktura pole je spolehlivá). Pole stejného typu jsou v celé databázi definována konzistentně.
- Integrita na úrovni vztahů se také nazývá referenční integrita a zajišťuje spolehlivost vztahu mezi dvěma tabulkami. Záznamy v obou tabulkách jsou synchronizovány a v případě změn dat v jedné tabulce se změny projeví i ve druhé.
- Jako další typ integrity dat bývají označována i business pravidla, která zavádějí určitá omezení ze strany zadavatele. Většinou se týkají způsobu získávání a používání dat. Do integrity dat jsou zahrnuta proto, jelikož mohou ovlivňovat některé aspekty návrhu databáze. [11]

2.5 Datové typy v MS Access

Datový typ definuje typ dat ukládaných do sloupce. Následující tabulka popisuje datové typy v MS Access 2010, které jsou použity při návrhu relačního databázového modelu.

Datový typ	Použití	Velikost
Text	text nebo kombinace textu	255 znaků
Memo	delší texty a čísla	65 536 znaků
Číslo	ukládání dat, která jsou určena pro matematické výpočty	1, 2, 4, 8 nebo 12 bajtů
Měna	ukládání peněžních hodnot	8 bajtů

Automatické číslo	zadávaní jedinečné hodnoty (např. primární klíč)	4 bajty (16 bajtů při použití pro ID replikace)
Datum/čas	datum a čas	8 bajtů
ANO/NE	pro data, která mohou nabývat pouze jedné ze dvou možných hodnot	1 bit
Objekt OLE	určen pro objekty OLE (jiné aplikace systému Microsoft Windows)	1 GB
Hypertextový odkaz	ukládání hypertextových odkazů (cesta UNC nebo adresa URL)	1 GB znaků nebo 2 GB úložiště
Příloha	používá se pro připojení více souborů (obrázky, záznamy)	komprimované přílohy 2GB, nekomprimované přílohy přibližně 700 kB

Tabulka 1: Datové typy v MS Access

[14]

Pro každý datový typ můžeme nastavit například tyto vlastnosti:

- formát pole,
- počet desetinných míst,
- vstupní masku,
- titulek,
- výchozí hodnotu,
- ověřovací pravidlo,
- ověřovací text.

2.6 Struktura jazyka UML

UML (Unified Modeling language) je unifikovaný modelovací jazyk a slouží k vizuálnímu modelování systému. Nejčastěji bývá spojován s modelováním objektově orientovaných systémů, ale jeho využitelnost je mnohem širší. Jazyk UML nenabízí žádný druh metodiky modelování, avšak poskytuje vizuální syntaxi, kterou můžeme využít při sestavování modelů.

Proč unifikovaný

Jazyk UML se snaží o unifikaci různých domén:

- Nabízí vizuální syntaxi pro modelování během celého vývojového cyklu softwarového projektu.
- Byl vytvořen jako nástroj pro modelování čehokoliv systému zasazenými do reálného času počínaje a podpůrnými systémy rozhodování konče.
- Je nezávislý na jakémkoliv programovacím jazyce a na jakékoliv platformě. Podporuje OOP jazyky (Java, C#), ale je také vhodný pro hybridní objektově orientované jazyky (C++) i pro jazyky založené na objektech (Visual Basic). Může se také použít pro modelování v neobjektových jazycích (C).
- Podporuje různé metodiky vývoje.
- Snaží se o vnitřní jednotu a konzistenci prostřednictvím malé množiny interních procesů, ale není vždy v této oblasti úspěšný. [5]

Struktura jazyka UML se skládá ze tří součástí:

- Stavební bloky tvoří základní prvky modelu (relace a digramy).
- Obecná mechanika popisuje obecné způsoby, pomocí nich v jazyce UML dosahujeme specifických cílů.
- Architektura vyjadřuje pohled na navrhovaný systém.

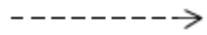

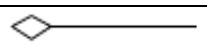

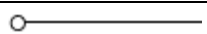
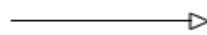
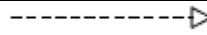
2.6.1 Stavební bloky jazyka UML

Jsou sestaveny z předmětů, vztahů a diagramů.

Předměty se dělí do čtyř skupin:

- Strukturní abstrakce jsou vlastně podstatná jména modelu UML, mezi které patří třídy, rozhraní, spolupráce, případ užití, aktivní třída, komponenta a uzel.
- Chování jsou slovesa modelu UML (interakce, stav).
- Seskupení jsou balíčky, které slouží k používání a seskupování významně souvisejících prvků modelu do soudržných jednotek.
- Poznámky jsou anotace, které lze k modelu připojit s úmyslem zachytit informaci sestavenou jen k tomuto účelu. [5]

Vztahy (relace) ukazují na modelu, jaký je vztah mezi dvěma předměty. Umožňují zachytit významový vztah mezi předměty. Vztahují se na strukturní abstrakce a seskupování. [5]

Typ relace	Syntaxe UML		Popis
	Zdroj	Cíl	
Závislost			Změna v určitém předmětu ovlivňuje význam závislého předmětu
Asociace			Popis množiny spojení mezi objekty
Agregace			Cílový prvek je součástí zdrojového prvku
Kompozice			Silnější forma agregace (má více omezení)
Ochranná nádoba			Zdrojový prvek obsahuje cílový prvek
Zobecnění			Jeden prvek je specializací jiného prvku a lze jej nahradit obecnějším prvkem
Realizace			Asociace mezi klasifikátory, kde jeden klasifikátor určuje dohodu, jejíž uskutečnění zaručuje druhý klasifikátor

Tabulka 2: Stavební bloky jazyka UML

Diagramy

Jsou okna nebo pohledy na model. Diagram však není model, protože předměty a relace lze z diagramu odstranit. Celkem existuje 13 typů digramů. Lze je rozdělit na statické a dynamické. Statické diagramy zachycují předměty a strukturní asociace mezi předměty. Do této skupiny patří diagram tříd, složené struktury, komponent, nasazení, objektů a balíčků. Dynamický model popisuje způsob, jímž na sebe jednotlivé předměty navzájem působí, aby bylo dosaženo požadovaného chování softwarového systému. Patří mezi ně diagram aktivit, interakce, případů užití a stavu. Diagram interakce dále dělíme na diagram posloupnosti, komunikace, sekvence a časování.

Není pevně stanoveno, v jakém pořadí bychom měli UML diagramy vytvářet. Přesto obvykle začínáme diagramem případu užití, který definuje rozsah platnosti navrhovaného systému. Ve skutečnosti se pracuje s několika různými diagramy zároveň. Každý z nich je postupně tvořen a doplňován.[5]

V této práci je použit *diagram případů užití, tříd, sekvence a komunikace*.

Diagram případů užití vyjadřuje rámečkem subjekt, který definuje hranice systému modelovaného pomocí případů užití. Aktéry zobrazujeme vně subjektu. Případy užití, které utvářejí chování systému, umísťujeme uvnitř subjektu. Vztah mezi aktérem a případem užití je znázorněn pomocí plné čáry (symbol přiřazení). U tohoto typu přiřazení vždy předpokládáme implicitní stereotyp, který označuje, že aktér a případ užití spolu určitým

způsobem komunikují. Aktér nemusí být pouze fyzická osoba, ale například i jiný systém či hardwarové zařízení. Modelování případů užití by mělo probíhat v následujících krocích:

- nalezení hranic systému,
- vyhledání aktérů,
- nalezení případů užití,
- specifikace případů užití,
- určení alternativních scénářů,
- opakování předchozích bodů, dokud nedojde k ustálení případů užití, aktérů a hranic systému. [5]

Diagram tříd je velmi důležitou součástí realizace případu užití. Jeho smyslem je popsat stavební prvky systému (třídy) a jejich vzájemné vztahy. Základem diagramu tříd jsou:

- třídy objektů,
- atributy definují vlastnosti tříd,
- metody (operace, funkce) vytvářející chování objektu a pomocí nichž se realizuje požadovaná funkcionality systému,
- asociace definují vztahy na další objekty.

Nosným prvkem sekvenčního diagramu jsou instance, nikoliv třídy. Vyjadřuje detailně posloupnost zpráv mezi objekty (instancemi) v čase. Těmito objekty mohou být buď objekty z programování (JAVA, C#) anebo vyšší implementační celky jako jsou komponenty, servery, subsystémy apod. Hlavním smyslem je popsat časový scénář posloupnosti zpráv mezi objekty v technologiích. Používá se:

- v technologickém návrhu,
- ve vzorech objektového programování, tj. pokud vzor obsahuje dynamiku a je žádoucí ji popsat.
- pro modelování všech situací v návrhu informačního systému, které podléhají dichotomii třída-instance,
- při zobrazování časové posloupnosti zpráv. [15]

Diagram komunikace je velmi podobný sekvenčnímu diagramu. Rozdíl je v tom, že diagram komunikace neukazuje přesně časovou posloupnost, ale zřetelněji zobrazuje strukturu objektu. Tento diagram použijeme, pokud chceme znázornit strukturu objektu. Vyjadřuje detailně posloupnost zpráv mezi objekty (instancemi) v čase. [15]

2.6.2 Obecná mechanika jazyka UML

Jazyk UML obsahuje čtyři společné mechanismy, které se používají konzistentně. Popisují čtyři strategie cesty k modelování objektů, jež jsou opakovaně používány v různých kontextech v celém jazyce UML. Mezi tyto mechaniky patří specifikace, ornamenty, podskupiny a mechanismy rozšiřitelnosti.

a) Specifikace

Modely UML mají nejméně dva rozměry. Grafický umožňuje vytvářet model prostřednictvím diagramů a symbolů (ikon). Textový se skládá ze specifikací různých prvků modelu. Specifikace jsou textovým popisem sémantiky jednotlivých prvků. [5]

b) Ornamenty

Každý prvek modelu je v jazyce UML vyjádřen jednoduchým symbolem, který lze obohatit řadou ornamentů, je-li třeba prostřednictvím diagramu zobrazit více informací.

Tvorba složeného modelu může začít pomocí několika základních symbolů s jedním nebo dvěma ornamenty. Později však můžeme model vylepšit dalšími ornamenty a takto postupujeme do té doby, dokud příslušný prvek neobsahuje dostatečný počet podrobností.

Ornamenty používáme pouze v případě, kdy zvyšují srozumitelnost a čitelnost diagramu, nebo když zdůrazňují určitou důležitou funkci modelu. Není třeba v diagramu zobrazovat všechny podrobnosti. Důležité je, aby byl diagram srozumitelný, aby znázorňoval přesně ty cíle, kterých chceme dosáhnout, a aby byl snadno čitelný. [5]

c) Podskupiny

Podskupiny popisují různé způsoby vidění světa. Jazyk UML rozlišuje podskupiny na skupinu klasifikátorů a instancí a skupinu rozhraní a implementací. [5]

- Klasifikátor a instance – abstraktním vyjádřením typu předmětu je klasifikátor, zatímco konkrétní předměty jsou považovány za instance. Názvy instancí jsou na svém symbolu podtrženy.
- Rozhraní a implementace – Řídíme se tím, co předmět vykonává (jeho rozhraní), od toho, jak to vykonává (jeho implementace). [5]

d) Mechanismy rozšiřitelnosti

Autoři jazyka UML si byli vědomi toho, že není možné vytvořit univerzální modelovací jazyk, který by vyhovoval všem uživatelům. Proto zahrnuli do jazyka tři mechanismy pro jeho rozšiřitelnosti a to omezení, stereotypy a označené hodnoty.

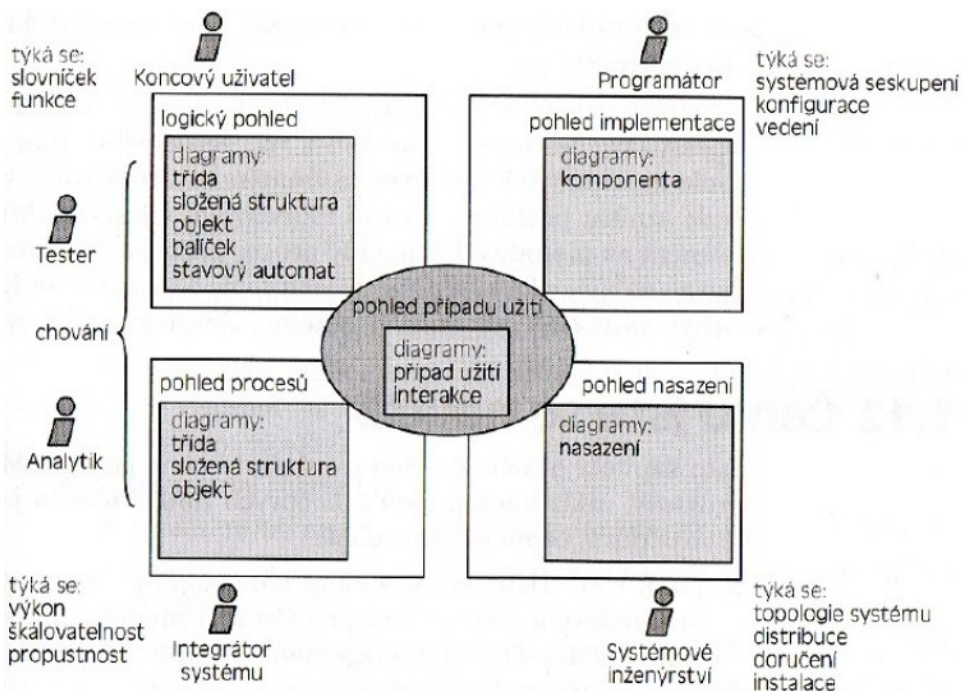
Omezení stanoví omezující podmínky rozšiřující sémantiku prvku tím, že umožní k němu přidávat nová pravidla.

Stereotyp umožňuje definovat nový prvek modelu UML, jenž je založen na existujícím prvku, takže definujeme sémantiku stereotypu. Stereotypy přidávají do metamodelu jazyka UML nové prvky.

Označené hodnoty umožňují rozšířit specifikaci prvku tím, že k takovému prvku přidáme informaci sestavenou jen k tomuto účelu. [5]

2.6.3 Architektura

Zachycuje strategické aspekty vyšší struktury systému. Lze na ní nahlížet mnoha způsoby, ale nejčastěji se na ni díváme pohledem „4+1.“ Abychom byli schopni zachytit všechny aspekty architektury daného systému, definuje jazyk UML čtyři různé pohledy na systém, mezi které patří logický pohled, pohled procesů a pohled implementace a pohled nasazení. Všechny tyto pohledy jsou integrovány do pátého pohledu, což je znázorněno na následujícím obrázku.[5]



Obrázek 3: Architektura UML [5]

Logický pohled zachycuje slovník oblasti problému jako množinu tříd a objektů. Důraz je přitom kladen především na zobrazení způsobu, jakým objekty a třídy tvořící základ systému implementují jeho chování.

Pohled procesů modeluje spustitelná vlákna a procesy jako aktivní třídy. Je to procesově orientovaná varianta logického pohledu, která obsahuje stejné artefakty.

Pohled implementace modeluje soubory a komponenty, které utvářejí hotový kódový základ systému. Slouží jednak ke znázornění závislostí mezi jednotlivými komponentami, jednak toho, jak spravovat konfiguraci množin vytvořených z těchto komponent. Umožňuje definice verze systému.

Pohled nasazení komponent na množinu fyzických výpočetních uzlů (počítačů a periferních zařízení) umožňuje modelování distribuce komponent na příslušné uzly distribuovaného systému.

Všechny pohledy jsou odvozeny od pohledu případů užití. Tento pohled zachycuje základní požadavky kladené na příslušný systém jako na množinu případů užití a utváří tak základ tvorby všech dalších pohledů. [5]

3 Analýza současného stavu

Kapitola analýza současného stavu popisuje, jaké moduly databázových aplikací se používají při správě místních poplatků a také definuje jejich nedostatky z pohledu uživatele. Dále se zmiňuje o zákonu o místních poplatcích, který musí být při výběru těchto poplatků dodržován.

3.1 Zákon o místních poplatcích

Obce vybírají tyto místní poplatky:

- poplatek ze psů,
- poplatek za lázeňský nebo rekreační pobyt,
- poplatek za užívání veřejného prostranství,
- poplatek ze vstupného,
- poplatek z ubytovací kapacity,
- poplatek za povolení k vjezdu s motorovým vozidlem do vybraných míst a částí měst,
- poplatek za provoz systému shromažďování, sběru, přepravy, třídění, využívání a odstraňování komunálních odpadů,
- poplatek za zhodnocení stavebního pozemku možností jeho připojení na stavbu vodovodu nebo kanalizace. [16]

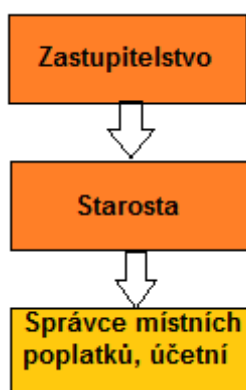
3.2 Obce prvního stupně

Podle rozsahu státní správy, kterou vykonávají, a rozsahu působnosti, kterou mají, se obce dělí na:

- obce (obce 1. stupně, vesnice),
- obce s pověřeným osobním úřadem (obce 2. stupně, malá města),
- obce s rozšířenou působností (obce 3. stupně, větší města s velkým správním obvodem).

Tato práce se zabývá návrhem databázové aplikace pro obce prvního stupně, které jsou pověřeny správou místních poplatků.

3.2.1 Organizační struktura obcí 1. stupně



Obrázek 4: Organizační struktura

Zastupitelstvo schvaluje obecně závazné vyhlášky, které se týkají konkrétní výše jednotlivých místních poplatků.

Starosta předkládá OZV, kde je stanovena výše místních poplatků, zastupitelstvu obce ke schválení.

Správce místních poplatků je pověřen výběrem místních poplatků a musí se řídit OZV obce, zákonem 565/1990. Sb. a daňovým řádem.

3.2.2 Současný stav databázových aplikací

Současný stav databázových aplikací byl zkoumán na základě dotazníkového šetření a také pomocí osobní konzultace s úředníky, kteří se správou místních poplatků zabývají.

Dotazníkové šetření se týkalo obcí 1. stupně. Bylo osloveno 21 obecních úřadů a nakonec přišlo 10 odpovědí, které berme jako reprezentativní vzorek. Na základě těchto odpovědí vypadá vyhodnocení dotazníků takto:

Má váš informační systém modul, který eviduje poplatek za: (otázka)	Ano (odpověď)
Psy	80 %
Lázeňský nebo rekreační pobyt	20 %
Užívání veřejného prostranství	70 %
Vstupné	10 %
Ubytovací kapacity	20 %
Povolení vjezdu s motorovým vozidlem	10 %
Komunální odpad	90 %
Zhodnocení stavebního pozemku	30 %

Tabulka 3: Současný stav databázových aplikací

Závěr analýzy současného stavu:

Většina úřadů 1. stupně má moduly pro správu poplatků za komunální odpad, užívání veřejného prostranství a psy. Zbylé moduly většina úřadů nemá. Pokud ale nastane situace, kdy chybějící moduly potřebují, zapisují informace do sešitu nebo textového editoru, což je nepřehledné a zmatené. Databázová aplikace by pomohla celý systém výběru více zjednodušit.

Mezi další zjištěné nedostatky pomocí dotazníků a osobních konzultací patří:

- chybějící moduly,
- nepřehlednost tlačítek a tabulek,
- zbytečná textová a vyhledávací pole,
- chybné generování sestav,
- nepřívětivé uživatelské rozhraní.

Cíl:

Navrhnout přehledný informační systém, který by umožnil efektivně spravovat každý typ poplatků a generoval by sestavy podle potřeb uživatele.

4 Návrh a implementace databázového systému

Tato kapitola se zabývá návrhem a implementací databázové aplikace. Při návrhu databázové aplikace je vycházeno z předchozí kapitoly zabývající se analýzou současného stavu. Snaží se vyhnout zjištěným nedostatkům a navrhnout takovou aplikaci, která by umožnila evidovat všechny typy poplatků a byla by reálně nasazena do provozu. Je zde vytvořen návrh pomocí modelovacího jazyka UML, logický model a část funkcionality fyzického modelu.

4.1 Návrh databázové aplikace pomocí modelovacího jazyka UML

V této podkapitole je vytvořen návrh databázové aplikace v jazyku UML. Pomocí analýzy požadavků je vytvořen slovní příběh, který umožňuje stanovit požadavky na systém. Analýza požadavků byla provedena důsledně, komunikací s uživateli existujících systému, studiem norem, zákonů a dotazníkem. Také je navržen komunikační diagram, sekvenční diagram a diagram tříd, z kterých dále vychází logické a fyzické modelování.

4.1.1 Zpracování uživatelských požadavků

Zpracování uživatelských požadavků je velmi důležitou součástí návrhu databázové aplikace. S databázovou aplikací bude pracovat běžný uživatel a také správce. Úkolem správce je nastavování uživatelských rolí a přístupových práv. Správce má také za úkol aktualizovat data za každé účetní období.

Nejčastěji bude s databázovou aplikací pracovat uživatel, který je zaměstnancem obecního úřadu a je pověřen správou místních poplatků. Do jeho kompetencí spadá výběr poplatků ze psů, poplatků za lázeňský nebo rekreační pobyt, poplatků za užívání veřejného prostranství, poplatků ze vstupného, poplatků z ubytovací kapacity, poplatků za povolení k vjezdu s motorovým vozidlem do vybraných míst a částí měst, poplatků za provoz systému shromažďování, sběru, přepravy, třídění, využívání a odstraňování komunálních odpadů, poplatků za zhodnocení stavebního pozemku možností jeho připojení na stavbu vodovodu nebo kanalizace.

Cílem databázové aplikace je, aby efektivně evidovala tyto poplatky. Každý typ poplatku je možné zaplatit nebo v případě jejich nezaplacení evidovat dluh. Jedná-li se o vracení poplatků, netýká se to všech místních poplatků. Některé poplatky nelze vrátet, protože se jedná pouze o jednorázovou platbu. Následující tabulka zobrazuje uživatelské požadavky na systém.

Poplatek za:	Evidence plateb	Evidence vratek	Evidence dluhu
Pes	Ano	Ano	Ano
Lázeňský nebo rekreační pobyt	Ano	Ne	Ano
Užívání veřejného prostranství	Ano	Ne	Ano
Ze vstupného	Ano	Ne	Ano
Z ubytovací kapacity	Ano	Ne	Ano
Povolení vjezdu s motorovým vozidlem	Ano	Ano	Ano
Komunální odpad	Ano	Ano	Ano
Zhodnocení stavebního pozemku	Ano	Ano	Ano

Tabulka 4: Uživatelské požadavky

V následující tabulce jsou shrnuty všechny požadavky uživatelů. Mají také určenou prioritu a funkčnost.

Označení R _x	Formulace	Priorita	Typ
R ₁	Přihlášení uživatele	1	funkční
R ₂	Evidence FO a PO	1	funkční
R ₃	Správa poplatků za psů	1	funkční
R ₄	Správa poplatků za lázeňský nebo rekreační pobyt	2	funkční
R ₅	Správa poplatků za užívání veřejného prostranství	1	funkční
R ₆	Správa poplatků ze vstupného	2	funkční
R ₇	Správa poplatků z ubytovací kapacity	2	funkční
R ₈	Správa poplatků za povolení vjezdu s motorovým vozidlem	2	funkční
R ₉	Správa poplatků za komunální odpad	1	funkční
R ₁₀	Správa poplatků za zhodnocení stavebního pozemku	2	funkční

Tabulka 5: Priority a typy uživatelských požadavků

R₁ – Přihlášení – uživatel se do databázové aplikace přihlásí v případě zadání správného uživatelského jména a hesla.

R₂ - Evidence FO a PO – databázová aplikace bude evidovat fyzické osoby a právnické osoby. V případě narození nebo přistěhování občana se zaeviduje do aplikace nová FO, pokud se občan odstěhuje nebo zemře, dojde k vymazání (pokud nemá žádné pohledávky). PO se vytvoří, pokud chce uhradit některý z místních poplatků a doposud není vedena v databázové aplikaci obce.

R₃ - Správa poplatků za psy – zaplatí FO nebo PO, která má trvalý pobyt nebo sídlo v obci. Poplatky ze psů mohou být za určitých podmínek vráceny nebo v případě neplacení vymáhány.

R₄ - Správa poplatků za lázeňský nebo rekreační pobyt – zaplatí ve stanovené výši FO nebo PO, která přechodné ubytování poskytla. Tyto poplatky jsou v případě nezaplacení vymáhány.

R₅ - Správa poplatků za užívání veřejného prostranství - zaplatí FO nebo PO, která užívá veřejné prostranství k provádění výkopových prací, umístění dočasných staveb a zařízení sloužících pro poskytování prodeje a služeb, pro umístění atrakcí a skládek, k vyhrazení trvalého parkovacího místa a k užívání veřejného prostranství pro kulturní akce. V případě nezaplacení poplatků bude poustoupeno k vymáhání.

R₆ – Správa poplatku ze vstupného – platí FO nebo PO, které akci pořádají. Pokud by nebyly poplatky uhrazeny, budou vymáhány.

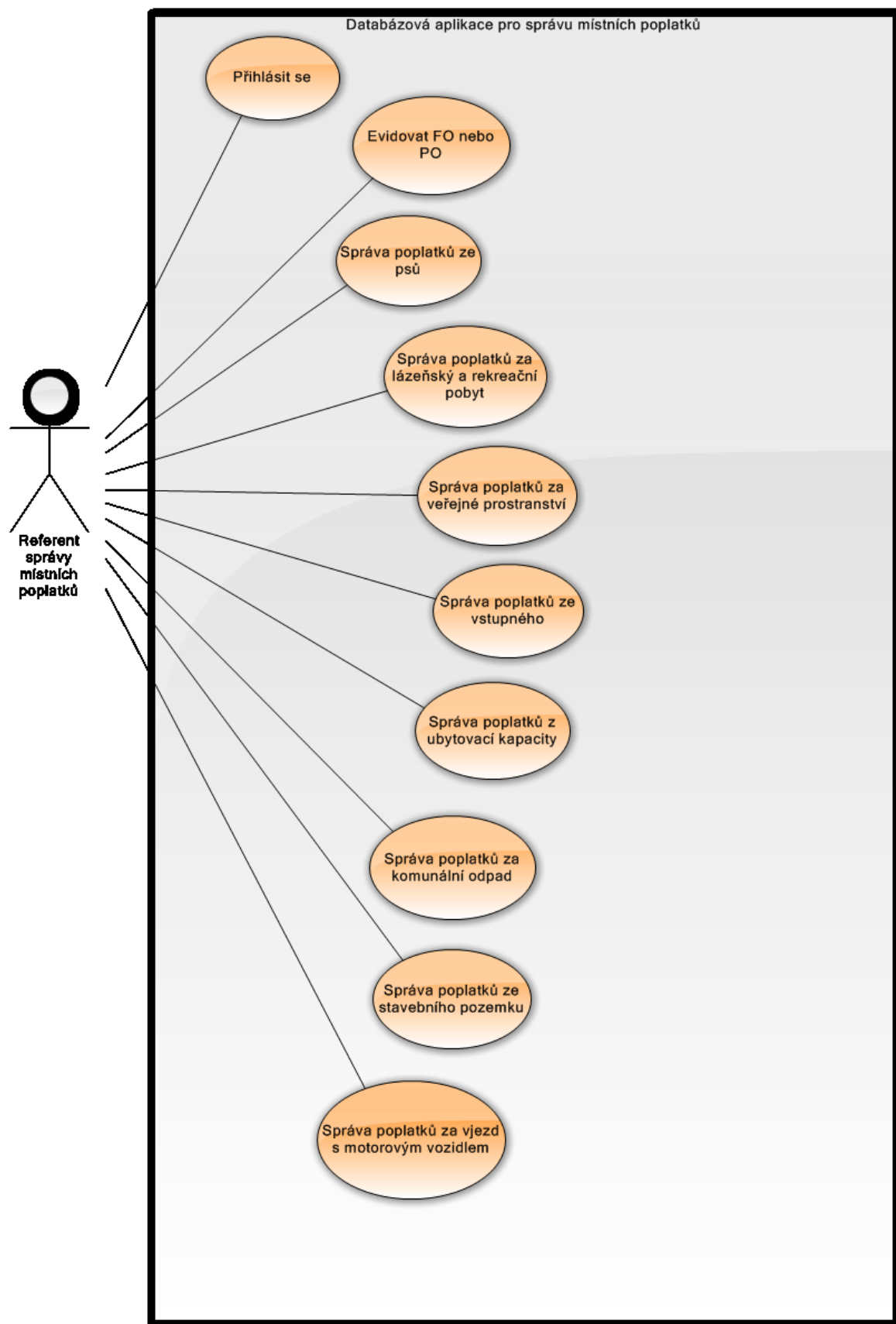
R₇ - Správa poplatku z ubytovací kapacity – platí FO nebo PO, která je ubytovatel. Sazba poplatku z ubytovací kapacity může být až 6 Kč za každé využití lůžko a den (rok 2014). Pokud jej ubytovatel neuhradí, budou poplatky vymáhány.

R₈ – Správa poplatků za povolení k vjezdu s motorovým vozidlem do vybraných míst a částí měst - platí FO nebo PO, kterým bylo vydáno povolení k vjezdu s motorovým vozidlem. Pokud plátce poplatek zaplatí paušální částkou na delší období, než pro které bude povolení potřebovat, bude mu vypočtená částka vrácena zpět. Pokud poplatek nezaplatí, bude vymáhán.

R₉ – Správa poplatků za komunální odpad - platí fyzická osoba, která má v obci trvalý pobyt, nebo která má ve vlastnictví stavbu určenou k individuální rekreaci. Poplatky je možno za určitých podmínek vrátit, ale při neplacení jsou vymáhány.

R₁₀ – Správa poplatku za zhodnocení stavebního pozemku – platí vlastník stavebního pozemku za možnost připojení k vodovodu nebo kanalizaci. Poplatek se platí obci, na jejímž území se nachází stavební pozemek. V případě vrácení pozemku obci může bývalý vlastník pozemku požádat o vrácení poplatku. Pokud vlastník pozemku poplatek neuhradí, je vymáhán.

4.1.2 Diagram případu užití



Obrázek 5: Diagram případů užití

Diagram případů užití zachycuje vztah aktéra (uživatele) k jednotlivým případům užití v navrhované databázové aplikaci.

Každý případ užití má svůj hlavní a alternativní scénář, vstupní a výstupní podmínky, primárního aktéra, rozsah a úroveň. Všechny případy užití s těmito vlastnostmi jsou dále podrobně rozebírány.

<p>Případ užití: UC1</p> <p>Stručný popis: Přihlášení do databázové aplikace</p> <p>Primární aktér: Referent správy místních poplatků</p> <p>Rozsah: Databázová aplikace</p> <p>Úroveň: Uživatelský cíl</p>
<p>Vstupní podmínky:</p> <p>Uživatel je pověřen správou místních poplatků.</p> <p>Administrátor databázové aplikace nastavil uživateli přihlašovací údaje.</p>
<p>Hlavní scénář:</p> <ol style="list-style-type: none">1 Případ užití začíná, chce-li se uživatel přihlásit do aplikace.2 Uživatel zadá přihlašovací jméno a své heslo.3 Systém uživatele přihlásí.
<p>Výstupní podmínky:</p> <p>Uživatel je úspěšně přihlášen.</p>
<p>Alternativní scénáře:</p> <ol style="list-style-type: none">1 Uživatel zadá přihlašovací jméno a heslo.2 Uživatel se nemůže přihlásit, protože platnost hesla vypršela, a proto je vyzván aplikací ke změně.3 Uživatel změní heslo.4 Uživatel zadá nové heslo a úspěšně se přihlásí do databázové aplikace.

<p>Případ užití: UC2</p> <p>Stručný popis: Evidence FO a PO</p> <p>Primární aktér: Referent správy místních poplatků</p> <p>Rozsah: Databázová aplikace</p> <p>Úroveň: uživatelský cíl</p>
<p>Vstupní podmínky: Uživatel je úspěšně přihlášen.</p>

Hlavní scénář:

1 Příklad užití začíná, pokud v zobrazeném menu vybere uživatel položku „evidence FO a PO“.

2 **IF** uživatel vybere "přidat FO nebo PO".

2.1 Databázová aplikace zobrazí formulář pro přidání nové FO nebo PO.

2.2 Uživatel vyplní formulář a klikne na tlačítko uložit.

2.3 Přidána FO nebo PO je úspěšně uložena do databáze.

3 **IF** uživatel vybere možnost "odstranit FO nebo PO".

3.1 Databázová aplikace uživateli zobrazí formulář s dotazem, jestli chce FO nebo PO skutečně odstranit.

3.2 Databázová aplikace smaže FO nebo PO, jen pokud nemá žádné pohledávky.

4 **IF** uživatel vybere možnost upravit FO nebo PO.

4.1 Databázová aplikace zobrazí formulář, který umožní editovat údaje.

4.2 Uživatel edituje údaje ve formuláři a poté klikne na tlačítko uložit.

4.3 Databázová aplikace uloží provedené změny.

Výstupní podmínky:

Obecní úřad úspěšně zaevidoval provedené úkony u PO nebo FO.

Příklad užití: UC3

Stručný popis: Správa poplatků ze psů

Primární aktér: Referent správy místních poplatků

Rozsah: Databázová aplikace

Úroveň: uživatelský cíl

Vstupní podmínky: Uživatel je úspěšně přihlášen.

Hlavní scénář:

1 Příklad užití začíná, pokud v zobrazeném menu vybere uživatel položku „správa poplatků ze psů“.

2 **INCLUDE** vyhledat FO nebo PO.

3 **IF** uživatel vybere „úhrada poplatku ze psů“.

3.1 Uživatel zadá datum, ve kterém byl poplatek zaplacen.

3.2 Uživatel zadá částku, kterou občan zaplatil.

3.3 Údaje se úspěšně uloží do databáze.

4 **IF** uživatel vybere „vratka poplatku za psa“.

4.1 Uživatel zadá datum, kdy došlo k úmrtí psa nebo odstěhování jeho majitele.

4.2 Databázová aplikace vypočítá vratnou částku.

4.3 Databázová aplikace psa vymaže z databázové aplikace.

5 **IF** uživatel vybere možnost „nedoplatky ze psů“.

5.1 Databázová aplikace na základě požadavku uživatele vygeneruje sestavy po datu splatnosti a zobrazí seznam dlužících FO a PO.

5.2 Databázová aplikace zobrazí u jednotlivých FO a PO dlužnou částku s datem splatnosti.

5.3 Uživatel každé FO a PO vyměří poplatek platebním výměrem, který lze zvýšit až na trojnásobek původní částky a stanoví nové datum splatnosti.

5.3.1 **IF** FO nebo PO neuhradí poplatek dle platebního výměru ve stanovené výši a termínu, je zahájeno exekuční řízení dle zákona 280/2009 Sb. Uživatel zadá nové datum splatnosti a výši exekučního poplatku.

5.4 Změny se uloží do databázové aplikace.

Výstupní podmínky:

Obecní úřad přijímá, vrací nebo vymáhá poplatek ze psů ve správné výši a v daném termínu.

Alternativní scénáře:

1 Aplikace nevyhledala FO nebo PO.

1.1 Uživatel zadal neexistující údaj.

1.2 **OR** FO nebo PO není zatím v obecní databázi, je nutné ji vložit podle ROB.

2 Údaje za uhrazení poplatku ze psů se nepodařilo uložit.

2.1 Uživatel nezadal údaje podle správné vstupní masky.

2.2 **OR** nevyplnil povinné údaje.

3 Uživatel osvobodil občana od platby.

3.1 Občan je držitelem průkazu ZTP/P 3. stupně.

<p>Případ užití: UC4</p> <p>Stručný popis: Správa poplatků za lázeňský nebo rekreační pobyt</p> <p>Primární aktér: Referent správy místních poplatků</p> <p>Rozsah: Databázová aplikace</p> <p>Úroveň: uživatelský cíl</p>
<p>Vstupní podmínky: Uživatel je úspěšně přihlášen.</p>
<p>Hlavní scénář:</p> <p>1 Případ užití začíná, pokud v zobrazeném menu vybere uživatel položku „správa poplatků za lázeňský nebo rekreační pobyt“.</p> <p>2 INCLUDE vyhledat FO nebo PO.</p> <p>3 IF uživatel vybere „úhrada poplatku za lázeňský nebo rekreační pobyt“.</p> <p> 3.1 Uživatel zadá částku, kterou obci odvede ubytovatel.</p> <p> 3.2 Uživatel vyplní dobu ubytování, účel ubytování, jména a adresy trvalého bydliště ubytovaných.</p> <p> 3.3 Databázová aplikace údaje úspěšně uloží.</p> <p>4 IF uživatel vybere možnost „nedoplatky za lázeňský nebo rekreační pobyt“.</p> <p> 4.1 Databázová aplikace na základě požadavku uživatele vygeneruje sestavy po datu splatnosti a zobrazí seznam dlužících ubytovatelů.</p> <p> 4.2 Databázová aplikace zobrazí u jednotlivých ubytovatelů dlužnou částku s datem splatnosti.</p> <p> 4.3 Uživatel každého ubytovatele vyměří poplatek platebním výměrem, který lze zvýšit až na trojnásobek původní částky a stanoví nové datum splatnosti.</p> <p> 4.3.1 IF ubytovatel neuhradí poplatek dle platebního výměru ve stanovené výši a termínu, je zahájeno exekuční řízení dle zákona 280/2009 Sb. Uživatel zadá nové datum splatnosti a výši exekučního poplatku.</p> <p> 4.4 Změny se uloží do databázové aplikace.</p>
<p>Výstupní podmínky:</p> <p>Obecní úřad přijímá nebo vymáhá poplatek za lázeňský nebo rekreační pobyt ve správné výši a v daném termínu.</p>
<p>Alternativní scénáře:</p> <p>1 Aplikace nevyhledala FO nebo PO.</p> <p> 1.1 Uživatel zadal neexistující údaj.</p> <p> 1.2 OR FO nebo PO není zatím v obecní databázi, je nutné ji vložit podle ROB.</p>

2 Údaje za uhrazení lázeňského nebo rekreačního poplatku se nepodařilo uložit.

2.1 Uživatel nezadal údaje podle správné vstupní masky.

2.2 **OR** nevyplnil povinné údaje.

3 Uživatel osvobodil občana od platby.

3.1 Občan je držitelem průkazu ZTP/P 3. stupně **OR** mladší 18 let **OR** starší 70 let.

Případ užití: UC5

Stručný popis: Správa poplatků za užívání veřejného prostranství

Primární aktér: Referent správy místních poplatků

Rozsah: Databázová aplikace

Úroveň: uživatelský cíl

Vstupní podmínky: Uživatel je úspěšně přihlášen.

Hlavní scénář:

1 Případ užití začíná, pokud v zobrazeném menu vybere uživatel položku „správa poplatků za užívání veřejného prostranství“.

2 **INCLUDE** vyhledat FO nebo PO.

3 **IF** uživatel vybere „úhrada poplatku za užívání veřejného prostranství“.

3.1 Uživatel zadá datum, ve kterém byl poplatek zaplacen.

3.2 Aplikace vypočítá částku, kterou musí občan zaplatit (až 10 Kč za m²/den).

3.3 Uživatel vyplní, o jaký typ poplatku se jedná (týdenní, měsíční nebo roční).

3.4 Údaje se uloží do databázové aplikace.

4 **IF** uživatel vybere možnost „nedoplatky za užívání veřejného prostranství“.

4.1 Databázová aplikace na základě požadavku uživatele vygeneruje sestavy po datu splatnosti a zobrazí seznam dlužících FO a PO.

4.2 Databázová aplikace zobrazí u jednotlivých FO a PO dlužnou částku s datem splatnosti.

4.3 Uživatel každé FO a PO vyměří poplatek platebním výměrem, který lze zvýšit až na trojnásobek původní částky a stanoví nové datum splatnosti.

4.3.1 **IF** FO a PO neuhradí poplatek dle platebního výměru ve stanovené výši a termínu, je zahájeno exekuční řízení dle zákona 280/2009 Sb.

<p>Uživatel zadá nové datum splatnosti a výši exekučního poplatku. 4.4 Změny se uloží do databázové aplikace.</p>
<p>Výstupní podmínky: Obecní úřad přijímá nebo vymáhá poplatek za užívání veřejného prostranství ve správné výši a v daném termínu.</p>
<p>Alternativní scénáře:</p> <p>1 Aplikace nevyhledala FO nebo PO.</p> <p>1.1 Uživatel zadal neexistující údaj. 1.2 OR FO nebo PO není zatím v obecní databázi, je nutné ji vložit podle ROB.</p> <p>2 Údaje za uhrazení poplatku za užívání veřejného prostranství se nepodařilo uložit.</p> <p>2.1 Uživatel nezadal údaje podle správné vstupní masky. 2.2 OR nevyplnil povinné údaje.</p>

<p>Případ užití: UC6</p> <p>Stručný popis: Správa poplatků ze vstupného</p> <p>Primární aktér: Referent správy místních poplatků</p> <p>Rozsah: Databázová aplikace</p> <p>Úroveň: uživatelský cíl</p>
<p>Vstupní podmínky: Uživatel je úspěšně přihlášen.</p>
<p>Hlavní scénář:</p> <p>1 Případ užití začíná, pokud v zobrazeném menu vybere uživatel položku „správa poplatků ze vstupného“.</p> <p>2 INCLUDE vyhledat FO nebo PO.</p> <p>3 IF uživatel vybere položku „úhrada poplatku ze vstupného“.</p> <p>3.1 Uživatel zadá datum konání akce. 3.2 Uživatel vypočítá částku ze vstupného (20% z vybraného vstupného nebo paušální částka).</p> <p>4 IF uživatel vybere možnost „nedoplatky ze vstupného“.</p> <p>4.1 Databázová aplikace na základě požadavku uživatele vygeneruje sestavy po datu splatnosti a zobrazí seznam dlužících FO a PO. 4.2 Databázová aplikace zobrazí u jednotlivých FO a PO dlužnou částku s datem splatnosti. 4.3 Uživatel každé FO a PO vyměří poplatek platebním výměrem, který lze</p>

zvýšit až na trojnásobek původní částky a stanoví nové datum splatnosti.

4.3.1 **IF** FO a PO neuhradí poplatek dle platebního výměru ve stanovené výši a termínu, je zahájeno exekuční řízení dle zákona 280/2009 Sb. Uživatel zadá nové datum splatnosti a výši exekučního poplatku.

4.4 Změny se uloží do databázové aplikace.

Výstupní podmínky:

Obecní úřad přijímá nebo vymáhá poplatek ze vstupného ve správné výši a v daném termínu.

Alternativní scénáře:

1 Aplikace nevyhledala FO nebo PO.

1.1 Uživatel zadal neexistující údaj.

1.2 **OR** FO nebo PO není zatím v obecní databázi, je nutné ji vložit podle ROB.

2 Údaje za uhrazení poplatku ze vstupného se nepodařilo uložit.

2.1 Uživatel nezadal údaje podle správné vstupní masky.

2.1 **OR** nevyplnil povinné údaje.

3 FO nebo PO je osvobozena od platby, protože se jedná o charitativní akci.

Případ užití: UC7

Stručný popis: Správa poplatků z ubytovací kapacity

Primární aktér: Referent správy místních poplatků

Rozsah: Databázová aplikace

Úroveň: uživatelský cíl

Vstupní podmínky: Uživatel je úspěšně přihlášen.

Hlavní scénář:

1 Případ užití začíná, pokud v zobrazeném menu vybere uživatel položku „správa poplatků z ubytovací kapacity“.

2 **INCLUDE** vyhledat FO nebo PO.

3 **IF** uživatel vybere „úhrada poplatku z ubytovací kapacity“.

3.1 Uživatel zadá datum, ve kterém ubytovatel částku zaplatil.

3.2 Uživatel zadá částku, kterou občan zaplatil (stanovena OZV).

3.3 Údaje se úspěšně uloží do databáze.

4 **IF** uživatel vybere možnost „nedoplatky z ubytovací kapacity“.

<p>4.1 Databázová aplikace na základě požadavku uživatele vygeneruje sestavy po datu splatnosti a zobrazí seznam dlužících ubytovatelů.</p> <p>4.2 Databázová aplikace zobrazí u jednotlivých ubytovatelů dlužnou částku s datem splatnosti.</p> <p>4.3 Uživatel každého ubytovatele vyměří poplatek platebním výměrem, který lze zvýšit až na trojnásobek původní částky a stanoví nové datum splatnosti.</p> <p>4.3.1 IF FO a PO neuhradí poplatek dle platebního výměru ve stanovené výši a termínu, je zahájeno exekuční řízení dle zákona 280/2009 Sb. Uživatel zadá nové datum splatnosti a výši exekučního poplatku.</p> <p>4.4 Změny se uloží do databázové aplikace.</p>
<p>Výstupní podmínky:</p> <p>Obecní úřad přijímá nebo vymáhá poplatky z ubytovací kapacity ve správné výši a v daném termínu.</p>
<p>Alternativní scénáře:</p> <p>1 Aplikace nevyhledala FO nebo PO.</p> <p>1.1 Uživatel zadal neexistující údaj.</p> <p>1.2 OR FO nebo PO není zatím v obecní databázi, je nutné ji vložit podle ROB.</p> <p>2 Údaje za uhrazení poplatku z ubytovací kapacity se nepodařilo uložit.</p> <p>2.1 Uživatel nezadal údaje podle správné vstupní masky.</p> <p>2.3 OR nevyplnil povinné údaje.</p> <p>3 Uživatel osvobodil občana od platby.</p> <p>3.1 Zařízení slouží k přechodnému ubytování žáků a studentů.</p>
<p>Případ užití: UC8</p> <p>Stručný popis: Správa poplatků za povolení k vjezdu s motorovým vozidlem</p> <p>Primární aktér: Referent správy místních poplatků</p> <p>Rozsah: Databázová aplikace</p> <p>Úroveň: uživatelský cíl</p>
<p>Vstupní podmínky: Uživatel je úspěšně přihlášen.</p>
<p>Hlavní scénář:</p> <p>1 Případ užití začíná, pokud v zobrazeném menu vybere uživatel položku „správa poplatků za povolení k vjezdu s motorovým vozidlem.“</p>

2 **INCLUDE** vyhledat FO nebo PO.

3 **IF** uživatel vybere „úhrada poplatku za povolení k vjezdu s motorovým vozidlem“.

3.1 Uživatel zadá datum, ve kterém byl poplatek zaplacen.

3.2 Uživatel zadá částku, kterou občan zaplatil (až 20 Kč/den nebo paušální poplatek).

3.3 Údaje se úspěšně uloží do databáze.

4 **IF** uživatel vybere „vratka poplatku za povolení k vjezdu s motorovým vozidlem“.

4.1 Uživatel zadá datum, kdy byl poplatek uhrazen.

4.2 Databázová aplikace vypočítá vratnou částku.

4.3 Databázová aplikace vymaže poplatek.

5 **IF** uživatel vybere možnost „nedoplatky za povolení k vjezdu s motorovým vozidlem“.

5.1 Databázová aplikace na základě požadavku uživatele vygeneruje sestavy po datu splatnosti a zobrazí seznam dlužících FO a PO.

5.2 Databázová aplikace zobrazí u jednotlivých FO a PO dlužnou částku s datem splatnosti.

5.3 Uživatel každé FO a PO vyměří poplatek platebním výměrem, který lze zvýšit až na trojnásobek původní částky a stanoví nové datum splatnosti.

5.3.1 **IF** FO a PO neuhradí poplatek dle platebního výměru ve stanovené výši a termínu, je zahájeno exekuční řízení dle zákona 280/2009 Sb. Uživatel zadá nové datum splatnosti a výši exekučního poplatku.

5.4 Změny se uloží do databázové aplikace.

Výstupní podmínky:

Obecní úřad přijímá nebo vrací poplatek za povolení vjezdu s motorovým vozidlem ve správné výši a v daném termínu.

Alternativní scénáře:

1 Aplikace nevyhledala FO nebo PO

1.1 Uživatel zadal neexistující údaj

1.2 **OR** FO nebo PO není zatím v obecní databázi, je nutné ji vložit podle ROB

2 Údaje za uhrazení poplatku za povolení k vjezdu s motorovým vozidlem se nepodařilo uložit.

2.1 Uživatel nezadal údaje podle správné vstupní masky

2.3 **OR** nevyplnil povinné údaje

3 Uživatel osvobodil občana od platby.

3.1 Uživatel je osoba ZTP.

3.2 **OR** osoba má trvalý pobyt nebo vlastní nemovitost ve vybraném místě.

Případ užití: UC9

Stručný popis: Správa poplatků za komunální odpad

Primární aktér: Referent správy místních poplatků

Rozsah: Databázová aplikace

Úroveň: uživatelský cíl

Vstupní podmínky: Uživatel je úspěšně přihlášen.

Hlavní scénář:

1 Případ užití začíná, pokud v zobrazeném menu vybere uživatel položku „správa poplatků za komunální odpad“.

2 **INCLUDE** vyhledat FO.

3 **IF** uživatel vybere „úhrada poplatku za komunální odpad“.

3.1 Uživatel zadá datum, ve kterém byl poplatek zaplacen.

3.2 Uživatel zadá částku, kterou občan zaplatil.

3.3 Údaje se úspěšně uloží do databáze.

4 **IF** uživatel vybere „vratka poplatku za komunální odpad“.

4.1 Uživatel zadá datum, kdy došlo k úmrtí nebo odstěhování FO.

4.2 Databázová aplikace vypočítá vratnou částku.

4.3 Uživatel vymaže tento poplatek z databázové aplikace.

5 **IF** uživatel vybere možnost „nedoplatky za komunální odpad“.

5.1 Databázová aplikace na základě požadavku uživatele vygeneruje sestavy po datu splatnosti a zobrazí seznam dlužících FO a PO.

5.2 Databázová aplikace zobrazí u jednotlivých FO a PO dlužnou částku s datem splatnosti.

5.3 Uživatel každé FO a PO vyměří poplatek platebním výměrem, který lze zvýšit až na trojnásobek původní částky a stanoví nové datum splatnosti.

5.3.1 **IF** FO a PO neuhradí poplatek dle platebního výměru ve stanovené výši a termínu, je zahájeno exekuční řízení dle zákona 280/2009 Sb. Uživatel zadá nové datum splatnosti a výši exekučního poplatku.

5.4 Změny se uloží do databázové aplikace.

Výstupní podmínky:

Obecní úřad přijímá, vrací nebo vymáhá poplatek za komunální odpad ve správné výši a v daném termínu.

Alternativní scénáře:

1 Aplikace nevyhledala FO.

1.1 Uživatel zadal neexistující údaj.

1.2 **OR** FO není zatím v obecní databázi, je nutné ji vložit podle ROB.

2 Údaje za uhrazení poplatku za komunální odpad se nepodařilo uložit.

2.1 Uživatel nezadal údaje podle správné vstupní masky.

2.2 **OR** nevyplnil povinné údaje.

Případ užití: UC10

Stručný popis: Správa poplatků za zhodnocení stavebního pozemku

Primární aktér: Referent správy místních poplatků

Rozsah: Databázová aplikace

Úroveň: uživatelský cíl

Vstupní podmínky: Uživatel je úspěšně přihlášen.

Hlavní scénář:

1 Případ užití začíná, pokud v zobrazeném menu vybere uživatel položku „správa poplatků za zhodnocení stavebního pozemku“.

2 **INCLUDE** vyhledat FO nebo PO.

3 **IF** uživatel vybere „úhrada poplatku za zhodnocení stavebního pozemku“.

3.1 Uživatel zadá datum, ve kterém byl poplatek zaplacen.

3.2 Uživatel zadá částku, kterou občan zaplatil.

3.3 Údaje se úspěšně uloží do databáze.

4 **IF** uživatel vybere „vratka poplatku za zhodnocení stavebního pozemku“.

4.1 Uživatel zadá datum, kdy došlo ke zrušení vlastnickým práv plátce.

4.2 Databázová aplikace vypočítá vratnou částku.

4.3 Uživatel vymaže poplatek z databázové aplikace.

5 **IF** uživatel vybere možnost „nedoplatky za zhodnocení stavebního pozemku“.

5.1 Databázová aplikace na základě požadavku uživatele vygeneruje sestavy po datu splatnosti a zobrazí seznam dlužících FO a PO.

5.2 Databázová aplikace zobrazí u jednotlivých FO a PO dlužnou částku

s datem splatnosti.

5.3 Uživatel každé FO a PO vyměří poplatek platebním výměrem, který lze zvýšit až na trojnásobek původní částky a stanoví nové datum splatnosti.

5.3.1 **IF** FO a PO neuhradí poplatek dle platebního výměru ve stanovené výši a termínu, je zahájeno exekuční řízení dle zákona 280/2009 Sb. Uživatel zadá nové datum splatnosti a výši exekučního poplatku.

5.4 Změny se uloží do databázové aplikace.

Výstupní podmínky:

Obecní úřad přijímá, vrací nebo vymáhá poplatek za zhodnocení stavebního pozemku ve správné výši a v daném termínu.

Alternativní scénáře:

1 Aplikace nevyhledala FO nebo PO.

1.1 Uživatel zadal neexistující údaj.

1.2 **OR** FO nebo PO není zatím v obecní databázi, je nutné ji vložit podle ROB.

2 Údaje za uhrazení poplatku za zhodnocení stavebního pozemku se nepodařilo uložit.

2.1 Uživatel nezadal údaje podle správné vstupní masky.

2.2 **OR** nevyplnil povinné údaje.

4.1.3 Matice RTM

Matice sledovatelnosti požadavků slouží k ověřování konzistence. Pokud není požadavek mapován na žádný případ užití, znamená to, že v modelu chybí případ užití. Platí to samozřejmě i naopak. Z následující tabulky lze vyčíst, že všechny požadavky se shodují s případem užití, a proto by měla databázová aplikace poskytovat všechny důležité funkcionality.

Požadavky	UC ₁	UC ₂	UC ₃	UC ₄	UC ₅	UC ₆	UC ₇	UC ₈	UC ₉	UC ₁₀
R ₁	x									
R ₂		x								
R ₃			x							

R ₄				x						
R ₅					x					
R ₆						x				
R ₇							x			
R ₈								x		
R ₉									x	
R ₁₀										x

Tabulka 6: Matice RTM

4.1.4 Analytický model tříd

Diagram tříd (Class Diagram) představuje statický pohled na modelovaný systém. Jeho úkolem je znázornit typy objektů v systému a jejich vztahy. Návrh tříd, jejich odpovědností a následné vytvoření tohoto diagramu patří mezi základní kroky návrhu.

Analýza slovního příběhu slouží k rozpoznání **tříd**, *atributů* a **metod**. Podstatná jména v příběhu reprezentují třídy a jejich atributy, slovesa reprezentují metody.

Analýza slovního příběhu:

Databázová aplikace pro správu místních poplatků slouží k evidenci **osob**, správě poplatků za **psy**, **rekreační pobyt**, **veřejné prostranství**, **vstupné**, **ubytovací kapacitu**, **povolení vjezdu s motorovým vozidlem**, **komunální odpad** a **zhodnocení stavebního pozemku**.

Fyzické nebo právnické osoby lze přidávat, odstraňovat nebo u nich editovat parametry, mezi které patří *titul*, *jméno*, *příjmení*, *datum narození*, *rodné číslo*, *obec*, *ulice*, *číslo popisné*, *PSC*, *název společnosti*, *obchodní jméno*, *IČO*.

Při správě jednotlivých typů poplatků musí referent místních poplatků vyhledat osobu, která je plátcem. Osoby lze vyhledávat podle rodného čísla, příjmení, čísla popisného, názvu společnosti nebo podle IČO.

Poplatek ze psů platí držitel psa. Držitelem je fyzická nebo právnická osoba. Pokud držitel přihlásí psa, může referent správy místních poplatků zadat jeho *jméno, číslo známky, rasu, stáří*, musí stanovit *výši sazby* a určit *typ poplatníka*.

Poplatek za lázeňský nebo rekreační pobyt platí fyzické osoby, které přechodně a za úplatu pobývají v lázeňských místech a v místech soustředěného turistického ruchu za účelem léčení nebo rekreace. Poplatek za lázeňský nebo rekreační pobyt ve stanovené výši vybere a obci odvede ubytovatel. Referent v tomto případě může zaevidovat u ubytovaných *počet dní, účel pobytu, datum zahájení pobytu, jméno, adresu, číslo občanského průkazu*, musí zadat *sazbu* a určit *typ poplatníka*.

Poplatek za užívání veřejného prostranství platí fyzické i právnické osoby, které užívají veřejné prostranství. Při platbě tohoto poplatku referent zadá *účel záboru, umístění, zabranou plochu, datum užívání, dobu užívání, sazbu a určí typ poplatníka*.

Poplatek ze vstupného platí fyzické a právnické osoby, které akci pořádají. V tomto případě zadá referent do databázové aplikace *účel akce, místo konání, datum konání akce, počet dní, povolení, částku* za prodané vstupenky a určí *typ poplatníka*.

Poplatek z ubytovací kapacity platí ubytovatel, kterým je fyzická nebo právnická osoba, která přechodně ubytování poskytla. V tomto případě může referent zadat datum *zahájení pobytu a počet dní, sazbu, typ poplatníka*. Dále může být v databázové aplikaci evidováno *jméno, příjmení, adresa a číslo občanského průkazu* ubytovaného.

Pokud chce fyzická nebo právnická osoba zaplatit **poplatek za povolení vjezdu s motorovým vozidlem**, referent správy místních poplatků zadá *datum a délku udělení povolení, ulici, číslo parcely, sazbu a typ poplatníka*. Také může referent zadat do aplikace konkrétní *vozidlo* a jeho *poznávací značku*.

Poplatek za komunální odpad platí fyzická osoba, která má v obci trvalý pobyt. Při platbě poplatku zadá referent konkrétní *sazbu, typ poplatníka a popis případu*.

V případě, že chce vlastník stavebního pozemku zaplatit **poplatek za zhodnocení stavebního pozemku** za připojení k vodovodu nebo kanalizaci, referent správy místních poplatků zaeviduje *ulici, číslo parcely, cenu pozemku a konkrétní sazbu*.

Tyto poplatky lze v případě jejich úhrady po určitém časovém intervalu odstranit.

Každý typ poplatku má svůj **závazek**. U těchto závazků eviduje referent při platbě *typ závazku, způsob úhrady, částku v Kč* (shodná se sazbou), *platbu* (kolik poplatník aktuálně zaplatil), *datum platby, datum splatnosti, variabilní symbol platby*. Pokud chce referent zobrazit aktuální účetní stav za dané období, zobrazí se mu ve formuláři *datum stavu, stav k určitému datu a zaplacení* (zaškrtně, zda bylo zapláceno včas a ve správné výši). Má-li být

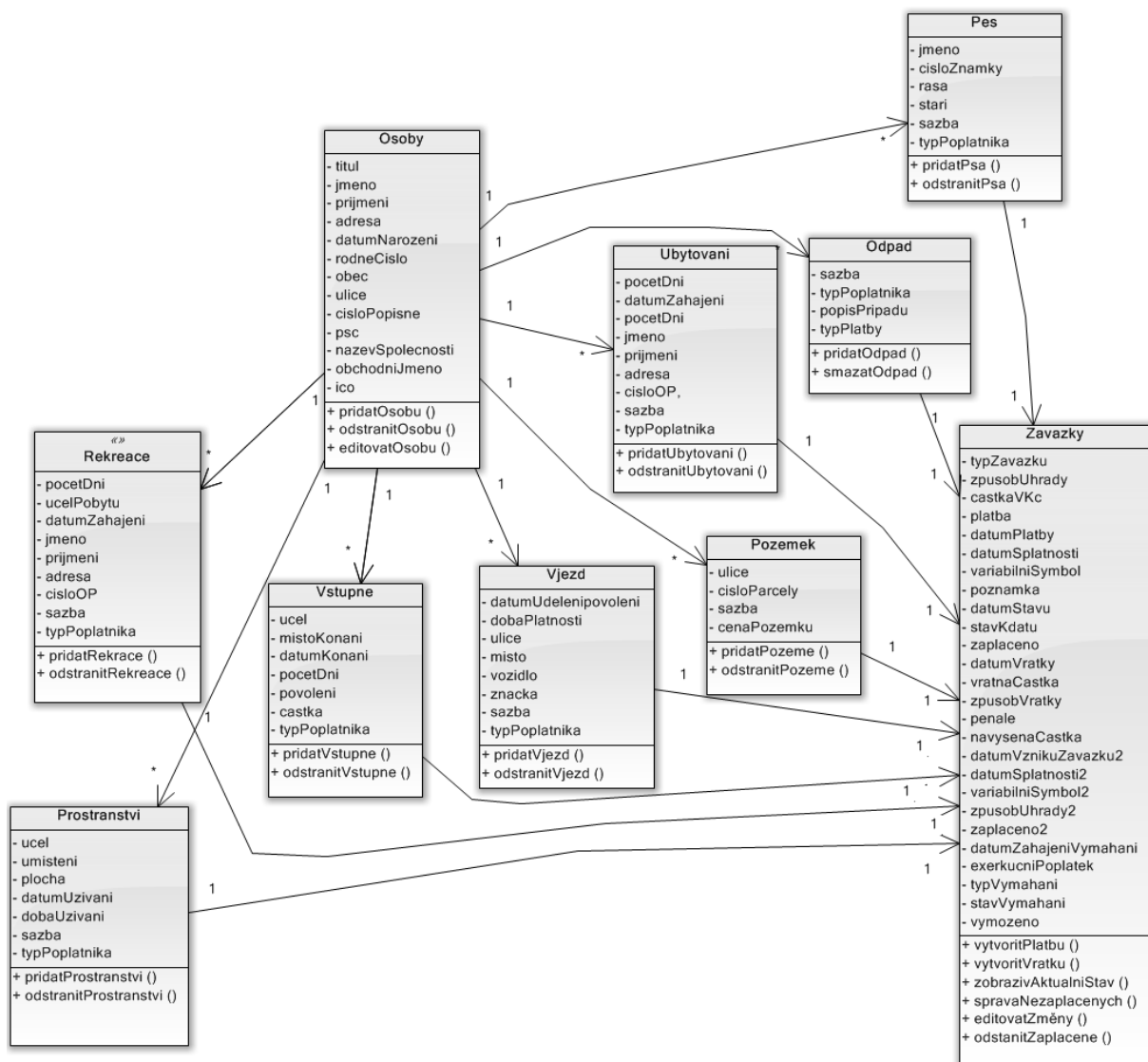
občanovi poplatek vrácen, aplikace vypočítá *vratnou částku*, referent zadá *datum vratky* a *způsob vratky*.

Jestliže nebude poplatek uhrazen včas a ve správné výši, nastává proces vymáhání. Referent v tomto případě postupuje podle zákona č. 280/2009 Sb. Musí zadat penále, *navýšenou částku v Kč*, *datum vzniku závazku*, *datum splatnosti*, *variabilní symbol* a *způsob úhrady*. Pokud dlužník ani poté poplatek neuhradí, nastaví referent stav zaplacení na hodnotu *ne*. V tomto případě referent zadá *datum zahájení vymáhání*, *exekční poplatek* (500 Kč za rok), *typ vymáhání*, *stav vymáhání*. Je-li *vymoženo* nastaveno na hodnotu *ano*, je poplatek uhrazen.

Třída	<i>Atributy</i>	Operace	Spolupráce
Osoby	titul, jmeno, prijmeni, datumNarozeni, rodneCislo, obec, ulice, cisloPopisne, PSC, nazevSpolecnosti, obchdoniJmeno, ico	pridatOsobu, odstranitOsobu, editovatOsobu, vyhledatPodleRc, vyhledatPodlePrijmeni, vyhledatPodleCP, vyhledatPodleNazvu, vyhledatPodleDic	Pes, Rekrece, Prostranstvi, Vstupne, UbytovaciKapacita, PovoleniVjezdu, Odpad, Pozemek
Pes	jmeno, cisloZnamky, rasa, stari, sazba, typPoplatnika	pridatPsa, odstranitPsa	Osoby, Zavazek
Rekreace	pocetDni, ucelPobytu, datumZahajeniPobytu, jmeno, prijmeni, adresa, cisloOP, sazba, typPoplatnika	pridatRekreace, odstranitRekreace	Osoby, Zavazek
Prostranstvi	ucel, umisteni, plocha, datumZahajeniUzivani, dobaUzivani, sazba, typ poplatnika	pridatProstranstvi, odstranitProstranstvi	Osoby, Zavazek
Vstupne	ucel, mistoKonani, datumKonani, pocet dni povoleni, castka, typPoplatnika	pridatVstupne, odstranitVstupne	Osoby, Zavazek
Ubytovaci Kapacita	pocetDni, datumZahajeniUbytovani,	pridatKapacitu, odstranitKapacitu	Osoby, Zavazek

	jmeno, prijmeni, adresa, cisloOP, dobaUkonceni, sazba, typPoplatnika		
PovoleniVjezdu	datumUdeleniPovoleni, dobaPlatnosti, ulice, misto, vozilo, znacka, sazba, typPoplatnika	pridatPovoleni, odstranitPovoleni	Osoby, Zavazek
Odpad	sazba, typPoplatnika, popisPripadu	pridatOdpad, odstranitOdpad	Osoby, Zavazek
Pozemek	ulice, cisloParcely, cenaPozemku, sazba	pridatPozemek, odstranitPozemek	Osoby, Zavazek
Zavazek	typZavazku, zpusobUhrady platby – (castkavKc, datumVzniku, datumSplatnosti, variabilniSymbol, zpusobUhrady, poznamka,) stavy – (datumStavu, stavKdatu, stavZaplaceni) vratky – (vratnaCastka, datumVratky, zpusobVratky) typPohledavky, vymáhání – (penale, castkavKc, datumVzniku, datumSplatnosti, variabilniSymbol, zpusobUhrady, poznamka, stavZaplaceni, datumZahajeniVymahani, exekucniPoplatek, typVymahani, stavVymahani vymozeno)	vytvoritPlatbu, vytvoritVratku, zobrazitStav, odstranitZaplacene, spravovatVymahani, editovatZmeny,	Pes, Prostranstvi, vstupne, Ubytovaci Kapacita, Povoleni vjezdu, Odpad, Pozemek,

Tabulka 7: Zpracování příběhu analytických tříd



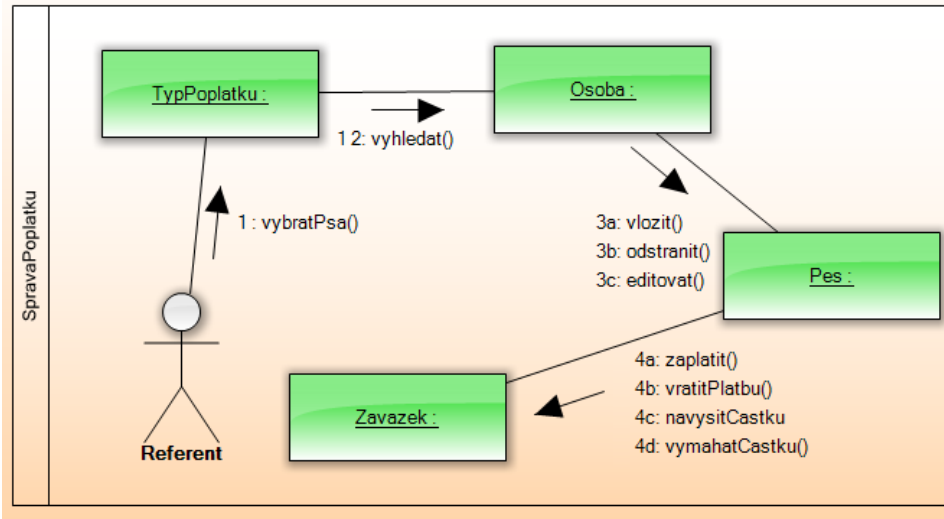
Obrázek 6: Diagram tříd

4.1.5 Realizace případu užití

V procesu realizace případu užití jsou modelovány interakce mezi objekty. Analytické třídy umožňují modelovat statickou strukturu systému, zatímco realizace případů užití slouží k popisu spolupráce instancí analytických tříd pro dosažení požadovaného chování systému. Diagramy realizace případů užití tímto umožňují dynamický pohled na systém. V této podkapitole jsou navrženy sekvenční diagramy, komunikační diagram a diagram aktivit.

Komunikační diagram

Následující obrázek znázorňuje komunikační diagram případu užití UC3, který popisuje správu místních poplatků ze psů. Pro UC4 - UC10 platí naprosto stejný postup komunikace, s výjimkou toho, že v prvním bodě referent vybere jiný typ poplatku.

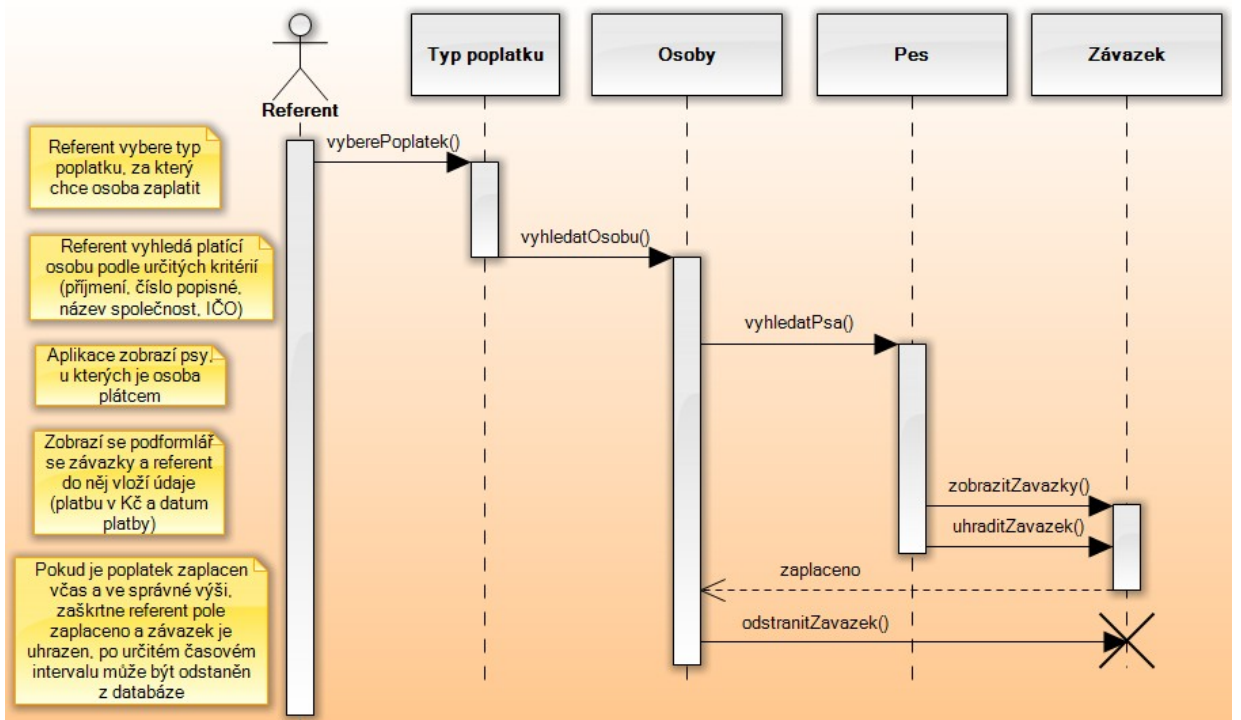


Obrázek 7: Komunikační diagram

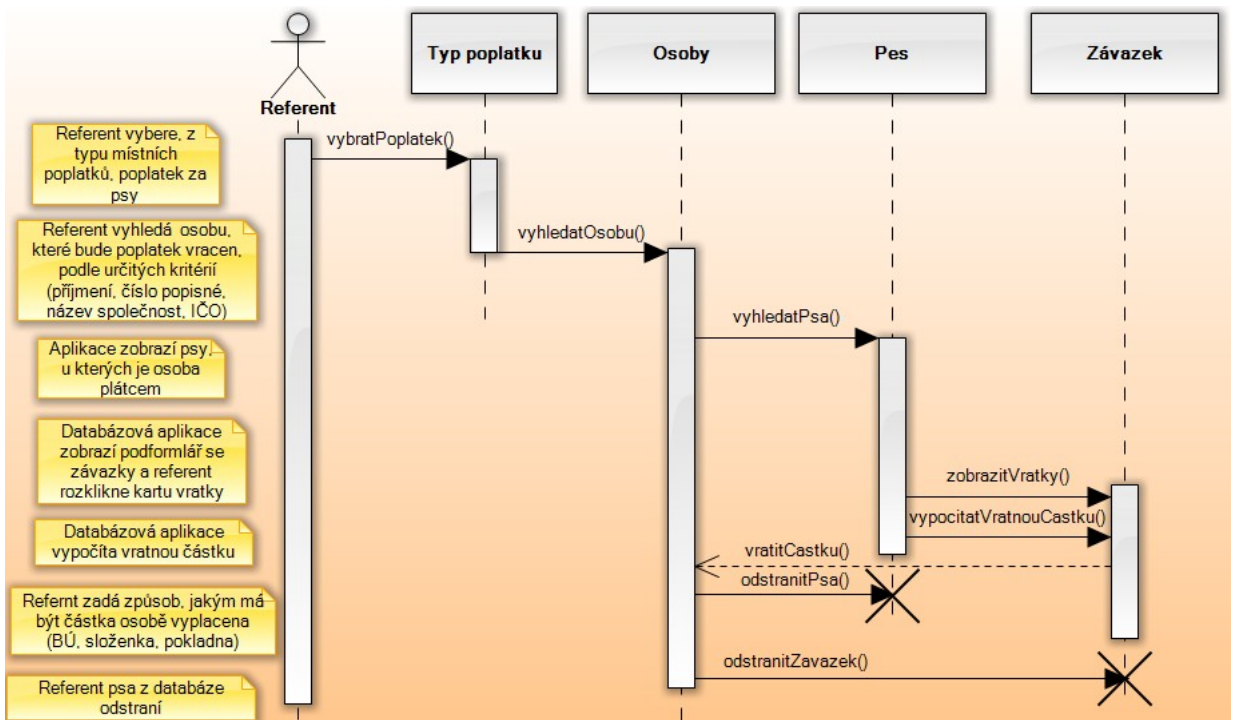
Komunikační zpráva vrátit platbu (4. b) neplatí pro UC4 – UC7.

Sekvenční diagram

Sekvenční diagram popisuje diagram případu užití pro správu místních poplatku ze psů (UC3). Podle rozpracovaného případů užití je zřejmé, že osoby mohou tento poplatek zaplatit, nebo jim může být poplatek vrácen. V případě nedodržení data splatnosti následuje proces vymáhání. Proto jsou sekvenční diagramy rozděleny na tři části, které zvlášť zachycují každý typ operace (platba, vratka a vymáhání dluhu).

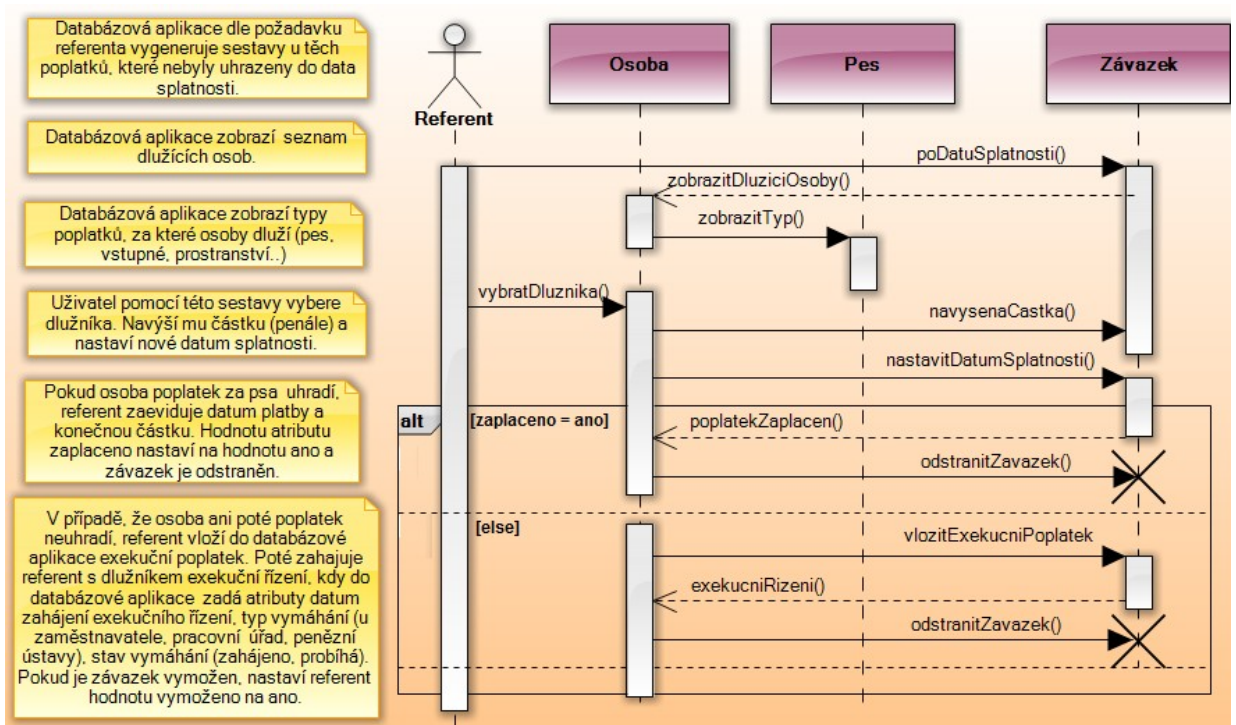


Obrázek 8: Sekvenční diagram - platba



Obrázek 9: Sekvenční diagram - vratka

Sekvenční diagram pro vrácení poplatků neplatí pro UC4 – UC7.

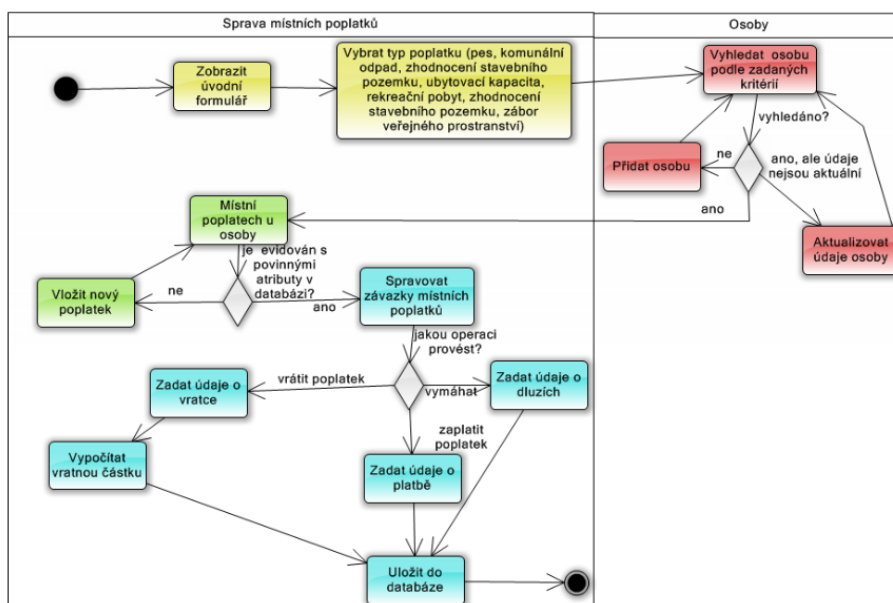


Obrázek 10: Sekvenční diagram – vymáhání dluhu

Stejný postup interakce platí i pro diagramy případů užití (UC4-UC10), liší se pouze výběrem typů poplatků.

Diagram aktivit

Obrázek diagramu aktivit zachycuje celý proces správy místních poplatků, podle rozpracovaných scénářů případů užití UC2 – UC10. Postupuje se podle hlavního scénáře, alternativní scénář není v tomto diagramu zahrnutý.



Obrázek 11: Diagram aktivit

4.2 Logický návrh databáze

V této kapitole je vytvořen logický návrh databáze. Diplomová práce se zabývá relačním modelem dat, a proto logický návrh databáze reprezentuje důležité entity a relace jako množinu základních tabulek. Tento návrh slouží jako zdroj informací pro fázi fyzického návrhu databáze.

4.2.1 Tabulky

Tabulky vycházejí z diagramu tříd a případů užití. Mají definovány své primární a cizí klíče, každý atribut je charakterizován určitými vlastnostmi (datový typ, velikost pole, povinnost pole). U datových typů automatické číslo, měna, datum a čas, ano/ne, není stanovena velikost pole, ale u fyzického návrhu databáze se těmto typům nastaví formát.

Tabulka *tblOsoba* - je navržena podle třídy Osoba.

Název pole	Datový typ	Velikost pole	Je nutno zadat
IDosoba (PK)	Automatické číslo	-	ano
titul	Text	8	ne
jmeno	Text	15	ne
prijmeni	Text	20	ne
datumNarozeni	Datum a čas	-	ne
rodneCislo	Číslo	10	ne
obec	Text	15	ano
ulice	Text	40	ano
cisloPopisne	Číslo	6	ano
nazevSpolecnosti	Text	40	ne
obchodniJmeno	Text	40	ne
ico	Číslo	8	ne

Tabulka 8: Tabulka *tblOsoba*

Tabulka *tblPes* - navržena třídy Pes.

Název pole	Datový typ	Velikost pole	Je nutno zadat
<i>IDosoba (FK)</i>	Číslo	-	ano
IDpsa (PK)	Automatické číslo	-	ano
jmenoPsa	Text	10	ano
cisloZnamky	Číslo	6	ano
rasa	Text	10	ne

stari	Číslo	2	ne
sazba	Měna	4	ano
typPoplatnika	Ano/ne	-	ano

Tabulka 9: Tabulka tblPes

Tabulka *tblRekreace* - navržena podle třídy Rekreace.

Název pole	Datový typ	Velikost pole	Je nutno zadat
<i>IDosoba (FK)</i>	Číslo	-	ano
IDrekrace (PK)	Automatické číslo	-	ano
pocetDni	Číslo	4	ano
ucelPobytu	Text	20	ne
datumZahajeni	Datum a čas	-	ano
jmeno	Text	15	ne
prijmeni	Text	20	ne
adresa	Text	50	ne
cisloOP	Číslo	10	ne
sazba	Měna	-	ano
typPoplatnika	Ano/ne	-	ano

Tabulka 10: Tabulka tblRekreac

Tabulka *tblProstranstvi* - navržena podle třídy Prostranství.

Název pole	Datový typ	Velikost pole	Je nutno zadat
<i>IDosoba (FK)</i>	Číslo	-	ano
IDprostranstvi (PK)	Automatické číslo	-	ano
ucel	Text	20	ano
plocha	Číslo	5	ano
datumUzivani	Datum a čas	-	ano
dobaUzivani	Číslo	-	ano
sazba	Měna	-	ano
typPoplatnika	Ano/ne	-	ano

Tabulka 11: Tabulka tblProstranstvi

Tabulka *tblVstupne* - navržena podle třídy Vstupne.

Název pole	Datový typ	Velikost pole	Je nutno zadat
<i>IDosoba (FK)</i>	Číslo	-	ano
IDvstupne (PK)	Automatické číslo	-	ano

mistoKonani	Text	20	ano
datumKonani	Datum a čas	-	ano
pocetDni	Číslo	2	ano
sazba	Měna	-	ano
povoleni	Ano/ne	-	ano
typPoplatnika	Ano/ne	-	ano

Tabulka 12: Tabulka tblVstupne

Tabulka *tblUbytovani* - navržena podle třídy Ubytovani.

Název pole	Datový typ	Velikost pole	Je nutno zadat
IDubytovani (PK)	Automatické číslo	-	ano
<i>IDosoba (FK)</i>	Číslo	-	ano
datumZahajeni	Datum a čas	-	ano
pocetDni	Číslo	2	ano
jmeno	Text	15	ne
prijmeni	Text	20	ne
adresa	Text	50	ne
cisloOP	Číslo	10	ne
sazba	Měna	-	ano
typPoplatnika	Ano/ne	-	ano

Tabulka 13: Tabulka tblUbytovani

Tabulka *tblPovoleniVjezdu* - navržena podle třídy Vjezd.

Název pole	Datový typ	Velikost pole	Je nutno zadat
<i>IDosoba (FK)</i>	Číslo	-	ano
IDvjezd (PK)	Automatické číslo	-	ano
datumUdeleniPovoleni	Datum a čas	-	ano
dobaPlatnostiPovoleni	Datum a čas	-	ano
ulice	Text	50	ano
misto	Text	20	ano
vozidlo	Text	10	ne
znacka	Text	10	ne
sazba	Měna	-	ano
typPoplatnika	Ano/ne	-	ano

Tabulka 14: Tabulka tblPovoleniVjezdu

Tabulka *tblOdpad* - navržena podle třídy Odpad.

Název pole	Datový typ	Velikost pole	Je nutno zadat
<i>IDosoba (FK)</i>	Číslo	-	ano
IDodpad (PK)	Automatické číslo	-	ano
sazba	Měna	-	ano
platce	Text	10	ano
popisPripadu	Text	50	ne

Tabulka 15: Tabulka *tblOdpad*

Tabulka *tblPozemek* - navržena podle třídy Pozemek.

Název pole	Datový typ	Velikost pole	Je nutno zadat
IDpozemek (PK)	Automatické číslo	-	ano
<i>IDosoba (FK)</i>	Číslo	-	ano
ulice	Text	50	ano
cisloParcely	Číslo	10	ano
cenaPozemku	Měna	-	ano
sazba	Měna	-	ano

Tabulka 16: Tabulka *tblPozemek*

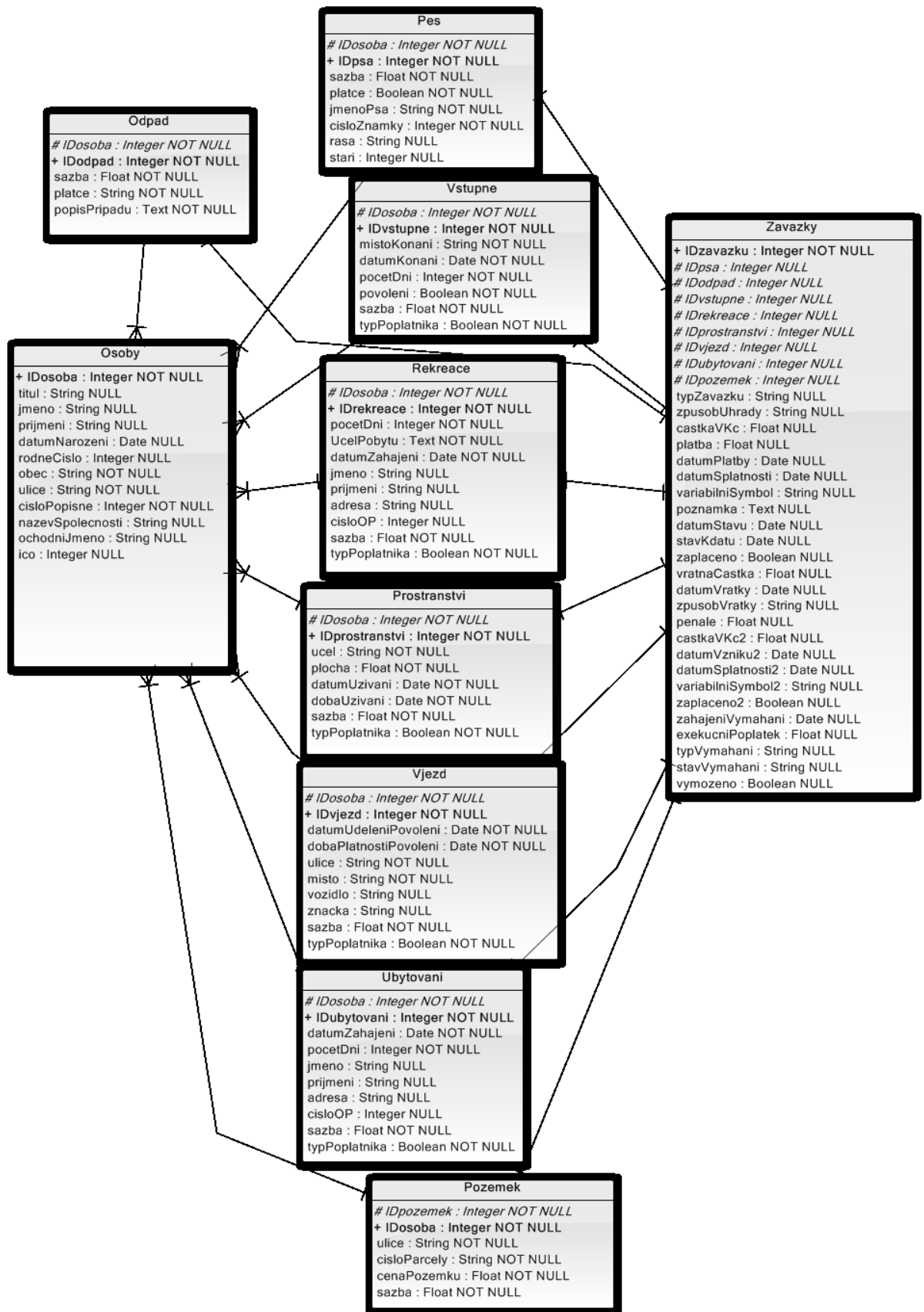
Tabulka *tblZavazky* – navržena podle třídy Zavazky.

Název pole	Datový typ	Velikost pole
IDzavazku (PK)	Automatické číslo	-
<i>IDpsa (FK)</i>	Číslo	-
<i>IDodpad (FK)</i>	Číslo	-
<i>IDrekreace (FK)</i>	Číslo	-
<i>IDprostranstvi (FK)</i>	Číslo	-
<i>IDvstupne (FK)</i>	Číslo	-
<i>IDvyjezd (FK)</i>	Číslo	-
<i>IDubytovani (FK)</i>	Číslo	-
<i>IDpozemek (FK)</i>	Číslo	-
typZavazku	Text	10
zpusobUhrady	Text	20
castkaVkc	Měna	-
platba	Měna	-
datumPlatby	Datum a čas	-

datumSplatnosti	Datum a čas	-
variabilniSymbol	Číslo	10
poznamka	Text	50
datumStavu	Datum a čas	-
stavKdatu	Datum a čas	-
zaplaceno	Ano/ne	-
vratnaCastka	Měna	-
datumVratky	Datum a čas	-
zpusobPlatby2	Text	20
penale	Měna	-
castkaVKc2	Měna	-
datumVzniku2	Datum a čas	-
datumSplatnosti2	Datum a čas	-
variabilniSymbol2	Číslo	-
Zaplaceno2	Ano/ne	-
zahajeniVymahani	Datum a čas	-
exekucniPoplatek	Měna	-
typVymahani	Text	50
stavVymahani	Text	10
vymozeno	Ano/ne	-

Tabulka 17: Tabulka tblZavazky

4.2.2 ER diagram



Obrázek 12: ER diagram

V této práci je vytvořen ER diagram, který je nezávislý na databázové technologii. I když předpokládám, že databázová aplikace bude dokončena v MS Accessu, programátor může zvolit jiný databázový server.

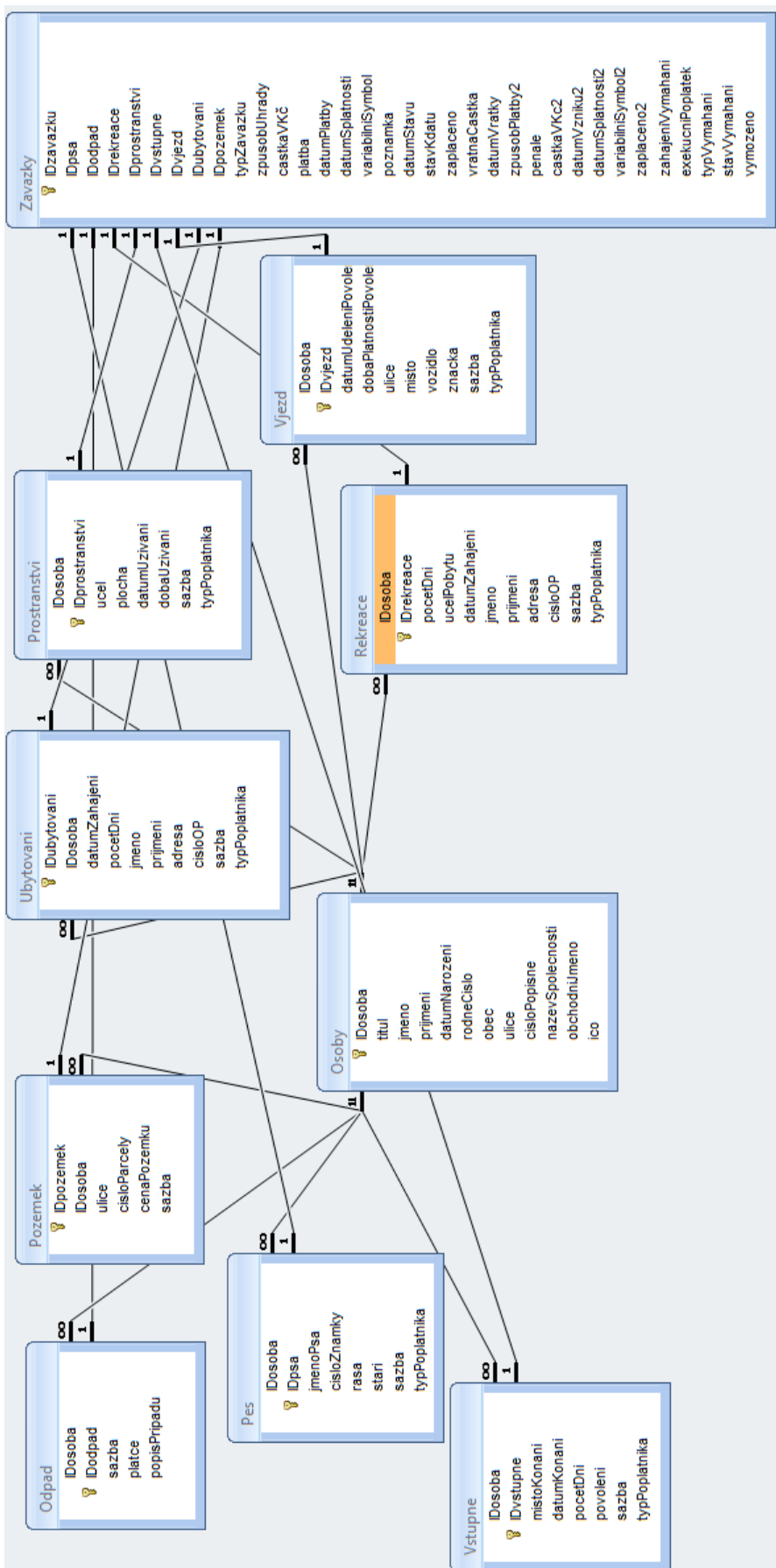
Navržené tabulky jsou mapovány do ER diagramu. Tabulky jsou vzájemně propojeny a jejich vztahy jsou definovány takto:

- Osoby, Pes (1:N)
- Osoby, Odpad (1:N)
- Osoby, Vstupne (1:N)
- Osoby, Pozemek (1:N)
- Osoby, Ubytovani (1:N)
- Osoby, Rekrece (1:N)
- Osoby, Vjezd (1:N)
- Osoby, Prostranstvi (1:N)
- Pes, Zavazky (1:1)
- Odpad, Zavazky (1:1)
- Vstupne, Zavazky (1:1)
- Pozemek, Zavazky (1:1)
- Ubytovani, Zavazky (1:1)
- Rekrece, Zavazky (1:1)
- Vjezd, Zavazky (1:1)
- Prostranstvi, Zavazky (1:1)

4.3 Implementace databázové aplikace

Kapitola implementace databázové aplikace se zabývá návrhem relačního modelu, formulářů a sestav v MS Accessu 2007.

Relační model



Obrázek 13: Relační model v MS Access

Návrh atributů vychází z diagramu tříd, primární a cizí klíče se řídí ER diagramem. Datové typy a velikost polí byly navrženy podle kapitoly tabulky (4.2.1).

Vazby mezi diagramy také vycházejí z ER diagramu. K jedné osobě může být přiřazeno více poplatků jednoho typu, proto je tento vztah definován 1:N. Každý typ poplatku má právě jeden závazek, proto jsou tyto relace ve vztahu 1:1.

4.3.1 Formuláře a sestavy

Tato podkapitola popisuje návrh a vzhled některých formulářů a sestav databázové aplikace.

Úvodní formulář

Referent se přihlásí do databázové aplikace. Při zadání správného přihlašovacího jména a hesla se zobrazí úvodní formulář. Tento formulář obsahuje správu místních poplatků, generování sestav, dluhů a evidenci osob.



Obrázek 14: Úvodní formulář

Po kliknutí na tlačítko osoby se zobrazí formulář s fyzickými a právnickými osobami, které lze vyhledávat, přidávat, editovat a mazat. Údaje, týkající se FO a PO, musí referent průběžně kontrolovat a aktualizovat podle registru obyvatel.

Platba za psa

Pokud osoba uhradí poplatek za psa, referent musí tyto údaje zaevidovat do databáze. V úvodním formuláři klikne na tlačítko pes a poté se zobrazí následující formulář:

Správa psů

Správa poplatků za psy

Titul: Název firmy:

Jméno: Obchodní jméno:

Příjmení: IČO:

Datum narození: Příjmení:

Rodné číslo: Rodné číslo:

Obec: Číslo popisné:

Ulice: Obchodní jméno:

Číslo popisné: IČO:

Jméno psa: Číslo známky:

Rasa: Stáří:

Sazba: Plátce:

Záznam: 1 z 5 Bez filtru

Platba Vratka Upomínka Vymáhání

způsobUhrady	castkaVKč	platba	datumPlatb	datumSplatnosti	variabilniSym	zaplaceno
složenka	300,00 Kč	300,00 Kč	11.4.2014	27.4.2014	1479638529	<input checked="" type="checkbox"/>
*						<input type="checkbox"/>

Záznam: 1 z 1 Bez filtru Vyhledávání

Obrázek 15: Formulář - platba poplatků

Nejprve musí referent osobu vyhledat. Aby byla osoba vyhledána co nejrychleji, lze vyhledávat podle různých kritérií. U fyzické osoby to je podle příjmení, rodného čísla nebo čísla popisného. V případě právnické osoby může referent zvolit buď obchodní jméno firmy nebo její IČO. Po vyhledání musí referent zkontrolovat, jestli jsou osobní údaje plátce správné. Pokud nejsou již aktuální, lze tyto údaje editovat přímo v textových polích.

Protože se jedná o platbu za psy, je do řídicího formuláře (*podle tblOsoby*) vložen podformulář pro správu psů (*tblPes*), který je umístěn vpravo nahoře. Tento podformulář zobrazuje jméno psa, číslo známky, rasu, sazbu a typ poplatníka. Do textového pole sazba zadá referent hodnotu ano nebo ne. Pokud je zadaná hodnota ne, osoba poplatek neplatí (např. invalidita třetího stupně). V tomto případě nastaví databázová aplikace sazbu na 0 Kč a poplatek je považován za zaplacený. Jestliže osoba platí za psa poprvé, podformulář pro správu psů je prázdný a referent musí vyplnit chybějící údaje.

Na kartě platba se eviduje způsob úhrady (složenka, běžný účet, hotově), částka v Kč (shodná se sazbou), platba, datum platby, datum splatnosti a variabilní symbol.

Plátce může částku uhradit najednou nebo se může domluvit na splátkovém kalendáři. To znamená, že celková částka bude rozdělená do několika plateb a každá z těchto plateb bude mít stanovené své datum splatnosti. Pokud je poplatek uhrazen včas a v plné výši, musí referent zaškrtnout pole zaplacené.

Doba splatností a výše sazeb jsou různé u každého typu poplatku. Sazby se určují podle typu poplatku a poplatníka. Liší se také podle obcí, kde jsou stanoveny obecně závaznou vyhláškou. Protože databáze je navržena univerzálně pro všechny obce 1. stupně, nelze přesně stanovit datum splatnosti a výši sazby. Z toho důvodu není ve formuláři u jednotlivých poplatků přednastavena sazba a datum splatnosti.

Vratka poplatku

V případě, že se občan odstěhuje a zaplatil již poplatek za dobu, za kterou službu neužívá, poplatek mu bude vrácen. Databázová aplikace vypočítá vratnou částku a referent zadá datum, kdy byl tento poplatek vrácen. Tyto údaje se evidují na kartě vratka.

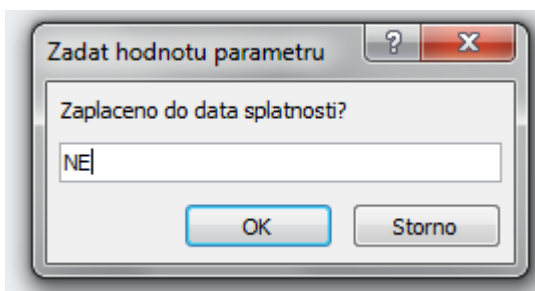
The screenshot shows a web application interface for 'Správa komunálního odpadu' (Municipal Waste Management). The main heading is 'Správa poplatků za komunální odpad'. The form is divided into several sections:

- Personal Information:** Titul (Bc.), Jméno (Hana), Příjmení (Kazmířová), Datum narození (12.3.1988), Rodné číslo (8853129847), Obec (Vítovka), Ulice, Číslo popisné (7).
- Company Information:** Název firmy, Obchodní jméno, IČO.
- Payment and Refund Details:** Sazba (500,00 Kč), Typ poplatníka (trvalý pobyt), Popis případu.
- Refund Selection:** A set of dropdown menus for selecting the refund recipient's details: Příjmení, Číslo popisné (Kazmířová), Rodné číslo, Obchodní jméno, and IČO.
- Navigation and Table:** Buttons for 'Platba', 'Vratka', 'Upomínka', and 'Vymáhání'. Below is a table with columns: datumPlatby, datumSplatnosti, datumVratky, vratnaCastka. The first row shows: 2.1.2014, 31.8.2014, 28.2.2014, 250,00 Kč. A second row is marked with an asterisk (*).
- Footer:** Záznam: 1 z 1, Bez filtru, Vyhledávání.

Obrázek 16: Formulář - vratka poplatku

Vymáhání poplatku

Aby mohl referent zjistit, která osoba dluží, je třeba vygenerovat sestavu s dlužníky. Referent v úvodním formuláři klikne na tlačítko dluhy a databázová aplikace zobrazí formulář s tímto dotazem:



Obrázek 17: Zadání hodnoty parametru

Pokud referent zadá do pole hodnotu ne, zobrazí se mu všechny osoby, které za poplatek nezaplatili včas. Sestava seřadí osoby vzestupně podle data splatnosti.

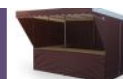
Sestava závazky a platby				
Datum Splatnosti:	Příjmení	Název společnosti:	Typ závazku	Zaplaceno
31.8.2012	Veselý		komunální odpad	<input type="checkbox"/>
31.8.2013	Knichalová		komunální odpad	<input type="checkbox"/>
1.2.2014	Adámek		veřejné prostranství	<input type="checkbox"/>
8.2.2014	Káňová		pes	<input type="checkbox"/>
3.4.2014		Firma a.s.	ubytovací kapacita	<input type="checkbox"/>
9.4.2014		Okna a.s.	vstupné	<input type="checkbox"/>
9.4.2014	Zych		vjezd s motorovým vozidlem	<input type="checkbox"/>
9.4.2014	Kopecká		pes	<input type="checkbox"/>

Obrázek 18: Sestava závazky a platby

Pokud chce referent poplatky vymáhat, vrátí se zpět na úvodní formulář, vybere typ poplatku a vyhledá konkrétní osobu.

Na kartě vymáhání zadá referent datum zahájení vymáhání, exekuční poplatek, typ vymáhání (zaměstnavatel, bankovní ústav, úřad práce) a stav vymáhání. Databázová aplikace vypočítá celkovou dlužnou částku jako součet sazby, penále a exekučního poplatku.

Správa poplatků za veřejné prostranství



Titul:	<input type="text"/>	Název firmy:	<input type="text"/>	Účel:	<input type="text" value="Prodej zboží"/>
Jméno:	Adéla	Obchodní jméno:	<input type="text"/>	Plocha:	<input type="text" value="5"/>
Příjmení:	Bartošová	IČO:	<input type="text"/>	Datum:	<input type="text" value="1.4.2014"/>
Datum narození:	10.4.1990	Příjmení:	<input type="text" value="Bartošová"/>	Doba:	<input type="text" value="3"/>
Rodné číslo:	9084748596	Rodné číslo:	<input type="text"/>	Sazba:	<input type="text" value="300,00 Kč"/>
Obec:	Střítež	Číslo popisné:	<input type="text"/>		
Ulice:	<input type="text"/>	Obchodní jméno:	<input type="text"/>		
Číslo popisné:	78	IČO:	<input type="text"/>		

Platba	Vratka	Upomínka	Vymáhání									
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	datumSplatnosti	penale	castkaVKc2	zahajeniVymahani	exekucniPoplate	typVymahan	stavVy	vymozen	Celkem
				27.4.2013	100,00 Kč	400,00 Kč	30.4.2014	500,00 Kč	na plat	v řízení	<input type="checkbox"/>	1 000,00 Kč
											<input type="checkbox"/>	

Obrázek 19: Vymáhání dluhu

5 Závěr

Tato diplomová práce se zabývá návrhem a implementací databázové aplikace pro správu místních poplatků. Jako cíl jsem si v úvodu stanovila, že se pokusím navrhnout takovou databázovou aplikaci, která by pomohla výběr těchto poplatků racionalizovat. Návrh byl směřován na malé obce, kde takové aplikace jsou nekompletní nebo dokonce zcela chybí. Tento předpoklad se také potvrdil v analýze současného stavu. Téměř žádná obec neměla aplikaci pro správu všech typů místních poplatků.

Aby bylo dosaženo stanoveného cíle, bylo nutno pochopit samotnou správu místních poplatků řídicí se určitými pravidly. Zjednodušeně lze říct, že princip správy místních poplatků spočívá v jejich platbě, vrácení a vymáhání. Na těchto úkolech byla postavená celá diplomová práce.

Analýza uživatelských požadavků byla provedena pomocí strukturovaného rozhovoru, který jsem vedla s jednotlivými referenty správy poplatků. Na základě této analýzy bylo vytvořeno několik diagramů případů užití. Pro pochopení toho, co má systém dělat, byly vytvořeny diagramy interakce a diagram aktivit. Z diagramu tříd potom vycházel samotný relační model.

Samotná implementace databázové aplikace je provedena v MS Access. Byly navrženy konkrétní formuláře a sestavy, které umožňují konečnému uživateli efektivní správu místních poplatků. Hlavní funkcionalita systému je tedy funkční a může být nasazena do testovací verze. Do budoucna může být databázová aplikace doplněna další funkcionalitou, například automatickými výpočty sazeb nebo vratných částek podle různých kritérií.

Návrh databázové aplikace byl proveden, implementace základního modulu správy byla vytvořena. Proto lze cíle považovat za splněné.

Seznam použité literatury

- [1] RIORDAN, Rebecca. *Vytváříme relační databázové aplikace*. Brno: Computer Press, 2000. ISBN 80-7726-360-9.
- [2] CONOLLY, T., C. BEGG and R. HOLOWCZAK. *Mistrovství databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.
- [3] Kaluža, Jindřich a Kalužová, L. *Informatika*. Praha: Ekopress, 2012. ISBN 978-80-86929-83-5.
- [4] Kaluža, Jindřich a Kalužová, L. *Modelování dat v informačních systémech*. Praha: Ekopress, 2012. ISBN: 978-80-86929-81-1.
- [5] ARLOW, Jim a NEUSTADT, Ila. *UML 2 a unifikovaný proces vývoje aplikací: průvodce analýzou a návrhem objektově orientovaného softwaru*. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
- [6] HORDĚJČUK, Vojtěch. *Jazyk SQL*. [online]. [cit. 2014-03-14] Dostupné z: <http://voho.cz/wiki/informatika/jazyk/sql/>.
- [7] NOVÁK, Vítězslav. *Přednášky k předmětu: Diplomový seminář*. [online]. 2014. [cit. 2014-03-14]. Dostupné z: <https://portalekf.wps.vsb.cz/wps/portal/>.
- [8] NOVÁK, Vítězslav. *Přednášky k předmětu: Databázové aplikace*. [online]. 2011. [cit. 2014-03-15]. Dostupné z: <https://portalekf.wps.vsb.cz/wps/portal/>.
- [9] Kapitola 2: Model vztahu mezi entitami. *fi.muni.cz*. [online]. [cit. 2014-03-12]. Dostupné z: <http://www.fi.muni.cz/~xnovak8/teaching/PB154/pb154-cesky-02.pdf>.
- [10] Teorie relačních databází: Normalizace. *manualy.net*. [online]. 2. 8. 2007. [cit. 2014-03-14]. Dostupné z: <http://www.manualy.net/article.php?articleID=13>.

[11] Pojmy integrity – integrita dat. *managed-dedicated-servery.net*. [online]. [cit. 2014-03-12]. Dostupné z: <http://www.managed-dedicated-servery.net/pojmy-integrity.html>.

[12] Datové modely. *wiki.cs.vsb.cz*. [online]. [cit. 2014-03-16]. Dostupné z: http://wiki.cs.vsb.cz/images/c/c6/Er_model07.pdf.

[13] Internetové technologie - klíče a relace. *kubenka.org*. [online]. [cit. 2014-03-16]. Dostupné z: http://www.kubenka.org/PEF/1_rocnik/Internetove_technologie/tema_2_opory_a_soubory/cviceni_2_2_klice_a_relace.pdf.

[14] Microsoft Office 2010. *Stručný přehled vlastností pole*. [online]. [cit. 2014-03-16]. Dostupné z: <http://office.microsoft.com/cs-cz/access-help/strucny-prehled-vlastnosti-pole-HA010231953.aspx#BM0c>.

[15] MINISTR, Jan. *Přednášky k předmětu: Objektově orientované modelování*. [online]. 2012. [cit. 2014-03-18]. Dostupné z: <https://portalekf.wps.vsb.cz/wps/portal/>.

[16] Portál veřejné správy. 565/1990 Sb., o místních poplatcích. *portal.gov.cz*. [online]. [cit. 2014-04-03]. Dostupné z: <http://portal.gov.cz/app/zakony/zakonPar.jsp?idBiblio=38937&nr=565~2F1990&rpp=15#local-content>.

Seznam zkratek

DCL - Data Control Language
DDL - Data Definition Language
DML - Data Manipulation Language
ER – Entity Relationship
FK - Foreign Key
FO – Fyzická osoba
IBM - International Business Machines Corporation
IČO - Identifikační číslo organizace
OOP - Objektivě orientované programování
OZV – Obecně závazná vyhláška
PC – Personal Computer
PK – Primary Key
PO – Právnícká osoba
PSČ – Poštovní směrovací číslo
ROB – Registr obyvatel
RTM - Requirements traceability matrix
SQL - Structured Query Language
SŘBD - Systém Řízení Báze Dat
TCL - Transaction Control Language
UC – Use Case
UML - Unified Modeling Language

Prohlášení o využití výsledků diplomové

Prohlašuji, že

- jsem byl(a) seznámen(a) s tím, že na mou diplomovou (bakalářskou) práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevydělečně, ke své vnitřní potřebě, diplomovou (bakalářskou) práci užít (§ 35 odst. 3);
- souhlasím s tím, že diplomová (bakalářská) práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího diplomové (bakalářské) práce. Souhlasím s tím, že bibliografické údaje o diplomové (bakalářské) práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo, diplomovou (bakalářskou) práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne 25.4.2014

SEZNAM PŘÍLOH

Příloha 1: Formulář pro správu místních poplatků za zhodnocení stavebního pozemku

Příloha 2: Formulář pro správu místních poplatků z ubytovací kapacity

Příloha 3: Formulář pro správu místních poplatků za povolení vjezdu s motorovým vozidlem

Příloha 4: Formulář pro správu místních poplatků za rekreační a lázeňský pobyt

PŘÍLOHY

Příloha 1: Formulář pro správu místních poplatků za zhodnocení stavebního pozemku

The screenshot shows a web application window with the title 'Úvodní formulář' and 'frmZhodnoceniSprava'. The main heading is 'Správa místních poplatků za zhodnocení stavebního pozemku'. The form contains several input fields and dropdown menus for personal and company information. On the right, there is a separate section for parcel details. Below the form is a table with columns for 'IDpozemek', 'typZavazku', 'zpusobUhrady', 'castkaVKČ', 'platba', 'datumPlatby', 'datumSplatno', and 'variabilniSy'. The first row of the table has the value '4' in the 'IDpozemek' column.

IDpozemek	typZavazku	zpusobUhrady	castkaVKČ	platba	datumPlatby	datumSplatno	variabilniSy
* 4							

Příloha 2: Formulář pro správu místních poplatků z ubytovací kapacity

The screenshot shows a web application window with the title 'Úvodní formulář' and 'frmUbytovaciSprava'. The main heading is 'Správa místních poplatků z ubytovací kapacity'. The form contains several input fields and dropdown menus for personal and company information. On the right, there is a separate section for accommodation details. Below the form is a table with columns for 'IDubytovani', 'zpusobUhrady', 'castkaVKČ', 'platba', 'datumPlatby', 'datumSplatno', and 'zaplaceno'. The first row of the table has the value '4' in the 'IDubytovani' column.

IDubytovani	zpusobUhrady	castkaVKČ	platba	datumPlatby	datumSplatno	zaplaceno
* 4						<input type="checkbox"/>

Příloha 3: Formulář pro správu místních poplatků za povolení vjezdu s motorovým vozidlem

The screenshot shows a software application window with the title 'Správa místních poplatků za povolení vjezdu'. The form contains several input fields for personal and company information, including name, address, birth date, and identification numbers. There are also dropdown menus for 'Příjmení', 'ČP', 'RČ', 'Obchodní jméno', and 'IČO'. A right-hand panel contains fields for 'Datum povolení', 'Platnost povolení do', 'Ulice', 'Místo', 'Vozidlo', 'Značka', and 'Sazba', along with a checkbox for 'typPoplatnika'. At the bottom, there is a table with columns: IDvjezd, typZavazku, zpusobUhrady, castkaVKČ, platba, datumPlatby, and datumSp. The table has one row with the value '4' in the first column.

Příloha 4: Formulář pro správu místních poplatků za rekreační a lázeňský pobyt

The screenshot shows a software application window with the title 'Správa poplatků za rekreační a lázeňský pobyt'. The form contains several input fields for personal and company information, including name, address, birth date, and identification numbers. There are also dropdown menus for 'Příjmení', 'ČP', 'RČ', 'Obchodní jméno', and 'IČO'. A right-hand panel contains fields for 'pocetDni', 'uceIPobytu', 'datumZahajeni', 'jmeno', 'prijmeni', 'adresa', 'cisloOP', and 'sazba', along with a checkbox for 'typPoplatnika'. At the bottom, there is a table with columns: IDrekreace, zpusobUhrady, castkaVKČ, platba, datumPlatby, datumSplatno, variabilniSyrr, and z. The table has one row with the value '4' in the first column.