

Univerzita Pardubice
Fakulta ekonomicko-správní

Návrh části informačního systému pro parkování

Daniel Voják

Bakalářská práce

2009

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Daniel VOJÁK**

Studijní program: **B6209 Systémové inženýrství a informatika**

Studijní obor: **Informatika ve veřejné správě**

Název tématu: **Návrh části informačního systému pro parkování**

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

1. Problematika parkování a mobilních přenosů dat
2. Popis použité metodiky
3. Návrh informačního systému pro parkování
4. Popis výsledného návrhu IS

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

- [1]KANISOVÁ, H., MÜLLER, M., UML srozumitelně. 2. aktualiz. vyd. Brno : Computer Press, 2006. 176 s. ISBN 80-251-1083-4.
- [2]KEOGH, J. E., GIANNINI, M., MATĚJŮ, V. OOP bez předchozích znalostí : průvodce pro samouky. 1. vyd. Brno : Computer Press, 2006. 222 s. ISBN 80-251-0973-9
- [3]IParking ,Inteligentní systém parkování [online]. 2007 [cit. 2008-12-12]. Dostupný z WWW: <http://gis.vsb.cz/GIS_Ostrava/GIS_Ova_2007/sbornik/Referaty/Sekce4/GISiParking.pdf >
- [4]RNDR. ILJA KRAVAL. Objektové modelování pomocí UML v praxi [online]. 2005 [cit. 2008-12-12]. Dostupný z WWW: <<http://library.upce.cz/Sources/E1.pdf>>
- [5]Parkovací systémy s naváděním [online]. 2006 [cit. 2008-12-12]. Dostupný z WWW: <<http://www.slavel.cz/dnsptech.htm>>
- [6]Parkovací systémy [online]. 2008 [cit. 2008-12-12]. Dostupný z WWW: <<http://www.azd.cz/produkty/systemy-pro-silnicni-dopravu/produkty/parkovaci-systemy/> >


Vedoucí bakalářské práce:


Ing. Milan Tomeš

Ústav systémového inženýrství a informatiky

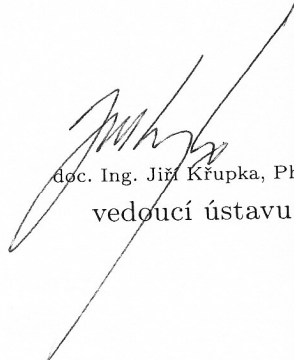
Datum zadání bakalářské práce: **6. října 2008**

Termín odevzdání bakalářské práce: **1. května 2009**


doc. Ing. Renáta Myšková, Ph.D.

děkanka

L.S.


doc. Ing. Jiří Krupka, Ph.D.

vedoucí ústavu

V Pardubicích dne 6. října 2008

Prohlašuji:

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 20. 8. 2009

Daniel Voják

ANOTACE

BAKALÁŘSKÁ PRÁCE JE VĚNOVÁNA NÁVRHU ČÁSTI INFORMAČNÍHO SYSTÉMU PRO PARKOVÁNÍ PROSTŘEDNICTVÍM METODIKY UML. PO NASTÍNĚNÍ SOUČASNÉ SITUACE V OBLASTI PARKOVÁNÍ SE PRÁCE ZAMĚŘUJE NA SEZNÁMENÍ S JEDNOTLIVÝMI ČÁSTMI UML. CÍLEM PRÁCE JE NÁVRH SYSTÉMU A ZNÁZORNĚNÍ ITERATIVNÍHO POSTUPU PŘI JEHO VYTVÁŘENÍ.

KLÍČOVÁ SLOVA

PARKOVACÍ SYSTÉM, UML, DIAGRAMY, PARKOVÁNÍ, CLASS DIAGRAM, USE CASE, SEKVENČNÍ DIAGRAM, NÁVRH SYSTÉMU

TITLE

THE DESIGN OF THE PART OF THE INFORMATION SYSTEM FOR PARKING

ANOTATION

THE BACHELOR THESIS IS DONATED TO THE DESIGN OF THE PART OF THE INFORMATION SYSTEM FOR THE PARKING BY MEANS OF UML METHODOLOGY.

FOR THE ADUMBRATION OF THE CURRENT SITUATION IN THE FIELD OF PARKING THIS THESIS IS CONCENTRATED ON THE IDENTIFICATION WITH THE SINGLE PARTS OF UML.

THE OBJECTIVE OF THE THESIS IS THE DESIGN OF THE SYSTEM AND THE ILLUSTRATION OF THE ITERATIVE PROCEDURE DURING ITS FORMATION.

KEYWORDS

PARKING SYSTEM, UML, DIAGRAMS, PARKING, CLASS DIAGRAM, USE CASE, SEQUENCE DIAGRAM, DRAFT OF THE INFORMATION

OBSAH

1	Úvod	8
2	Problematika parkování	8
3	UML	10
3.1.	Unifikace UML.....	10
3.2.	Struktura jazyka UML	11
3.3.	Relace (relationships)	11
3.3.1	Relace <<include>>	11
3.3.2	Relace <<extend>>.....	12
3.3.1	Agregace	12
3.3.2	Generalizace.....	12
3.3.3	Asociace	13
3.4.	Diagramy	14
3.5.	Modelování případů užití.....	16
3.5.1	Hranice systému.....	17
3.5.2	Aktéři.....	17
3.5.3	Případy užití	17
3.6.	Diagram případu užití	18
3.6.1	Specifikace případu užití	18
3.6.2	Název případu užití	19
3.6.3	ID případu užití.....	19
3.6.4	Stručný popis	19
3.6.5	Aktéři.....	19
3.6.6	Vstupní a výstupní podmínky.....	19
3.6.7	Tok událostí.....	20
3.7.	Class diagram (diagram tříd)	20
3.8.	Sekvenční diagram	20
4	Návrh systému	22
4.1.	Obecný popis systému.....	24
4.2.	První iterace v návrhu systému	25

4.2.1	Use Case	25
4.2.2	Scénáře k případům užití	27
4.2.3	Sekvenční diagramy	36
4.2.4	Class diagram	37
4.3.	Finální iterace v návrhu systému.....	39
4.3.1	Use Case	39
4.3.2	Scénáře případu užití	41
4.3.3	Sekvenční diagram	49
4.3.4	Class diagram	49
4.4.	Možnosti budoucího rozšíření.....	51
5	Závěr.....	52
6	Použitá literatura	53
	SEZNAM OBRÁZKŮ.....	55
	SEZNAM TABULEK.....	56

1 Úvod

Se vzrůstajícím počtem osobních automobilů na českých silnicích přichází problém, kam všechna vozidla zaparkovat. Celkově se po českých silnicích pohybuje 4 409 508 [1] osobních automobilů. Není ani v lidských silách, aby na každé auto připadalo jedno parkovací místo. O to více je důležité, co nejefektivněji využívat nabízená parkovací místa. S tím však bývá většinou ve velkých městech problém. Denně projíždí těmito velkými městy statisíce aut, jejichž řidiči následně hledají místo, kam by své vozidlo zaparkovali. Velmi často bývá problémem samotná sobeckost řidičů, kdy je pro ně důležité, aby zaparkovali oni sami. Na ostatní se tak již neohlížejí a je častým jevem na českých parkovištích vidět auto zaparkované přes dvě místa. Dalším problémem bývá umístění parkovišť, kdy pro řidiče, který nepochází z daného města je někdy takřka nemožné parkoviště najít. Dnešní doba však nabízí velké možnosti v oblasti využití moderních informačních technologií v oblasti parkování.

Vše začíná již při příjezdu do města. Světelné tabule, které by informovaly řidiče o stavu volných a obsazených míst na nejbližším parkovišti se nevyskytují téměř nikde. Řidič by již dopředu věděl, zda se mu vyplatí vážit cestu na parkoviště či má zvolit jiné.

Rozvoj moderních technologií v České republice umožňuje vytvořit velmi efektivní parkovací systém, který by řidiči umožňoval prostřednictvím internetu zjistit, které parkoviště poblíž jeho aktuálního umístění je volné. Řidič by si mohl dokonce i parkovací místo zarezervovat, aby se vyhnul zbytečné cestě.

Tímto problémem se zabývá tato bakalářská práce, jejímž cílem je prostřednictvím UML navrhnout část parkovacího systému, jež bude tvořit základ budoucího informačního systému a usnadní tak samotný proces parkování.

Úvodní kapitola je zaměřena na seznámení se současnou problematikou v parkování. Poté následuje seznámení s použitou metodikou UML, pomocí které bude probíhat samotný návrh parkovacího systému. Nejobjemnější kapitola je věnována cíli této práci, tedy návrhu parkovacího systému. Tato kapitola znázorňuje iterativní přístup při vytváření konceptu systému. Na jejím konci budou probrány možnosti budoucího rozšíření. Tuto práci uzavře závěrečná kapitola, jež bude zaměřena na zhodnocení celé práce.

2 Problematika parkování

Současný stav řešení dopravy v klidu je velmi neutěšený. Dle nejpoužívanějšího internetového vyhledávače Google [2], je možno nalézt pod klíčovým slovem „parkování“

1 700 000 tematických odkazů, které se vztahují k parkování. Z tohoto vysokého čísla vyplývá, že o parkování je nejenom v České republice velký zájem. Řešením této situace z tohoto důvodu zabývají jak ve veřejné správě, tak v oblasti soukromého podnikání.

Hlavní motivací pro neustálé zkvalitňování služeb v oblasti parkování pro veřejnou správu je častá nespokojenost občanů. Města v České republice postupují ve své parkovací politice různorodě. V Hlavním městě Praha začalo budování systému tzv. modrých zón [3], kdy jsou jednotlivé části Prahy rozděleny a ve vybraných úsecích je následně možné parkování pouze se zakoupenou kartou do příslušné zóny. Tento systém je však velmi nedokonalý. Parkovací kartu pro danou zónu si může zakoupit pouze řidič, který v ní má vedeno trvalé bydliště. Ostatní řidiči, musí vyhledat parkovací místa, kde se dá využít platební automat. Těchto míst se však nachází poskrovnu. Nedochází tedy k účinnému vyřešení problému pro řidiče, kteří nemohou zaparkovat své vozidlo.

V dalších městech (Brno, Jihlava, Ostrava) je plánována výstavba parkovacích domů, které by velmi ulevili řidičům, parkujícím především v centru. S výstavbou těchto domů je však spojeno spousta problémů. Mezi největší samozřejmě patří vysoké náklady na samotné vybudování. Dalším problémem je výběr správné lokality, která by umožnila maximalizaci využití parkovacího domu. Velmi často se objeví i problém se schválením stavebního povolení, kdy často bývají proti památkáři. I vzhledem k výše uvedeným problémům probíhá výstavba parkovacích domů velmi pomalu.

Se vzrůstající poptávkou roste i nabídka, a proto se na trhu pohybuje stále více firem, které se snaží řešit problém s parkováním. Zabývají se provozem informačních systémů, které se orientují na parkování.

Jedním z velmi rozšířených je parkovací systém GPP Variant od firmy Green Center [4], který je užíván mimo jiné v systému záchytných parkovišť P + R v Praze, ve Fakultní nemocnici v Motole, v Kongresovém centru v Praze a dokonce je využíván i v zahraničí na letišti v polské Poznani. Výrobce jej dle [4] označuje za nejprodávanější parkovací systém za posledních 10 let v České republice. Dalším velmi rozšířeným parkovacím systémem je CrossPark, který nabízí firma Cross [5], jenž je využíván na parkovištích v městech, jako jsou Zlín, Bratislava, Ostrava a systém je provozován i v městě Monterrey, jenž se nachází v Mexiku.

Třetím parkovacím systémem, jenž lze zařadit mezi nejrozšířenější je univerzální parkovací systém ReadyPark 500 [6] od firmy Nesity. Tento parkovací systém je využíván v Hradci Králové, Pardubicích, Jablonci nad Nisou a spoustě dalších.

Pomoc by mohla být poskytnuta od navrhovaného parkovacího systému, který by prostřednictvím moderních technologií zefektivnil samotný proces parkování.

3 UML

Tato kapitola se zaměřuje na seznámení s UML, jenž bude následně použit pro návrh části informačního systému pro parkování. Tvorba parkovacího systému prostřednictvím UML byla zvolena díky možnosti detailního a maximálně efektivního samotného návrhu.

Jazyk UML (Unified Module Language), unifikovaný modelovací jazyk) je dle [7] univerzální jazyk pro vizuální modelování systémů. Přestože je nejčastěji spojován s modelováním objektově orientovaných softwarových systémů, má mnohem širší využití, což vyplývá z jeho zabudovaných rozšiřovacích mechanismů.

Jazyk UML byl navržen proto, aby spojil nejlepší existující postupy modelovacích technik a softwarového inženýrství. Jako takový je explicitně navržen takovým způsobem, aby jej mohly implementovat všechny nástroje CASE (computer-aided software engineering). Zmíněná koncepce vychází z pochopení skutečnosti, že se rozsáhlé softwarové systémy obvykle bez podpory nástrojů CASE neobejdou. Diagramy vytvořené v jazyku UML jsou srozumitelné pro lidi, ale navíc je mohou snadno interpretovat i programy CASE.

Je nesmírně důležité uvědomit si, že jazyk UML nenabízí žádný druh metodiky modelování. Přirozeně, určité aspekty metodiky můžeme najít v každém z prvků, z nichž se model UML skládá. Samotný jazyk UML však poskytuje pouze vizuální syntaxi, kterou můžeme využít při sestavování svých modelů.

Unified process již ovšem metodikou je. Sděluje nám, jaké pracovníky musíme využít, jaké činnosti vykonat a jaké produkty vyrobit, aby se nám podařilo sestavit model funkčního softwarového systému.

Jazyk UML není vázán na žádnou specifickou metodiku nebo životní cyklus. Lze jej použít společně se všemi existujícími metodami. Unified process využívá jazyk UML jako vlastní syntaxi pro vizuální modelování.

3.1. Unifikace UML

Unifikace UML nevychází pouze z historických měřítek. Jazyk UML se snaží o unifikaci různých domén [7].

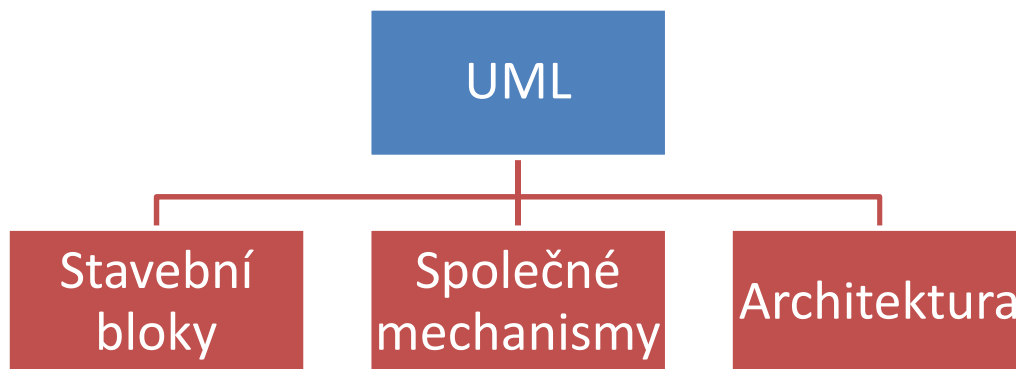
- Vývojový cyklus – jazyk UML nabízí vizuální syntaxi pro modelování během celého vývojového cyklu softwarového projektu
- Aplikační domény – jazyk UML byl vytvořen jako nástroj pro modelování čehokoliv

- Implementační jazyky a platformy – jazyk UML je nezávislý na jakémkoliv programovacím jazyku a na jakékoliv platformě
- Vlastní interní pojmy – jazyk UML se o vnitřní jednotu a konzistenci snaží prostřednictvím malé množiny interních pojmů

3.2. Struktura jazyka UML

Struktura jazyka UML obsahuje dle [7] tyto součásti, viz Obrázek 1

- Stavební bloky – jsou to základní prvky modelu, relace a diagramy
- Společné mechanismy – obecné způsoby, jimiž v jazyku UML lze dosáhnout specifických cílů
- Architektura – pohled v jazyku UML na architekturu navrhovaného systému



Obrázek 1 Struktura jazyka UML, zdroj: autor

3.3. Relace (relationships)

Relace umožňuje ukázat na modelu, jaký je vztah mezi dvěma předměty. Znameníou analogií role, kterou relace hrají v modelech UML, je rodina a vztahy mezi jejími jednotlivými členy.

Relace umožňují zachytit významový vztah mezi předměty. Vztahují se na strukturní abstrakce a seskupování [7].

3.3.1 Relace <<include>>

Dle [7] objevuje se tam, kde existuje podobná nebo stejná část sekvence scénáře opakující se ve více případech užití. V takovém případě není vhodné udržovat více kopií shodných částí scénářů, ale doporučuje se vyčlenit samostatný případ užití obsahující opakující se část scénáře. Jedná se tedy o vyčlenění společného chování ze scénářů základních případů užití. Použitému případu užití je následně jedno, kdo jej používá. Dodává svoje chování do základního případu užití, který pro tento účel musí definovat přesný bod, v němž je řízení

předáno do rozšiřujícího případu užití. Podstatné je, že základní případ užití není bez rozšiřujícího případu užití kompletní.

3.3.2 Relace <<extend>>

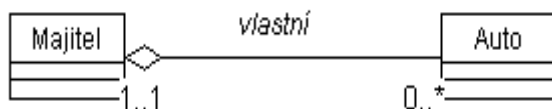
Tento typ relace podle [7] přidává rozšiřující případ užití nové, doplňkové chování do základního případu užití. Podstatným rozdílem oproti relaci typu <<include>> (14) však je, že základní případ užití je sám o sobě zcela soběstačný. Deklaruje pouze tzv. body rozšíření, které ovšem nejsou součástí scénáře, ale pouze ukazují místo ve scénáři, kde může být funkčnost případně rozšířena.

Rozšiřující případ užití tedy přidává chování k základnímu případu užití právě v tomto bodě rozšíření. Vztah <<extend>> modeluje volitelné části případu užití. Případ užití může mít více bodů rozšíření a rozšiřující případ užití může rozšiřovat jeden nebo více těchto bodů rozšíření.

3.3.1 Agregace

Jednou z nejčastěji vyskytujících se vazeb v modelování objektových tříd je agregace. Vazba typu agregace říká [8], že jedna třída je částí druhé třídy, např. motor je částí auta, takže můžeme určit vazbu mezi objektovou třídou Auto a objektovou třídou Motor jako vazbu typu agregace. Obrázek 2 zobrazuje konstrukci agregace.

Speciálním typem agregace je kompozice, kdy je dáno, že podřízený objekt nemůže existovat samostatně bez nadřazeného objektu.

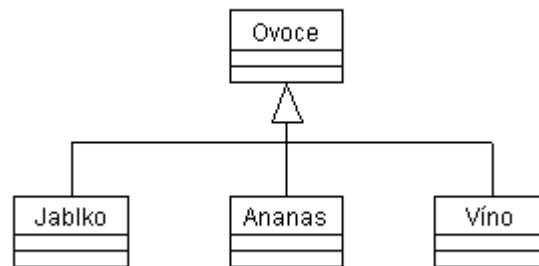


Obrázek 2 Agregace, zdroj: [9]

3.3.2 Generalizace

Velmi důležitým vztahem, který je používán ke statickému pohledu na třídy, je vztah generalizace [8]. Generalizace je klíčový koncept objektivě orientované analýzy a návrhu. Pokud se nachází v analýze tento typ vztahu, bývá implementován v objektovém prostředí jako dědění. Princip dědění umožňuje objektovým třídám sdílet jejich charakteristiky včetně hierarchie dědění a uchovat jejich rozdíly.

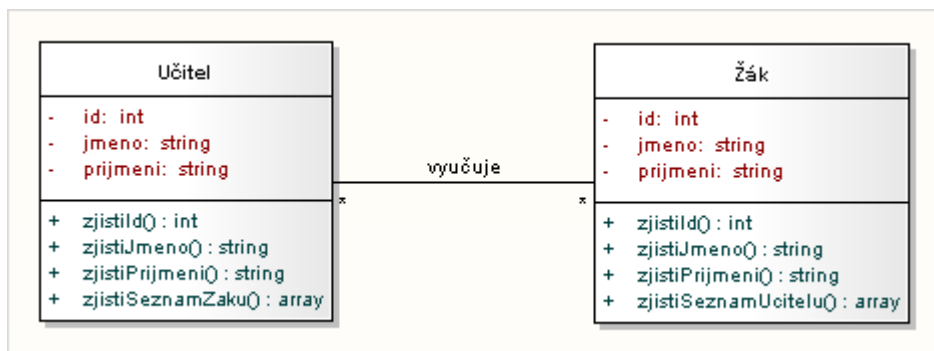
Generalizace se používá především z důvodu opakovaného použití, aby se nemusely opakovat společné prvky. Dědičnost může být i ve více než jedné úrovni. Základní typ generalizace zobrazuje Obrázek 3.



Obrázek 3 Generalizace, zdroj: [10]

3.3.3 Asociace

Dle [7] asociace znárodňuje vztahy mezi jednou či více třídami, které jsou abstrakcí množiny spojení mezi instancemi (objekty) těchto tříd. Jinak řečeno, asociace mezi dvěma třídami říká, že třídy mají mezi sebou „nějaký přímý vztah“. U asociací předpokládáme, že jsou v podstatě obousměrné, pokud nejsou explicitně specifikovány jako jednosměrné, jak zachycuje Obrázek 4. Jméno asociace implikuje směr a mělo by být v aktivní slovesné formě. Jasné asociace však nemusí být pojmenovávány.



Obrázek 4 Asociace, zdroj: [11]

Možné typy asociací znázorňuje Tabulka 1.

Tabulka 1 Možné typy asociací, autor – upraveno na základě: [8]

Typ vazby asociace	Popis vazby
$A\ 1\dots 1\ B$	Instance třídy A má asociaci právě na jednu instanci třídy B
$A\ 1\dots * B$	Instance třídy A má asociaci na žádnou, jednu nebo více instancí třídy B
$A\ 1\dots 0..1\ B$	Instance třídy A má asociaci na nulu nebo jednu in třídy B
$A\ 1\dots 1..* B$	Instance třídy A má asociaci na jednu nebo více instancí třídy B
$A\ 1\dots 1\dots 5\ B$	Instance třídy A má asociaci na jednu až pět instancí třídy B

3.4. Diagramy

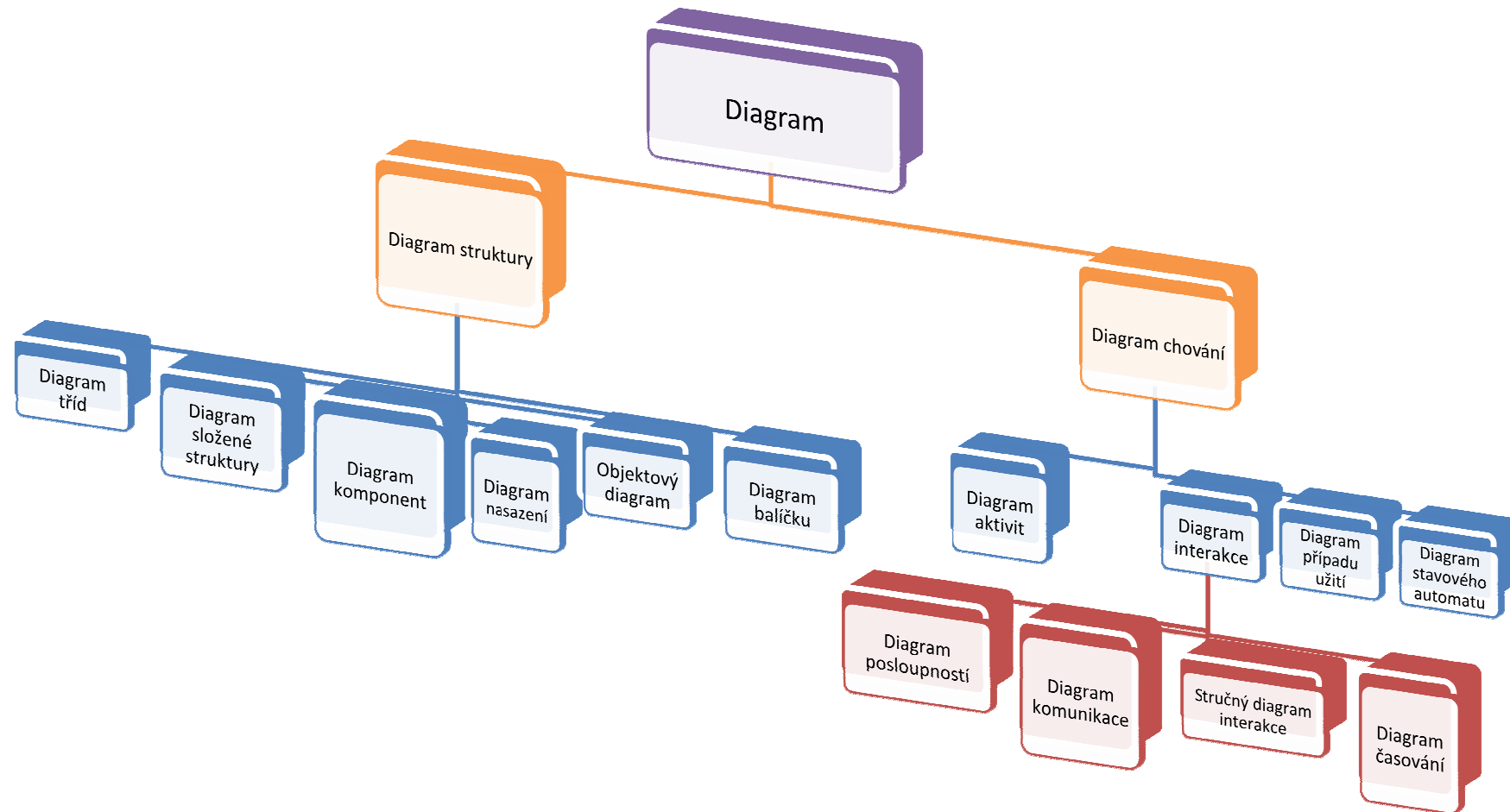
Diagramy jsou podle [7] okna nebo pohledy na model. V tom je veliký rozdíl, protože předměty a relace lze z diagramu odstranit. Lze je dokonce odstranit ze všech diagramů, ale v modelu mohou stále existovat. Ve skutečnosti zůstanou až do té doby, dokud nebudou explicitně vymazány z modelu. Velmi častou chybou začínajících při tvorbě projektu v jazyku UML je odstranění předmětu z diagramů, ale jeho ponechání v modelu.

Celkem existuje čtrnáct různých typů diagramů UML a všechny zachycuje **Chyba! Nenalezen zdroj odkazů.** Každý mnohoúhelník zastupuje jeden typ diagramu.

Množinu diagramů lze rozdělit na ty, které modelují statickou strukturu systému (takzvaný diagram struktury), a na ty, které modelují dynamickou strukturu systému (diagram chování). Diagramy struktury zachycují předměty a strukturní asociace mezi předměty. Diagramy chování naproti tomu zachycují způsob, jímž na sebe jednotlivé předměty navzájem působí, aby bylo dosaženo požadovaného chování softwarového systému.

Neexistuje žádné pevně stanovené pořadí, v němž by měli být diagramy UML vytvářet. Přesto se obvykle začíná diagramem případu užití, který definuje rozsah platnosti navrhovaného systému.

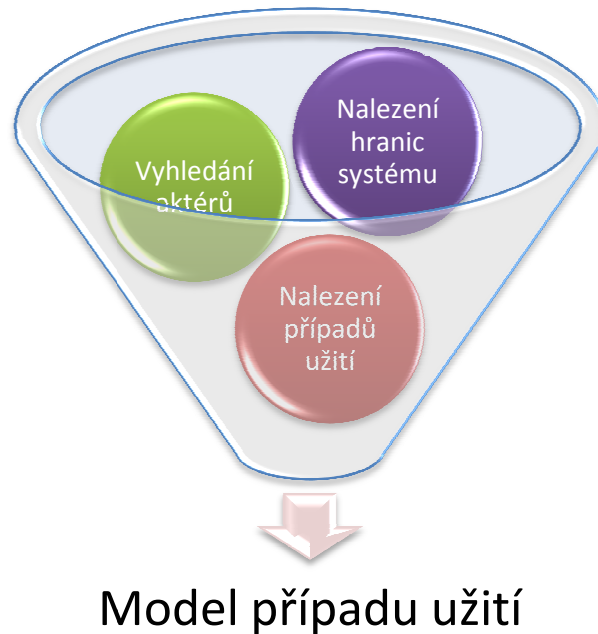
I když existují i další typy diagramů, které znázorňuje **Chyba! Nenalezen zdroj odkazů.**, následující kapitoly se budou zabývat popisem diagramu případu užití, diagramem tříd a sekvenčním diagramem. Jedná se o diagramy, které byly použity v této bakalářské práci, a proto bude pozornost zaměřena pouze na ně.



Obrázek 5 Diagramy UML - zdroj: autor

3.5. Modelování případů užití

Modelování případů užití je dle [7] jednou z forem inženýrství požadavků. Modelování případů užití je jiným, doplňkovým způsobem získávání a dokumentování požadavků a skládá se z následujících aktivit.



Obrázek 6 Model případu užití, zdroj: autor

Výstupem aktivit, jež znázorňuje Obrázek 6, je model případu užití, jenž obsahuje tři komponenty [7].

- Hranice systému – ohraničení zobrazené kolem případů užití, jež je vyznačením území nebo hranic modelovaného systému
- Aktéři – role, přidělené osobám nebo předmětům používajícím daný systém
- Případy užití (Use cases) – činnosti, které mohou aktéři se systémem vykonávat

Modely případů užití jsou prvotním vstupem k modelování tříd.

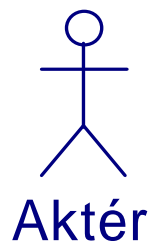
3.5.1 Hranice systému

První věcí při tvorbě softwarového systému, je stanovení jeho hranic. Dle [7] musí být určeno, co je součástí systému a co již není jeho součástí. Hranice systému definuje ten, kdo systém používá (aktéři), a to, co specifikuje přínos systému aktérům.

3.5.2 Aktéři

Aktér specifikuje roli, kterou určitá externí entita přijímá v okamžiku, kdy začíná daný systém bezprostředně používat.

Aktéři jsou v jazyce UML znázorněni tak, jak je zachycuje Obrázek 7.



Obrázek 7 Aktér, zdroj: autor

3.5.3 Případy užití

Případ užití je něco, co aktér od systému očekává. Je to „případ užití systému specifickým aktérem“.

Případy užití jsou vždy iniciovány aktérem, vždy napsány z pohledu aktéra a obvykle jsou považovány za součást systému.

Symbol UML pro případ užití znázorňuje Obrázek 8.



Obrázek 8 Use case, zdroj: autor

Modelování případů užití je iterativní proces a postupuje vpřed postupným upřesňováním. Nejprve se začíná s pouhým názvem případu užití. Později jsou k názvu připojovány další podrobnosti.

Zmiňované podrobnosti se skládají z počátečních krátkých popisů, které jsou nakonec upřesněny do úplné specifikace.

3.6. Diagram případu užití

Celá tato kapitola, včetně všech podkapitol čerpá informace ze zdroje [7]. V diagramu případu užití vyjadřujeme rámečkem hranice systému. Rámeček je hranicí systému, a jak bylo zmíněno v kapitole 3.5.1 aktéři jsou znázorněni vně hranic systému, zatímco případy užití, které utvářejí chování systému, jsou umístěny uvnitř hranic systému.

Vztah mezi aktérem a případem užití je znázorněn pomocí plné čáry, což je v jazyku UML symbol přiřazení.

3.6.1 Specifikace případu užití

Pro specifikace případů užití neexistuje žádný standard UML. Obvykle se však používá šablona, jež znázorňuje Tabulka 2. Existuje sice mnoho komplexnějších šablon, ale obecně se doporučuje při modelování případů užití držet se té nejjednodušší formy.

Je důležité rozhodnout se pro standard, jenž bude konzistentně používán v průběhu celého projektu. V případě, že by takový standard nebyl dodržován či neexistoval, byl by celý proces modelování případů užití zbytečně složitý.

Scénář jednoduché specifikace případů užití obsahuje následující informace:

Tabulka 2 Scénář případů užití, zdroj: autor

Název případu užití
Jedinečný identifikátor případu užití
Stručný popis - zachycuje podstatu případu užití
Aktéři zapojení do případu užití
Vstupní podmínky - musí být splněny ještě před spuštěním případu užití
Hlavní scénář - popisuje jednotlivé kroky případu užití
Výstupní podmínky - musí být splněny na konci případu užití
Alternativní scénáře - seznam alternativ hlavního scénáře

3.6.2 Název případu užití

Jazyk UML nemá pro pojmenování případů užití žádný standard. Případy užití by však měly být pojmenovávány vždy VelbloudímZápisem. Takové sousloví by se mělo skládat ze sloves nebo slovesných frází a každé slovo takového sousloví by mělo začínat velkým písmenem.

Vždy by měla být snaha o nalezení krátkého výstižného názvu. Běžný čtenář by měl snadno získat dobrý přehled o provozních funkcích a o procesu modelovaném pomocí případu užití již podle názvu případu užití.

Název případu užití je jedinečným identifikátorem případu užití v rámci modelu.

3.6.3 ID případu užití

Přestože názvy případů užití musí být v rámci modelu jedinečné, lze je postupem času měnit. Z tohoto důvodu se často vyplatí přiřadit případům užití také jedinečné a neměnné číselné identifikátory.

3.6.4 Stručný popis

Jako stručný popis by měl stačit jeden odstavec, v němž bývá shrnut cíl případu užití. Snahou by mělo být zachycení podstaty případu užití – obchodní přínos pro aktéry.

3.6.5 Aktéři

Z pohledu konkrétního případu užití existují dva typy aktérů:

- Hlavní (primární) aktéři – ve skutečnosti spouštějí případ užití
- Vedlejší aktéři – jsou v interakci s případem užití po jeho spuštění

Všechny případy užití jsou spouštěny vždy jen jedním aktérem. Ovšem stejný případ užití může být spuštěn různými aktéry v různých okamžicích. Všichni aktéři, kteří mohou spustit případ užití, jsou hlavními aktéry. Všichni ostatní jsou aktéry vedlejšími.

3.6.6 Vstupní a výstupní podmínky

Vstupní a výstupní podmínky jsou omezením.

- Vstupní podmínky omezují stav systému předtím, než je možné případ užití spustit. Mohou být vnímány jako omezení, které brání aktérům v aktivaci případu užití, dokud nebudou splněny všechny definované podmínky

- Výstupní podmínky omezují stav systému po skončení případu užití.

Vstupní a výstupní podmínky pomáhají v návrhu správné funkce systému. Měly by být jednoduchými výrazy týkajícími se stavu systému, které budou vyhodnoceny jako pravda nebo nepravda. Zmiňované podmínky jsou rovněž označovány jako booleovské podmínky.

3.6.7 Tok událostí

Jednotlivé kroky případu užití jsou uvedeny v toku událostí. Případ užití lze považovat za rozvětvenou říční deltu. Každý případ užití má pomyslné hlavní koryto a spoustu vedlejších ramen. Menší ramena jsou alternativními toky. V nich lze zachycovat chyby, rozvětvení nebo přerušení hlavního toku. Hlavní tok je také označován za hlavní scénář, vedlejší toky jsou označovány za alternativní scénáře.

3.7. Class diagram (diagram tříd)

Pro návrh objektových systémů se dle [8] předpokládá, že je třeba navrhnout model tříd objektů (class model), který nezobrazuje jednotlivé objekty, ale šablony pro jejich vytvoření. Vztah mezi třídou a objekty znázorňuje Obrázek 4. Třída objektů je definována svými atributy, pouze se určí jejich název a typ. Při vzniku instance objektu (skutečný objekt) se atributům přiřadí skutečné hodnoty.

Diagramy tříd zobrazují statickou stránku systému, především vztahy mezi třídami. Vztahy, které jednotlivé třídy navzájem pojí, jsou asociace, agregace, kompozice a generalizace. Podřízenost jednoho objektu vůči druhému je v analýze chápána dvojím způsobem, buď jako agregace, nebo jako kompozice. V obou případech se však jedná o vztah dvou objektů, z nichž podřízený objekt má svou objektovou referenci vloženu do prvního objektu.

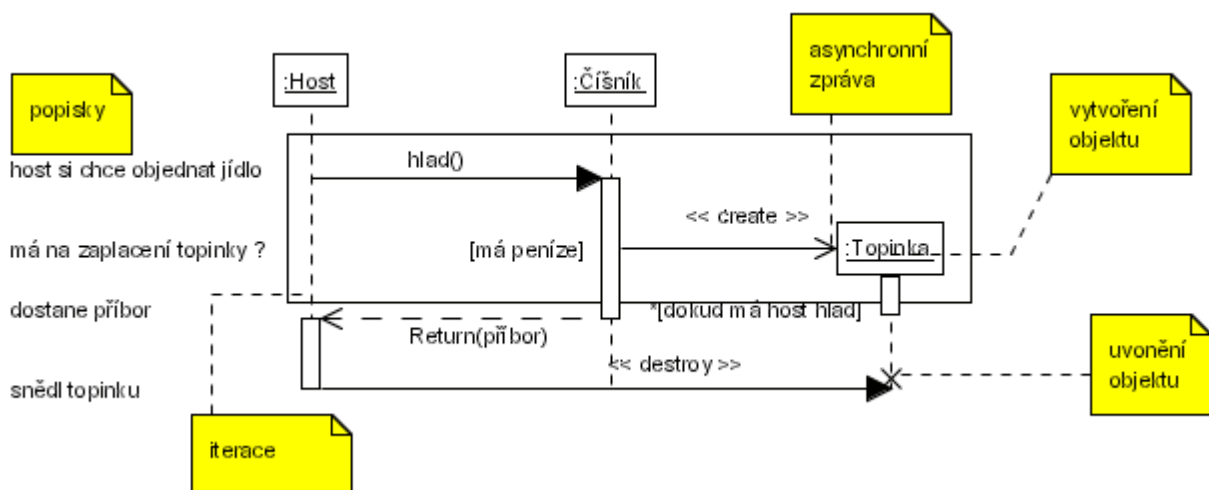
3.8. Sekvenční diagram

Sekvenční diagramy dle [7] znázorňují interakce mezi čarami života jako časově uspořádanou posloupnost událostí. Tyto diagramy jsou nejbohatší a nejpružnější formou diagramů interakce.

Zdůrazňují časově orientovanou posloupnost zpráv předávaných mezi objekty. Uživatelé většinou lépe a rychleji porozumí sekvenčním diagramu než diagramům komunikačním, protože jsou přehlednější. Sekvenční diagram znázorňuje Obrázek 9.

Dle [12] se sekvenční diagramy vytvářejí většinou přímo z diagramů případů užití. K jednomu případu užití může existovat několik sekvenčních diagramů, které modelují interakci objektů v rámci komunikace aktéra se systémem.

Jak je docela zřejmé, sekvenční diagramy jsou zaměřeny výhradně na dynamickou stránku systému. Jsou vhodné pro nalezení jednotlivých objektů a zobrazení komunikace mezi nimi. S tvorbou sekvenčního diagramů se postupně vynořují jednotlivé objekty (resp. kandidáti na objekty), které jsou rovnou zapracovávány do diagramů tříd. Je vhodné zdůraznit, že objekty nalezené v počátečních fázích modelování systému se zřídka v původní podobě uplatní též jako softwarové objekty implementované v cílovém prostředí. Zprvu se jedná spíše o koncepty, pojmy, které se postupem času, zejména ve fázi designu, vyhodnocují a dochází zpravidla k částečné redukci a konsolidaci objektových modelů. Sekvenční diagram obsahuje dvě dimenze. V horizontální rovině se zobrazují jednotlivé objekty, zatímco vertikální rovina představuje tok času. Zprávy posílané mezi jednotlivými identifikovanými objekty mohou být různého druhu. Záleží-li na jejich bližším odlišení, lze klasifikovat zprávy asynchronní, vnořené, zprávy představující návratové hodnoty apod. S rozpracováváním sekvenčního diagramu může poměrně rychle vzrůstat jeho celková komplexnost, což je vcelku spolehlivým příznakem příliš obecného případu užití, ke kterému je sekvenční diagram vytvářen.



Obrázek 9 Vzor sekvenčního diagramu, zdroj: [13]

4 Návrh systému

V této kapitole bude rozebrán návrh systém pro parkování na základě metodiky uvedené v kapitole 3. Nejprve bude definován systém jako celek, poté budou definovány jeho jednotlivé části.

Celý návrh systému probíhal iterativně. Z důvodu velkého množství iterací bude v této práci graficky zobrazena první iterace a poté iterace poslední. Ostatní části systému, které vznikly v průběhu návrhu tohoto systému, budou popsány textovou formou.

Společně se stále se zvyšujícím množstvím vozidel v městských aglomeracích vzrůstá také závažnost řešení problematiky parkování. Je nepřijatelné, aby se touto skutečností nikdo nezabýval a možnosti kam zaparkovat své vozidlo by tak stagnovaly. Parkovací systémy zaznamenaly v posledních letech velký pokrok. Ve velkých městech bývá častým jevem provoz parkoviště plně automaticky bez nutnosti zásahu lidského faktoru.

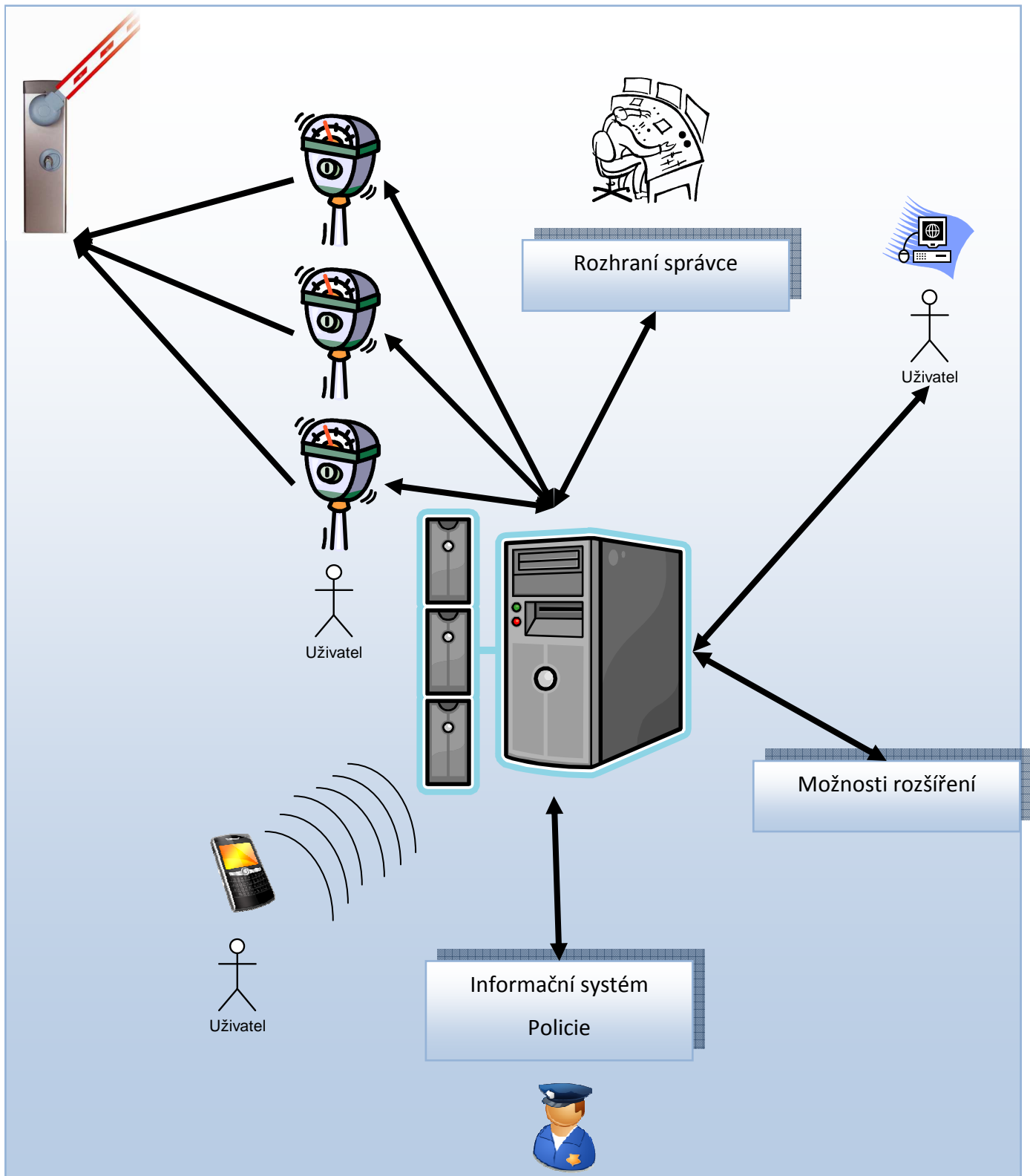
Z důvodu situace, kdy řidič nemůže ani po dlouhých minutách nalézt vhodné místo na zaparkování se nabízí řešení zavedení parkovacího systému, který sníží nároky na životní prostředí, zvýší komfort a také bezpečnosti při parkování. Tyto situace bývají zejména v době tzv. dopravní špičky, kdy jsou veškerá parkoviště zaplněná a na silnicích se začínají tvořit kolony vozidel, které následně brání běžnému provozu.

Navrhovaný parkovací systém je ideální systémové řešení umožňující komplexní realizaci od registrace jednotlivých zákazníků, přes výběr poplatků za parkovné až po implementaci naváděcího parkovacího systému vozidel na parkovací plochy oblasti.

V systému je možnost instalovat neomezený počet vjezdových a výjezdových terminálů, provozovatel tak není limitován v případě zájmu o expanzi.

Parkovací systém umožňuje široké možnosti sestavení podle konkrétních požadavků provozovatele parkoviště. Provozovatel si tak sám zvolí, co je pro něj důležité a co dokáže využít tak, aby maximalizoval své zisky a minimalizoval náklady. Mezi možnostmi volby patří přizpůsobení tarifů, rozsahu nabízených služeb, možnost nastavení rezidentního parkování s následnou kontrolou dlouhodobých abonentů.

Navrhovaný parkovací systém se uplatní ve všech velkých i malých městech bez speciálních požadavků, ale i na místech, kde je vyžadována kontrola pohybu automobilů ve vyhrazeném prostoru, regulace jejich počtu a detailní přehled o provozu celého parkoviště.



Obrázek 10 Návrh systému, zdroj: autor

4.1. Obecný popis systému

Jak znázorňuje Obrázek 10 systém je velmi komplexní a umožňuje přístup různým druhům uživatelů. Samozřejmostí je zde přístup pro správce, který má možnost fyzicky či vzdáleně kontrolovat celý provoz parkoviště a provádět různé úpravy. Parkovací systém umožňuje přístup uživatelům prostřednictvím mobilního telefonu, internetu či přístup přímo na místě.

Příjezdové automaty jsou osazeny barevným dotykovým LCD panelem, který usnadňuje práci se systémem i pro méně zdatné v oblasti moderních technologií. Veškeré ovládání je intuitivní a samozřejmě je zde zahrnuta i nápověda v případě nesnází. Automat, který by mohl být využit pro tento parkovací systém, zachycuje Obrázek 11.



Obrázek 11 Parkovací automat, zdroj: [14]

Na příjezdech na parkoviště jsou nainstalovány moderní závory, které mají lehkou avšak odolnou konstrukci a doba zvednutí se pohybuje v řádech sekund. Jednou z možných závor splňující výše uvedené je závora od firmy CrossPark [5] jenž znázorňuje Obrázek 12.



Obrázek 12 Parkovací závora, zdroj: [15]

Přístup Policie České republiky do parkovacího systému, se ukázal při postupném návrhu jako důležitý faktor. Zvyšuje pocit bezpečí u řidičů a současně pomáhá provozovateli parkoviště s neplatícími řidiči. Důležité pro tuto spolupráci je zajištění kontroly nad parkovištěm bez nutnosti přítomnosti strážníka na parkovišti.

Systém obsahuje velké možnosti pro dodatečné rozšíření. Tímto tématem se zabývá kapitola 4.4.

4.2. První iterace v návrhu systému

V následujících podkapitolách budou znázorněny postupy tvorby prostřednictvím diagramu případu užití, scénářů k případům užití, sekvenčních diagramů a class diagramu, jejichž teoretický postup konstrukce byl vysvětlen v kapitole 3.4. Jedná se o velmi raný návrh systému, který zachycuje úplné počátky při tvorbě. Veškeré diagramy byly vytvořeny v programu Pacestar UML Diagrammer.

4.2.1 Use Case

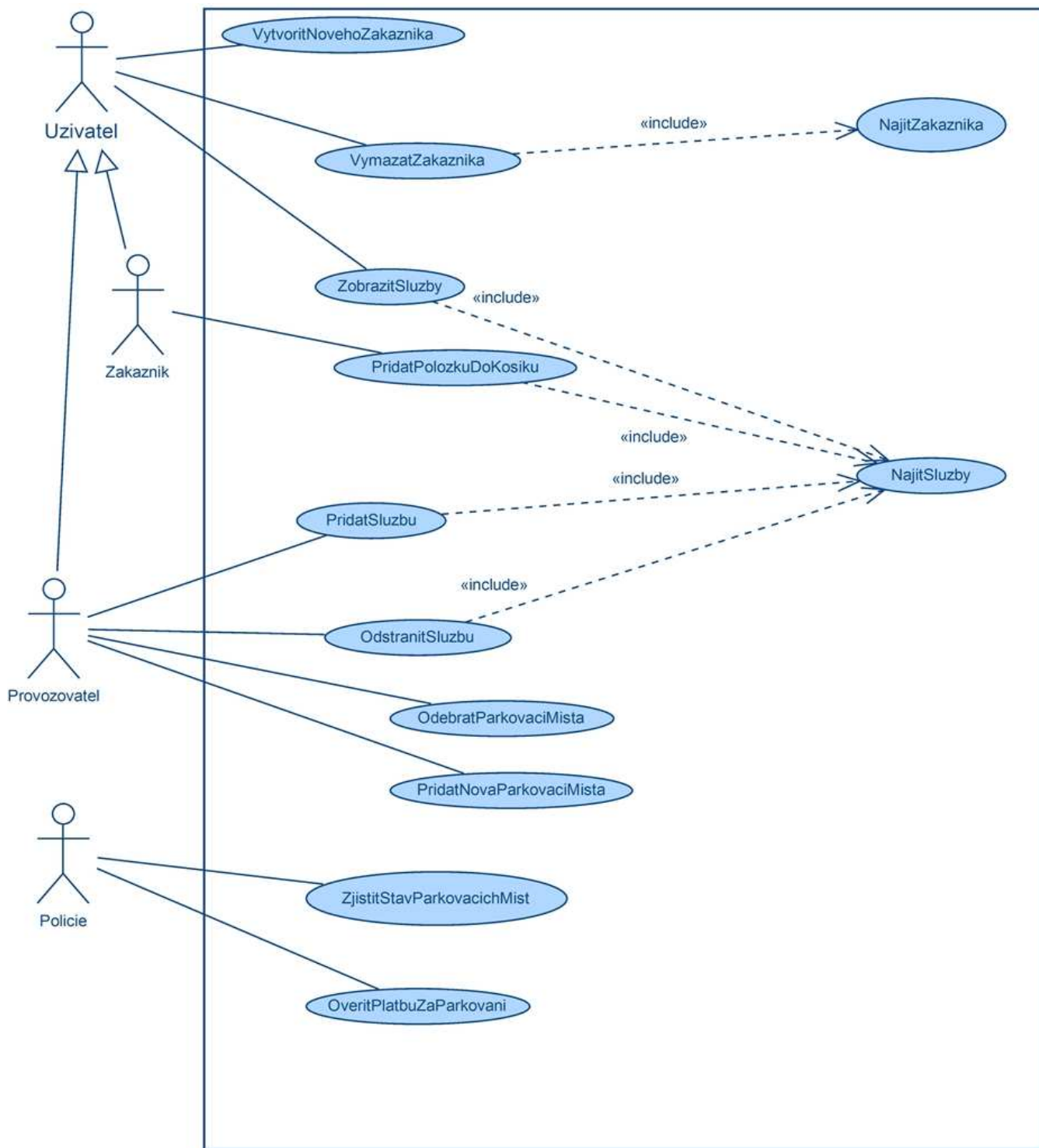
Při návrhu systému se dle [7] doporučuje začít s diagramem případu užití. Z tohoto důvodu i tento systém začal s návrhem prostřednictvím tohoto diagramu.

Zpočátku museli být stanovení aktéři, kteří budou navrhovaný systém využívat. Nejprve byl navrhnout zákazník, jenž by měl systém využívat nejvíce. Dále provozovatel, pro kterého je vlastně daný systém navrhován. Zpočátku bylo ještě uvažováno o aktérovi správce, ale poté došlo ke sloučení do aktéra provozovatele. Z důvodu využívání systému více zákazníky a provozovateli, byla provedena generalizace. Vznikl aktér Uživatel, jehož vlastnosti podědili Provozovatel se Zákazníkem.

Další aktérem byla stanovena Policie, s níž bylo počítáno od začátku plánování.

Jednotlivé případy užití jsou podrobněji popsány prostřednictvím jejich scénářů v kapitole 4.2.2.

Prvotní návrh systému zachycuje Obrázek 13.



Obrázek 13 První iterace – Use Case, zdroj: autor

4.2.2 Scénáře k případům užití

Tato kapitola je věnována popisům jednotlivým případům užití prostřednictvím jejich scénářů. Následuje vždy krátké zdůvodnění, proč byl daný případ užití do systému zahrnut a poté je zobrazen pomocí tabulky samotný scénář k popisovanému případu užití.

Případ užití VytvoritNovéhoZakaznika se zabývá vytvořením uživatelského účtu, který následně zůstane v systému uložen pro opakované přihlášení do systému. Systém nabídne uživateli formulář, který musí být vyplněn se zvýrazněnými poli e-mail, heslo a potvrzení hesla. Po vyplnění formuláře provede systém kontrolu zadaných údajů a v případě, že je vše vyplněno správně, vytvoří nový uživatelský účet. V případě, že uživatel zadá neplatné heslo, začíná alternativní scénář, kdy je uživatel informován o chybně vyplněné e-mailové adrese a následně je zobrazen znovu úvodní formulář. Další možnou chybou může být zadání špatného hesla. Kdy je uživatel požádán o opětovné vyplnění formuláře. Poslední možností startu alternativního scénáře, je po stisknutí tlačítka storno, čímž je přerušena tvorba uživatelského účtu. Z důvodu možného uklepnutí však systém bude vyžadovat od zákazníka potvrzení své volby. V případě, že uživatel potvrdí svoji volbu, končí celý případ užití tím, že systém zobrazí úvodní stránku parkovacího systému. Nedojde-li však k potvrzení tlačítkem „Ano“ případ užití pokračuje v místě přerušení.

Tabulka 3 Příklad užití ID 1, zdroj: autor

Idenifikátor	ID1	
Název	VytvoritNovehoZakaznika	
Popis	Systém vytvoří nového zákazníka	
Vstupní podmínky		
Výstupní podmínky	Byl vytvořen nový účet pro zákazníka	
Uživatelé	Zakaznik, Provozovatel	
Základní posloupnost	Krok	Činnost
	1.	Případ je Uživatelem spuštěn příkazem "Vytvořit nového zákazníka"
	2.	Dokud jsou údaje zákazníka neplatné:
	2.1	Systém žádá uživatele, aby vyplnil Jméno, Příjmení, Adresu, Telefon, Automobil, Typ vozidla, SPZ, Heslo, Znovu potvrzení hesla a E-mail
	2.2	Systém ověří údaje uživatele
	3.	Uživatel potvrdí zadané údaje
	4.	Systém vytvoří nový účet zákazníka a uloží zadané údaje do databáze
Alternativní posloupnost	Krok	Činnost
	2.3	Uživatel chybně vyplnil heslo
	2.4	Systém informuje uživatele, že zadal neplatné heslo a scénář pokračuje bodem 2. 1
	2.3	Uživatel vyplnil chybně e-mailovou adresu
	2.4	Systém informuje uživatele, že zadal neplatnou e-mailovou adresu a scénář pokračuje bodem 2. 1
	1.	Alternativní scénář může začít kdykoli
	2.	Uživatel zruší tvorbu účtu stiskem tlačítka „Storno“
	3.	Systém požádá uživatele o potvrzení této volby tlačítky „Ano“ a „Ne“
	4.	POKUD uživatel potvrdí svoji volbu tlačítkem „Ano“
	5.	Systém zruší proces tvorby nového účtu
6.	JINAK případ užití pokračuje hlavním scénářem	

Z důvodu velkého množství potenciačních zákazníků je jednotlivým uživatelům přidělován jedinečný identifikátor, pod kterým budou v systému vystupovat. V případě užití (Tabulka 4) zadá uživatel svoje ID, na jehož základě poté systém vyhledá údaje o zákazníkovi a zobrazí je.

Tabulka 4 Příklad užití ID 2, zdroj: autor

Idenifikátor	<<include>> ID 2	
Název	NajitZakaznika	
Popis	Uživatel hledá údaje o uživatelském účtu	
Vstupní podmínky	Uživatel je přihlášen v systému	
Výstupní podmínky	Systém našel údaje o zákazníkovi	
Uživatelé	Zakaznik, Provozovatel	
Základní posloupnost	Krok	Činnost
	1.	Uživatel zadá ID zákazníka
	2.	Systém vyhledá na základě ID údaje o zákazníkovi
	3.	Systém zobrazí vyhledané údaje
Alternativní posloupnost	3.1	Systém nenašel hledaného uživatele a vrací se k bodu 1.

V případě, že uživatel zůstane dlouho neaktivní, či je to jeho vlastní žádost, umožňuje systém prostřednictvím případu užití VymazatZakaznika smazat uživatelský účet. I do tohoto případu užití je zahrnuta relace <<include>> NajitZakaznika, pomocí které uživatel nalezne svůj účet. Po stisknutí tlačítka „Odstranit účet“ systém z důvodu bezpečnosti požádá zákazníka o ověření hesla. Pokud dojde k úspěšnému ověření, bude uživatelský účet smazán. V opačném případě je spuštěn alternativní scénář, kdy systém informuje zákazníka o špatně zadaném hesle s žádostí o jeho znovu vyplnění.

Tabulka 5 Případ užití ID 3, zdroj: autor

Identifikátor	ID 3	
Název	VymazatZakaznika	
Popis	Uživatel smaže uživatelský účet	
Vstupní podmínky	Uživatel je přihlášen v systému	
Výstupní podmínky	Uživatelský účet byl smazán	
Uživatelé	Zákazník, Provozovatel	
Základní posloupnost	Krok	Činnost
	1.	<<include>> NajitZakaznika
	2.	Systém zobrazí informace o uživatelském účtu
	3.	Uživatel stiskne tlačítko "Odstranit účet"
	4.	Systém požádá uživatele o potvrzení a o opětovné zadání hesla
	5.	Uživatel zadá heslo a potvrdí příkazem "Ano"
Alternativní posloupnost	6.	Systém ověří heslo a následně smaže zákaznický účet
	Krok	Činnost
	6.1	Systém při ověřování hesla narazil na chybu informuje o tom uživatele
	6.2	Případ užití poté pokračuje bodem 4. hlavního scénáře

Hlavní věcí, která zajímá každého zákazníka, je rozsah a cena nabízených služeb. Právě rozsah nabídky a příznivost ceny rozhodují o tom, že zákazník využije zrovna ten daný parkovací systém. Velmi často směřují první kroky v systému právě do seznamu nabízených služeb. Tento okamžik ošetřuje případ užití ZobrazitSluzby (Tabulka 6).

Případ užití ZobrazitSluzby umožňuje uživateli zobrazit služby, které sám vyhledává, ale i všechny služby, které systém nabízí. Je zde zahrnut <<include>> NajitSluzby (Tabulka 8), pomocí kterého uživatel vyhledává jednotlivé služby.

Tabulka 6 Příklad užití ID 4, zdroj: autor

Identifikátor	ID 4	
Název	ZobrazitSluzby	
Popis	Systém zobrazí nabízené služby	
Vstupní podmínky	Uživatel je přihlášen v systému	
Výstupní podmínky	Systém zobrazil nabízené služby	
Uživatelé	Zákazník, Provozovatel	
Základní posloupnost	Krok	Činnost
	1.	<<include>> NajitSluzby
	2.	Systém zobrazí informace o nabízených službách
	3.	Uživatel má možnost prohlédnout si nabízené služby

Základním kamenem celého systému je uživatelská přívětivost při výběru vybraných služeb. Navrhovaný systém by mohl obsahovat nejmodernější technologie, ale v případě špatné intuitivnosti při ovládání by většinu zákazníků odradil. Systém z tohoto důvodu využívá, dnes již zákazníci zažitou věc na většině internetových obchodů, metodu nákupního košíku, kdy jednotlivé jeho položky jsou samotné služby. Jde sice o princip poněkud zastaralý, ale stále efektivní. Přidávání jednotlivých položek do košíku je zobrazeno v tabulce (Tabulka 7).

Tabulka 7 Příklad užití ID 5, zdroj: autor

Identifikátor	ID 5	
Název	PridatPolozkuDoKosiku	
Popis	Uživatel vloží položky do košíku	
Vstupní podmínky	Zákazník je přihlášen v systému	
Výstupní podmínky	Systém vložil vybrané položky do košíku a zobrazil košík	
Uživatelé	Zákazník	
Základní posloupnost	Krok	Činnost
	1.	<<include>> NajitSluzby
	2.	DOKUD Zakaznik nestiskne "Ukončit výběr"
	2.1	Systém zobrazuje služby, které jsou momentálně k dispozici
	2.2	Zákazník vybere službu, zvolí den, čas a počet den
	2.3	Systém vloží položku do košíku a informuje zprávou "Položka vložena do košíku"
3.	Zákazník ukončí výběr	

Uživateli umožňuje vyhledat požadovanou službu případ užití NajitSluzby. Na základě zadaného textu prohledá systém fulltextovým vyhledáváním celou nabídku, a v případě, že existuje pro daný výraz položka či několik položek zobrazí je. Pokud zadaným kritériím neodpovídá žádná nabízená služba, informuje systém uživatele, že pro zadané neexistuje žádná nabízená služba a zobrazí zákazníkovi úvodní stránku. Veškeré nabízené služby budou v systému uloženy v kompatibilitě SEO¹ aby umožňovali vyhledávání i pomocí internetových vyhledávačů, čímž umožní větší přísun potencionálních zákazníků.

Tabulka 8 Případ užití ID 6, zdroj: autor

Identifikátor	<<include>> ID 6	
Název	NajitSluzby	
Popis	Uživatel hledá požadovanou službu	
Vstupní podmínky	Uživatel je přihlášen v systému	
Výstupní podmínky	Systém našel službu a zobrazil ji uživateli	
Uživatelé	Zákazník, Provozovatel	
Základní posloupnost	Krok	Činnost
	1.	Uživatel zadá podmínky pro vyhledání služby
	2.	Systém hledá na základě uživatelem zadané podmínky v celé databázi
	3.	Najde-li systém odpovídající službu, PAK
	3.1	Systém zobrazí seznam nalezených služeb
3.2	NEBO systém sdělí uživateli, že zadané podmínky neodpovídá žádná nabízená služba	

Žádný systém není dokonalý a není snad ani možné, aby hned zpočátku systém obsahoval veškeré nabízené služby. Dále je nutné, aby byl provozovatel systému schopen reagovat na požadavky klientů, kteří bývají velmi nároční. A neméně důležitá je schopnost reagovat na konkurenční nabídku, čímž se zamezí odlivu současných i potencionálních zákazníků. Případ užití PridatSluzbu, který je znázorněn (Tabulka 9), umožňuje provozovateli přidávat nové nabízené služby, které nabízí řešení výše uvedeným problémům.

¹ **Search Engine Optimization**, *optimalizace pro vyhledávače*) je metodika vytváření a upravování webových stránek takovým způsobem, aby jejich forma a obsah byly vhodné pro automatizované zpracování v internetových vyhledávačích. [17]

Tabulka 9 Příklad užití ID 7, zdroj: autor

Idenifikátor	ID 7	
Název	PridatSluzbu	
Popis	Provozovatel přidá do systému novou službu	
Vstupní podmínky	Provozovatel je přihlášen v systému	
Výstupní podmínky	Systém vložil do systému novou službu	
Uživatelé	Provozovatel	
Základní posloupnost	Krok	Činnost
	1.	Provozovatel vloží do systému novou službu
	2.	Systém službu uloží, přidá ji do seznamu všech služeb a informuje zprávou "Služba uložena"

Z důvodu velkého množství automobilů, jejichž řidiči dennodenně hledají místo, kde by své vozidlo zaparkovali, je pouze otázka času, kdy nastane pro provozovatele otázka, jak uspokojit nově přichozí zákazníky, když kapacita parkoviště je již plně obsazena. Řešení je nasnadě, provozovatel může expandovat a navýšit stávající parkovací kapacitu. Samozřejmě v případě, že není limitován velikostí pozemku, na kterém je parkoviště provozováno. Z důvodu možné expanze je v systému use case PridatNovaParkovaciMista, jež znázorňuje Tabulka 10.

Tabulka 10 Příklad užití ID 8, zdroj: autor

Idenifikátor	ID 8	
Název	PridatNovaParkovaciMista	
Popis	Provozovatel přidá do systému nové parkovací místo	
Vstupní podmínky	Provozovatel je přihlášen v systému	
Výstupní podmínky	Systém vložil do systému nové parkovací místo	
Uživatelé	Provozovatel	
Základní posloupnost	Krok	Činnost
	1.	Provozovatel vloží do systému nová parkovací místa
	2.	Systém parkovací místa uloží a navýší o tato místa celkovou kapacitu

Vzhledem k běžné situaci, že ne každá nabízená služba je úspěšná, nastane občas situace, kdy provozovatel dojde k rozhodnutí, že danou službu již dále nebude svým zákazníkům nabízet a rozhodne o její odstranění ze systému. Use case OdstranitSluzbu (Tabulka 11) ošetřuje tento možný požadavek provozovatele parkovacího systému. Důležitá věc, na kterou musí provozovatel dbát je skutečnost, že při procesu odstraňování služby ze systému může nastat situace, kdy je služba již zarezervována zákazníkem. V tomto případě je nutné zákazníkovi o zrušení služby informovat. Z důvodu užívání moderních technologií v celém parkovacím systému, bude zákazník informován e-mailem o stažení jím rezervované služby z nabídky a zároveň mu bude nabídnuta nejvíce podobná nabízená služba jako náhrada. Poté je již jen pouze na zákazníkovi, zda nabízenou službu využije, rozhodne se pro jinou či půjde ke konkurenci.

Tabulka 11 Příklad užití ID 9, zdroj: autor

Identifikátor	ID 9	
Název	OdstranitSluzbu	
Popis	Provozovatel vymaže službu ze systému	
Vstupní podmínky	Provozovatel je přihlášen v systému	
Výstupní podmínky	Systém vymazal službu ze systému	
Uživatelé	Provozovatel	
Základní posloupnost	Krok	Činnost
	1.	<<include>> Najít_Službu
	2.	Systém zobrazí požadovanou službu
	3.	Provozovatel zvolí možnost "Odstranit službu"
	4.	Systém vyžádá potvrzení k provedení akce
	5.	Provozovatel potvrdí svoji volbu stisknutím tlačítka "Ano"
6.	Systém odstraní službu ze systému a v případě blokáce služby zákazníkem pošle oznamovací e-mail o zrušení služby a nutnosti výběru jiné	

Případ užití OdebratParkovaciMista (Tabulka 12) popisuje odebrání parkovacích míst ze systému. Tuto situaci může přivodit více faktorů. Provozovatel systému se může dostat do situace, kdy na základě provedené analýzy zjistí, že kapacita parkoviště je dlouhodobě nenaplněvaná. Rozhodne tedy o snížení počtu parkovacích míst.

Dalším důvodem, proč odebrat parkovací místo ze systému může být rekonstrukce parkoviště, kdy může být určitá část parkoviště dočasně nedostupná. V tomto případě provozovatel dočasně vyřadí parkovací místa z nabídky a po úspěšně dokončené rekonstrukci je opět navrátí zpět pomocí případu užití, jež zachycuje Tabulka 10.

Tabulka 12 Případ užití ID 10, zdroj: autor

Identifikátor	ID 10	
Název	OdebratParkovaciMista	
Popis	Provozovatel vymaže parkovací místo ze systému	
Vstupní podmínky	Provozovatel je přihlášen v systému	
Výstupní podmínky	Systém vymazal parkovací místo ze systému	
Uživatelé	Provozovatel	
Základní posloupnost	Krok	Činnost
	1.	Systém zobrazí všechna parkovací místa, která nabízí k parkování
	2.	Provozovatel vybere místo, které chce z nabídky odstranit
	3.	Systém požádá provozovatele o potvrzení volby
	4.	Provozovatel potvrdí svoji volbu stisknutím tlačítka "Ano"
5.	Systém odstraní parkovací místo ze systému a v případě blokace parkovacího místa zákazníkem pošle oznamovací e-mail o zrušení parkovacího místa a nutnosti přeoobjednání	

Spolupráce s ať už městskou či státní policií je důležitým faktorem v otázce bezpečnosti. Policie se dennodenně setkává se situací, kdy díky nemodernosti většiny českých parkovišť nemá přehled o naplněnosti jejich kapacity. Z důvodu využití moderních mobilních technologií u navrhovaného parkovacího systému si může Policie pro přihlášení do systému vyžádat stav parkovacích míst, kdy se od systému dočká feedbacku v podobě výpisu stavu volných a obsazených míst. Není tedy nutné, aby policisté byli fyzicky přítomni přímo na parkovišti, ale vše vidí buď na služebních PDA či noteboocích. Tímto šetří svůj čas a mohou se věnovat i jiným povinnostem. Dále přidává možnost v případě plně obsazené kapacity parkoviště reagovat na aktuální situaci na silnicích. Touto situací se zabývá Tabulka 13.

Tabulka 13 Příklad užití ID 11, zdroj: autor

Identifikátor	ID 11	
Název	ZjistitStavParkovacichMist	
Popis	Policie si vyžádá informace o stavu volných míst na parkovišti	
Vstupní podmínky	Policie je přihlášena v systému	
Výstupní podmínky	Systém zobrazil stav volných parkovacích míst	
Uživatelé	Policie	
Základní posloupnost	Krok	Činnost
	1.	Policie si vyžádá stav volných a obsazených míst na parkovišti
	2.	Systém zobrazí aktuální stav parkovacích míst

Dalším důležitým bodem kooperace provozovatele parkovacího systému s policií je otázka placení parkovného a následné jednání s dlužníky. Častým jevem na českých parkovištích je situace, kdy řidič své vozidlo zaparkuje na placené stání, ale zapomene navštívit parkovací automat. I přes značný počet strážníků se nepodaří každé vozidlo pokutovat. Tabulka 14 umožňuje Policii ověřit veškeré platby za zaparkovaná vozidla. Systém ji ulehčí práci s tím, že existuje-li nějaké vozidlo, které nemá zapláceno parkovné, zvýrazní tyto vozidla na přehledném plánu.

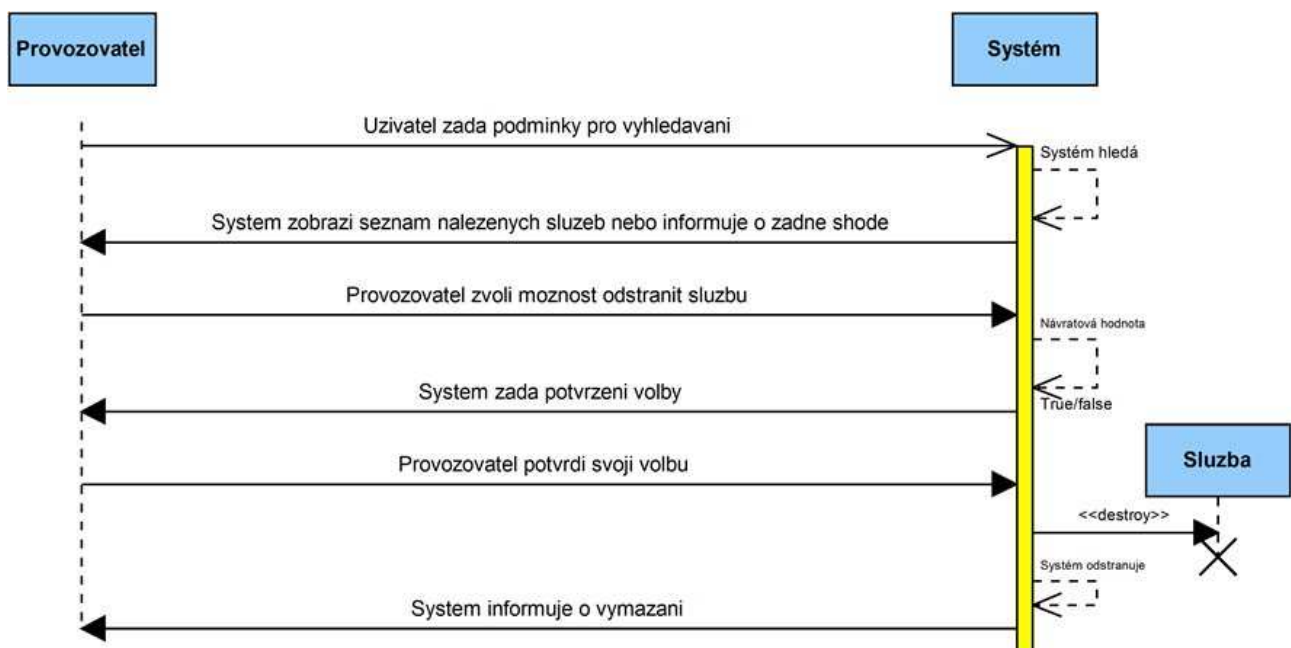
Tabulka 14 Příklad užití ID 12, zdroj: autor

Identifikátor	ID 12	
Název	OveritPlatbuZaParkovani	
Popis	Policie ověří, zda vozidla mají zaplacenou parkovné	
Vstupní podmínky	Policie je přihlášená v systému	
Výstupní podmínky	žádné	
Uživatelé	Policie	
Základní posloupnost	Krok	Činnost
	1.	Policie si vyžádá k jednotlivým zaparkovaným vozidlům informace, do kdy mají zaplacené stání
	2.	System zobrazí jednotlivá parkovací místa a zvýrazní ty, které parkují, ale nemají zaplacenou
	3.	POKUD parkovné není zaplacenou
	3.1	<<exclude>>UlozitPokutu
	4.	JINAK konec případu užití

4.2.3 Sekvenční diagramy

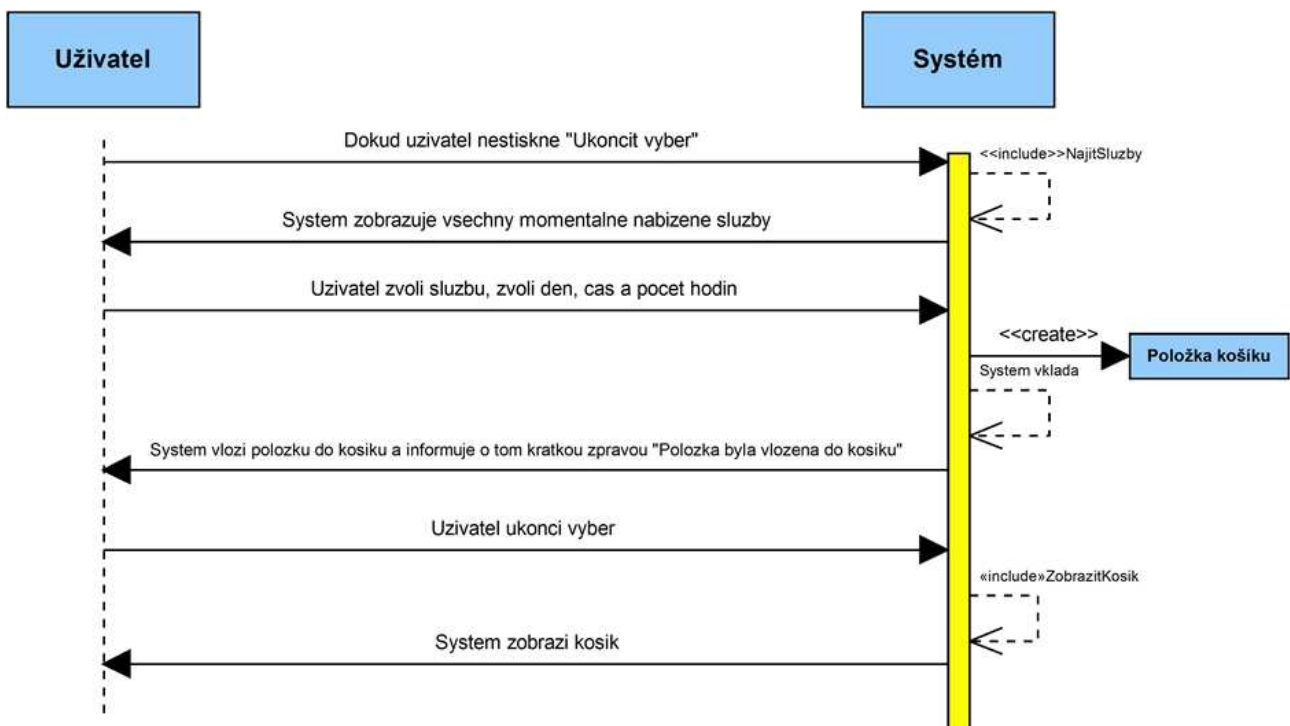
Sekvenční diagramy jsou důležitým faktorem pro návrh diagramu tříd. Z důvodu velké repetitivnosti při tvorbě jednotlivých diagramů, budou v této kapitole znázorněny pouze dva i když při návrhu systému jich vznikalo větší množství.

Sekvenční diagram pro odstranění služby ze systému, jejíž scénář užití zachycuje Tabulka 11, zobrazuje komunikaci mezi provozovatelem a systémem se zachycením časové posloupnosti. Tento sekvenční diagram zachycuje Obrázek 14. Nejprve provozovatel zadá požadovanou službu, jež po nalezení požaduje odstranit. Systém, jenž si vyžádá potvrzení této volby, následně provede odstranění služby ze systému prostřednictvím funkce <<destroy>> a informuje o tom provozovatele.



Obrázek 14 Sekvenční diagram - Odstranit službu, zdroj: autor

Druhý, v této práci uvedený sekvenční diagram, který zobrazuje Obrázek 15, popisuje akci mezi uživatelem a systémem při vkládání položky do košíku. Systém poté, co uživatel zvolí den, čas a počet hodin vytvoří pomocí funkce <<create>> novou položku v košíku a informuje o vložení položky do košíku uživatele. Sekvenční diagram popisuje případ užití Přidat položku, jehož scénář znázorňuje Tabulka 7 a ve finální iteraci Tabulka 17.

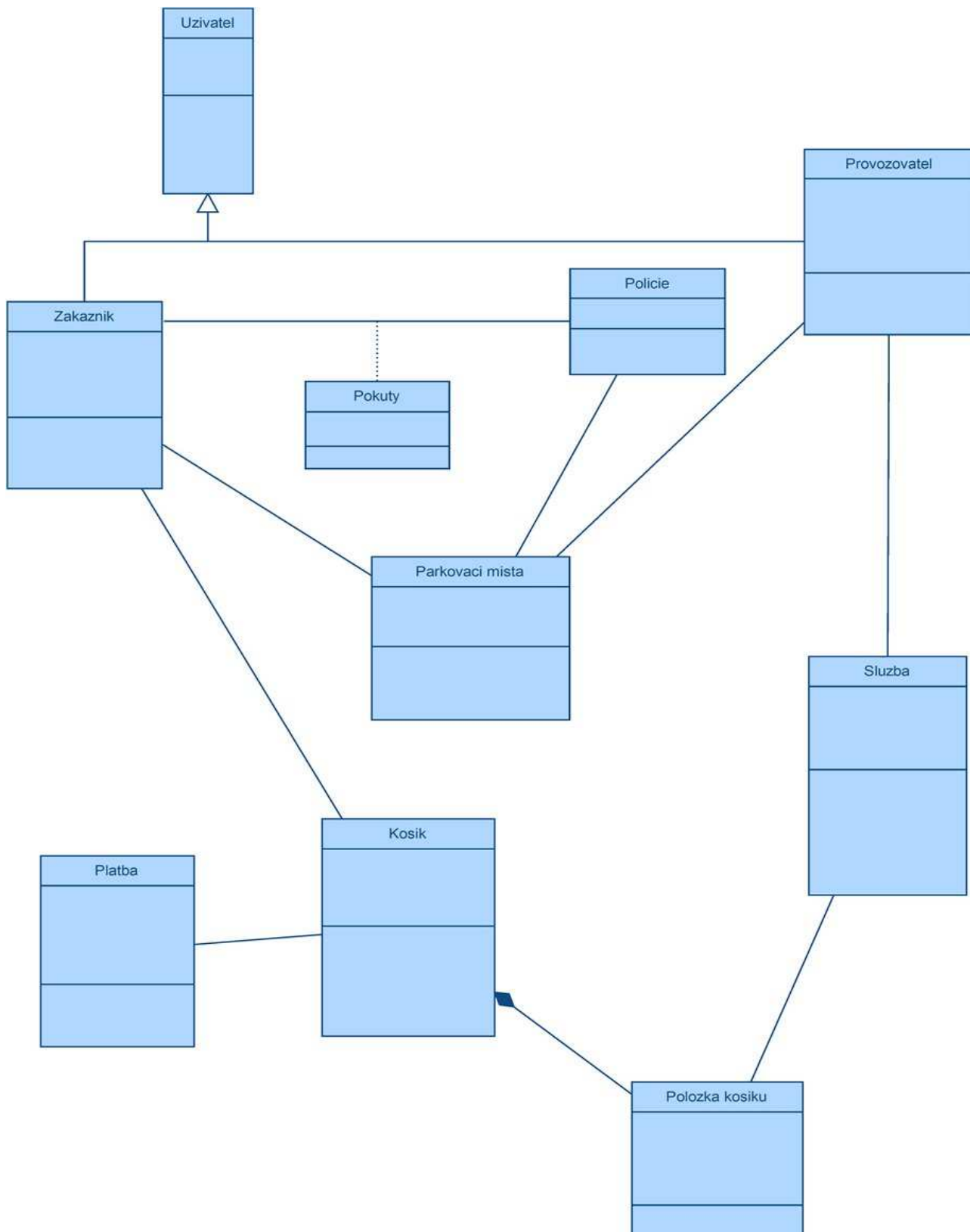


Obrázek 15 Sekvenční diagram - Přidat položku, zdroj: autor

4.2.4 Class diagram

Výstupem z celého návrhu systému je diagram tříd (class diagram). S diagramem tříd pracuje programátor, který na jeho základě vytváří softwarově samotný systém.

Původní class diagram, který zobrazuje Obrázek 16, je pouze hrubý náčrt. Neobsahuje žádné asociace, atributy jednotlivých tříd ani jejich funkce. Byl stvořen jako základ pro představu o celém systému. Základním stavebním kamenem pro jeho vytvoření byly sekvenční diagramy.



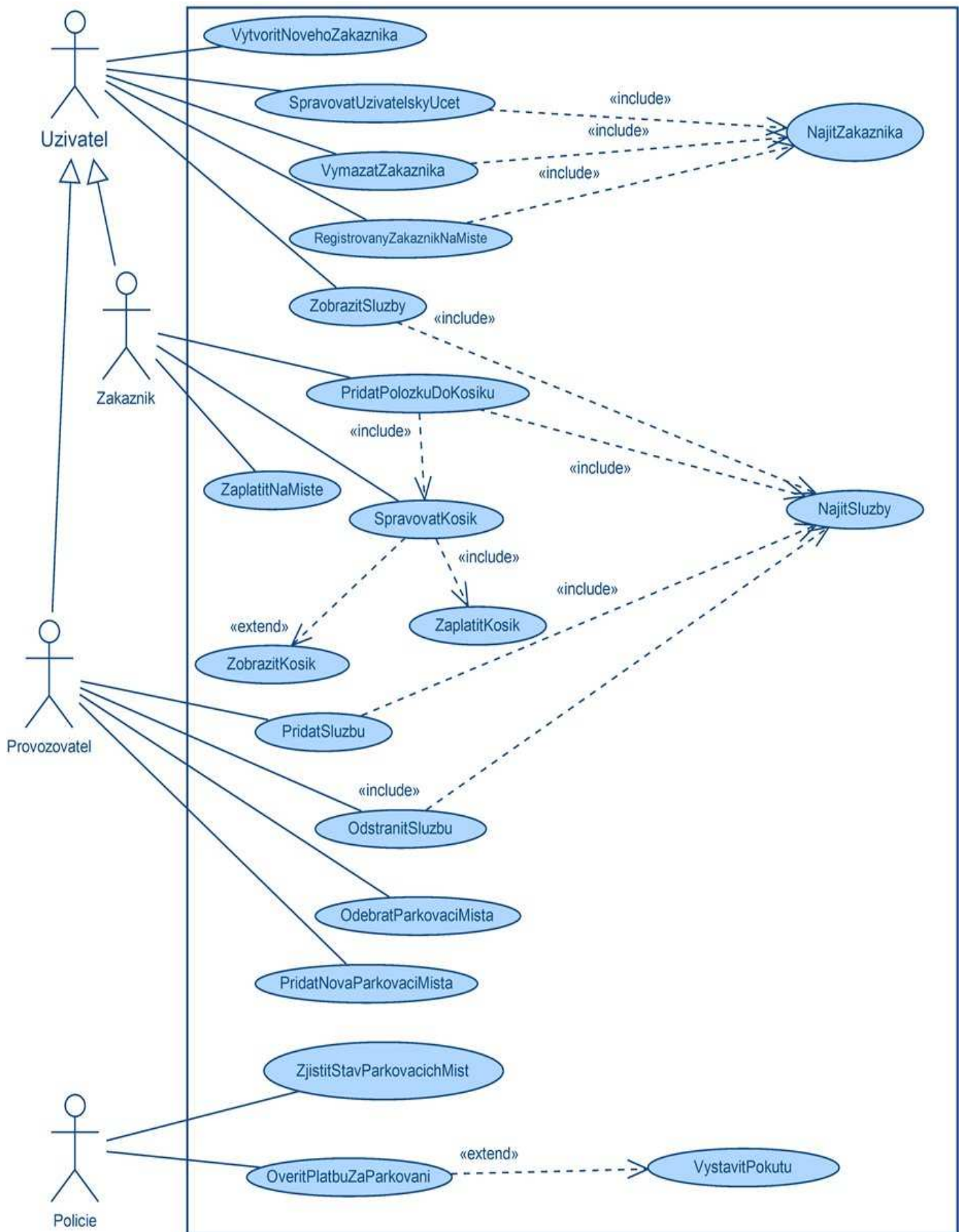
Obrázek 16 První iterace - class diagram, zdroj: autor

4.3. Finální iterace v návrhu systému

V této kapitole bude popisován finální návrh systému, který od úplného začátku, který byl zobrazen v kapitole 4.2 dospěl ke konci. Cesta ke konečnému návrhu vedla přes velký počet postupných kroků, kdy se postupně vyjevovaly další užitečné vlastnosti, které by měl systém umět, a naopak se objevovaly chyby, které bylo nutné opravit.

4.3.1 Use Case

Obrázek 17 zachycuje změny, ke kterým došlo v průběhu návrhu a zobrazuje finální návrh parkovacího systému. Došlo zde k doplnění několika případů užití, úpravě jednotlivých vazeb a vztahů mezi aktéry a celkové změně v principu fungování parkovacího systému, kdy bylo původně zamýšleno využít parkoviště pouze pro registrované zákazníky. S přibývajícím postupem bylo od tohoto řešení odstoupeno a parkovací systém je tak přístupný i pro neregistrované zákazníky. Dále byla přidána možnost uložení pokuty pro Policii, což umožňuje trestat neukázněné řidiče. Byly doplněny možnosti pro činnosti s nákupním košíkem, kdy má uživatel nabízet široké možnosti. Nutným případem užití se ukázala možnost správy uživatelského účtu. Byl to dokonce poslední případ užití, který byl do návrhu systému přidán.



Obrázek 17 Finální iterace - Use Case, zdroj: autor

4.3.2 Scénáře případu užití

V této kapitole jsou zachyceny ostatní scénáře k případům užití, které byly do systému postupným vývojem doplněny.

Z důvodu, že registrovaný zákazník nemá vždy možnost provést registraci s předstihem, byla do systému doplněna možnost provést vše na místě. Aby nebyl limitován nutností předchozí rezervace, umožňuje systém přihlášení do systému přímo na parkovišti prostřednictvím automatu.

Existuje zde tak možnosti pro registrovaného zákazníka, který nemohl provést rezervaci zaparkovat i pokud přijede neplánovaně. Zákazník tak již po přihlášení pouze zvolí parkovací místo, zvolí počet hodin a může parkovat. Platbu následně může provést, až zaparkuje, či dokonce opustí parkoviště pomocí zaslání textové zprávy s vygenerovaným kódem.

Tento případ užití popisuje případ užití RegistrovanýZakaznikNaMiste (Tabulka 15) jenž se ukázal být nezbytným pro spokojenost zákazníka.

Tabulka 15 Případ užití ID 13, zdroj: autor

Identifikátor	ID 13	
Název	RegistrovanýZakaznikNaMiste	
Popis	Registrovaný zákazník přijede bez předchozí rezervace	
Vstupní podmínky	žádné	
Výstupní podmínky	Registrovaný zákazník zaparkoval	
Uživatelé	Zákazník, Provozovatel	
Základní posloupnost	Krok	Činnost
	1.	<<include>>NajitZakaznika
	1.	Zákazník zvolí na automatu možnost "Registrovaný zákazník"
	2.	Systém vyzve zákazníka aby vyplnil své heslo
	3.	Zákazník zadá své heslo
	4.	Systém vyhodnotí zadané údaje jako správné
	5.	Zobrazí zákazníkovi aktuálně volná parkovací místa
	6.	Zákazník si vybere parkovací místo a nastaví počet hodin
	7.	Systém zobrazí kód a telefonní číslo, na které má kód zaslat
	8.	Zákazník potvrdí stiskem tlačítka "OK"
Alternativní posloupnost	9.	Systém zvedne závoru a případ užití končí
	4.1	Systém vyhodnotil zadané údaje jako chybné a vyzve zákazníka k opětovnému zadání
	8.1	Pokud nedojde ke stisknutí tlačítka "OK" je zákazník vyzván k jeho stisknutí. V případě, že není stisknuto tlačítko do 1 minuty, případ užití končí a systém zobrazí úvodní schéma

Samotné vystavení pokuty již je klasickým postupem, který je již zažit po mnoho let, pouze s tím rozdílem, že policie zadává podrobnosti o vystavené pokutě přímo do parkovacího systému a opět nemusí být fyzicky přítomna. Celá tato operace může probíhat pouze elektronicky. Registrovaný zákazník obdrží v případě vystavení pokuty SMS zprávu o nutnosti úhrady této pokuty před odjezdem z parkoviště. Neregistrovaný zákazník musí před opuštěním parkoviště nechat zkontrolovat při příjezdu vystavený lístek v automatu, ve kterém následně může provést úhradu v hotovosti, pomocí platební karty či zasláním zadaného kódu pomocí textové zprávy. Jakmile je pokuta zaplacená, vyplní policie v systému úhradu pokuty u zákazníka a následně zašle registrovanému zákazníkovi potvrzující e-mail a v případě, že se jedná o neregistrovaného zákazníka textovou zprávu s potvrzením. Příklad užití VystavitPokutu jenž řeší výše uvedené, popisuje Tabulka 16.

Tabulka 16 Příklad užití ID 14, zdroj: autor

Identifikátor	<<extend>> ID 14	
Název	VystavitPokutu	
Popis	Policie vloží do systému pokutu a následně ji vybere	
Vstupní podmínky	Policie je přihlášena do systému	
Výstupní podmínky	Policie zaznamenala a vybrala pokutu	
Uživatelé	Policie, Zákazník	
Základní posloupnost	Krok	Činnost
	1.	Policie zadá podrobnosti o pokutě do systému
	2.	Systém uloží pokutu do systému
	3.	Zákazník uhradí pokutu a doplatí parkovné
	4.	Policie přijme platbu pokuty a zadá do systému úhradu pokuty
	5.	Systém vytiskne doklad o zaplacení
	6.	Policie předá doklad o zaplacení zákazníkovi

U případu užití, jež zachycuje Tabulka 17, došlo k přidání <<include>> ZobrazitKosik a k celkovému předělání koncepce s prací s jednotlivými položkami. Jakmile zákazník ukončí výběr požadovaných služeb, systém automatiky zobrazuje nákupní košík. K této změně došlo v průběhu návrhu z důvodu větší komfortnosti pro zákazníka.

Tabulka 17 Případ užití ID 5, zdroj: autor

Identifikátor	ID 5	
Název	PridatPolozkuDoKosiku	
Popis	Uživatel vloží položky do košíku a následně se zobrazí celý nákupní košík	
Vstupní podmínky	Zákazník je přihlášen v systému	
Výstupní podmínky	Systém vložil vybrané položky do košíku a zobrazil košík	
Uživatelé	Zákazník	
Základní posloupnost	Krok	Činnost
	1.	<<include>> NajitSluzby
	2.	DOKUD Zakaznik nestiskne "Ukončit výběr"
	2.1	Systém zobrazuje služby, které jsou momentálně k dispozici
	2.2	Zákazník vybere službu, zvolí den, čas a počet den
	2.3	Systém vloží položku do košíku a informuje zprávou "Položka vložena do košíku"
	3.	Zákazník ukončí výběr a systém zobrazí košík
	4.	<<include>>ZobrazitKosik

Po ukončení výběru služeb, které zákazníka zaujaly, zobrazí systém celý nákupní košík. Zákazník si tak bude moci lehce ověřit, zda systém přidal všechny vybrané služby a následně provést kontrolu. Zákazník tak dostane velmi rychle a přehledně souhrn veškerých vybraných služeb. Případ užití ZobrazitKosik (Tabulka 18) je vazba <<include>> případu užití (Tabulka 17).

Tabulka 18 Případ užití ID 15, zdroj: autor

Identifikátor	<<include>> ID 15	
Název	ZobrazitKosik	
Popis	Systém zobrazí obsah nákupního košíku	
Vstupní podmínky	Zákazník je přihlášen v systému	
Výstupní podmínky	Systém zobrazil informace o košíku	
Uživatelé	Zákazník	
Základní posloupnost	Krok	Činnost
	1.	Zákazník vybere možnost "Zobrazit košík"
	2.	Nejsou-li v košíku žádné položky PAK
	2.1	Systém zobrazí zprávu "Váš košík je prázdný"
	2.2	Případ užití končí
	3.	Pro všechny položky v košíku
3.1	Systém zobrazí informační tabulku, která obsahuje ID služby, počet hodin, den, tarif na hodinu a místo	

Nezbytným prvkem systému je možnost správy košíku. Zákazník musí mít vždy možnost úpravy vybraných služeb, protože nezdá se, že si svůj výběr rozmyslí. Proto systém obsahuje případ užití SpravovatKosik (Tabulka 19), pomocí kterého uživatel dokáže měnit jednotlivé položky košíku. Je zde možnost odstranění přidané položky, protože existuje možnost chyby při výběru a bylo by tedy nesmyslné neumožnit následné odebrání vybrané služby z košíku.

Dále je zde volba změny počtu hodin, protože zákazník mohl chybně nahlédnout do diáře a tak by se mohl stát vlastním nedopatřením subjektem, se kterým by situaci musela řešit Policie, samozřejmě v případě, že by zaplatil menší počet hodin, než by následně využil. V opačném případě snad neexistuje žádný zákazník, který by platil dobrovolně za službu, kterou maximálně nevyužije. Z důvodu snahy o usnadnění ovládání systému zákazníkovi, provádí systém při změnách automatickou kontrolu možnosti její provedení a v případě nemožnosti nabídne systém sám uživateli nejbližší možnou volbu. Analogicky systém umožňuje změnu parkovacího místa a dne.

Veškeré tyto změny mají za snahu co nejvíce zpříjemnit a zjednodušit práci s parkovacím systémem uživateli, kdy i při špatném počátečním výběru jednotlivých služeb následuje snadná možnost nápravy.

Tabulka 19 Příklad užití ID 16, zdroj: autor

Identifikátor	ID 16	
Název	SpravovatKosik	
Popis	Zákazník mění obsah svého košíku	
Vstupní podmínky	Zákazník je přihlášen v systému a má zobrazen nákupní košík	
Výstupní podmínky	Zákazník upravil položky v košíku	
Uživatelé	Zákazník	
Základní posloupnost	Krok	Činnost
	1.	DOKUD zákazník upravuje obsah košíku
	1.1	Zákazník vybere položku z nákupního košíku
	1.2	POKUD zákazník vybere příkaz "Odstranit položku"
	1.2.1	Systém zobrazí žádost o potvrzení volby
	1.2.2	Zákazník potvrdí svoji volbu
	1.2.3	Systém odstraní položku z košíku, přepočítá položky v košíku a zobrazí aktualizovaný košík
	1.3	POKUD zákazník upraví hodnotu u položky počet hodin
	1.3.1	Systém zkontroluje, zda je možnost rozšíření současného výběru o požadovaný nový počet hodin
	1.3.2	POKUD je možnost změny počtu hodin systém tuto změnu uloží a informuje o tom uživatele
	1.3.3	JINAK zobrazí chybovou hlášku o blokaci a nabídne zákazníkovi jiné volné parkovací místo
	1.4	POKUD zákazník zvolí nové parkovací místo
	1.4.1	Systém ověří možnost rezervace, a pokud není obsazeno, provede změnu, o které informuje zákazníka
	1.5	POKUD zákazník vybere jiný den
	1.5.1	Systém provede kontrolu obsazení parkovacího místa v daný den a v případě obsazení nabídne jiné parkovací místo se žádostí o potvrzení změny
	1.5.2	POKUD zákazník potvrdí změnu parkovacího místa
	1.5.2.1	Systém provede změnu a informuje zákazníka
1.5.2.2	JINAK systém oznámí zákazníkovi o nemožnosti provedení změny a případ užití končí	

Jakmile dojde k ukončení správy košíku, již nic nebrání jeho zaplacení. Příklad užití zabývající se placením za košík je věnována Tabulka 20. Z důvodu spolupráce provozovatele parkovacího systému s bankami a mobilními operátory, je zde možnost platby pomocí SMS a prostřednictvím platební karty.

Při platbě pomocí zaslání krátké textové zprávy systém zákazníkovi vygeneruje unikátní kód, pomocí kterého následně správně přiřadí platbu. Současně s kódem dojde k zobrazení telefonního čísla, na které musí být daný kód zaslán. Cena zaslání SMS zprávy je dle tarifu mobilního operátora, není zde tedy cena nesmyslně nadsazená, jak bývá zvykem u jiných systémů.

V případě, že se zákazník rozhodne využít možnost platby pomocí platební karty, spustí systém samostatný formulář, kde celý postup probíhá. Od zvolení typu platební karty přes zadání jejího čísla až po ověřovací kód CVV2². Samotný CVV2 kód je požadavkem pro platby kartou na internetu od samotných kartových asociací VISA a MasterCard.

Tabulka 20 Příklad užití ID 17, zdroj: autor

Identifikátor	ID 17	
Název	ZaplatitKosik	
Popis	Zákazník zaplatí košík	
Vstupní podmínky	Zákazník je přihlášen v systému	
Výstupní podmínky	Zákazník provedl platbu	
Uživatelé	Zákazník	
Základní posloupnost	Krok	Činnost
	1.	Zákazník ukončí výběr dalších položek a přejde k pokladně
	2.	Systém zobrazí všechny vybrané položky, celkovou částku a nabídne způsob výběru placení
	3.	POKUD zákazník vybere možnost platby přes SMS
	3.1.	PAK systém zobrazí kód a telefonní číslo, na které zákazník zašle tento kód. Po obdržení platby zašle zákazníkovi potvrzující sms a e-mail
	4.	POKUD zákazník vybere možnost platby pomocí platební karty
	4.1	Systém zobrazí formulář, kde je vyplněna celková částka a požádá zákazníka o výběr typu platební karty, číslo platební karty, ověřovací kód CVV2 a dobu vypršení platnosti karty
	4.2	Zákazník vyplní požadované údaje a potvrdí svoji platbu
	4.3	Systém ověří zadané údaje a informuje zákazníka o proběhnutí platby
	4.4.	Systém zašle potvrzovací e-mail
Alternativní posloupnost	Krok	Činnost
	4.4	Alternativní scénář začíná krokem 4.3 hlavního scénáře
	4.5	Systém při ověřování zadaných údajů našel chybu a nemohl autorizovat platbu
	4.6	Systém zobrazí znovu formulář se zvýrazněnými údaji, u kterých přišel na chybu
	4.7	POKUD uživatel zadá nové údaje, případ užití pokračuje krokem 4.3 hlavního scénáře
	5.	JINAK případ užití končí

² CVV2 kód je jedním z bezpečnostních prvků, který se u platebních karet používá. Pro internetové platby to má ten význam, že proto, aby někdo znal CVV2 kód, tak musí mít fyzicky platební kartu k dispozici, jelikož CVV2 kód je na zadní straně platební karty (na podpisovém proužku) [16].

Pro obě možnosti plateb systém po úspěšně provedené platbě zašle potvrzovací e-mail a následně i SMS zprávu, které obsahují kód, jenž bude zákazníkem zadán při příjezdu na parkoviště.

Parkovací systém byl původně navržen pouze pro registrované zákazníky, kteří by si dopředu místo rezervovali. Poté však došlo k přehodnocení celé situace, z důvodu možné ztráty potenciálních zákazníků, kteří by o nutné registraci nevěděli a byli by tím vším značně znevýhodněni.

Systém tedy umožňuje pomocí případu užití ZaplatitNaMiste (Tabulka 21) provést výběr parkovacího místa a následně platbu přímo při příjezdu na parkoviště. Celá tato činnost probíhá prostřednictvím automatu, kdy zákazník postupuje dle pokynů systému a je ukončeno zasláním kódu na vyplněné mobilní telefonní číslo. Z důvodu uložení dat je zaručena platba od zákazníka.

Tabulka 21 Příklad užití ID 18, zdroj: autor

Identifikátor	ID 18	
Název	ZaplatitNaMiste	
Popis	Na parkoviště přijede neregistrovaný zákazník a chce parkovat	
Vstupní podmínky	žádné	
Výstupní podmínky	Neregistrovaný zákazník zaparkoval	
Uživatelé	Zákazník	
Základní posloupnost	Krok	Činnost
	1.	Zákazník přijede na parkoviště, kde na automatu zvolí možnost zaparkovat bez předchozí rezervace
	2.	Systém následně provede kontrolu, zda je na parkovišti ještě volné parkovací místo a poté vybídne zákazníka k volbě hodin, po které bude využívat parkoviště
	3.	Zákazník zvolí počet hodin
	4.	Systém na základě zadaného počtu hodin vybere pro zákazníka nejvhodnější místo na zaparkování, zákazníkovi ho vyznačí na interaktivní mapce a vyzve zákazníka k vyplnění svého jména a telefonního čísla.
	5.	Zákazník vyplní své jméno a telefonní číslo
	6.	Systém poté uloží vyplněné údaje a zašle zákazníkovi na uvedené telefonní číslo kód s číslem, na které jej má zaslat
	7.	Pokud zákazník zvolí možnost zobrazit znovu parkovací místo
	7.1	Systém znovu vyznačí místo na interaktivní mapě
8.	JINAK případ užití končí	

Po úspěšně ukončeném procesu tvorby uživatelského účtu může u uživatele nastat zájem o změnu údajů, které vyplnil při registraci ať už z důvodu koupě nového auta, či změně hesla. Z tohoto důvodu byl do systému doplněn případ užití SpravovatUzivatelskyUcet (Tabulka 22), který umožňuje provést změnu jakéhokoliv vyplněného údaje.

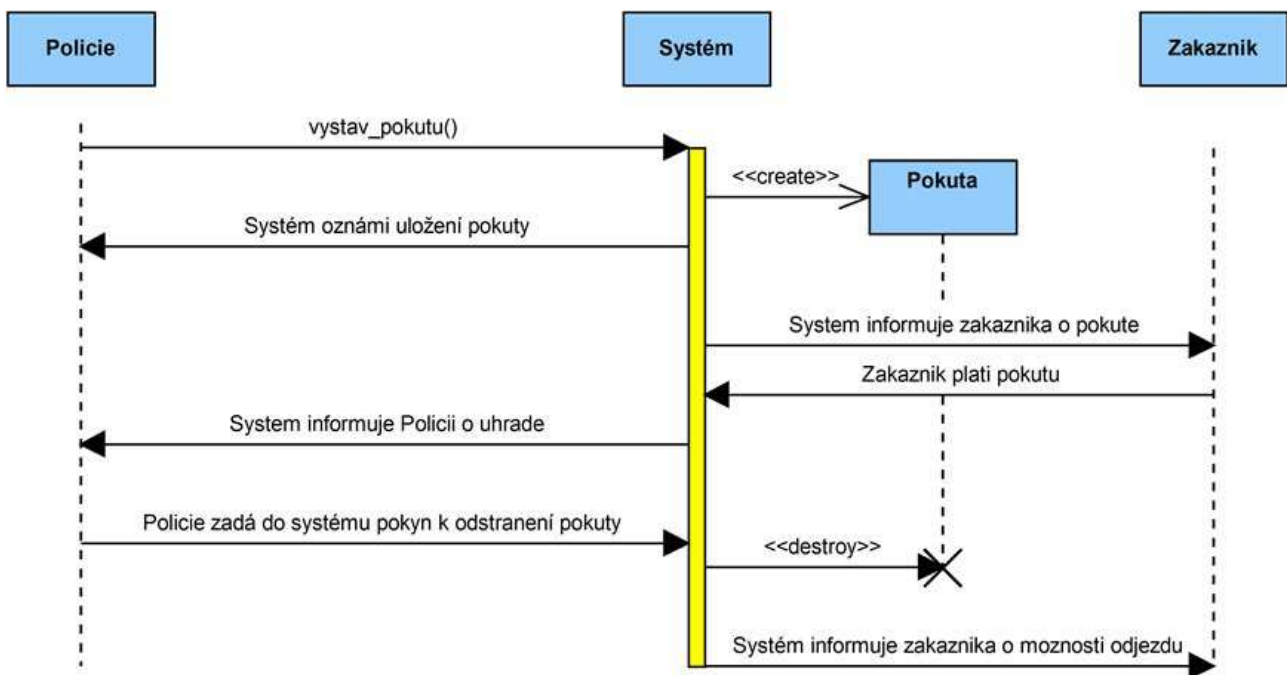
Dále zde je možnost nahlédnout ke všem provedeným objednávkám, ať již pro kontrolu, či z důvodu zapomnětlivosti zákazníka.

Tabulka 22 Příklad užití ID 19, zdroj: autor

Identifikátor	ID 19	
Název	SpravovatUzivatelskyUcet	
Popis	Uživatel upravuje údaje v uživatelském účtu a může nahlédnout uzavřených objednávek	
Vstupní podmínky	Uživatel je přihlášen v systému	
Výstupní podmínky	Údaje v uživatelském účtu byly změněny nebo byly zobrazeny objednávky	
Uživatelé	Zákazník, Provozovatel	
Základní posloupnost	Krok	Činnost
	1.	<<include>> Najít_Zákazníka
	2.	POKUD Uživatel zvolí možnost "Uzavřené objednávky"
	2.1	System zobrazí v přehledné tabulce všechny jeho objednávky
	3.	POKUD Uživatel zvolí možnost "Aktualizovat uživatele"
	3.1	System zobrazí údaje o uživatelském účtu
	3.2	Uživatel provede změnu u vybrané položky
	3.3	System provede kontrolu, zda jsou nově zadané údaje správné
	4.	POKUD jsou nově zadané údaje správné
	4.1	System zadané změny uloží a následně informuje zákazníka
Alternativní posloupnost	5.	JINAK případ užití končí
	3.4	System při kontrole zjistil chybu a zvýraznil chybně vyplněné položky
	3.5	Uživatel znovu vyplnil měněné údaje a scénář pokračuje 3.3

4.3.3 Sekvenční diagram

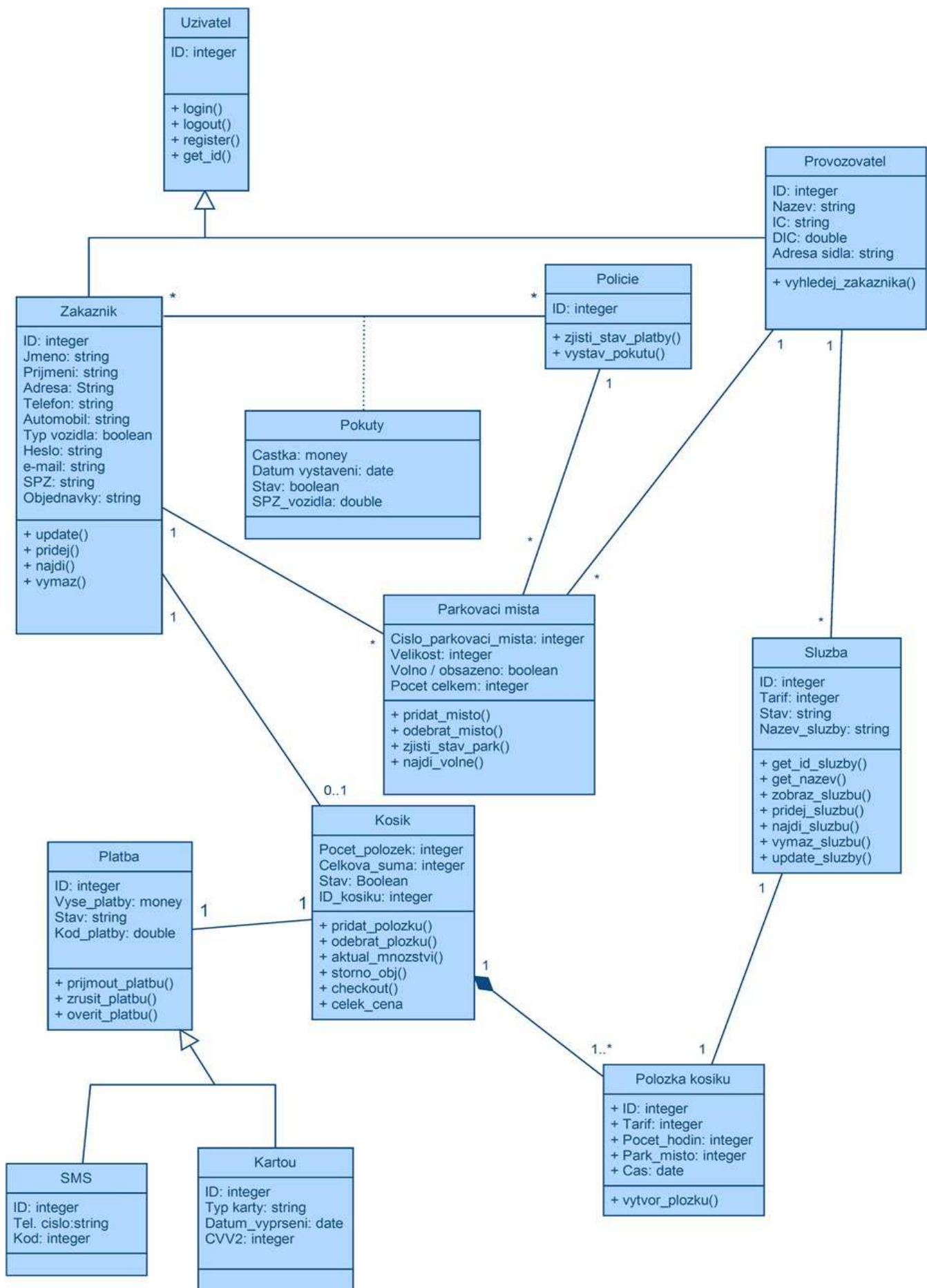
Jak bylo zmíněno v kapitole 4.2.3, scénáře případu užití se často opakují a z toho důvodu bude v této podkapitole zachycen pouze jeden sekvenční diagram, který se od ostatních liší. Tento sekvenční diagram zachycuje komunikaci mezi třemi aktéry. Mezi policií, systémem a zákazníkem. Do tohoto diagramu ještě vstupuje třída pokuty, které ukládá policie jednotlivým zákazníkům. Tato třída je pro daného zákazníka vytvořena a po přijaté platbě je následně uvolněna. Sekvenční diagram vystavení pokuty zachycuje Obrázek 18.



Obrázek 18 Sekvenční diagram – Vystavit pokutu, zdroj: autor

4.3.4 Class diagram

Konečným výstupem pro návrh systému je diagram tříd a jemu věnován Obrázek 19. Tento diagram již obsahuje veškeré asociační vazby mezi jednotlivými třídami. Třídy obsahují atributy a funkce, jež byly definovány především pomocí scénářů případů užití a sekvenčních diagramů.



Obrázek 19 Finální iterace - class diagram, zdroj: autor

4.4. Možnosti budoucího rozšíření

Vzhledem k situaci, že se tato bakalářská práce zabývá návrhem pouze části informačního systému pro parkování, zaměří se tato kapitola na možné budoucí rozšíření navrhovaného systému.

Systém by prováděl automatickou kontrolu integrity dat s ukládáním vzniklých chyb, které by následně usnadnili práci technikům a minimalizovali by tak čas, po který by parkoviště bylo mimo provoz. Další možností rozšíření je zavedení systému do navigačních přístrojů, kde by prostřednictvím bodů zájmů umožňovali navigaci přímo na parkoviště, či prostřednictvím dopravních informací, které i v České republice, začínají nacházet své uplatnění. Možným rozšířením jsou i navigační tabule, které při příjezdu do města informují o stavu obsazených míst na vybraném parkovišti. Zajímavým rozšířením z hlediska bezpečnosti je rozpoznávání SPZ parkujících vozidel, které by následně byly uloženy v systému. Své uplatnění by určitě našly různé předplacené čipové karty, které by vlastnili dlouhodobě parkující abonenti a ještě více tak usnadnili zákazníkům jejich parkování. Využití by se určitě našlo pro hlasový modul, který by sloužil pro komunikaci s parkovací obsluhou. Ať už přítomnou na parkovišti, či vzdálenou v centrále společnosti provozující parkovací systém. Velmi častým a chytrým prvkem současných parkovišť jsou navigační směrové šipky uvnitř parkoviště. Velmi efektním a efektivním avšak ne až tak důležitým budoucím rozšířením by mohla být také světelná signalizace nad jednotlivými volnými místy, kdyby tak příjíždějící řidič rovnou věděl, kam má zamířit. Tím by tak odpadlo většinou zdlouhavé hledání volného parkovacího místa a urychlilo by celkový proces parkování.

Možností budoucího rozšíření je mnoho. Výše popsané uvádí pouze nejzákladnější možné rozšíření. Při bližším zkoumání a především po zavedení systému do běžného provozu by se samozřejmě vyskytly další možnosti, jak navrhovaný systém obohatit a ještě více tak naplnit různé zákaznické tužby.

5 Závěr

Cílem této bakalářské práce byl návrh části informačního systému pro parkování. Nejprve došlo k seznámení se současnou problematikou parkování, kdy byl zmíněn současný neutěšený stav řešení dopravy v klidu s následným seznámením se současnými parkovacími systémy. Problematika parkování se ukázala jako řešitelná pomocí UML a informačních systémů.

Následuje seznámení s UML, kdy jsou postupně popisovány principy tvorby, až po jednotlivé diagramy, které jsou v práci použity.

Poté je přistoupeno k samotnému návrhu parkovacího systému. Vzhledem k iterativnímu zpracování je zde zachycena iterace první a finální. V této kapitole je systém představen nejprve obecně jako celek a poté je přistoupeno k popisování diagramů, které byly při návrhu celého systému použity. Popis je prostřednictvím jednotlivých diagramů grafickou i textovou formou. V obou popsáních iteracích začíná návrh případy užití, přes scénáře k případům užití, poté následují sekvenční diagramy a výstupem každé iterace je diagram tříd, jenž zachycuje systém ze softwarového hlediska. Během celé práce bylo navrženo 19 případů užití a jejich scénářů, 10 sekvenčních diagramů a 12 tříd.

Poslední kapitola se zaměřuje na možnosti budoucího rozšíření navrhovaného systému, kdy jsou načrtnuty některé možnosti pro další práci se systémem.

Výsledkem bakalářské práce je návrh části systému, který by mohl být lékem na neutěšený stav na českých silnicích a parkovištích. Opírá se o rozvoj moderních technologií a růst celkové počítačové gramotnosti. Vzhledem k postupnému vývoji celého návrhu a k jeho konečnému řešení byl naplněn cíl této práce.

6 Použitá literatura

- [1] *Počet registrovaných vozidel* [online]. 2009 [cit. 2009-07-25]. Dostupný z WWW: <<http://zpravy.kurzy.cz/182652-pocet-registrovanых-automobilu-v-cr-v-prvnim-pololetimirne-poklesl-stejne-jako-purmerny-vek-aut/>>
- [2] *Počet dotazů na téma "Parkování" ve vyhledávači Google* [online]. 2009 [cit. 2009-08-10]. Dostupný z WWW: <<http://www.google.cz/search?hl=cs&client=firefox-a&rls=org.mozilla%3Acs%3Aofficial&q=parkov%C3%A1n%C3%AD+&btnG=Hledat&lr=>>>
- [3] *Modré zóny* [online]. 2007 [cit. 2009-08-08]. Dostupný z WWW: <<http://magistrat.prahamesto.cz/default.aspx?id=73897&ido=8639&sh=65591000>>
- [4] *GPP Variant* [online]. 2009 [cit. 2009-07-30]. Dostupný z WWW: <<http://www.green.cz/stranka-parkovaci-systemy-2>>
- [5] *CrossPark* [online]. 2009 [cit. 2009-08-07]. Dostupný z WWW: <http://www.cross.cz/index.php?dc=2_2_54_0>.
- [6] *ReadyPark 500* [online]. 2009 [cit. 2009-08-06]. Dostupný z WWW: <<http://www.nessy.cz/parkoviste.php?lg=cz>>.
- [7] ARLOW, Jim, NEUSTADT, Ila. *UML 2 a unifikovaný proces vývoje aplikací*. [s.l.] : [s.n.], 2008. 567 s.
- [8] KANISOVÁ, Hana, MÜLLER, Miroslav. *UML srozumitelně*. [s.l.] : [s.n.], 2007. 176 s.
- [9] *Agregace* [online]. 2008 [cit. 2009-07-07]. Dostupný z WWW: <<http://nb.vse.cz/~zelenyj/it380/eseje/xbohm07/image2.gif>>.
- [10] *Generalizace* [online]. 2008 [cit. 2009-07-07]. Dostupný z WWW: <<http://nb.vse.cz/~zelenyj/it380/eseje/xbohm07/image3.gif>>.
- [11] *Asociace* [online]. 2008 [cit. 2009-07-10]. Dostupný z WWW: <<http://kubanek.org/blog/wp-content/uploads/2009/02/d2.png>>.
- [12] *Sekvenční diagramy* [online]. 2008 [cit. 2009-07-30]. Dostupný z WWW: <http://www.komix.cz/Tisk/Clanky/Historie/Uvod_UML.aspx>.
- [13] *Sekvenční diagram* [online]. 2008 [cit. 2009-07-24]. Dostupný z WWW: <http://mpavus.wz.cz/uml/img/seq_2.gif>.
- [14] *Parkovací automat* [online]. 2009 [cit. 2009-08-05]. Dostupný z WWW: <http://autogard.testujeme.cz/data/img/park_systemy/EPA700.jpg>.
- [15] *Parkovací závora* [online]. 2009 [cit. 2009-08-08]. Dostupný z WWW: <<http://www.cross.cz/katalog/fckeditor/editor/filemanager/connectors/php/image/AZ.jpg>>.

- [16] *CVV2 kód* [online]. 2009 [cit. 2009-08-20]. Dostupný z WWW: <<http://www.mesec.cz/poradna/placeni-na-internetu-duben-2005/13544/>>.
- [17] *Http://cs.wikipedia.org/wiki/Search_Engine_Optimization* [online]. 2009 [cit. 2009-08-19]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Search_Engine_Optimization>.

SEZNAM OBRÁZKŮ

<i>Obrázek 1</i> Struktura jazyka UML, zdroj: autor.....	11
<i>Obrázek 2</i> Agregace, zdroj: [9].....	12
<i>Obrázek 3</i> Generalizace, zdroj: [10]	13
<i>Obrázek 4</i> Asociace, zdroj: [11].....	13
<i>Obrázek 5</i> Diagramy UML - zdroj: autor	15
<i>Obrázek 6</i> Model případu užití, zdroj: autor	16
<i>Obrázek 7</i> Aktér, zdroj: autor.....	17
<i>Obrázek 8</i> Use case, zdroj: autor.....	17
<i>Obrázek 9</i> Vzor sekvenčního diagramu, zdroj: [13].....	21
<i>Obrázek 10</i> Návrh systému, zdroj: autor.....	23
<i>Obrázek 11</i> Parkovací automat, zdroj: [14].....	24
<i>Obrázek 12</i> Parkovací závora, zdroj: [15].....	24
<i>Obrázek 13</i> První iterace – Use Case, zdroj: autor	26
<i>Obrázek 14</i> Sekvenční diagram - Odstranit službu, zdroj: autor	36
<i>Obrázek 15</i> Sekvenční diagram - Přidat položku, zdroj: autor	37
<i>Obrázek 16</i> První iterace - classs diagram, zdroj: autor.....	38
<i>Obrázek 17</i> Finální iterace - Use Case, zdroj: autor.....	40
<i>Obrázek 18</i> Sekvenční diagram – Vystavit pokutu, zdroj: autor	49
<i>Obrázek 19</i> Finální iterace - class diagram, zdroj: autor.....	50

SEZNAM TABULEK

<i>Tabulka 1 Možné typy asociací, autor – upraveno na základě: [8]</i>	14
<i>Tabulka 2 Scénář případů užití, zdroj: autor</i>	18
<i>Tabulka 3 Příklad užití ID 1, zdroj: autor</i>	28
<i>Tabulka 4 Příklad užití ID 2, zdroj: autor</i>	28
<i>Tabulka 5 Příklad užití ID 3, zdroj: autor</i>	29
<i>Tabulka 6 Příklad užití ID 4, zdroj: autor</i>	30
<i>Tabulka 7 Příklad užití ID 5, zdroj: autor</i>	30
<i>Tabulka 8 Příklad užití ID 6, zdroj: autor</i>	31
<i>Tabulka 9 Příklad užití ID 7, zdroj: autor</i>	32
<i>Tabulka 10 Příklad užití ID 8, zdroj: autor</i>	32
<i>Tabulka 11 Příklad užití ID 9, zdroj: autor</i>	33
<i>Tabulka 12 Příklad užití ID 10, zdroj: autor</i>	34
<i>Tabulka 13 Příklad užití ID 11, zdroj: autor</i>	35
<i>Tabulka 14 Příklad užití ID 12, zdroj: autor</i>	36
<i>Tabulka 15 Příklad užití ID 13, zdroj: autor</i>	41
<i>Tabulka 16 Příklad užití ID 14, zdroj: autor</i>	42
<i>Tabulka 17 Příklad užití ID 5, zdroj: autor</i>	43
<i>Tabulka 18 Příklad užití ID 15, zdroj: autor</i>	43
<i>Tabulka 19 Příklad užití ID 16, zdroj: autor</i>	45
<i>Tabulka 20 Příklad užití ID 17, zdroj: autor</i>	46
<i>Tabulka 21 Příklad užití ID 18, zdroj: autor</i>	47
<i>Tabulka 22 Příklad užití ID 19, zdroj: autor</i>	48