

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Diplomová práce

Analýza a návrh informačního systému v UML

Bc. Markéta Brauneisová

© 2021 ČZU v Praze

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Markéta Brauneisová

Systemové inženýrství a informatika
Informatika

Název práce

Analýza a návrh informačního systému v UML

Název anglicky

Analysis and design of information system in UML

Cíle práce

Cílem diplomové práce je analyzovat a navrhnout model informačního systému v jazyce UML. Modelovaný informační systém bude poskytovat podporu při zprostředkování služeb v oblasti prodeje nemovitostí.

Dílní cíle diplomové práce jsou:

- vytvořit literární rešerši v oblasti návrhu informačních systémů, analýzy uživatelských požadavků a objektového modelování informačních systémů,
- vytvořit analýzu uživatelských požadavků a datový slovník,
- navrhnout objektový model, stavový model a model interakcí.

Metodika

Metodika diplomové práce je založena na studiu odborných informačních zdrojů. Získané znalosti budou využity pro návrh daného informačního systému. Pro návrh modelu informačního systému budou použity metody softwarového inženýrství a jazyk UML (Unified Modeling Language). Na základě shrnutí teoretických poznatků a výsledků vlastní části práce budou formulovány závěry diplomové práce.

Doporučený rozsah práce

60-80

Klíčová slova

informační systém, UML, modelování informačního systému, objektivně orientovaný přístup

Doporučené zdroje informací

BOOCH, Grady, James RUMBAUGH a Ivar JACOBSON. The unified modeling language user guide. 2nd ed. Upper Saddle River: Addison-Wesley, 2005. ISBN 9780321267979.

BRUCKNER, Tomáš. Tvorba informačních systémů: principy, metodiky, architektury. Praha: Grada, 2012. Management v informační společnosti. ISBN isbn978-80-247-4153-6.

BUCHALCEVOVÁ, Alena a Iva STANOVSKÁ. Příklady modelů analýzy a návrhu aplikace v UML. Praha: Oeconomica, 2013. ISBN 978-80-245-1922-7.

COAD, P. Object oriented analysis. New Jersey: Prentice-Hall, 1991. ISBN 0136299814.

PAGE-JONES, Meilir. Základy objektově orientovaného návrhu v UML. Praha: Grada, 2001. Moderní programování. ISBN 80-247-0210-x.

VRANA, Ivan. Projektování informačních systémů s UML. V Praze: Česká zemědělská univerzita, Provozně ekonomická fakulta, 2008. ISBN 978-80-213-1817-5.

Předběžný termín obhajoby

2020/21 LS – PEF

Vedoucí práce

Ing. Jan Tyrychtr, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 19. 11. 2020

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 19. 11. 2020

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 26. 02. 2021

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Analýza a návrh informačního systému v UML" jsem vypracovala samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autorka uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušila autorská práva třetích osob.

V Praze dne 30.3.2021

Poděkování

Ráda bych touto cestou poděkovala vedoucímu diplomové práce Ing. Janu Tyrychtrovi, Ph.D. za trpělivost, cenné rady a čas, který mi věnoval v průběhu psaní této práce.

Analýza a návrh informačního systému v UML

Abstrakt

Tato diplomová práce se zabývá analýzou a návrhem informačního systému v jazyce UML. Modelovaný informační systém slouží pro podporu služeb v oblasti prodeje nemovitostí. Systém je koncipován tak, aby efektivně podporoval proces prodeje nemovitostí a zároveň snížil administrativní náročnost nezbytných činností spojených s tímto procesem.

Teoretická část práce má rešeršní charakter a věnuje se problematice analýzy uživatelských požadavků, návrhu informačních systémů a objektového modelování informačních systémů. Dále jsou zde uvedeny a popsány standardy jazyka UML.

Úvodem do praktické části je stručné představení navrhovaného systému. Následně je navrhovaný systém porovnán s existujícími konkurenčními řešeními, a to za účelem ověření konkurenceschopnosti systému na trhu. V rámci analytické části je provedena analýza uživatelských požadavků. Ze stanovených požadavků vychází návrh systému, který tvoří model tříd, stavový model a model interakcí. V závěru práce jsou pro vizualizaci systému navrženy wireframes a jsou zde formulovány doporučení pro implementaci.

Klíčová slova: informační systém, analýza informačního systému, návrh informačního systému, objektově orientovaný přístup, UML, modelování informačního systému, model tříd, stavový model, model interakcí, wireframe

Analysis and design of information system in UML

Abstract

This diploma thesis deals with the analysis and design of an information system in the UML language. The modeled information system is used to support services in the area of real estate sales. The system is designed to effectively support the process of selling real estate and at the same time reduce the administrative complexity of the necessary activities associated with this process.

The theoretical part of the work has a research character and deals with the analysis of user requirements, design of information systems and object modeling of information systems. Furthermore, UML standards are listed and described here.

The introduction to the practical part is a brief introduction to the proposed system. Subsequently, the proposed system is compared with existing competing solutions to verify the competitiveness of the system in the market. Within the analytical part, the analysis of user requirements is performed. The design of the system, which consists of a class model, a state model and a model of interactions, is based on the identified requirements. At the end of the work, wireframes are designed for system visualization and recommendations for implementation are formulated.

Keywords: information system, information system analysis, information system design, object-oriented approach, UML, information system modeling, class model, state model, interaction model, wireframe

Obsah

1 Úvod	13
2 Cíl práce a metodika.....	14
2.1 Cíl práce.....	14
2.2 Metodika.....	14
2.2.1 Vícekriteriální analýza variant.....	14
2.2.2 Analýza uživatelských požadavků	15
2.2.3 Návrh systému	16
3 Teoretická východiska	19
3.1 Informační systém.....	19
3.1.1 Klasifikace informačních systémů dle úrovně řízení	20
3.2 Podnikové informační systémy.....	21
3.3 Životní cyklus informačního systému	23
3.3.1 Modely životního cyklu informačního systému	27
3.4 Metodiky vývoje informačních systémů	32
3.4.1 Rigorózní metodiky.....	33
3.4.2 Agilní metodiky	35
3.4.3 Porovnání agilních a rigorózních metodik	36
3.5 Analýza požadavků na informační systém.....	37
3.5.1 Sběr požadavků.....	38
3.5.2 Typy požadavků.....	38
3.6 Přístupy k analýze informačních systémů	39
3.6.1 Strukturovaný přístup.....	39
3.6.2 Objektově orientovaný přístup.....	40
3.6.3 Porovnání strukturovaného a objektivně orientovaného přístupu	41
3.7 UML (Unified Modeling Language).....	41
3.7.1 Historie UML.....	42
3.7.2 Způsoby užití UML.....	42
3.7.3 Základní charakteristiky a struktura UML	43
4 Vlastní práce	51
4.1 Obecný popis systému.....	51
4.2 Dostupná konkurenční řešení na trhu.....	52
4.2.1 Realko.....	53
4.2.2 Poski REAL.....	54

4.2.3	Softreal	54
4.2.4	Ostatní řešení	55
4.2.5	Shrnutí a výběr nejlepšího řešení	55
4.3	Analýza požadavků	58
4.3.1	Funkční požadavky	58
4.3.2	Nefunkční požadavky.....	64
4.4	Návrh systému	67
4.4.1	Model tříd	67
4.4.2	Stavový model	72
4.4.3	Model interakcí	77
4.4.4	Wireframes	90
4.4.5	Návrh realizace	92
5	Výsledky a diskuse	94
5.1	Zhodnocení návrhu.....	94
5.2	Diskuse	95
6	Závěr	96
7	Seznam použitých zdrojů	98
8	Přílohy	102

Seznam obrázků

Obrázek 1: Formát pro vyjádření požadavků (Zdroj: Arlow, Neustadt, 2007)	16
Obrázek 2: Klasifikace IS dle úrovní řízení (Zdroj: Vymětal, 2009).....	20
Obrázek 3: Vzájemné vztahy stěžejních aplikací ERP II (Zdroj: Basl, Blažíček, 2012)	22
Obrázek 4: Vodopádový model (Zdroj: Sommerville, 2013).....	28
Obrázek 5: Iterativní model (Zdroj: Myslín, 2016)	29
Obrázek 6: Prototypový model (Zdroj: Šmíd, 2005)	30
Obrázek 7: Spirálový model (Zdroj: Šmíd, 2005)	31
Obrázek 8: Inkrementální model (Zdroj: Požár, 2010)	32
Obrázek 9: Pracovní postupy v rámci metody UP (Zdroj: Arlow, Neustadt, 2007).....	34
Obrázek 10: Srovnání agilních a rigorózních metodik (Zdroj: Buchalcevo va , 2005)	36
Obrázek 11: Diagramy UML (Zdroj: Arlow, Neustadt, 2007).....	46
Obrázek 12: Obohacený prvek (Zdroj: Arlow, Neustadt, 2007)	47
Obrázek 13: Architektura 4+1 (Zdroj: Arlow, Neustadt, 2007)	49
Obrázek 14: Zjednodušené schéma systému	52
Obrázek 15: Náhled aplikace Realko (Zdroj: Realko, 2021)	53
Obrázek 16: Náhled aplikace Poski REAL (Zdroj: Poski REAL, 2021)	54
Obrázek 17: Náhled aplikace Softreal (Zdroj: Softreal, 2021).....	55
Obrázek 18: Diagram tříd	68
Obrázek 19: Stavový diagram pro třídu Uživatel	73
Obrázek 20: Stavový diagram pro třídu Nabídka	75
Obrázek 21: Stavový diagram pro třídu ObchodníPřípad	76
Obrázek 22: Diagram případů užití – Celkový pohled na systém	78
Obrázek 23: Dekompozice případů užití Správa šablon a Správa smluv.....	79
Obrázek 24: Sekvenční diagram – Přidat šablonu	81

Obrázek 25: Sekvenční diagram – Editovat šablonu	83
Obrázek 26: Sekvenční diagram – Vytvořit smlouvu	85
Obrázek 27: Sekvenční diagram – Připojit smlouvu k obchodnímu případu	87
Obrázek 28: Diagram aktivit – Vložení šablony do systému	88
Obrázek 29: Diagram aktivit – Připojení smlouvy k obchodnímu případu.....	89
Obrázek 30: Wireframe – Domovská stránka.....	90
Obrázek 31: Wireframe – Správa obchodních případů	91
Obrázek 32: Wireframe – Detail obchodního případu	92
Obrázek 33: Diagram nasazení	93

Seznam tabulek

Tabulka 1: Kritéria pro požadavky (Zdroj: Arlow, Neustadt, 2007)	16
Tabulka 2: Životní cyklus IS dle různých autorů (Zdroj: Šmíd, 2005, Řepa, 1999, Polák et al., 2003, Kendall, 1991, Král, 1998)	24
Tabulka 3: Porovnání funkcí popsaných konkurenčních řešení	56
Tabulka 4: Pořadí variant určené metodou TOPSIS	57
Tabulka 5: Seznam funkčních požadavků	58
Tabulka 6: Seznam nefunkčních požadavků	64
Tabulka 7: Scénář případu užití – Přidat šablonu	80
Tabulka 8: Scénář případu užití – Editovat šablonu	82
Tabulka 9: Scénář případu užití – Vytvořit smlouvu	84
Tabulka 10: Scénář případu užití – Připojit smlouvu k obchodnímu případu.....	86

Seznam použitých zkratek

API – Application Programming Interface
BI – Business Intelligence
CPM – Corporate Performance Management
CRM – Customer Relationship Management
CSS – Cascading Style Sheets
DSS – Decision Support Systems
EIS – Executive Information System
ERP – Enterprise Resources Planning
HTML – HyperText Markup Language, značkovací jazyk
HTTPS – Hypertext Transfer Protocol Secure
IS – informační systém
JS – JavaScript, skriptovací jazyk
MIS – Management Information System
OIS – Office Information System
OMG – Object Management Group
OOP – objektově orientovaný přístup
OPEN – Object-oriented Process, Environment and Notation
PHP – Hypertext Preprocessor, skriptovací programovací jazyk
PIS – podnikový informační systém
REST – Representational State Transfer
RUP – Rational Unified Process
SCM – Supply Chain Management
SOAP – Simple Object Access Protocol
TLS – Transport Layer Security
TPS – Transaction Processing System
UML – Unified Modeling Language
UP – Unified Process
XML-RPC – protokol pro vzdálené volání procedur

1 Úvod

V současné době se fungování téměř žádné firmy na trhu neobejde bez podpory informačních a komunikačních technologií, tedy i informačních systémů. Informační systém ve firmě by měl podporovat firemní procesy, tak aby zvýšil jejich efektivitu a zároveň tak zoptimalizoval chod celé firmy. Taková firma snadněji dosahuje stanovených cílů a může získat i konkurenční výhodu na trhu. V neposlední řadě znamená optimální podpora procesů pro firmu úsporu provozních i personálních nákladů. Není tomu jinak ani v oblasti prodeje nemovitostí, na kterou se tato práce zaměřuje.

Tato diplomová práce se zabývá analýzou a návrhem informačního systému pro podporu prodeje nemovitostí, tedy informačním systémem určeným pro realitní kanceláře. V práci jsou promítnuty i profesní zkušenosti autorky, které získala během působení na administrativní pozici v nejmenované realitní kanceláři.

Jednou z klíčových věcí při uzavírání obchodu je dostatečná komunikace s klientem. Proto musí být informační systém navržený tak, aby efektivně podporoval firemní procesy a zejména snižoval administrativní náročnost činností souvisejících s uzavíráním obchodů. Zaměstnanci realitních kanceláří tak nebudou muset trávit tolik času vyřizováním administrativy, ale budou mít více času věnovat se svým klientům. Jednou z časově nejnáročnějších administrativních činností při prodeji nemovitostí je příprava a evidence smluv. Při ručním vyplňování údajů do smluv navíc často dochází k chybám. Pokud dojde k nalezení chyb ve smlouvě klientem nebo obchodním partnerem, není to pro realitní kancelář nejlepší vizitkou a v některých případech může dojít i k ukončení vzájemné spolupráce. Proto navrhovaný systém přípravu smluv automatizuje. Na trhu navíc v současné době neexistuje konkurenční řešení, které by nabízelo automatizaci tohoto procesu.

Návrh systému je realizován prostřednictvím grafického modelovacího jazyka UML (Unified Modeling Language), který umožňuje zprostředkovat pohled na navrhovaný systém z různých úhlů a v různých úrovních abstrakce. Jednou z dalších výhod UML je také to, že návrh je srozumitelný nejen pro vývojáře ale i pro obchodníky a manažery.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem této práce je analyzovat a navrhnout model informačního systému v jazyce UML. Modelovaný informační systém bude poskytovat podporu při zprostředkování služeb v oblasti prodeje nemovitostí. Cílem teoretické části práce je vytvořit literární rešerši v oblasti tvorby informačních systémů, analýzy uživatelských požadavků a objektového modelování informačních systémů. Cílem praktické části práce je vytvořit analýzu uživatelských požadavků a datový slovník. Na základě provedené analýzy bude navržen model tříd, stavový model a model interakcí.

2.2 Metodika

Metodika diplomové práce je založena na studiu odborných informačních zdrojů. Získané znalosti byly využity pro návrh daného informačního systému. Pro návrh modelu informačního systému byly použity metody softwarového inženýrství a jazyk UML (Unified Modeling Language). Na základě shrnutí teoretických poznatků a výsledků vlastní části práce byly formulovány závěry diplomové práce.

V následujících podkapitolách budou popsány jednotlivé metodiky využitě v praktické části práce.

2.2.1 Vícekriteriální analýza variant

Jako úlohy vícekritériálního rozhodování se nazývají ty rozhodovací úlohy, ve kterých se důsledky rozhodnutí posuzují dle více kritérií. Pokud je množina přípustných variant konečná, poté se hovoří o úloze vícekritériálního hodnocení variant (Fiala, Maňas, 1994).

Při hodnocení variant se zpravidla postupuje tak, že se nejprve vytvoří množina hodnotících kritérií a následně se určí váhy jednotlivých kritérií. Na závěr se stanoví pořadí variant podle některé z metod vícekritériálního rozhodování. V rámci této práce byla pro stanovení vah kritérií zvolena bodovací metoda, pro určení pořadí variant byla zvolena metoda TOPSIS.

Bodovací metoda

Při stanovení vah kritérií bodovací metodou je nejprve potřeba stanovit bodovou stupnici (např. od 0 do 10). Hodnocení každého kritéria je následně vyjádřeno počtem bodů. Platí, že čím je kritérium důležitější, tím více bodů obdrží, a naopak. Výpočet vah z bodového hodnocení je proveden dle následujícího vztahu:

$$v_i = \frac{b_i}{\sum_{i=1}^k b_i}$$

Tato metoda se používá především v případech, kdy kritéria hodnotí více expertů (Brožová, Houška, 2002).

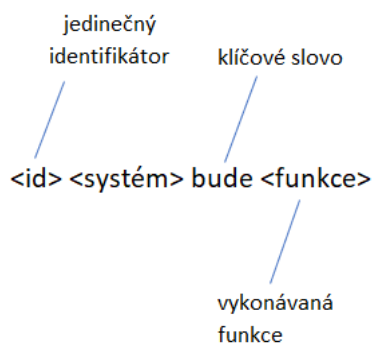
Metoda TOPSIS

Metoda TOPSIS je založena na výběru varianty, která je nejbližší tzv. ideální variantě, jež je charakterizována vektorem nejlepších kritériálních hodnot, a současně nejdále od tzv. bazální varianty, která je naopak reprezentována vektorem nejhorších kritériálních hodnot. Tato metoda vyžaduje, aby všechna kritéria byla maximalizační (Jablonský, 2007).

Všechny kroky metody TOPSIS jsou podrobně popsány v publikaci *Operační výzkum: kvantitativní modely pro ekonomické rozhodování* od Jablonského (2007).

2.2.2 Analýza uživatelských požadavků

Analýza požadavků na systém byla zpracována v souladu s teoretickými východisky uvedenými v této práci. Sběr požadavků byl uskutečněn dle popsáných způsobů zjišťování požadavků. Zjištěné požadavky byly rozděleny do dvou kategorií – na funkční a nefunkční požadavky. Pro vyjádření požadavků byl zvolen formát, jež doporučují Arlow a Neustadt (2007), zachycený na obrázku č. 1.



Obrázek 1: Formát pro vyjádření požadavků (Zdroj: Arlow, Neustadt, 2007)

Pro zachycení dodatečných informací k jednotlivým požadavkům byl zvolen atribut *priorita*, který popisuje relativní prioritu požadavku vůči ostatním požadavkům. Pro ohodnocení priority požadavků byla využita množina kritérií popsána v tabulce č. 1.

Hodnoty atributu <i>priorita</i>	Sémantika
Nezbytný (Must have)	Povinné požadavky, jež jsou základem systému.
Možný (Should have)	Důležité požadavky, které lze však vynechat.
Eventuální (Could have)	Požadavky, jež jsou opravdu nepovinné.
Chceme mít (Want to have)	Požadavky, které mohou být zahrnuty do dalších verzí systému.

Tabulka 1: Kritéria pro požadavky (Zdroj: Arlow, Neustadt, 2007)

2.2.3 Návrh systému

V dalších podkapitolách bude uvedena stručná metodika zpracování jednotlivých diagramů UML použitých při modelování systému. Podrobný popis a notace jednotlivých diagramů jsou uvedeny v publikaci *Unified modeling language user guide* od Booch et al. (1999).

Diagram tříd

Diagram tříd zobrazuje typy objektů v systému a různé druhy statických vztahů, které mezi nimi existují (Fowler, 2009). Třída objektů popisuje skupinu objektů s podobnými vlastnostmi (atributy), chováním (operacemi) a stejnými vztahy k jiným objektům. V diagramu tříd se typicky vyskytují vztahy asociace, generalizace, agregace a jejího

zvláštního případu kompozice (Vrana, 2008). Správná klasifikace je klíčem k úspěšně provedené analýze a návrhu.

Stavový diagram

Stavový diagram, také někdy označovaný jako stavový automat, reprezentuje všechny stavy, do kterých se objekt může někdy dostat, a také podmínky přechodů mezi jednotlivými stavy. Přechody mezi jednotlivými stavy jsou realizovány prostřednictvím událostí (Kanisová, Müller, 2004). Stavové diagramy byly v této práci modelovány pouze pro třídy, jež vykazují zajímavé chování, a pouze tam, kde je jejich modelování užitečné.

Diagram užití

Diagramy užití formálně znázorňují strukturu funkčnosti systému, hranice systému a jeho interakce s okolním světem. Hlavním obsahem diagramů užití jsou případy užití, vztahy mezi nimi a aktéři, kteří případy užití provádí. Zejména pro velké aplikace je vhodné případy užití strukturovat pomocí vztahů `<<include>>`, `<<extend>>` a zobecnění (Vrana, 2008). Vztah `<<include>>` začleňuje jeden případ užití do chování jiného případů užití, vztah `<<extend>>` pak přidává do existujícího případu užití nové chování (Arlow, Neustadt, 2007).

Sekvenční diagram

Sekvenční diagram typicky zachycuje chování jednoho případu užití tak, že zobrazuje interakci objektů zúčastněných v daném případě užití. U každého účastníka je v diagramu zobrazena svisle orientovaná čára života. Mezi těmito čarami života následně probíhá interakce prostřednictvím zasílání zpráv, přičemž časová posloupnost těchto zpráv je zobrazována shora dolů (Fowler, 2009). Mezi účastníky mohou být zasílány různé typy zpráv, přehledně je popisují Arlow a Neustadt (2007).

Diagram aktivit

Diagram aktivit zobrazuje aktivity a přechody mezi nimi. Tyto diagramy podporují jak sekvenční, tak paralelní chování. Paralelní chování je modelováno pomocí rozvětvení a spojení hlavního toku (Arlow, Neustadt, 2007). Kanisová a Müller (2004) uvádějí, že

diagram aktivit je v podstatě variantou stavového diagramu, proto je terminologie a notace diagramů aktivit velmi podobná stavovým diagramům. Diagramy aktivit se používají k modelování všech typů procesů (Arlow, Neustadt, 2007).

Diagram nasazení

Diagram nasazení mapuje architekturu softwaru na fyzickou architekturu systému, na níž bude systém spouštěn. V UML existují dvě verze diagramu nasazení – diagram nasazení a diagram konkrétního nasazení. V této práci byla vytvořena pouze první verze diagramu nasazení, která obvykle vzniká již v etapě návrhu. Diagram nasazení zobrazuje komponenty, uzly a vztahy mezi jednotlivými uzly, přičemž uzel reprezentuje typ hardwaru a komponenta zastupuje typ softwaru (Arlow, Neustadt, 2007).

3 Teoretická východiska

3.1 Informační systém

Informační systémy a informační a komunikační technologie označované zkratkou IS/ICT jsou v současnosti neoddelitelnou součástí lidského světa. Jedná se o dynamicky rozvíjející se odvětví, které zasahuje do širokého spektra lidských činností. Klíčovou roli hrají informační systémy ve fungování organizací všeho druhu především pak v byznysu, kde pomáhají při dosahování stanovených cílů a zlepšení konkurenceschopnosti podniku na trhu (Bruckner et al., 2012).

V současné době lze v odborné literatuře nalézt mnoho různých definic informačního systému. Obecně je možné říct, že *„informační systém (IS) lze chápat jako systém vzájemně propojených informací a procesů, které s těmito informacemi pracují (Požár, 2010).“* Přesnější definici lze nalézt v publikaci od Molnára (2009), který popisuje informační systém jako *„soubor lidí, technických prostředků a metod (programů), zabezpečujících sběr, přenos, zpracování, uchování dat, za účelem prezentace informací pro potřeby uživatelů činných v systémech řízení.“*

Z uvedených definic vyplývá, že každý informační systém musí disponovat prostředky pro sběr, kontrolu a uchování dat. Data slouží pro reprezentaci faktů, atributů, odrazu dějů a věcí. Aby data poskytovala informace, musejí být pro koncové uživatele zobrazena ve srozumitelné a použitelné formě. Informace závisejí také na znalostech a potřebách konkrétních uživatelů. Různí uživatelé informačních systémů využívají ke svým činnostem jiné informace (Sklenák, 2001, Král, 1998).

Informační systém je obvykle navržen jak nástroj podporující jisté činnosti. Funkce informačního systému jsou podporovány vhodnými technickými prostředky. Informační systém nutně nemusí využívat pouze automatizované činnosti (počítačem podporované), naopak často využívá také neautomatizované (ruční) způsoby práce. Volba optimálního poměru automatizovaných a neautomatizovaných činností je často zásadním problémem při návrhu informačního systému (Král, 1998).

3.1.1 Klasifikace informačních systémů dle úrovně řízení

V každé organizaci nalezneme několik organizačních (řídících) úrovní. Každá z nich zpravidla vyžaduje specifický způsob zpracování informací či specifický druh informací. Z toho důvodu je často používána klasifikace, která pomocí hierarchického modelu rozlišuje operativní (transakční), taktické a strategické informační systémy, jak lze vidět na obrázku č. 2 (Vymětal, 2009).



Obrázek 2: Klasifikace IS dle úrovně řízení (Zdroj: Vymětal, 2009)

- Operativní úroveň – zajišťuje zpracování informací týkajících se rutinní podnikové agendy např. z oblasti výroby, prodeje nebo logistiky. Informační systémy operativní úrovně řízení reagují na plnění každodenních činností a sledují tok transakcí napříč organizací. Jejich hlavním úkolem je poskytnout přesné, aktuální a jednoduše dostupné informace (Sodomka, 2006).
- Taktická úroveň – požaduje informace nutné k plnění administrativních úkolů a podpoře rozhodování. Jako podklady k rozhodování používá informace z transakčních systémů, které jsou prostřednictvím MIS transformované do podoby

výstupních sestav obsahujících souhrn výsledků z požadované oblasti (tzv. reporting). MIS poskytují flexibilní nástroje umožňující definici a výpočet vlastních modelových agregovaných ukazatelů pro rutinní i nerutinní rozhodování (Sodomka, 2006, Vymětal, 2009, Molnár, 2009).

- Strategická úroveň – informační systémy strategické úrovně slouží k identifikaci dlouhodobých trendů uvnitř i vně organizace. Jejich hlavním úkolem je predikovat očekávané změny a určit, zda a jak je podnik schopen na tyto změny reagovat (Sodomka, 2006). Dle Vymětala (2009) strategická úroveň rozhodování využívá systémy EIS (Executive Information System). Bruckner et al. (2012) ale uvádí, že podklady pro strategická rozhodnutí zajišťují systémy CPM (Corporate Performance Management).

Strategická rozhodování vyžadují podporu datových skladů, systémů pro podporu rozhodování (DSS), ad hoc analýz a dalších postupů, které souhrnně označujeme jako Business Intelligence (BI). Systémy pro strategické rozhodování využívají data jak z interních, tak z externích zdrojů (Vymětal, 2009, Sodomka, 2006).

3.2 Podnikové informační systémy

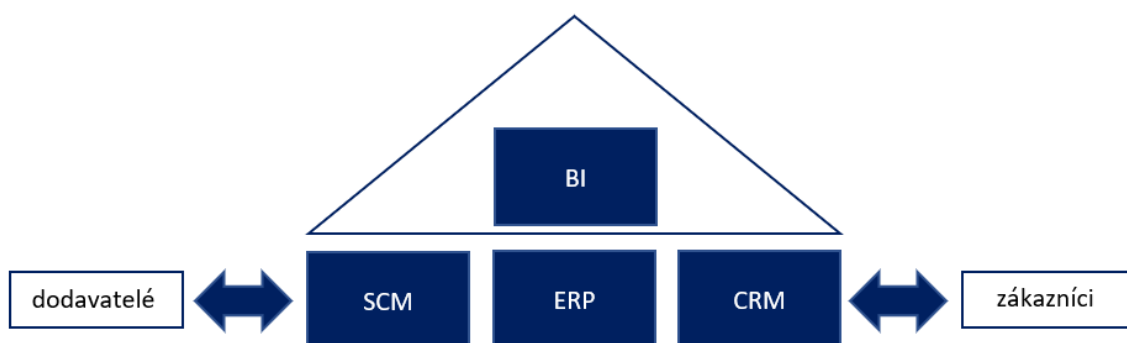
Základem pro fungování a prosperitu každého úspěšného podniku je efektivní zpracování informací a budování znalostní báze. Klíčovou technologií k dosažení tohoto cíle představuje podnikový informační systém (Sodomka, 2006).

Sodomka (2006) definuje podnikový informační systém (PIS) jako „*systém vytvářen lidmi, který prostřednictvím dostupných technologických prostředků a stanovené metodologie zpracovává podniková data a vytváří z nich informační a znalostní bázi organizace sloužící k řízení podnikových procesů, manažerskému rozhodování a správě podnikové agendy.*“

Hlavním úkolem PIS je zajistit adekvátní podporu podnikových procesů prostřednictvím informačních a komunikačních technologií. V minulosti byly v podnicích využívány systémy, pro které byl typický agendový způsob zpracování. Díky požadavkům trhu a větší dostupnosti hardwaru a softwaru vznikla v 90. letech potřeba integrovat dříve programově

oddělené moduly do jednoho společného řešení na bázi jednotné databázové platformy, což vedlo ke vzniku prvních celopodnikových systémů typu ERP (Enterprise Resource Planning) (Gála et al., 2009, Basl, Blažíček, 2012).

Postupem času se funkčnost ERP systémů rozšířila o řízení dodavatelských řetězců (SCM– Supply Chain Management), o řízení vztahů se zákazníky (CRM – Customer Relationship Management) a o manažerský informační systém BI (Business Intelligence). Toto rozšíření se označuje jako ERP II (Basl, Blažíček, 2012). Vzájemný vztah stěžejních aplikací ERP II lze vidět na obrázku č. 3.



Obrázek 3: Vzájemné vztahy stěžejních aplikací ERP II (Zdroj: Basl, Blažíček, 2012)

ERP (Enterprise Resource Planning)

ERP představuje jádro soudobého PIS. Úkolem ERP je pokrýt plánování a řízení hlavních interních podnikových procesů, tj. výroba, logistika, personalistika a ekonomika. Mezi nejdůležitější vlastnosti ERP systémů patří schopnost automatizovat a integrovat hlavní podnikové procesy, sdílet data a postupy přes celý podnik a vytvářet a zpřístupňovat informace v reálném čase (Molnár, 2009).

CRM (Customer Relationship Management)

CRM je aplikace určená k řízení vztahů se zákazníky, což je dnes považováno za jeden ze základních zdrojů konkurenceschopnosti podniků (Gála et al., 2009).

Umožňuje sbírat a uchovávat data o zákaznících, která lze pomocí analytických nástrojů zobrazit z různých pohledů. Díky těmto datům lze např. efektivněji navrhovat a realizovat marketingové kampaně a analyzovat zákaznické chování (Basl, Blažiček, 2012).

CRM nepřináší užitek jen podnikům, ale také zákazníkům. Díky zaznamenávání detailních informací o zákaznících, o jejich požadavcích či o stavu a průběhu jednotlivých obchodních případů jsou podniky schopné uplatnit individuální přístup k jednotlivým zákazníkům a zajistit jim kvalitní informační služby o jejich zakázkách (Gála et al., 2009).

SCM (Supply Chain Management)

SCM neboli řízení dodavatelských řetězců představuje soubor nástrojů a procesů, které slouží k optimalizaci řízení a k maximální efektivitě všech článků celého dodavatelského řetězce. SCM propojuje dodavatele a odběratele pomocí informačních technologií. Díky tomu mohou partneři v rámci řetězce spolupracovat, sdílet informace, optimalizovat tak celý postup a zároveň zvýšit akceschopnost a ziskovost celého řetězce (Basl, Blažiček, 2012).

BI (Business Intelligence)

Softwarové aplikace typu BI využívají data uložená ve standardním ERP, v CRM aplikacích nebo SCM. Lze říct, že BI aplikace integrují věcně či lokálně samostatné informační zdroje, které transformují podle požadavků konkrétního uživatele (manažera či analytika) do podoby vhodné pro podnikové analýzy a reporty. Aplikace BI nabízejí detailní i agregované informace za delší časové období formou přehledových tabulek a různých grafů, které zachycují trendy či korelace různých jevů. Takto zpracované informace výrazně přispívají ke zlepšení kvality a výkonnosti podnikového řízení a zvýšení konkurenceschopnosti podniku (Gála et al., 2009, Basl, Blažiček, 2012).

3.3 Životní cyklus informačního systému

Bruckner et al. (2012) uvádí tuto definici životního cyklu IS: „*Životní cyklus informačního systému je časový úsek, který začíná úmyslem vytvořit systém a končí, když se systém přestane používat.*“

Životní cyklus informačního systému bývá zpravidla organizován do několika fází, jejichž pojmenování, počet či pojetí se liší u různých autorů odborné literatury. Lze říct, že prakticky neexistuje jednotný pohled na definování a chápání životního cyklu IS.

Přehled fází životního cyklu IS, jak je popisují různí autoři, lze vidět v tabulce č. 2.

Fáze/Autor	Šmíd	Řepa	Polák	Kendall	Král
1.	Předběžná analýza neboli specifikace cílů	Informační strategie organizace	Zadání	Identifikace problémů, možností a cílů	Marketing a specifikace cílů
2.	Analýza systému neboli specifikace požadavků	Úvodní studie systému	Analýza	Definování informačních potřeb	Specifikace požadavků
3.	Projektová studie neboli návrh	Globální analýza a návrh	Návrh	Analýza systémových potřeb	Návrh systému
4.	Implementace	Detailní analýza a návrh	Implementace	Návrh doporučeného systému	Kódování
5.	Testování	Implementace	Testování a provoz	Vývoj a dokumentace k softwaru	Testování
6.	Zavádění systému	Zavedení		Testování a zavádění softwaru	Oživení a předání
7.	Zkušební provoz	Provoz, údržba a rozvoj		Údržba a hodnocení systému	Údržba
8.	Rutinní provoz a údržba				

Tabulka 2: Životní cyklus IS dle různých autorů (Zdroj: Šmíd, 2005, Řepa, 1999, Polák et al., 2003, Kendall, 1991, Král, 1998)

Šmíd (2005), který definuje životní cyklus IS nejpodrobněji, uvádí osm částí, které tvoří životní cyklus informačního systému a jejichž popisem se budou zabývat následující

podkapitoly. Pro lepší pochopení budou doplněné i o názory ostatních autorů zabývajících se touto problematikou.

1. Předběžná analýza neboli specifikace cílů

Základem každého návrhu, vývoje či úpravy informačního systému jsou požadavky uživatelů a cíle organizace. Proto je cílem této části dané požadavky shromáždit, v hrubých rysech rozebrat a odhadnout dobu realizace a náklady. Výstupem by měl být dokument, který specifikuje účel systému, identifikuje jeho uživatele a jejich požadavky, definuje části systému a navrhuje jejich řešení, obsahuje seznamy událostí a odhady datové základny a technického i softwarového zajištění (Šmíd, 2005).

Řepa (1999) dále zdůrazňuje důležitost provedení marketingového výzkumu v případě záměru vyvíjet IS pro trh. To zahrnuje zjištění nepokrytých míst na trhu, prozkoumání záměru konkurenčních podniků a vyhledání možných zákazníků.

2. Analýza systému neboli specifikace požadavků

Tato část životního cyklu systému je rozbořením předchozí části. Je zároveň klíčovou částí, neboť veškeré chyby v předchozí analýze, které se zde neodhalí, jsou později velmi obtížně odstranitelné (Šmíd, 2005).

Král (1998) uvádí, že tato část by měla být ukončena oponenturou (tzv. studií proveditelnosti) testující, zda požadavky odpovídají cílům projektu a zda jsou konzistentní a splnitelné v daných termínech a s danými zdroji.

3. Projektová studie neboli návrh

Tato část je výsledkem analýz provedených v předchozích etapách cyklu. Výstup této části by měl obsahovat podklady pro obsah smlouvy s externím firmou, časový harmonogram, finanční rámec projektu, konkrétní implementace systému, podmínky zavádění v organizaci, záruční servis a podmínky celkového předání informačního systému. Je zásadní, aby výsledná studie byla vypracována v dostatečně detailním provedení a v podobě, která bude srozumitelná všem aktérům. V případě dohody mezi zákazníkem

a tvůrci systému bude tvořit podklad pro realizaci systému, podmínky předání a testování (Šmíd, 2005).

4. Implementace

V této části probíhá vlastní programování navrženého informačního systému. Jako podklady slouží veškeré informace shromážděné v předchozích etapách a fyzický návrh systému. Na základě fyzického návrhu se definují vstupy a výstupy jednotlivých operací, naprogramují se veškeré funkce a doladí se jejich vzájemné propojení. Na závěr se realizované funkce ověří a připraví se testovací data (Šmíd, 2005).

Dle Bruckner et al. (2012) je součástí implementace také kompletace celé dokumentace (projektové, programové, uživatelské a administrátorské), která postupně vznikala v jednotlivých fázích projektu.

5. Testování

Buchalcevová a Stanovská (2013) uvádějí, že testování hraje při vývoji IS důležitou roli, neboť poskytuje zpětnou vazbu, zda vyvíjený systém uspokojuje definované požadavky a identifikuje nesrovnalosti a chyby v řešení.

Dle Šmída (2005) se testování provádí na informačním systému, který ale ještě není v reálném prostředí, neboť případné selhání by mohlo mít rozsáhlé následky (např. ve zdravotnických či leteckých systémech).

6. Zavádění systému

V této části probíhá instalace hotového systému a jeho zavedení do provozu organizace. V rámci toho dochází k transformaci původní datové základny, aby byla přístupná novému systému, poskytnutí manuálů a školení uživatelů. Školení uživatelů by nemělo být podceňováno, neboť je klíčové pro úspěch celého projektu (Šmíd, 2005).

Zavedení systému může v praxi proběhnout několika různými způsoby. Tyto způsoby podrobně popisují Šmíd (2005) a Molnár (2009).

7. Zkušební provoz

Bezprostředně po zavedení systému do provozu v organizaci následuje zkušební provoz systému, ve kterém je poskytovatel povinný zajistit okamžitý servis, odstranit chyby zjištěné během provozu nebo dořešit požadavky uživatelů v rámci původního návrhu (Šmíd, 2005).

Dle Vransky et al. (2005) by měl být provoz nového systému nejdříve spuštěn v omezeném rozsahu např. ve vybraném oddělení. V tomto omezeném provozu lze vyřešit vzniklé problémy a většina uživatelů se v ostrém provozu setká s již doladěným systémem.

8. Rutinní provoz a údržba

Poslední a pravděpodobně nejdelší částí je fáze, kdy je systém provozován a používán. V rámci údržby systému probíhá např. odstraňování chyb zjištěných za provozu, přizpůsobení nového hardwaru a změny v použitém systémovém softwaru (např. v databázových a operačních systémech). V neposlední řadě probíhá také vylepšování funkcí systému a jeho úpravy na základě aktuálních požadavků uživatelů. (Král, 1998, Šmíd, 2005)

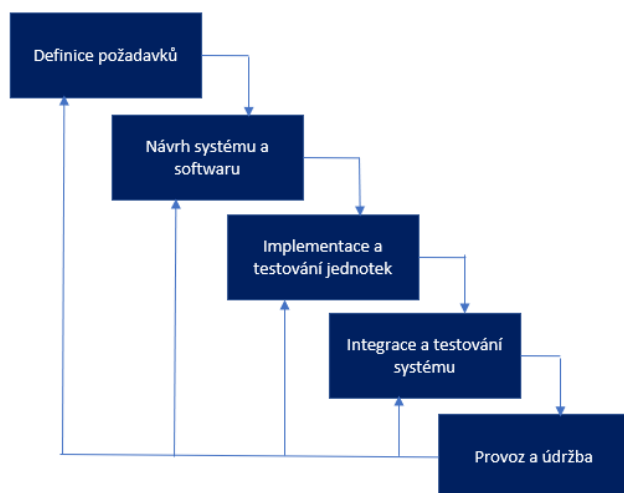
3.3.1 Modely životního cyklu informačního systému

V předešlé kapitole byly popsány jednotlivé části životního cyklu informačních systémů. Následující podkapitoly se budou zabývat nejznámějšími modely životního cyklu IS.

Vodopádový model

Vodopádový model se rozšířil zejména v 70. a 80. letech minulého století. Jedná se o vůbec nejstarší přístup k této problematice, který se dodnes používá i ve většině dalších inženýrských oborů. Podstatou vodopádového modelu je skutečnost, že existuje systematický postup, jak dojít od zadání k řešení pomocí řady na sebe navazujících činností, jež lze předem naplánovat (Merunka, 2006, Bruckner et al., 2012).

Jak lze vidět na obrázku č. 4, vodopádový model tvoří několik po sobě jdoucích etap činností.



Obrázek 4: Vodopádový model (Zdroj: Sommerville, 2013)

Použití vodopádového modelu je vhodné především v případech, kdy je možné v první fázi vývoje definovat všechny požadavky na systém, které se navíc v čase příliš nemění. V opačném případě nepřináší tento model dobré výsledky. Hlavní nevýhodou modelu je především pozdní integrace programového systému, která se provádí až po naprogramování všech modelů. Nedostatky a opomenutí specifikace požadavků se často projeví až v této fázi, což je příliš pozdě, neboť opravy a přeprogramování systému jsou finančně nákladné, a navíc vedou k celkovému zpoždění projektu (Bruckner et al., 2012, Král, 1998).

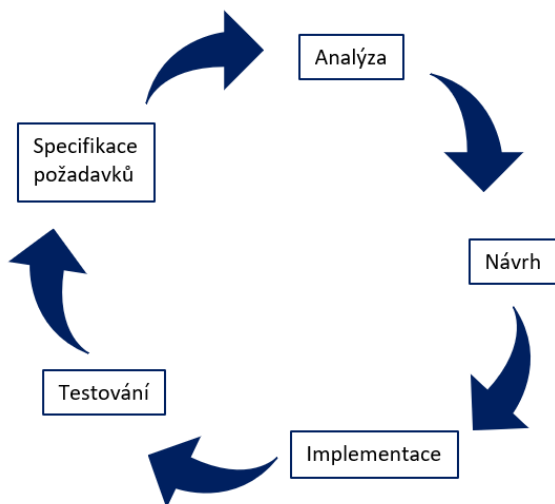
„Vodopádový model lze chápat jako univerzální model, který má své nevýhody, ale je podstatně lepší než náhodný, metodicky neřízený přístup k řešení systému“ (Šmíd, 2005).

Iterativní model

Myslín (2016) uvádí, že iterativní model nelze popsat jen jako jednu konkrétní metodiku, neboť se jedná o několik metodik vyznačujících se tím, že využívají iterativní přístup.

Iterativní vývoj je založen na principu rozložení projektu na řadu dílčích projektů tzv. iterací. Jak je patrné z obrázku č. 5, každá iterace zároveň obsahuje všechny fáze vývoje – specifikaci požadavků, analýzu, návrh, implementaci a testování. Iterace se provádějí až do doby, než je dosaženo konečného cíle. Jinými slovy je možné vracet se

do předchozích fází vývoje projektu za účelem zpřesnění zadání (Bruckner et al., 2012, Myslín, 2016, Merunka, 2006).



Obrázek 5: Iterativní model (Zdroj: Myslín, 2016)

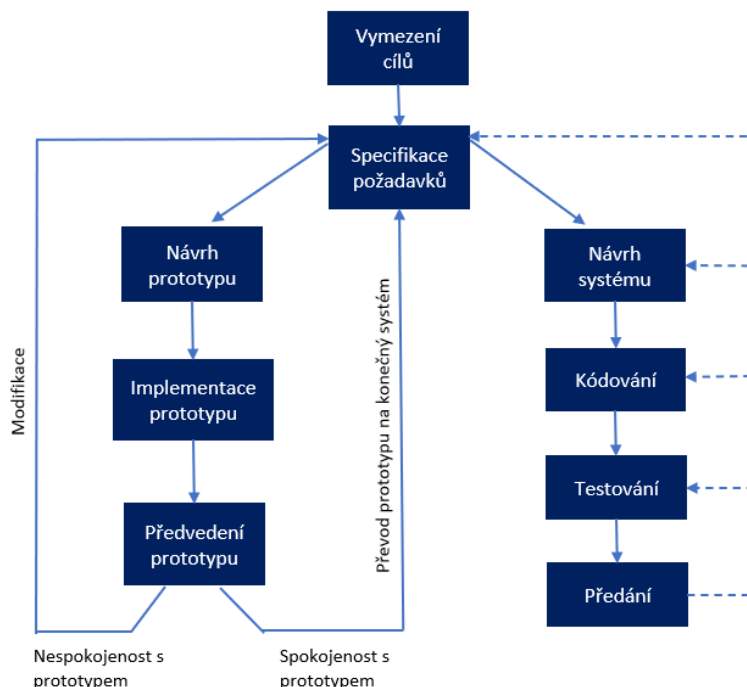
Výhodou iterativního modelu je, že umožňuje lépe sledovat postup projektu. Po každé iteraci je možné zhodnotit provedenou práci a odhalit včas případné problémy. Zákazník navíc na konci každé iterace vidí dílčí výsledky a může poskytnout rychlou zpětnou vazbu (Bruckner et al., 2012).

Iterativní přístup k vývoji software lze nalézt např. v prototypovém a spirálovém modelu (Merunka, 2006).

Prototypový model

Podstatou a také hlavní výhodou prototypového modelu je předpoklad změn funkčních požadavků a umožnění reakce na tyto změny, čímž se odlišuje od modelu vodopád, kde je nutné všechny požadavky přesně a neměnně definovat již v první fázi vývoje. Prototyp lze popsat jako zjednodušenou implementaci celého systému nebo jako plnou implementaci části systému. Hlavním cílem prototypu je seznámit zákazníka v co nejkratším čase s prvními verzemi systému, což mu umožňuje bezprostředně reagovat na výsledky a specifikovat další požadavky, na základě kterých je modifikován prototyp systému.

Po zanesení všech požadavků a připomínek následuje samotný návrh a implementace celého systému (Šmíd, 2005).

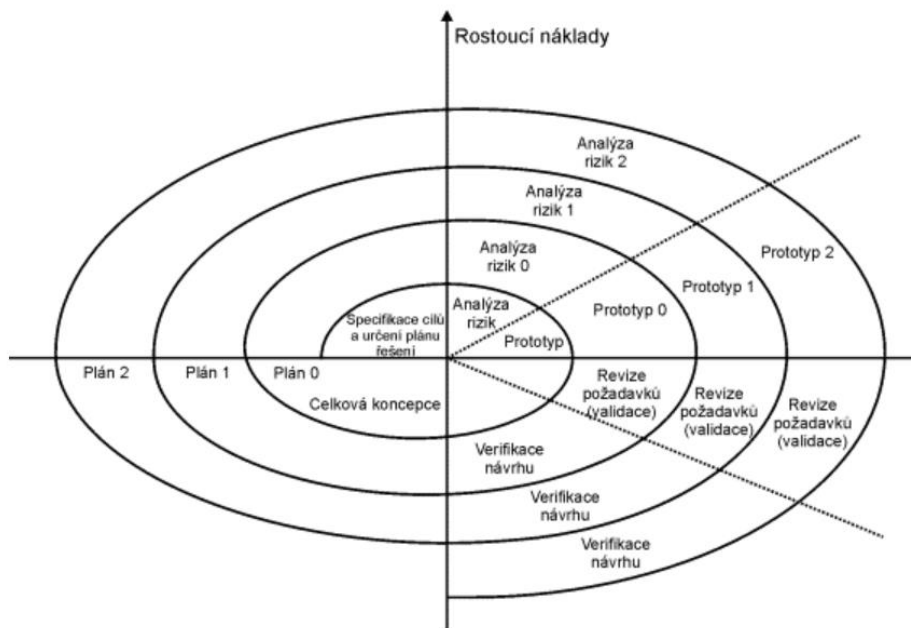


Obrázek 6: Prototypový model (Zdroj: Šmíd, 2005)

Hlavní nevýhodou prototypového modelu je skutečnost, že se málokdy podaří otestovat vlastnosti systému, které se projevují až při plném provozu (tzn. s plným objemem reálných dat a při plné interakci s uživateli). Není také ideální používat prototyp, který je určen pouze pro specifikace funkcí, jako cílové řešení. Porušení tohoto pravidla obvykle nevede k dobrým koncům (Král, 1998).

Spirálový model

Spirálový model, navržený B.W. Boehmem v roce 1988, je kombinací prototypového přístupu a analýzy rizik. Jak lze vidět na obrázku č. 7., Boehmův model má tvar spirály. Každá smyčka ve spirále představuje jeden krok procesu vývoje systému. Náklady a čas potřebné na realizaci vývoje jednotlivých kroků nebo celého systému jsou patrné z modelu, neboť úhlová dimenze udává časovou náročnost a radiální úroveň celkové náklady (Sommerville, 2013, Šmíd, 2005).



Obrázek 7: Spirálový model (Zdroj: Šmíd, 2005)

Dle Šmída (2005) je postup vývoje v jednotlivých krocích shodný s vodopádovým modelem. Každý krok (smyčka) se skládá z těchto částí:

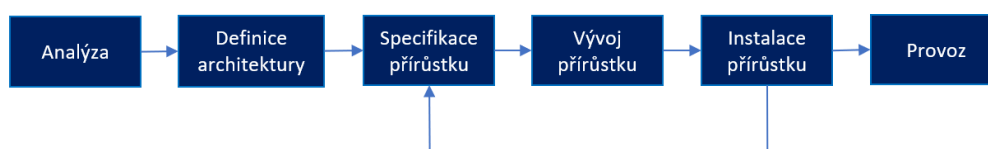
- specifikace cílů a určení plánu řešení,
- vyhodnocení alternativ řešení a analýza rizik,
- vývoj prototypu dané úrovně, jeho předvedení a vyhodnocení,
- revize požadavků,
- verifikace požadavků.

Důležitým rozdílem mezi spirálovým modelem a ostatními modely je využití analýzy rizik v procesu vývoje informačního systému, díky níž je možné vyhodnotit potenciální rizika a předejít tak zásadním chybám (Sommerville, 2013). Výhodou modelu je možnost specifikovat požadavky v jednotlivých krocích vývoje a modifikovat tak systém podle těchto požadavků. Naopak stinnou stránkou modelu může být to, že proces vývoje vyžaduje neustálou spolupráci zákazníka. Model tedy není vhodný pro systémy vyvíjené

na zakázku bez účasti budoucích uživatelů. Tento model celkově vyžaduje větší pracnost, proto je vhodný zejména pro vývoj velkých systémů (Král, 1998, Šmíd, 2005).

Inkrementální model

Inkrementální neboli přírůstkový model je založen na myšlence, že celý systém je na hrubé úrovni specifikován a navržen na začátku, ale následně se systém rozdělí na samostatně realizované části tzv. přírůstky. Každý přírůstek prochází všemi fázemi procesu vývoje samostatně. Při aplikaci inkrementálního modelu je klíčová otázka správného vymezení přírůstků (Požár, 2010, Bruckner et al., 2012).



Obrázek 8: Inkrementální model (Zdroj: Požár, 2010)

Výhodou tohoto modelu je existence rychlé zpětné vazby od zákazníka po dodání každého nového přírůstku. Naopak nevýhodou může být obtížné sledování a kontrola postupu prací na vývoji systému (Požár, 2010).

3.4 Metodiky vývoje informačních systémů

Metodiky vývoje informačních systémů se zabývají globálními aspekty vývoje informačních systémů. Definují jednotlivé etapy vývoje informačního systému a vztahy mezi nimi. Zjednodušeně lze uvést, že metodika nám říká „jaké“ kroky činit v jakém pořadí, „jak“ je provádět a především „proč“ to tak má být (Chlapek et al., 2011).

V současné době jsou při vývoji informačních systémů v praxi nejčastěji využívány rigorózní a agilní metodiky (Buchalceková, 2005).

3.4.1 Rigorózní metodiky

Rigorózní metodiky bývají podrobné a formální, jsou založené na přesvědčení, že procesy při budování IS lze popsat, plánovat, řídit a měřit. Velice přesně definují jednotlivé procesy, činnosti a vytvářené produkty, proto často bývají velmi robustní. Rigorózní metodiky bývají typicky založené na vodopádovém modelu životního cyklu. Mohou být ale také postaveny na iterativním nebo inkrementálním modelu životního cyklu. Mezi tyto metodiky lze typicky zařadit metodiku OPEN nebo metodiku Unified Process (UP) či Rational Unified Process (RUP) (Bruckner et al., 2012, Buchalceková, 2005).

OPEN (Object-oriented Process, Environment and Notation)

Metodika OPEN je veřejně přístupná metodika poskytující podporu pro celý životní cyklus softwarové aplikace. Je zaměřená především na vývoj objektově orientovaných a komponentových aplikací. Definuje procesní rámec tzv. OPEN Process Framework (OPF), což je procesní metamodel, ze kterého mohou být generovány instance specifické pro organizaci. Využívá jazyk UML (Unified Modeling Language) nebo OML (OPEN Modeling Language) (Buchalceková, 2005).

UP (Unified Process)

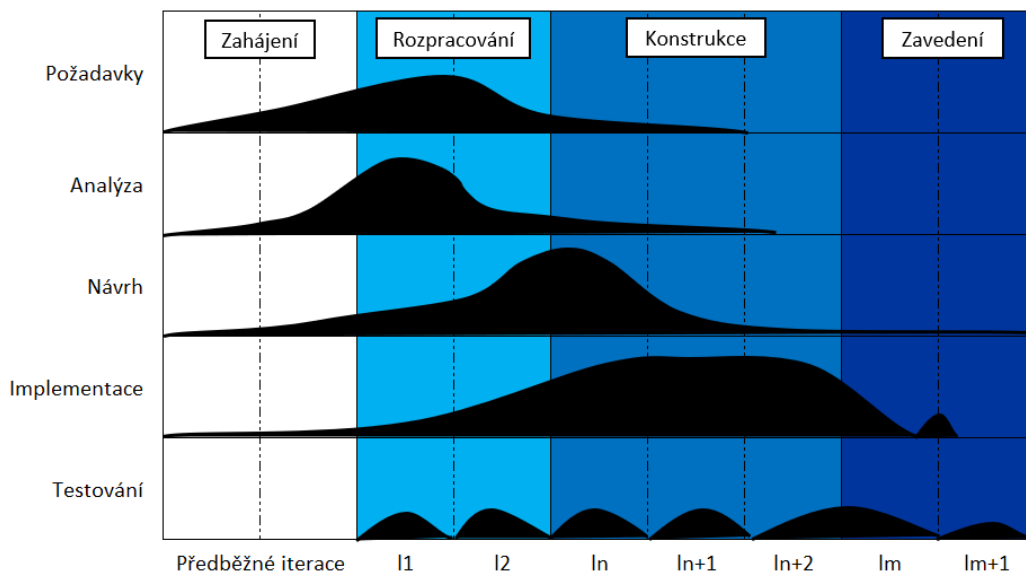
Metodika UP je obecnou metodikou tvorby softwaru. Je založena na iterativním a inkrementálním (přírůstkovém) přístupu. Projekt je tedy rozložen na řadu menších projektů (iterací). Každá iterace generuje vlastní základní linii (baseline), rozdíl mezi těmito liniemi je pak označován jako přírůstek (inkrement).

Dle Arlow a Neustadt (2007) se každá iterace skládá z pěti základních pracovních postupů (workflow), jsou jimi:

- požadavky – zachycují funkcionality systému,
- analýza – upřesnění požadavků a jejich strukturování,
- návrh – realizace požadavků v architektuře systému,

- implementace – tvorba softwaru,
- testování – ověření fungování implementace.

Životní cyklus projektu je v rámci metodiky UP rozdělen na čtyři fáze – zahájení, rozpracování, konstrukce a zavedení. Konec každé fáze představuje milník projektu, jež je definován podmínkami splnění a představuje indikátor pokroku v projektu. Každá fáze může obsahovat jednu nebo více iterací. V závislosti na konkrétní fázi projektu se také liší objem práce vykonané v jednotlivých pracovních postupech, což je patrné z obrázku č. 9 (Arlow, Neustadt, 2007).



Obrázek 9: Pracovní postupy v rámci metody UP (Zdroj: Arlow, Neustadt, 2007)

RUP (Rational Unified Process)

Metodika RUP je komerční verzí metodiky Unified Process. Lze říct, že metodika UP je otevřeným standardem. Naproti tomu RUP je specifickou komerční podtřídou, která určité vlastnosti a funkce UP překrývá nebo rozšiřuje. Největší rozdíl mezi oběma metodikami spočívá především v úplnosti a v jednotlivých detailech. Naopak pracovní postupy, sémantika a ideový obsah obou metodik jsou velmi podobné (Arlow, Neustadt, 2007).

3.4.2 Agilní metodiky

Díky změnám technologií, ekonomického prostředí a především díky tlaku na rychlost zavedení IS vznikla potřeba zásadní změny v metodikách. Tradiční rigorózní metodiky přestávají stačit požadavkům zákazníků a s jejich používáním se pojí řada problémů vedoucích často k překročení rozpočtu a harmonogramu projektu či k dodání řešení, které nevyhovuje uživatelům. V důsledku toho se začínají prosazovat metodiky, které umožňují vytvořit řešení velmi rychle a pružně jej přizpůsobit měnícím se požadavkům. Tyto metodiky se označují jako agilní (Buchalceová, 2005, Chlapek et al., 2011).

Dle Chlapek et al. (2011) v roce 2001 skupina firem podporujících tyto metodiky sepsala tzv. „agilní manifest“, v němž definovala hodnoty, na nichž je třeba postavit postupy tvorby informačních systémů a které upřednostňují:

- individualitu a komunikaci před procesy a nástroji,
- provozuschopnost software před obsažnou dokumentací,
- spolupráci se zákazníkem před sjednáváním kontraktu,
- reakci na změnu před plněním plánu.

Z toho vyplývá, že agilní metodiky kladou důraz především na komunikaci a spolupráci mezi lidmi a předpokládají změny v průběhu vývoje IS, ať už se jedná o změny priorit zákazníka nebo změny technologií. Naopak upozadují význam přesného plánu a jeho plnění či vypracování podrobné dokumentace (Chlapek et al., 2011).

Scrum

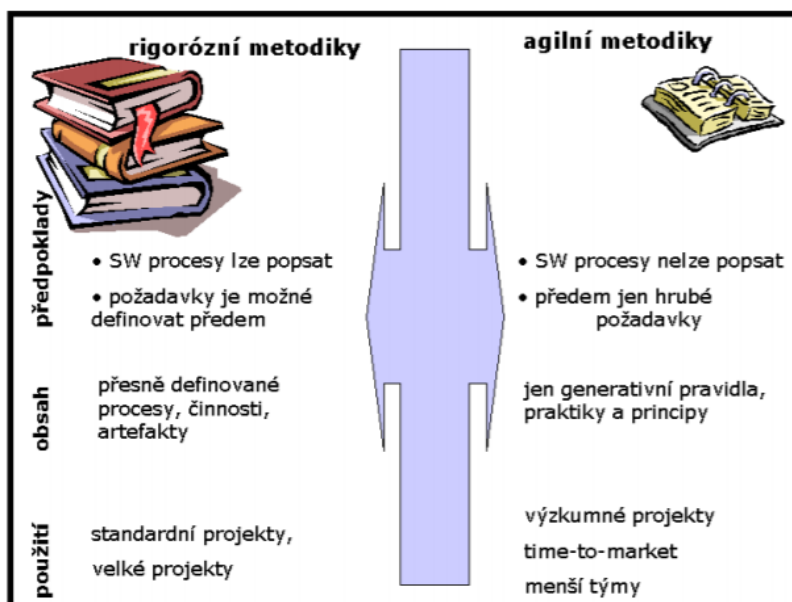
Metodika Scrum je nejznámější a nejpoužívanější agilní metodikou. Podstatou této metodiky je, že vývoj IS probíhá v iteracích nazývaných Sprint. Na konci každého Sprintu je dodána vybraná množina užitečných vlastností. Sprint zpravidla trvá 2 až 4 týdny. V rámci Sprintu má každý člen týmu přidělený úkol a účastní se každodenních patnáctiminutových schůzek tzv. Daily Scrum Meetings, jejichž podstatou je monitorovat stav projektu a identifikovat problémy a překážky ohrožující úspěch projektu (Bruckner et al., 2012).

Extrémní programování (XP)

Extrémní programování je metodikou, poprvé popsanou K. Beckem v roce 1999, jejíž využití je vhodné zejména pro vývoj IS, jehož zadání není jasné, anebo se mění. Tato metodika nezavádí převratné změny do stávajících metodik a technik, vychází z běžně používaných principů a postupů při vývoji IS, ale dovádí je však do extrémů. V důsledku toho zavádí do vývoje IS specifické praktiky, např. párové programování, refaktorizaci, testy před kódováním aj. Zároveň klade velký důraz na kvalitu technického řešení. Tato metodika je vhodná zejména pro malé až středně velké týmy (Buchalceová, 2005, Vymětal, 2009)

3.4.3 Porovnání agilních a rigorózních metodik

Rigorózní a agilní metodiky představují dvě skupiny metodik, které vycházejí z odlišných předpokladů a odlišného pohledu na vývoj software (Buchalceová, 2005). Podrobné srovnání obou metodik z různých hledisek lze vidět na obrázku č. 10.



Obrázek 10: Srovnání agilních a rigorózních metodik (Zdroj: Buchalceová, 2005)

Rigorózní metodiky, které vycházejí z předpokladu, že tvorba IS je definovaný proces, který lze přesně popsat a následně realizovat a zlepšovat, bývají často kritizovány kvůli

rozsáhlé nepotřebné dokumentaci, zbytečnému kreslení diagramů a oddalování vlastního programování (Merunka, 2006, Bruckner et al. 2012).

Agilní metodiky, vycházející z přesvědčení, že proces tvorby IS je empirický proces, naopak tyto činnosti vůbec neprovádějí a soustředí se výhradně na vlastní tvorbu softwaru (Merunka, 2006, Bruckner et al. 2012). Oba z těchto přístupů k vývoji IS mají své odpůrce a zastánce.

Merunka (2006) obhajuje použití rigorózní metodik. *„Pokud se podíváme na jiné oblasti inženýrských aktivit (stavebnictví, strojírenství, ...), tak uvidíme, že se všude plánuje, analyzuje, dokumentuje, prototypuje, testuje atd. Proč by tedy měla být oblast tvorby softwaru výjimečná?“* Nicméně připouští, že v některých případech vede k lepším výsledkům použití agilního přístupu.

Buchalcevová (2005) uvádí, že řešením tohoto problému může být kombinace obou přístupů. Rigorózní metodiky je možné odlehčit a aplikovat v rámci nich některé z agilních přístupů. Naopak, pokud se agilní přístup používá při vývoji větších projektů, je třeba více formalizovat a dokumentovat.

3.5 Analýza požadavků na informační systém

Analýza požadavků je proces, který slouží k popisu aktivit zapojených do zjišťování, dokumentování a údržby množiny požadavků na systém. Požadavky jsou základem všech systémů a v podstatě jsou vyjádřením toho, co by měl daný systém dělat, avšak nikoliv toho, jak by to měl systém dělat (Arlow, Neudstadt, 2007).

Chlapek et al. (2011) zdůrazňuje důležitost analýzy požadavků a uvádí, že výstup analýzy požadavků, tzv. katalog požadavků, je jedním ze základních dokumentů využívaných ve všech etapách vývoje IS. Je důležitý nejen pro analýzu a návrh IS, ale také pro tvorbu testovacích scénářů akceptačních testů. Chlapek et al. (2011) zároveň upozorňuje na chyby, které vznikají při nedůsledné analýze požadavků, jejichž oprava může být v pozdějších fázích vývoje IS velmi nákladná.

Dle Wiegerse (2008) by měl každý požadavek vykazovat následující vlastnosti – úplnost, správnost, proveditelnost, jednoznačnost, prioritu a ověřitelnost.

3.5.1 Sběr požadavků

Sběr požadavků je zajišťován zpravidla osobami, které jsou zapojeny do užívání a vývoje systému. Mezi tyto zainteresované osoby lze zařadit např. přímé uživatele, zákazníky, analytiky, vývojáře, vrcholové manažery, právníky apod. (Arlow, Neustadt, 2007).

V odborné literatuře je popsáno poměrně velké množství způsobů zjišťování požadavků. Arlow a Neustadt (2007) popisují jako jeden z nejefektivnějších způsobů zachycování požadavků tzv. dílnu požadavků. Jedná se o efektivní techniku zjišťování informací, která je zaměřená především na hledání nových nápadů. Tato technika spočívá v diskusi mezi zainteresovanými stranami, inženýrem požadavků a experty v oboru. Výsledkem diskuse by měly být nové nápady a nejdůležitější požadavky na systém, které jsou zaznamenány a později analyzovány.

Dalším možným způsobem zjišťování požadavků je konzultace (pohovor) se zainteresovanými osobami, která je vedena s každou jednotlivou osobou zvlášť. Užitečným doplňkem ke konzultacím mohou být dotazníky. Prostřednictvím dotazníků lze jednoduše ověřit správnost pochopení požadavků, které byly získány během osobních pohovorů. Výhodou dotazníků je především to, že je lze použít pro větší počet zainteresovaných osob a zároveň nejsou oproti pohovorům tak časově náročné (Arlow, Neustadt, 2007).

Sommerville (2013) dále uvádí, že jako zdroj pro požadavky lze využít také dokumentaci a specifikaci podobných systémů nebo systémů zainteresovaných stran.

3.5.2 Typy požadavků

Pro účely této práce budou rozlišovány dva druhy požadavků – funkční požadavky, které určují, jaké chování bude systém nabízet, a nefunkční požadavky, jež specifikují vlastnosti nebo omezující podmínky daného systému. Kromě dělení požadavků na funkční

a nefunkční bývá vhodné požadavky dále kategorizovat, a tím základní taxonomii rozšiřovat (Arlow, Neudstadt, 2007).

3.6 Přístupy k analýze informačních systémů

V průběhu času se v oblasti analýzy a návrhu IS objevily dvě různé metodologie – strukturovaný přístup a objektivně orientovaný přístup. Oba přístupy jsou založené na principu dekompozice, jinak mají ale odlišné charakteristiky.

3.6.1 Strukturovaný přístup

Pojem strukturovaný přístup jako první formuloval v 80. letech Edward Yourdon v publikaci *Modern Structured Analysis*. Jak už název napovídá, strukturovaný přístup je založený na strukturalizaci, která je základní pracovní metodou a prostředkem boje proti složitosti, která je pro vývoj IS typická (Řepa, 1999).

Strukturovaný přístup je charakteristický tím, že pomocí techniky dekompozice člení projekt na malé, dobře definované aktivity a určuje posloupnost a interakci těchto aktivit. Při strukturované analýze je systém zkoumán ze dvou základních pohledů, a to z pohledu zaměřeného na funkcionalitu systému a pohledu zaměřeného na datovou strukturu. Kombinace těchto přístupů umožňuje nahlížet na systém z různých úhlů (Ráček, 2006).

Dle Yourdona (1989) a jeho metodiky YMSA se rozlišují dva základní modely systému.

1. Esenciální model – vyjadřuje podstatu systému, skládá se ze dvou částí, kterými jsou:

- model okolí systému – reprezentuje vnější pohled na systém a popisuje hlavně rozhraní systému a události ve vnějším světě,
- model chování systému – soustředí se na vnitřní uspořádání systému a jeho vlastnosti.

2. Implementační model

Všechny tyto modely popisují systém z různých úhlů pohledu a se specifickou abstrakcí. Zároveň se z části překrývají, neboť popisují stejnou realitu. Aby byla zajištěna integrita

popisu jako celku, je nezbytné explicitně definovat, v čem se modely doplňují a co mají společné (Řepa, 1999).

3.6.2 Objektově orientovaný přístup

Vymětal (2009) uvádí tuto definici OOP: „*Objektivně orientovaný přístup, rozvíjený od 80. let minulého století, je založen na objektech. Objekt je určitý prvek systému, který vykazuje vlastní chování (reakci na vstupní impulsy – metody) a má určité vlastnosti (atributy). Objekty, které mají stejné chování a obdobné či stejné vlastnosti, tvoří třídu objektů.*“

Page – Jones (2001) dále uvádí, že objekty jsou obrazem reálného světa. Objekt je instancí (výskytem) třídy, která představuje základní konstrukční prvek OOP. Objekty odvozené ze stejné třídy sdílí stejnou strukturu a chování.

Coad (1991) a Page-Jones (2001) definovali hlavní koncepty OOP, jejichž popisem se budou zabývat následující podkapitoly.

Abstrakce

Podstatou abstrakce je ignorování irelevantních vlastností a operací objektu a soustředění se na ty, které relevantní jsou. Abstrakce odděluje důležité informace od nepodstatných (Coad, 1991).

Zapouzdření

Zapouzdřením se v OOP dle Page-Jonese (2001) rozumí zabalení operací a atributů, představujících nějaký stav, do jednoho typu objektu tak, že každá změna interně uložených informací objektu nebo přístup k nim se uskutečňuje prostřednictvím operací objektu.

Dědičnost

Jak už bylo zmíněno, objekt je instancí své třídy, a tudíž přebírá všechny vlastnosti třídy. Tento stav se nazývá dědičnost (Schmuller, 2001). Třídy mohou tvořit hierarchie

dědičnosti s nadřizenými a podřizenými třídami. Dědičnost umožňuje objektům jedné třídy používat prvky její libovolné nadřizené třídy (Page-Jones, 2001).

Polymorfismus

Polymorfismus nebo také mnohotvárnost je vlastnost, jejímž prostřednictvím se může operace chovat odlišně v různých třídách a v různých okamžicích. Ačkoliv má operace u různých tříd stejné jméno, může provádět pokaždé jinou operaci. Specifická implementace operace určitou třídou se nazývá metoda (Vrana, 2008, Schmuller, 2001).

Mezi objekty existují v OOP různé vztahy, které jsou podrobněji popsány v kapitole 3.8 *UML (Unified Modeling Language)*.

3.6.3 Porovnání strukturovaného a objektivně orientovaného přístupu

O vztahu těchto dvou přístupů se vedou diskuse. Část odborné veřejnosti tvrdí, že objektově orientované metody vytlačí metody strukturované. Vymětal (2009) uvádí, že strukturované metody narážejí na hranice svých možností, neboť vyžadují přesnou definici uvedených návazností v celém projektu, což vede k nekončícím změnám a špatné udržitelnosti dokumentace projektů. S tímto názorem nesouhlasí Král (1998), který uvádí, že existují oblasti, kde je strukturovaný pohled vhodnější nebo kde je vhodné kombinovat oba přístupy dle potřeby.

3.7 UML (Unified Modeling Language)

„Sjednocený modelovací jazyk (UML) je druh grafické notace podporovaný nezávislým meta-modelem, který umožňuje popisovat a navrhovat softwarové systémy, konkrétně systémy budované využitím objektově orientované (OO) metodiky.“ (Fowler, 2009)

Spolu s vývojem stále více složitějších a komplexnějších počítačových systémů vznikla potřeba kvalitních a spolehlivých návrhů. Programovací jazyky pro tyto účely neposkytují dostatečnou úroveň abstrakce, a navíc nejsou srozumitelné pro každého. Tak vznikla notace návrhu, následně přijatá jako standard. Touto notací je jazyk UML (Schmuller, 2001).

UML je relativně volným standardem řízeným uskupením Object Management Group (OMG), jež je volným konsorciem společností. OMG vzniklo za účelem vytváření standardů, které by podporovaly vzájemnou slučitelnost objektově orientovaných systémů (Fowler, 2009).

3.7.1 Historie UML

Do poloviny 90. let minulého století existovalo několik soupeřících jazyků a metodik pro vizuální modelování. Mezi nejúspěšnější z nich patřily metody Booch (Gradyho Booche), OMT (Jamese Rambaugh) a metodika Objectory (Ivara Jacobsona) (Arlow, Neustadt, 2007).

V roce 1994 nastoupil Rambaugh do společnosti Rational Software, kde již pracoval Booch. Jacobson se k nim přidal v roce 1995. Softwarovým průmyslem následně začaly kolovat první návrhy jazyka UML. Mnoho společností tušilo, že by jim jazyk UML mohl posloužit pro strategické účely, a tak vzniklo konsorcium jazyka UML. V roce 1997 poskytlo konsorcium verzi 1.0 jazyka UML skupině OMG (Object Management Group), a tak vznikl první průmyslový standard objektově orientovaného jazyka pro vizuální modelování. Jazyk UML se v podstatě stal standardem v softwarovém průmyslu a dále se vyvíjí (Arlow, Neustadt, 2007, Schmuller, 2001)

3.7.2 Způsoby užití UML

V současné době existují odlišné názory na to, jakým způsobem by mělo být UML ve vývoji softwaru používáno. Tři základní charakteristiky způsobu užití definovali nezávisle na sobě Mellor (2004) a Fowler (2009), kteří uvádějí, že UML je dnes zdaleka nejčastěji využíváno pro tvorbu náčrtků, méně často pak pro tvorbu podrobných návrhů. Rozdíl mezi náčrtem a návrhem spočívá ve skutečnosti, že náčrtky jsou záměrně nekompletní a zdůrazňují jen důležité informace o některých aspektech systému. Naproti tomu hlavní význam návrhu spočívá v jeho kompletnosti a zaměření na detaily. Náčrtky i návrhy lze využít v rámci dopředného i zpětného inženýrství. V rámci dopředného inženýrství je vytvářen kód na základě UML diagramu, zatímco zpětné inženýrství vytváří UML diagram na základě existujícího kódu. Třetí charakteristikou je způsob užití UML jako

programovacího jazyka. Podstata spočívá ve vytvoření UML diagramů, které jsou překládány přímo do spustitelného kódu. V tomto případě nemá dopředné ani zpětné inženýrství smysl, neboť UML a zdrojový kód jsou jedno a totéž.

3.7.3 Základní charakteristiky a struktura UML

Modelovací jazyk UML je souhrnem především grafických notací k vyjádření analytických a návrhových modelů. Žádný diagram UML nezachycuje navrhovaný systém jako celek, ale soustředí se vždy právě na jeden pohled na vyvíjený systém (Kanisová, Müller, 2004).

Dále je potřeba zmínit, že jazyk UML nenabízí žádný druh metody modelování, ale představuje pouze vizuální syntaxi, která se využívá při sestavování modelů (Fowler, 2009).

Funkci jazyka UML jako jazyka vizuálního lze nejlépe chápat z jeho struktury. Strukturu jazyka UML dle Booch et al. (1999) tvoří tři součásti:

- stavební bloky,
- společné mechanismy,
- architektura.

V následujících podkapitolách budou podrobněji popsány jednotlivé součásti jazyka UML.

Stavební bloky

Jazyk UML se dle Booch et al. (1999) skládá ze tří stavebních bloků:

- z předmětů (things),
- vztahů (relationships),
- diagramů (diagrams).

Předměty

Předměty neboli „věci“ či abstrakce jsou základními prvky modelu. V jazyku UML se dle Booch et al. (1999) dělí na:

- strukturální abstrakce (structural things) – jsou definovány jako podstatná jména modelu UML (např. třídy, rozhraní, spolupráce, případ užití, komponenta, uzel),
- chování (behavioural things) – je definováno jako slovesa modelu (např. interakce, stav),
- seskupení (grouping things) – jsou balíčky používané k seskupování významově souvisejících prvků do soudržných jednotek,
- poznámky (annotational things) – jedná se o komentáře, které lze využít k popisu či osvětlení jakéhokoliv prvku modelu.

Vztahy

Vztahy neboli relace jsou pojátkem mezi dvěma či více předměty, určují jejich vzájemnou souvislost. V UML existují čtyři typy vztahů: závislost, asociace, generalizace a realizace. Tyto vztahy jsou základními relačními stavebními bloky UML (Booch et al., 1999).

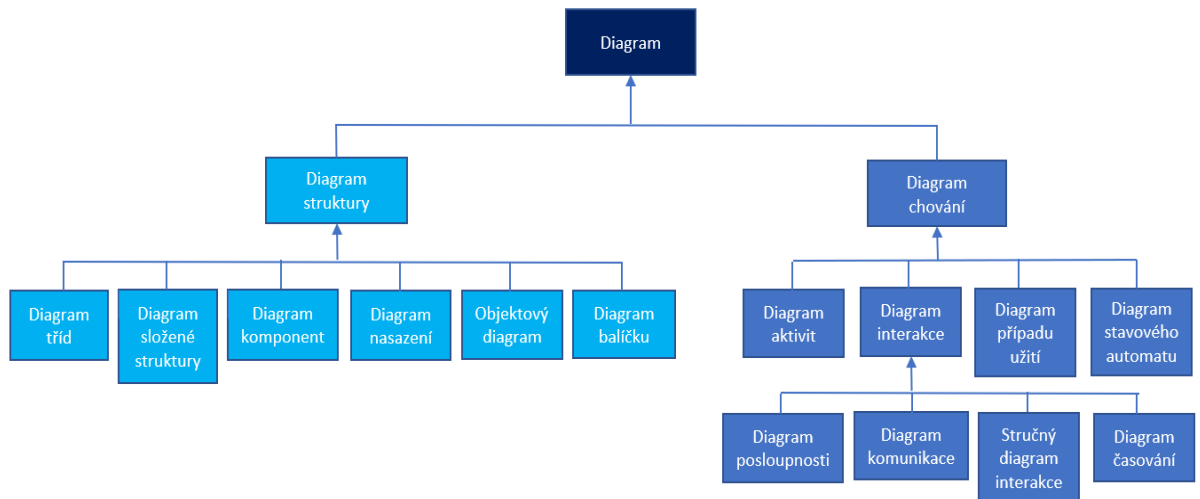
- Závislost – je relace mezi dvěma prvky, v níž se změna jednoho prvku promítá do druhého prvku. Znamená to, že prvky jsou na sobě do určité míry závislé, ale relace zároveň není fakticky asociací. Závislost se používá při modelování relací mezi třídami. Vyskytuje se ale rovněž také mezi balíčky a balíčky, objekty a třídami, a dokonce mezi operací a třídou. UML 2 definuje tři typy závislostí – užití (usage), abstrakci (abstraction) a oprávnění (permission). Graficky se závislost zobrazuje jako nevyzdobená tečkovaná šipka. Typ závislosti je často zřejmý již z kontextu, proto obvykle není potřeba užít stereotyp (Arlow, Neustadt, 2007).
- Asociace – popisují relace mezi jednou nebo více třídami, které jsou abstrakcí množiny spojení mezi instancemi (objekty) těchto tříd. Asociace jsou definovány svými vlastnostmi, kterými jsou název asociace, název role, násobnost

a průchodnost (Kanisová, Müller, 2004). Vylepšenou formou asociace je agregace a kompozice. Agregace je vztah typu „součást-celek“. Vazba typu agregace říká, že jedna třída je částí druhé třídy. Symbolem agregace je čára s diamantem na straně „celku“. Speciálním případem agregace je kompozice. Jedná se v podstatě o silnější formu agregace, která má více omezení. Podřízený objekt např. nemůže existovat samostatně bez nadřízeného objektu. Grafická značka kompozice je plný diamant (Kanisová, Müller, 2004, Vrana, 2008).

- Generalizace – neboli zobecnění je klíčový koncept objektově orientované analýzy a návrhu. Podstatou generalizace je relace mezi obecnou objektovou třídou (tzv. super class nebo parent) a více specifikovanou další objektovou třídou (subclass nebo child), která následuje v hierarchii dědičnosti na nižší úroveň. Podřízené objektové třídy dědí ze svého předka všechny vlastnosti, tj. atributy, operace, relace a omezení, a navíc mohou ke zděděnému fondu přidat novou charakteristiku nebo naopak mohou některé operace předchůdců předefinovat (Kanisová, Müller, 2004, Arlow, Neustadt, 2007). Graficky je vztah generalizace zobrazen jako plná čára s dutou šipkou směřující k nadřízené třídě (Booch et al., 1999).
- Realizace – je sémantický vztah mezi klasifikátory, kde jeden klasifikátor určuje dohodu, kterou uskutečňuje druhý klasifikátor. Nejčastěji se realizace používá k určení vztahu mezi rozhraním a třídou či komponentou, které poskytují operaci nebo službu. Rozhraní může být realizováno mnoha takovými třídami nebo komponentami, a naopak třída nebo komponenta může realizovat mnoho rozhraní. Realizaci lze také použít k určení vztahu mezi případem užití a Use Case, který tento případ užití realizuje. Významově je realizace jakousi kombinací mezi generalizací a závislostí, také grafická notace realizace je kombinací notací těchto dvou vztahů – vykresluje se jako tečkovaná čára s šipkou směřující ke klasifikátoru, který specifikuje dohodu (Booch et al., 1999).

Diagramy

Diagramy jsou grafickým znázorněním prvků modelu a jejich vztahů, vizualizují systém z různých perspektiv. Je důležité zdůraznit, že diagramy poskytují pouze pohledy na model, nejsou ale modelem samotným (Booch et al., 1999). V jazyce UML existuje třináct typů různých diagramů, které lze vidět na obrázku č. 11.



Obrázek 11: Diagramy UML (Zdroj: Arlow, Neustadt, 2007)

Zmíněné diagramy lze rozdělit do dvou kategorií – na diagramy struktury, které modelují statickou strukturu systému (tzv. statický model), a na diagramy chování, modelující dynamickou strukturu systému (dynamický model). Statický model zobrazuje předměty a strukturní asociace mezi nimi. Dynamický model naopak zachycuje způsob, jímž na sebe jednotlivé předměty navzájem působí, aby bylo dosaženo požadovaného chování systému (Arlow, Neustadt, 2007). Vrana (2008) definuje navíc model interakcí, který zobrazuje spolupráci objektů v systému pomocí diagramu případů užití, sekvenčního diagramu a diagramu aktivit.

Toto uspořádání diagramů není ale vyloženě neměnné. V UML lze najít mnoho způsobů, jak diagramy uspořádat a dále rozlišovat (Schmuller, 2001).

Společné mechanismy

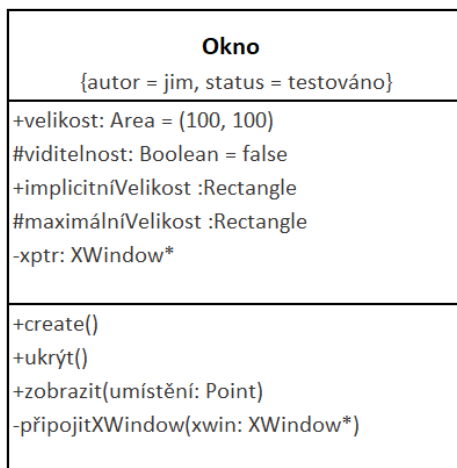
Jazyk UML obsahuje čtyři společné mechanismy, které platí konzistentně v celém jazyku. Popisují čtyři základní strategie cesty k modelování objektů.

Specifikace

Modely UML mají alespoň dva rozměry – grafický, který umožňuje vizualizovat model prostřednictvím diagramů a symbolů, a textový, jenž se skládá ze specifikací různých prvků modelu. Specifikace jsou textovým popisem sémantiky jednotlivých prvků. Jsou také jádrem a podstatou modelu UML, utváří sémantický podklad, který udržuje celý model pohromadě a dává mu smysl (Arlow, Neustadt, 2007, Booch et al., 1999).

Ornamenty

Každý prvek modelu UML je vyjádřen velmi jednoduchým symbolem. Je-li potřeba prostřednictvím diagramu zobrazit více informací za účelem zdůraznění či zvýraznění určitých důležitých detailů nebo za účelem zlepšení srozumitelnosti a čitelnosti diagramu, lze prvek obohatit řadou ornamentů (Arlow, Neustadt, 2007, Booch et al., 1999). Takto obohacený prvek lze vidět na obrázku č. 12.



Obrázek 12: Obohacený prvek (Zdroj: Arlow, Neustadt, 2007)

Podskupiny

Podskupiny popisují dva druhy vidění světa. V jazyku UML se dle Booch et al. (1999) rozlišují dvě podskupiny:

- klasifikátorů a instancí – klasifikátorem označujeme abstraktní vyjádření typu předmětu, naproti tomu instance je konkrétním výskytem abstraktní představy,
- rozhraní a implementací – rozhraní definuje dohodu, která zaručuje, čím se budou jednotlivé implementace řídit, implementace představuje jednu konkrétní realizaci této dohody.

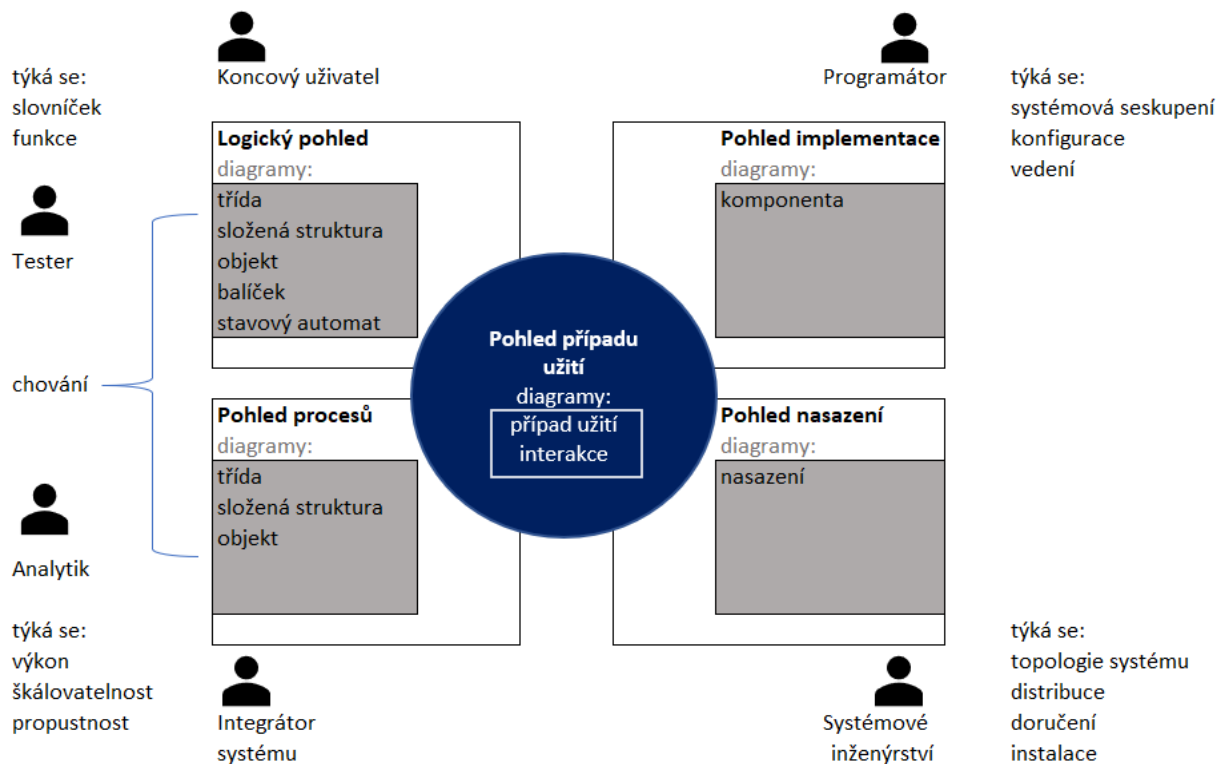
Mechanismy rozšiřitelnosti

Aby jazyk UML uspokojil potřeby všech uživatelů v současnosti i v budoucnu, zahrnuli autoři do jazyka tři jednoduché mechanismy, které umožňují jeho rozšiřitelnost.

- Omezení – specifikují určitou podmínku nebo pravidlo, týkající se prvku modelu, které musí být vyhodnoceno jako pravdivé. Omezující podmínka je v modelu vyznačena jako textový řetězec uzavřený do složených závorek (Arlow, Neustadt, 2007, Booch et al., 1999).
- Stereotypy – umožňují vytvářet nové prvky modelu založené na existujících prvcích. Nové prvky mají stejnou podobu (atributy a relace), ale používají se s jiným záměrem. Graficky se stereotyp v modelu zobrazuje jako symbol. Lze použít také název stereotypu uzavřený ve dvojitých lomených závorkách umístěný nad názvem prvku. Každý model může obsahovat nanejvýš jeden stereotyp (Ambler, 2005).
- Označené hodnoty – umožňují rozšiřovat vlastnosti prvků a vytvářet nové informace ve specifikaci daného prvku. Označené hodnoty jsou v modelu zobrazeny pod názvem prvku jako textový řetězec ve složených závorkách, přičemž řetězec obsahuje klíčové slovo s přidruženou hodnotou (např. {tag1=hodnota1}) (Arlow, Neustadt, 2007, Booch et al., 1999).

Architektura UML

Vizualizace, specifikace, konstrukce a dokumentace softwarově náročného systému vyžaduje nahlížet na systém z více různých perspektiv. Aby bylo možné reflektovat všechny podstatné aspekty systému, definuje jazyk UML pět různých pohledů na systém tzv. „4+1“ architekturu, kterou lze vidět na obrázku č. 13 (Arlow, Neustadt, 2007, Booch et al., 1999).



Obrázek 13: Architektura 4+1 (Zdroj: Arlow, Neustadt, 2007)

- Logický pohled – zachycuje slovník oblasti problému jako množinu tříd a objektů, důraz klade především na zobrazení způsobu, jakým třídy a objekty implementují chování systému. Logický pohled primárně podporuje funkční požadavky systému, tzn. služby poskytované uživatelům.
- Pohled procesů – modeluje spustitelná vlákna a procesy jako aktivní třídy, je to procesově orientovaná varianta logického pohledu. Tento pohled primárně řeší výkon, škálovatelnost a propustnost systému.

- Pohled implementace – zahrnuje soubory a komponenty, které se používají k vytvoření kódového základu systému. Tento pohled primárně řeší závislosti mezi jednotlivými komponentami systému a správu konfigurace množin vytvořených z těchto komponent.
- Pohled nasazení – modeluje fyzické nasazení komponent na množinu fyzických výpočetních uzlů, které tvoří hardwarovou topologii systému. Tento pohled zachycuje distribuci, dodávku a instalaci částí, které tvoří systém.
- Pohled případu užití – zachycuje základní požadavky kladené na příslušný systém jako na množinu případů užití. Všechny ostatní zmiňované pohledy jsou integrovány do pohledu případu užití (Arlow, Neustadt, 2007, Booch et al., 1999).

4 Vlastní práce

Vlastní část práce se zaměřuje na analýzu a návrh informačního systému pro podporu zprostředkování služeb v oblasti prodeje nemovitostí. Při analýze a návrhu autorka vychází částečně ze zkušeností, které získala na administrativní pozici v nejmenované realitní společnosti. Informační systém ve zmíněné společnosti nefungoval optimálně a měl řadu nedostatků, což bylo jedním z motivů k tvorbě této práce.

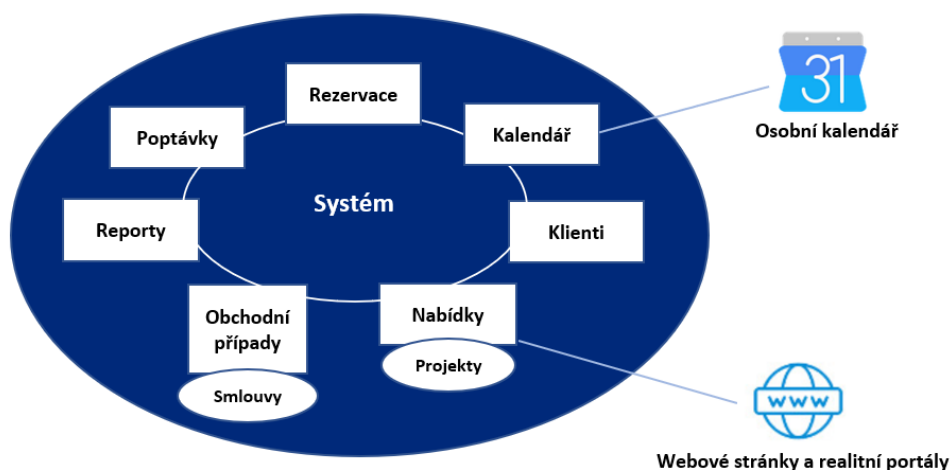
Jelikož je systém určen pro trh, a nikoliv pro jednoho konkrétního zákazníka, jsou v rámci předběžné analýzy prozkoumána na trhu dostupná konkurenční řešení za účelem ověření, zda bude systém konkurenceschopný a má tedy vůbec smysl ho navrhovat. V rámci předběžné analýzy jsou také připravovány otázky pro rozhovory s potenciálními uživateli systému.

Dalším krokem je analýza systému, která zahrnuje analýzu požadavků, v rámci které jsou zjištěny požadavky od potenciálních uživatelů systému. Následně jsou požadavky dále upřesňovány autorkou práce. Finální požadavky slouží jako podklad pro tvorbu modelu tříd, stavového modelu a modelu interakcí. Návrh je pro lepší představu o výsledné podobě systému doplněn o wireframes – modely uživatelského rozhraní. V závěru jsou formulována doporučení pro případnou implementaci systému.

4.1 Obecný popis systému

Navrhovaný systém efektivně podporuje celý proces prodeje nemovitostí, čímž snižuje administrativní zátěž kladenou na zaměstnance. V neposlední řadě zajišťuje bezpečnost uchovávaných dat. Díky přehledným reportům umožňuje také sledování firemních výsledků.

Navrhovaný informační systém je určen především pro realitní kanceláře, nikoliv pro developerské společnosti, které se sice také zabývají prodejem nemovitostí, ale mají velmi odlišné požadavky na informační systém.



Obrázek 14: Zjednodušené schéma systému

Na obrázku č. 14 je znázorněno zjednodušené analytické schéma navrhovaného systému. Jak je patrné, nedílnou součástí systému je správa klientů a správa poptávek po nemovitostech. V systému je možné evidovat a spravovat všechny nemovitosti, které má firma v nabídce. Spolu související nabídky je možné přiřazovat do projektů. Nabídky a projekty je možné publikovat na webové stránky realitní kanceláře nebo na realitní portály. Systém dále umožňuje správu obchodních případů, která zahrnuje evidenci plateb a kalkulaci provizí. Součástí systému je kalendář, kam se propisují termíny zadané v obchodních případech. Kalendář slouží také pro plánování schůzek uživatelů a je propojitelný s kalendáři třetích stran. Významnou funkcí systému je automatické generování smluv k obchodním případům z předem připravených šablon. Systém také umožňuje vytvářet různé druhy reportů.

4.2 Dostupná konkurenční řešení na trhu

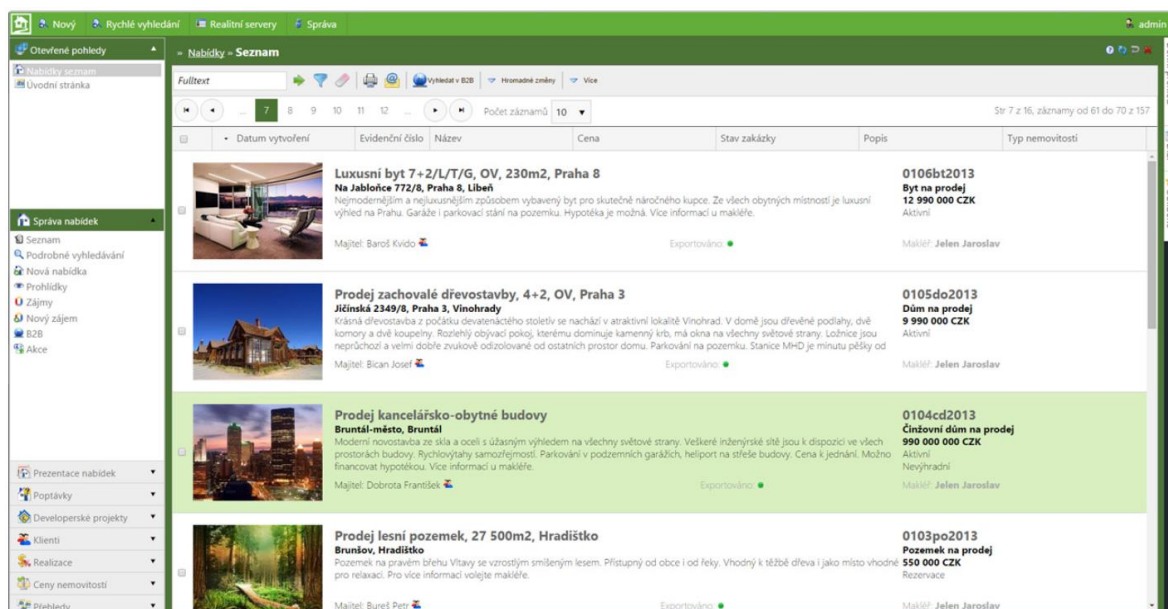
V této kapitole jsou popsána na trhu dostupná řešení určená pro podporu zprostředkování služeb v oblasti prodeje nemovitostí, cílená na realitní kanceláře. Nejdříve jsou vybraná řešení podrobně popsána a na konci kapitoly jsou porovnána v přehledné tabulce. Následně je na základě provedené vícekritériální analýzy variant vybráno nejlepší řešení.

Popsaná konkurenční řešení jsou dostupná ve více verzích lišících se funkcionalitou a cenou. Pro účely této práce se uvažují nejvyšší verze těchto řešení, nižší verze nejsou vzhledem k omezené funkcionalitě relevantní.

4.2.1 Realko

Realko je webová aplikace, která je určena pro malé i velké realitní kanceláře. Aplikace je optimalizována také pro mobilní zařízení (Realko, 2021).

V nejvyšší verzi aplikace umožňuje správu uživatelských účtů a nastavení oprávnění pro jednotlivé skupiny uživatelů. Aplikace dále umožňuje správu poptávek a klientů i jejich párování, správu nabídek a publikaci nabídek na realitní servery. Mezi další funkce aplikace patří zpracování reportů. Součástí aplikace je kalendář, který je možné propojit s aplikací Kalendář od Google. Kromě těchto funkcí nabízí aplikace další rozšiřující moduly, které ale nejsou součástí základní ceny. Za další poplatky je možné přikoupit modul pro správu projektů, modul topování a modul obchodních případů zahrnující i kalkulaci provizí. Dále je možné přikoupit také modul zajišťující publikaci nabídek na vlastní webové stránky (Realko, 2021).

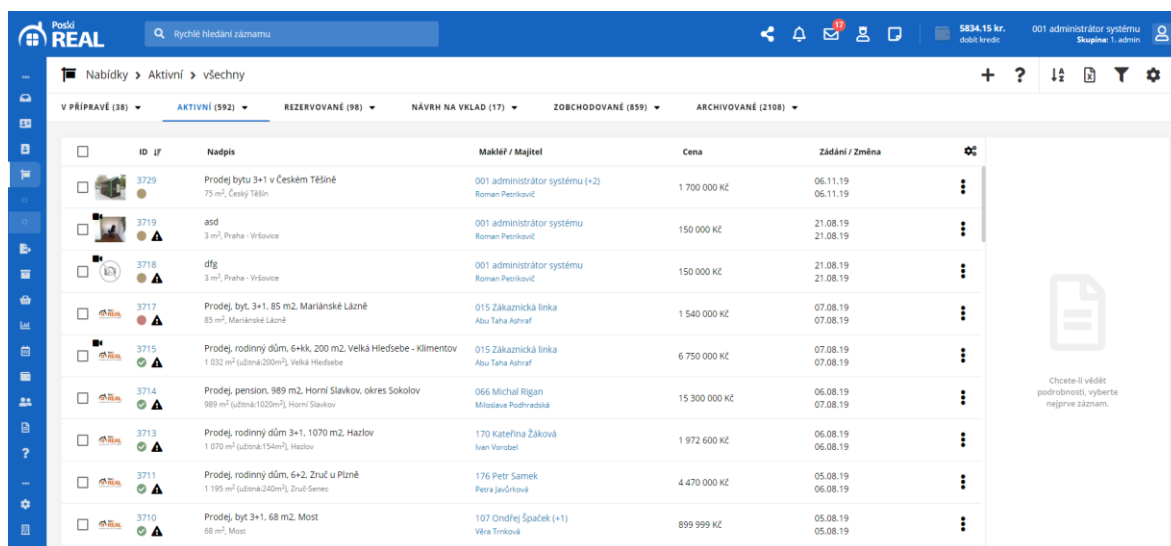


Obrázek 15: Náhled aplikace Realko (Zdroj: Realko, 2021)

4.2.2 Poski REAL

Poski REAL je stejně jako předchozí řešení webovou aplikací, která je optimalizována i pro mobilní zařízení. Aplikace je určena pro malé i velké realitní kanceláře (Poski REAL, 2021).

Nejvyšší verze aplikace obsahuje moduly pro správu uživatelských účtů, správu klientů, správu poptávek a modul pro práci s nabídkou a její export na webové stránky nebo realitní servery. Kromě toho je součástí nejvyšší verze také modul pro správu projektů, modul obchodních případů a modul financí, který zajišťuje sledování výdajů u obchodních případů i kalkulaci provizí. Dále aplikace nabízí také kalendář po plánování schůzek se synchronizací s aplikací Google Kalendář (Poski REAL, 2021).



The screenshot displays the Poski REAL web application interface. At the top, there is a navigation bar with the Poski REAL logo, a search bar, and various utility icons. Below the navigation bar, there are tabs for different offer statuses: 'V PŘÍPRAVĚ (38)', 'AKTIVNÍ (592)', 'REZERVOVANÉ (98)', 'NÁVRH NA VKLAD (17)', 'ZOBCHODOVANÉ (859)', and 'ARCHIVOVANÉ (2108)'. The main content area shows a table of active offers with columns for ID, title, agent/owner, price, and dates. A sidebar on the left contains navigation icons, and a right sidebar has a document icon and a note: 'Chcete-li vidět podrobnosti, vyberte nejprve záznam.'

ID	Nadpis	Makléř / Majitel	Cena	Zadání / Změna
3729	Prodej bytu 3+1 v Českém Těšíně 75 m², Český Těšín	001 administrátor systému (+2) Roman Perinčovič	1 700 000 Kč	06.11.19 06.11.19
3719	asd 3 m², Praha - Vršovice	001 administrátor systému Roman Perinčovič	150 000 Kč	21.08.19 21.08.19
3718	dfg 3 m², Praha - Vršovice	001 administrátor systému Roman Perinčovič	150 000 Kč	21.08.19 21.08.19
3717	Prodej byt. 3+1, 85 m², Mariánské Lázně 85 m², Mariánské Lázně	015 Zákaznická linka Abu Taha Ashraf	1 540 000 Kč	07.08.19 07.08.19
3715	Prodej, rodinný dům, 6+kk, 200 m², Velká Hledebe - Klimentov 1 032 m² (užtená:200m²), Velká Hledebe	015 Zákaznická linka Abu Taha Ashraf	6 750 000 Kč	07.08.19 07.08.19
3714	Prodej, pension, 989 m², Horní Slavkov, okres Sokolov 989 m² (užtená:1020m²), Horní Slavkov	066 Michal Rigan Milosava Podhradská	15 300 000 Kč	06.08.19 07.08.19
3713	Prodej, rodinný dům 3+1, 1070 m², Hazlov 1 070 m² (užtená:154m²), Hazlov	170 Kateřina Žáková Ivan Vorobeč	1 972 600 Kč	06.08.19 06.08.19
3711	Prodej, rodinný dům, 6+2, Zruč u Písně 1 195 m² (užtená:240m²), Zruč u Písně	176 Petr Samek Petra Javorková	4 470 000 Kč	05.08.19 06.08.19
3710	Prodej, byt 3+1, 68 m², Most 68 m², Most	107 Ondřej Špaček (+1) Věra Timková	899 999 Kč	05.08.19 05.08.19

Obrázek 16: Náhled aplikace Poski REAL (Zdroj: Poski REAL, 2021)

4.2.3 Softreal

Realitní program Softreal je webovou aplikací. Aplikace je rovněž optimalizována pro mobilní zařízení (Softreal, 2021).

Aplikace v nejvyšší verzi nabízí modul pro správu uživatelských účtů a databáze klientů, modul pro zpracování reportů a modul pro správu poptávek a nabídek. Aplikace dále umožňuje publikovat nabídky na realitní servery a webové stránky. Součástí aplikace je

také kalendář. Za další poplatky je možné přikoupit moduly, které nejsou základní součástí ani nejvyšší verze aplikace. Jedná se o modul pro správu developerských projektů a modul pro kalkulaci provizí a funkci sledování nákladů za inzerci u jednotlivých nabídek. (Softreal, 2021).

ID	Důležitost	Vytvořeno	Změněno	Makléř	Typ nemovitosti	Prodej / Pronájem	Titulek	Dispozice	Lokalita	Cena	Akce
N490	3	27.5.2013	28.7.2013	Zajímavé	Domy a vily	Prodej	Prodej domu v Nové Pace		Nová Páka, Lipová	2 500 000 Kč	Akce
N487	3	27.5.2013	27.5.2013	Zajímavé	Byty	Prodej	Prodej bytu 2+kk v Praze Hostalka	2+kk	Praha, Ptačická	3 500 000 Kč	Akce
N486	3	27.5.2013	27.5.2013	Zajímavé	Byty	Prodej	Prodej bytu 2+1 v Jihlavě	2+1	Jihlava, Nad Jihlavskou	1 630 750 Kč	Akce
N485	3	27.5.2013	27.5.2013	Zajímavé	Byty	Prodej	Prodej bytu 3+1 v Karlových Varech	3+kk	Karlovy Vary, Hedvičova	1 200 000 Kč	Akce
N484	3	27.5.2013	27.5.2013	Zajímavé	Byty	Prodej	Prodej bytu 1+1 na Praze 10	1+1	Praha, Ruzská	2 350 000 Kč	Akce
N483	3	27.5.2013	27.5.2013	Zajímavé	Byty	Prodej	Prodej bytu 2+1 v Pardubicích	2+1	Pardubice, Surovoty	1 150 000 Kč	Akce
N482	3	27.5.2013	27.5.2013	Zajímavé	Pozemky	Prodej	Zastřešený pozemek v Záhradě	zahrad	Záhrad	350 000 Kč	Akce
N481	3	27.5.2013	27.5.2013	Zajímavé	Domy a vily	Prodej	Bytový dům v Mělníku		Mělník, Husova	9 700 000 Kč	Akce
N480	3	27.5.2013	27.5.2013	Zajímavé	Byty	Prodej	Slunný byt na Praze 1	5+kk	Praha 1, Ve Smečkách	6 500 000 Kč	Akce
N479	3	27.5.2013	27.5.2013	Zajímavé	Byty	Prodej	Byt 6+1, Vysoká u Mělníka	6+1	Mělník	1 000 000 Kč	Akce
N471	3	5.5.2013	21.5.2013	Starší	Domy a vily	Prodej	Novostavba samostatného RD	7+kk	Křehovice, Slnovka	7 900 000 Kč	Akce
N461	3	26.10.2012	4.4.2013	Zajímavé	Byty	Prodej	Prodej bytu Praha 7	1+1	Praha 7, Komenského	2 200 000 Kč	Akce
N399	3	11.10.2012	11.10.2012	Zajímavé	Byty	Prodej	Krásný byt v osobním vlastnictví		Odolena Voda, V Malém háji	Info v RK	Akce
N373	3	14.8.2012	30.10.2012	Zajímavé	Byty	Pronájem	Pronájem bytu 4+1, České Budějovice	4+1	Praha 6, Mlády, Hrabovce	10 000 Kč	Akce
N365	3	27.7.2012	27.7.2012	Zajímavé	Byty	Prodej	Prodej bytu Ok. 3+1/L, Habrová ul., P3 - Žďev	3+1	Praha 2, Anýlská	3 500 000 Kč	Akce
N326	3	14.8.2012	3.10.2012	Zajímavé	Byty	Prodej	Byt 3+1+L, OV, 4. patro, cihlový dům, Ústí nad Labem	3+1	Praha 1, Legátova	Info v RK	Akce
N311	3	28.9.2012	12.7.2013	Zajímavé	Byty	Prodej	Prodej bytu 2+1 v OV, Liberec, Vratslavice n/Ú - Tamašská ul	2+1	Liberec	1 500 000 Kč	Akce
N240	3	2.11.2011		Zajímavé	Byty	Prodej	Prodej bytu Turnov Prodej bytu Turnov	2+kk	Turnov	1 300 000 Kč	Akce

Obrázek 17: Náhled aplikace Softreal (Zdroj: Softreal, 2021)

4.2.4 Ostatní řešení

Mezi další řešení dostupné na trhu patří např. aplikace Urbium, REALBrána nebo Realman. Tyto aplikace ale nabízejí pouze základní funkce jako zpracování reportů a evidenci a správu klientů, poptávek a nabídek. Vzhledem k omezené funkcionalitě nejsou pro srovnání relevantní a nebudou zde dále podrobněji popisovány.

4.2.5 Shrnutí a výběr nejlepšího řešení

Porovnání nejdůležitějších funkcí popsanych řešení lze vidět v tabulce č. 3. Zelená ikona označuje funkce, které řešení obsahuje v základním balíčku. Žlutá ikona označuje rozšiřující funkce dostupné za příplatek. Červená ikona pak označuje funkce, které nejsou součástí daného řešení.

Funkce	Realko	Poski REAL	Softreal	Navrhovaný IS
Správa uživatelských účtů	✓	✓	✓	✓
Správa klientů	✓	✓	✓	✓
Správa poptávek	✓	✓	✓	✓
Správa nabídek	✓	✓	✓	✓
Správa projektů	★	✓	★	✓
Správa obchodních případů	★	✓	✗	✓
Evidence plateb	✗	✓	✗	✓
Provize	★	✓	★	✓
Publikace nabídek na realitních serverech	✓	✓	✓	✓
Publikace nabídek na webových stránkách	★	✓	✓	✓
Topování	★	✓	✗	✓
Kalendář	✓	✓	✓	✓
Kalendář propojený s aplikacemi třetích stran	✓	✓	✗	✓
Reporty	✓	✓	✓	✓
Sledování nákladů na inzerci	✗	✗	★	✗
Přehled prohlídek	✓	✓	✗	✗
Automatická příprava smluv	✗	✗	✗	✓

Tabulka 3: Porovnání funkcí popsaných konkurenčních řešení

Výběr nejlepšího řešení je proveden na základě vícekritériální analýzy variant, konkrétně metodou TOPSIS. Kritéria, která jsou v analýze uvažována, jsou shodná s funkcemi jednotlivých řešení, které jsou uvedeny v tabulce č. 3. Stanovení důležitosti jednotlivých kritérií je provedeno na základě bodovací metody. Výběr kritérií a přidělení bodů jednotlivým kritériím byly provedeny po konzultaci se specialisty na prodej nemovitostí.

Jak už bylo zmíněno, pro výběr nejlepší varianty je použita metoda TOPSIS. Pro výpočet byl využit doplněk MS Excel – SANNA. V tabulce č. 4 je uvedeno výsledné pořadí popsanych řešení včetně hodnot vypočteného ukazatele relativní vzdálenosti c_i . Všechny dílčí výpočty jsou z důvodu jejich rozsahu uvedené v přílohách této práce.

Pořadí	Řešení	c_i
1.	Navrhovaný IS	0,62667
2.	Poski Real	0,47507
3.	Realko	0,36641
4.	Softreal	0,35888

Tabulka 4: Pořadí variant určené metodou TOPSIS

Jako největší výhodu svého řešení vidí autorka funkci umožňující automatické generování smluv z předdefinovaných šablon. Tato funkce je zásadní, neboť příprava smluv představuje velkou část administrativních činností v procesu prodeje nemovitostí a je časově náročná. Vzhledem k chybovosti, ke které dochází při manuálním doplňování údajů do smluv, prochází smlouvy často kontrolou jiného zaměstnance. V těchto případech pak dochází i ke zdvojnásobení času vynaloženého na přípravu smlouvy.

Pokud není chyba odhalena při kontrole smlouvy, ale je odhalena až klientem nebo obchodním partnerem, např. developerem, kterému je následně smlouva předkládána, znamená to v lepším případě pouze projev neprofesionality ze strany realitní kanceláře. V horším případě může dojít až k odstoupení od uzavření obchodu ze strany klienta nebo k ukončení spolupráce ze strany obchodního partnera, což může mít pro realitní kancelář závažné následky. Díky automatizaci procesu přípravy smluv lze těmto problémům předcházet.

Další výhodou navrhovaného řešení je také přehledná koncepce systému. Konkurenční řešení často obsahují množství funkcí, které ale v praxi nejsou příliš dobře využitelné a v konečném důsledku tak spíše snižují přehlednost celého systému a znesnadňují jeho používání. Proto je navrhovaný systém koncipován tak, aby byl přehledný a obsahoval

opravdu důležité funkce, které přispějí ke snížení administrativní náročnosti a zároveň větší efektivitě celého procesu prodeje nemovitostí.

4.3 Analýza požadavků

Obsahem této kapitoly je analýza požadavků na systém, která je základem pro návrh celého systému. Požadavky byly specifikovány kombinací různých způsobů sběru požadavků popsanych v teoretických východiscích této práce, konkrétně na základě rozhovorů s potenciálními uživateli, tedy osobami zabývajícími se prodejem nemovitostí, dále na základě autorčiných zkušeností s používáním obdobného systému a studiem dokumentace obdobného systému. Požadavky jsou přehledně rozděleny do dvou kategorií – na funkční a nefunkční požadavky.

4.3.1 Funkční požadavky

ID požadavku	Název požadavku	Priorita
F01	Systém bude umožňovat správu uživatelů.	M
F02	Systém bude umožňovat přihlášení do systému.	M
F03	Systém bude obsahovat kalendář.	S
F04	Systém bude umožňovat správu nabídek.	M
F05	Systém bude umožňovat správu projektů.	S
F06	Systém bude umožňovat správu klientů.	M
F07	Systém bude umožňovat správu poptávek.	M
F08	Systém bude umožňovat správu rezervací.	M
F09	Systém bude umožňovat správu šablon smluv.	M
F10	Systém bude umožňovat správu smluv.	M
F11	Systém bude umožňovat správu obchodních případů.	M
F12	Systém bude umět vytvářet reporty.	M

Tabulka 5: Seznam funkčních požadavků

F01 – Správa uživatelů

System bude umožňovat přidat a spravovat více uživatelů. Jelikož se předpokládá, že se systémem budou pracovat různé kategorie uživatelů, bude možné regulovat ovládání funkcionalit systému pomocí oprávnění. Pro jednoduchost nastavení oprávnění bude využito uživatelských rolí, které budou obsahovat soubor oprávnění typických pro určitou kategorii uživatelů. Správu uživatelů bude v systému zajišťovat administrátor, který bude mít přidělena nejvyšší možná oprávnění.

Požadované funkce:

- přidání uživatele,
- editace uživatele,
- odebrání uživatele,
- nastavení oprávnění,
- nastavení rolí.

F02 – Přihlášení do systému

Každému uživateli, který bude přihlášen do systému, bude umožněno ovládání systému na základě přiřazených oprávnění. Pokud uživatel nebude mít přiřazená žádná oprávnění, bude mít k dispozici základní funkce systému, jimiž budou:

- odhlášení ze systému,
- změna hesla.

F03 – Kalendář

System bude obsahovat modul kalendář, který bude umožňovat plánování událostí. Kalendář bude propojený s dalšími moduly systému, ze kterých budou do kalendáře odesílána data. Uživatel bude mít také možnost propojit modul kalendář se svým osobním kalendářem v aplikacích třetích stran.

Požadované funkce:

- plánování událostí,
- propojení s dalšími moduly v systému,
- propojení s aplikacemi třetích stran.

F04 – Správa nabídek

Systém bude obsahovat modul pro správu nabídek, který bude sloužit jako databáze nabízených nemovitostí. Uživateli umožní přidávat nabídky, odebírat nabídky nebo upravovat stávající nabídky. U každé nabídky bude možné přiřadit prodejce, který ji spravuje a šablonu smlouvy, díky čemuž bude následně možné vygenerovat ze systému hotovou smlouvu (viz požadavek F09 a F10). Jednou z klíčových funkcí správy smluv je také publikování nabídek na webových stránkách nebo realitních serverech a případně topování publikovaných nabídek. Systém umožní také vyhledávání nabídek podle stanovených kritérií.

Požadované funkce:

- přidání nabídky,
- editace nabídky,
- odebrání nabídky,
- přiřazení prodejce,
- přiřazení šablony,
- publikování nabídky,
- topování nabídky,
- vyhledání nabídky.

F05 – Správa projektů

Modul správa projektů bude umožňovat obdobné funkce jako modul správa nabídek. Jeho klíčovou funkcí je sdružení souvisejících nabídek do jednoho celku, respektive přiřazení nabídek k projektu, což následně umožňuje efektivnější správu takových nabídek a přehlednější vyhledávání nebo reporting.

Požadované funkce:

- přidání projektu,
- editace projektu,
- odebrání projektu,
- přiřazení prodejce,
- publikování projektu,
- topování projektu,
- vyhledání projektu,
- přiřazení nabídky k projektu.

F06 – Správa klientů

Modul správa klientů bude představovat databázi klientů. Klienty bude možné přidávat, editovat nebo odebrat. U každého klienta bude možné evidovat jeho osobní údaje, kontakt nebo prodejce, který se o daného klienta stará. Systém umožní také vyhledávání klientů podle stanovených kritérií.

Požadované funkce:

- přidání klienta,
- editace klienta,
- odebrání klienta,
- vyhledání klienta,
- přiřazení prodejce.

F07 – Správa poptávek

V modulu správy poptávek budou evidovány všechny poptávky klientů po nabízených nemovitostech nebo projektech. V rámci každé poptávky bude možné evidovat konkrétního prodejce, který poptávku vyřídil. Poptávky bude možné přidávat, editovat nebo odebrat. Poptávky bude také možné vyhledávat podle stanovených kritérií.

Požadované funkce:

- přidání poptávky,

- editace poptávky,
- odebrání poptávky,
- vyhledání poptávky,
- přiřazení prodejce.

F08 – Správa rezervací

Modul správy rezervací bude sloužit k časově omezené rezervaci nabídek. Bude stanovena standardní doba rezervace nabídky, uživatel bude ale moci rezervaci prodloužit, případně rezervaci smazat ještě před jejím vypršením. V systému bude také možné vyhledávat rezervace podle stanovených kritérií.

Požadované funkce:

- přidání rezervace,
- smazání rezervace,
- prodloužení rezervace,
- vyhledání rezervace.

F09 – Správa šablon smluv, F10 – Správa smluv

Systém bude umožňovat přidávat šablony smluv, které budou následně přiřazeny k příslušným nabídkám. Ze šablony uložené v systému bude možné vytvořit a exportovat kompletní smlouvu do textového editoru, na základě toho, že systém bude umět vyplnit prázdná pole v šabloně smlouvy konkrétními daty, tzn. osobními údaji klienta a parametry nabídky (např. výměra, cena apod.). Po podepsání smlouvy klientem bude možné smlouvu připojit k obchodnímu případu.

Požadované funkce:

- přidání šablony,
- nastavení polí,
- editace šablony,
- odebrání šablony,

- vytvoření smlouvy,
- export smlouvy do textového editoru,
- připojení smlouvy k obchodnímu případu.

F11 – Správa obchodních případů

Modul správy obchodních případů bude sloužit k evidenci realizovaných obchodů. U každého obchodního případu bude možné evidovat prodejce, který obchod zprostředkoval.

System bude umět vypočítat cenu obchodního případu na základě ceny nabídky a zadané výše DPH. Na základě ceny obchodního případu a zadané procentuální výše provize bude vypočtena provize pro realitní kancelář. Při úspěšném dokončení obchodu bude možné provést uzavření obchodního případu. V opačném případě bude systém umět obchodní případ stornovat, případně vymazat.

Součástí obchodního případu bude také platební kalendář, který bude sloužit k evidenci předepsaných i přijatých plateb. V obchodním případě bude možné také evidovat důležité termíny. Naplánované termíny pak bude uživatel moct nastavit jako událost v kalendáři, tato funkce bude zajištěna realizací požadavku F03.

Požadované funkce:

- | | |
|----------------------------------|-----------------------------------|
| • přidání obchodního případu, | • nastavení platebního kalendáře, |
| • editace obchodního případu, | • evidence plateb, |
| • storno obchodního případu, | • nastavení důležitých termínů, |
| • uzavření obchodního případu, | • odeslání termínů do kalendáře, |
| • odstranění obchodního případu, | • vypočtení celkové ceny, |
| • vyhledání obchodního případu, | • vypočtení provize. |
| • přiřazení prodejce, | |

F12 – Reporty

Funkce reporty bude sloužit k vytváření výstupních sestav ze systémových dat. System bude umožňovat tato data exportovat do tabulkového editoru.

Požadované funkce:

- zobrazení seznamu dat,
- export dat do tabulkového editoru.

4.3.2 Nefunkční požadavky

ID požadavku	Název požadavku	Priorita
N01	System bude dostupný z internetových prohlížečů.	M
N02	Připojení do systému bude realizováno prostřednictvím protokolu HTTPS.	M
N03	System bude provádět autentizaci uživatele.	M
N04	System bude umožňovat spravovat oprávnění.	M
N05	System bude respektovat zásady použitelnosti.	M
N06	System bude mít dostupnost minimálně 99 %.	M
N07	System bude mít nízkou latenci.	S
N08	System bude možné zálohovat.	M
N09	System bude respektovat obecné nařízení o ochraně osobních údajů (GDPR).	M
N10	System bude aktualizovat data na webových stránkách/realitních serverech.	M
N11	System bude rozšiřitelný.	C

Tabulka 6: Seznam nefunkčních požadavků

N01 – Dostupnost z internetových prohlížečů

System bude dostupný z webového serveru pomocí internetu s využitím webového prohlížeče jako tenkého klienta. System bude podporovat přístup z webových prohlížečů Chrome, Firefox, Opera, Edge a Safari. V těchto prohlížečích by měly být fungování a vzhled systému jednotné. Tento způsob byl zvolen především kvůli nezávislosti na operačním systému uživatele a zároveň kvůli dostupnosti systému z různých druhů zařízení bez nutnosti instalace.

N02 – Šifrované připojení

Připojení do systému a veškerá komunikace s webovým serverem budou realizovány prostřednictvím zabezpečeného protokolu HTTPS využívajícím protokol TLS. Tímto bude zajištěna bezpečnost komunikace a důvěrnost i integrita přenášených dat.

N03 – Autentizace uživatelů

Jelikož v systému bude uloženo velké množství citlivých dat, bude při přihlášení uživatele do systému provedena autentizace. Pokud bude uživatel nečinný déle jak hodinu, bude ze systému automaticky odhlášen a bude požadována opětovná autentizace. Autentizace bude provedena na základě zadání uživatelského jména a hesla, přičemž heslo každého uživatele musí tvořit minimálně osm znaků, které budou alespoň kombinací písmen a číslic. Expirace hesla uživatele je šest měsíců. Heslo bude zároveň ukládáno ve formě hashe, což zamezí zneužití hesel potenciálním útočníkem.

N04 – Správa oprávnění

V systému bude možné nastavit každému uživateli oprávnění k využívání různých funkcionalit systému. Jedním z důvodů pro udělování oprávnění je ochrana citlivých dat v systému. Vhodně nastavenými oprávněními je možné zamezit kopírování, čtení nebo přepisu dat neoprávněnými uživateli.

N05 – Respektování zásad použitelnosti

Jednou z klíčových vlastností systému je jednoduchost jeho používání a snadná naučitelnost ovládání systému. Proto je nezbytné při návrhu respektovat zažité konvence a předpokládané znalosti a zkušenosti budoucích uživatelů. Dobrá použitelnost systému znamená konkurenční výhodu a zároveň může výrazně snížit dobu potřebnou pro zaškolení nového uživatele. Jelikož bude systém určen nejen pro desktop, ale i pro mobilní zařízení, je nutné, aby design systému byl responzivní.

N06 – Dostupnost

System bude mít dostupnost minimálně 99 %, což představuje maximální dobu výpadku 87,6 hodin ročně. Doba obnovy při poruše systému zároveň nesmí přesáhnout 6 hodin. Do této doby není započítána nezbytná údržba systému.

N07 – Latence

System nesmí přesáhnout dobu odezvy 2 sekundy při běžných operacích a 30 sekund při exportu výstupních sestav. System by měl zároveň pracovat plynule. Konkrétní počet souběžně připojených uživatelů, při kterém bude systém ještě pracovat plynule bude stanoven podle konkrétní instance systému.

N08 – Zálohování

Jelikož se předpokládá, že v systému bude uchováváno poměrně velké množství dat, je nezbytné nastavit optimální frekvenci zálohování systému, a to zároveň s ohledem na finanční náklady na uchování záloh. Po zvážení bylo stanoveno, že systém bude denně zálohovat nová data. Interval pro zálohu obrazu celého systému pak bude stanoven dle konkrétní instance systému.

N09 – GDPR

System bude respektovat obecné nařízení o ochraně osobních údajů GDPR. Mezi osobní údaje se řadí např. jméno, pohlaví, věk, datum narození, adresa, fotografie apod. Ochrana osobních údajů se týká všech údajů vztahujících se k identifikovaným či identifikovatelným fyzickým osobám (GDPR, 2021), tedy jak údajů klientů, tak i údajů samotných uživatelů.

N10 – Aktualizace dat

Data o nabídkách a projektech publikovaná na realitních serverech nebo webových stránkách budou průběžně aktualizována. Konkrétní způsob zveřejnění a aktualizace těchto dat bude řešen v rámci implementace.

N11 – Rozšiřitelnost systému

Jelikož se část potenciálních zákazníků zabývá kromě prodeje nemovitostí současně i jejich pronájmem, měl by systém být připravený na případné budoucí rozšíření, které bude realizovat obchodní proces pronajímání nemovitostí.

4.4 Návrh systému

Na základě teoretických východisek práce a analýzy požadavků je vytvořen model tříd, stavový model a model interakcí. Model interakcí je zaměřen konkrétně na proces správy smluv, který je v porovnání s popsány konkurenčními řešeními inovativní. Modely jsou vytvořeny ve standardech jazyka UML v nástroji StarUML verzi 4.0.1. V závěru práce jsou navrženy wireframes systému a je nastíněn možný způsob implementace systému.

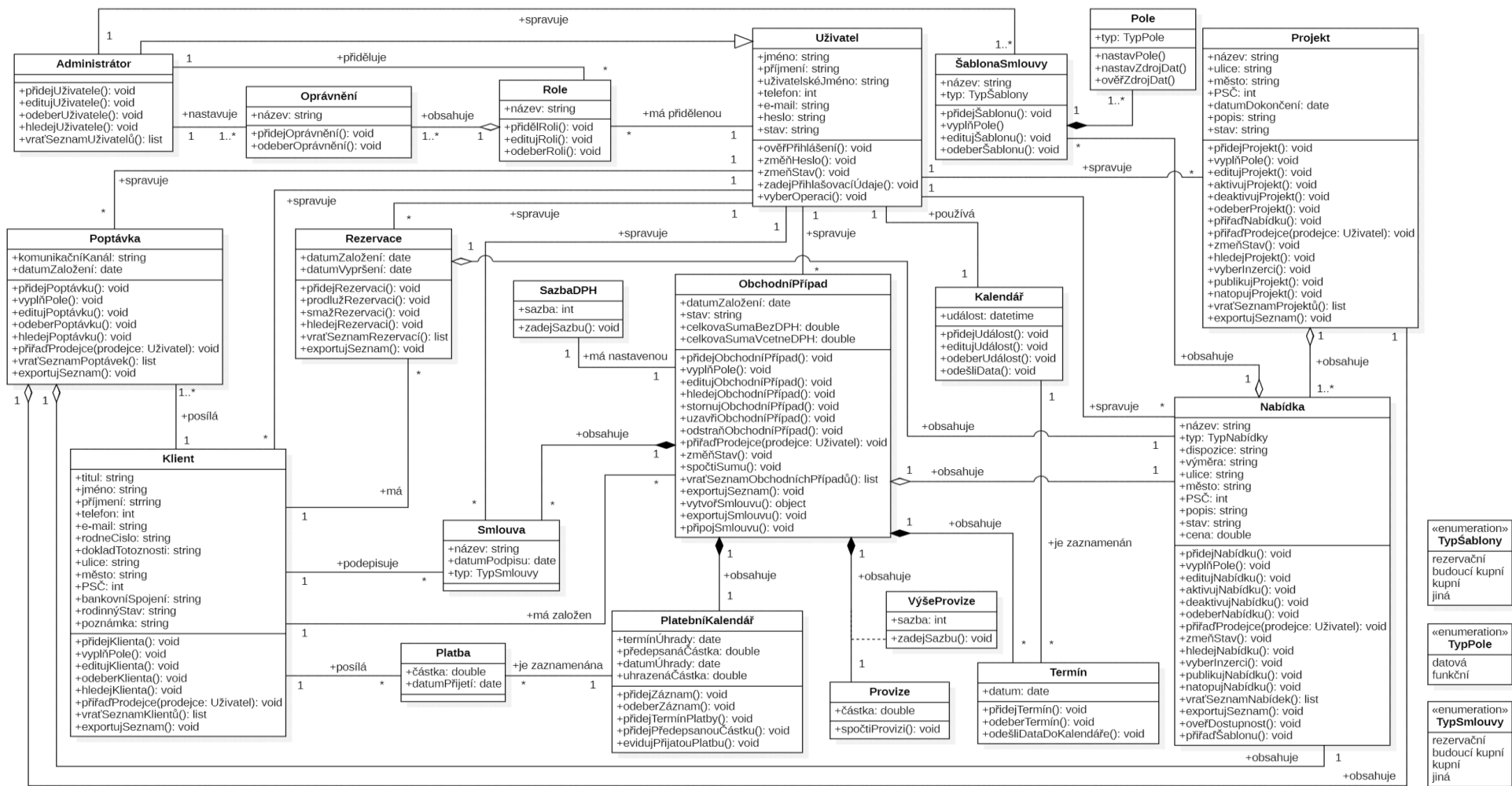
4.4.1 Model tříd

Model tříd představuje statický model systému, jehož hlavním cílem je zobrazení tříd objektů a vazeb mezi těmito třídami. Model tříd je reprezentován diagramem tříd, který lze vidět na obrázku č. 18.

Při tvorbě diagramu tříd byly v rámci analytického procesu nejprve identifikovány jednotlivé třídy. Při výběru názvů tříd byl brán zřetel na to, aby názvy byly jednoznačné a zároveň odpovídaly skutečným obchodním pojmům. Dalším krokem byla identifikace vazeb mezi třídami a přidání základních atributů. V diagramu tříd se vyskytují vazby typu generalizace, asociace, agregace a kompozice. Následně byla doplněna kardinalita jednotlivých vazeb a byly přidány operace tříd. V průběhu vytváření dalších modelů systému byl diagram tříd průběžně zpřesňován a doplňován o nové atributy a operace. Byly také doplněny datové typy atributů, metody operací a tam, kde to bylo vhodné, byly pro atributy vytvořeny číselníky.

Je však potřeba dodat, že při modelování diagramu tříd byl uplatněn princip zobecnění a diagram tak obsahuje pouze třídy nezbytně nutné k pochopení fungování celého systému. Detailní zachycení všech tříd, atributů a operací systému by bylo nad rámec této práce.

Diagram tříd doplňuje datový slovník, který obsahuje popis všech tříd v diagramu včetně jejich rozsahu v rámci systému.



Obrázek 18: Diagram tříd

Datový slovník

Třída Administrátor

Třída *Administrátor* je podtřídou třídy *Uživatel*, vyskytuje se mezi nimi tedy vztah typu generalizace. *Administrátor* dědí od třídy *Uživatel* všechny atributy a operace a zároveň přidává své vlastní. Oproti třídě *Uživatel* má přiřazená nejvyšší možná oprávnění, která mu umožňují spravovat všechny ostatní uživatele systému.

Třída Uživatel

Třída *Uživatel* představuje osobu, která je registrována v systému. Na základě přiřazených *Oprávnění* ovládá funkcionality systému.

Třída Oprávnění

Třída *Oprávnění* reguluje používání funkcí systému pro různé uživatele. Pro použití jednotlivých funkcí v systému je potřeba mít přidělená oprávnění, jejichž nastavení zajišťuje *Administrátor*.

Třída Role

Třída *Role* obsahuje souhrn oprávnění pro konkrétní kategorii uživatelů. Mezi třídami *Oprávnění* a *Role* se vyskytuje vazba typu agregace, neboť se jedná o typ vztahu „skládá se z“. Za přiřazení rolí uživatelům je zodpovědný *Administrátor*.

Třída Nabídka

Třída *Nabídka* představuje modul pro správu nabídek v systému. Konkrétní instance této třídy představuje konkrétní nabízenou nemovitost, jejíž parametry jsou popsány pomocí atributů třídy *Nabídka*. Nabídky v systému spravuje *Uživatel*.

Třída Projekt

Třída *Projekt* představuje modul pro správu projektů v systému. Konkrétní instance třídy *Projekt* představuje konkrétní nabízený projekt. Mezi třídami *Projekt* a *Nabídka* existuje vztah typu agregace – třída *Projekt* obsahuje přiřazené nabídky, ale třída *Nabídka* může

zároveň existovat nezávisle na třídě *Projekt*. Za správu projektů v systému je zodpovědný *Uživatel*.

Třída Klient

Třída *Klient* představuje modul pro správu klientů. Konkrétní instance této třídy představuje konkrétní fyzickou nebo právnickou osobu. Osobní údaje klientů jsou v systému evidovány pomocí atributů třídy *Klient*. Správu klientů v systému zajišťuje *Uživatel*.

Třída Poptávka

Třída *Poptávka* představuje modul pro správu poptávek v systému. Konkrétní instance této třídy představuje konkrétní poptávku po nabízené nemovitosti nebo projektu zaslanou klientem. Mezi třídou *Nabídka/Projekt* a třídou *Poptávka* je vztah typu agregace, neboť součástí každé instance třídy *Poptávka* musí být vždy konkrétní instance třídy *Nabídka* nebo třídy *Projekt*. Za správu poptávek je v systému zodpovědný *Uživatel*.

Třída Rezervace

Třída *Rezervace* představuje modul pro správu rezervací v systému. Konkrétní instance této třídy představuje konkrétní rezervaci nabízené nemovitosti – instanci třídy *Nabídka*. Nemovitost má rezervovanou klient, přičemž počet rezervací jednoho klienta není v systému omezen. Správu rezervací zajišťuje *Uživatel*.

Třída ŠablonaSmlouvy

Třída *ŠablonaSmlouvy* představuje modul pro správu šablon v systému, konkrétní entita této třídy pak představuje vzor smlouvy pro realizaci obchodu. Šablonu smlouvy je možné vkládat, editovat a odebírat, přičemž konkrétní šablona musí být spjata s konkrétní nabídkou, aby z ní bylo možné vytvořit smlouvu. Mezi třídami *Nabídka* a *ŠablonaSmlouvy* proto existuje vztah typu agregace. Šablony smluv má v systému na starosti *Administrátor*.

Třída Pole

Třída Pole představuje v systému pole v šabloně, do kterých jsou doplňována data ze systému při vytváření smlouvy. Mezi třídou *ŠablonaSmlouvy* a *Pole* je vztah typu kompozice, jelikož existence polí je vázána na existenci šablony.

Třída Smlouva

Třída *Smlouva* představuje v systému smlouvu vytvořenou ze šablony smlouvy. Vytvoření smlouvy je operací ve třídě *ObchodníPřípad*, jelikož na základě přiřazení klienta a nabídky k obchodnímu případu jsou do polí v šabloně smlouvy doplněná příslušná data. Tato třída zároveň představuje smlouvu podepsanou klientem, prostřednictvím které je realizován obchod. Třída *Smlouva* je spojena kompoziční vazbou s třídou *ObchodníPřípad*, neboť smlouva může být vytvořena až po přidání obchodního případu. V rámci jednoho obchodního případu může být uzavřeno více typů smluv např. smlouva rezervační a smlouva kupní aj. Správu smluv v systému zajišťuje *Uživatel*.

Třída ObchodníPřípad

Třída *ObchodníPřípad* představuje modul pro správu obchodních případů v systému, přičemž konkrétní instance představuje konkrétní obchod mezi klientem a realitní kanceláří. Součástí obchodního případu je třída *PlatebníKalendář*, *Provize* a *Termín*. Tyto třídy a třída *ObchodníPřípad* mají mezi sebou vztah typu kompozice, jelikož existence těchto tříd je svázána s existencí obchodního případu a bez něj zanikají. Součástí obchodního případu je také konkrétní instance třídy *Nabídka*. Mezi třídami *ObchodníPřípad* a *Nabídka* tudíž existuje vazba typu agregace. Správu obchodních případů zajišťuje v systému *Uživatel*.

Třída SazbaDPH

Třída *SazbaDPH* je třídou, která slouží k nastavení výše DPH. Třída *SazbaDPH* je spojena asociační vazbou s třídou *ObchodníPřípad* a umožňuje vypočítat celkovou cenu obchodního případu.

Třída PlatebníKalendář

Třída *PlatebníKalendář* je třídou, která slouží k evidenci předepsaných a přijatých plateb týkajících se obchodního případu.

Třída Platba

Třída *Platba* představuje peněžitou platbu, kterou zasílá klient na základě uzavřené smlouvy a která je evidována v platebním kalendáři.

Třída VýšeProvize

Třída *VýšeProvize* je vazební třídou, která umožňuje nastavení výše provize. Díky tomu lze pak pomocí operace v třídě *Provize* spočítat celkovou provizi pro realitní kancelář.

Třída Provize

Třída *Provize* představuje příjem z realizovaného obchodu plynoucí realitní kanceláři.

Třída Termín

Třída *Termín* slouží k evidování důležitých termínů obchodního případů. Prostřednictvím operace *odešliTermínDoKalendáře* pak lze konkrétní termíny přidat do kalendáře uživatele – třídy *Kalendář*.

Třída Kalendář

Třída *Kalendář* představuje modul pro plánování událostí uživatele. Zároveň umožňuje sdílet naplánované události s jinou kalendářní aplikací, např. s osobním kalendářem uživatele v jeho zařízení.

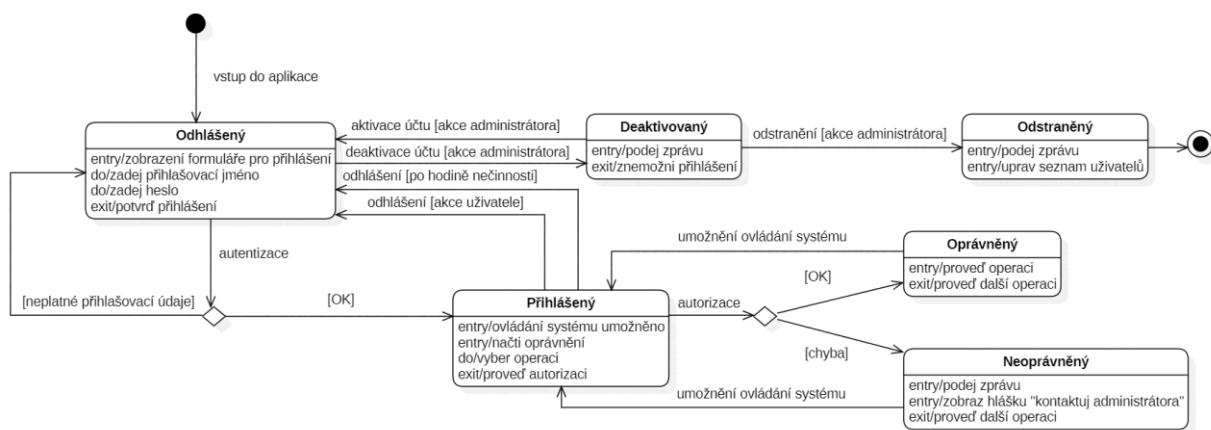
4.4.2 Stavový model

Stavový model se skládá ze stavových diagramů, přičemž pro každou třídu může existovat jeden nebo více stavových diagramů. Vždy je ale potřeba zvážit, zda tvorba stavového diagramu bude pro model užitečná. Pomůže-li tvorba diagramu v pochopení složitého

životního cyklu nebo chování, má takový stavový diagram význam. V této práci byly vytvořeny stavové diagramy pro třídy *Uživatel*, *Nabídka* a *Obchodní případ*.

Stavový diagram třídy *Uživatel*

Na obrázku č. 19 je zobrazen stavový diagram pro třídu *Uživatel*, který popisuje stavy, kterými třída v systému prochází. Stavový diagram zobrazuje stavy uživatele, který je v systému registrován. Třída *Uživatel* prochází v systému celkem šesti stavy – *Odhlášený*, *Přihlášený*, *Oprávněný*, *Neoprávněný*, *Deaktivovaný* a *Odstraněný*.



Obrázek 19: Stavový diagram pro třídu *Uživatel*

Výchozím stavem pro uživatele po vstupu do aplikace je stav *Odhlášený*. Po zadání správných přihlašovacích údajů a úspěšné autentizaci uživatel přejde do stavu *Přihlášený*. Při zadání neplatných přihlašovacích údajů uživatel zůstane ve stavu *Odhlášený*. Přihlášenému uživateli je umožněno ovládání systému a provádění operací. Potom, co uživatel vybere operaci, kterou chce provést, provede systém autorizaci uživatele. Po úspěšném provedení autorizace přejde uživatel do stavu *Oprávněný*. Systém provede zvolenou operaci, uživatel přejde do stavu *Přihlášený* a je mu umožněno provádět další operace v systému. V případě neúspěšné autorizace přejde uživatel do stavu *Neoprávněný*. Následně systém uživateli zobrazí zprávu o neúspěšné autorizaci a vyzve uživatele ke kontaktování administrátora. Nakonec uživatel přejde opět do stavu *Přihlášený*. Uživatel může být ze systému odhlášen dvěma způsoby – manuálně nebo automaticky, a to v případě, že je více jak hodinu v systému nečinný. Uživateli může být administrátorem deaktivován účet, tím přejde uživatel do stavu

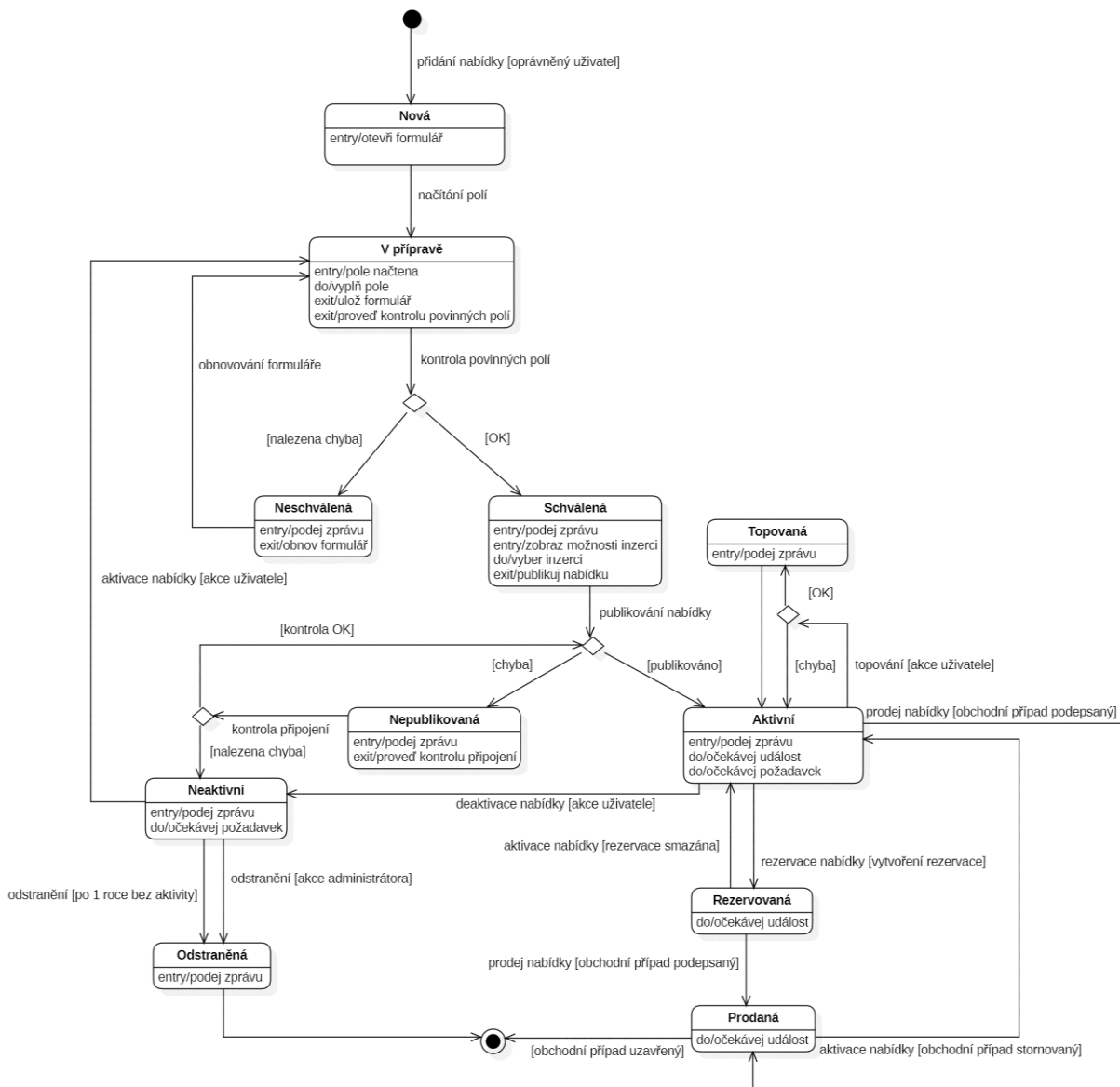
Deaktivovaný. Pokud je uživatel ve stavu *Deaktivovaný*, je mu znemožněno přihlášení do systému. Ze stavu *Deaktivovaný* může uživatel přejít do stavu *Odhlášený*, pokud je administrátorem aktivován, anebo do stavu *Odstraněný*, pokud je administrátorem odstraněn ze systému. Stav *Odstraněný* je konečným stavem třídy *Uživatel*.

Stavový diagram třídy Nabídka

Na obrázku č. 20 je zobrazen stavový automat třídy *Nabídka*. Třída *Nabídka* může ve svém životním cyklu v systému procházet celkem jedenácti stavy – *Nová*, *V Přípravě*, *Neschválená*, *Schválená*, *Nepublikovaná*, *Aktivní*, *Neaktivní*, *Topovaná*, *Rezervovaná*, *Prodaná* a *Odstraněná*.

Po otevření formuláře pro přidání nové nabídky do systému se nachází nabídka ve stavu *Nová*. Následně po načtení všech polí formuláře přejde do stavu *V přípravě*. V tomto stavu dochází k vyplnění polí formuláře a k uložení formuláře. Následně je provedena kontrola povinných polí. Pokud je při kontrole nalezena chyba, přechází nabídka do stavu *Neschválená* a dojde k obnovení formuláře, čímž nabídka přechází opět do stavu *V přípravě*. V případě, že kontrola povinných polí proběhla úspěšně, přejde nabídka do stavu *Schválená*. V tomto stavu dochází k výběru inzertních míst a následuje publikování nabídky. Pokud se nabídka v pořádku publikuje, přechází do stavu *Aktivní*. V opačném případě přechází do stavu *Nepublikovaná* a je provedena kontrola připojení k inzertním serverům/webovým stránkám. V případě, že kontrola proběhne v pořádku, zahájí se opětovný pokus o publikování nabídky. V opačném případě přechází nabídka do stavu *Neaktivní*. Do tohoto stavu může přejít také ze stavu *Aktivní* prostřednictvím požadavku od uživatele.

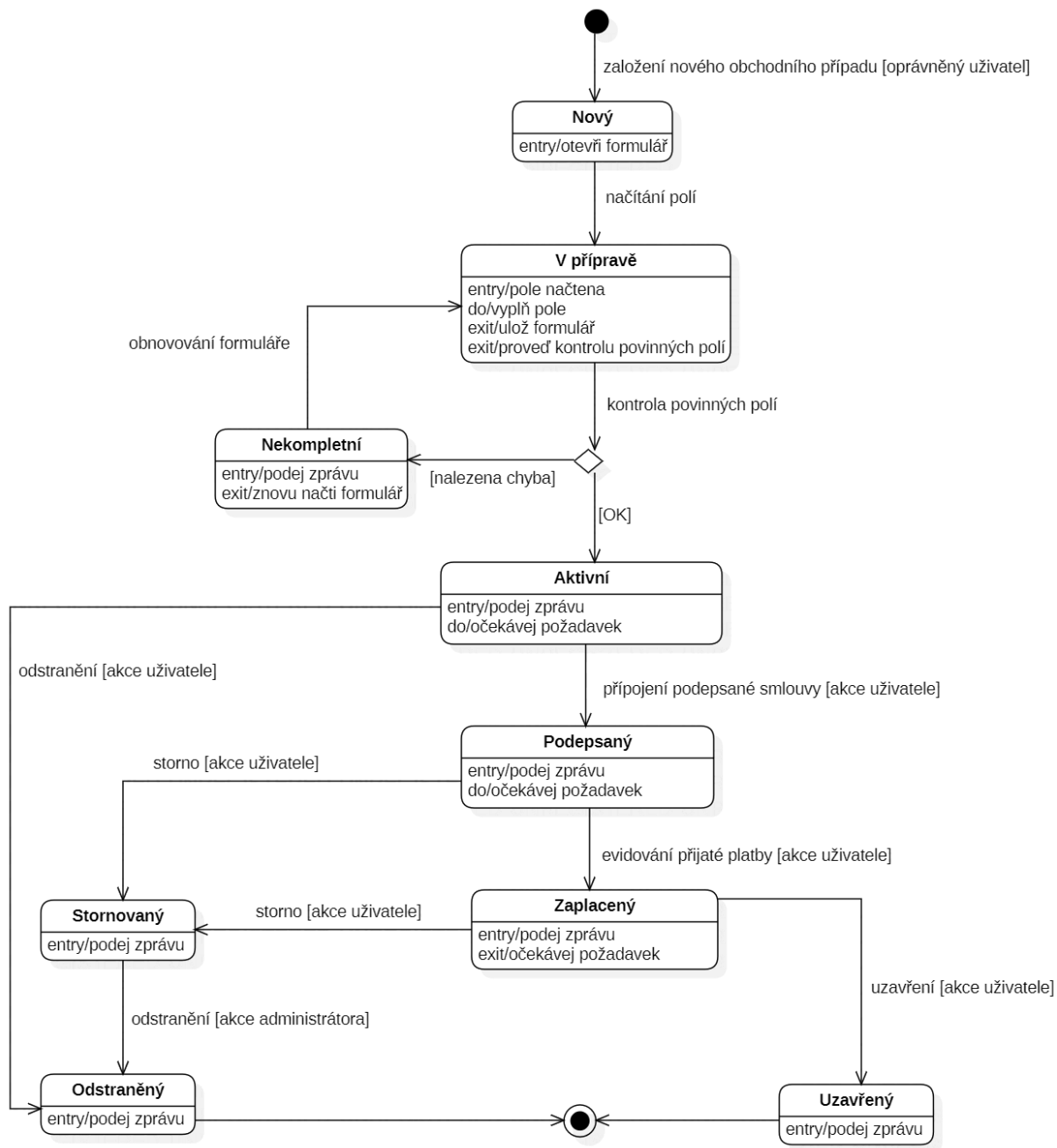
Ve stavu *Aktivní* může být nabídka topována, čímž přechází do stavu *Topovaná*. Ve stavu *Aktivní* nabídka interaguje s jinými objekty v systému – *Rezervace* a *Obchodní případ*. V případě, že dojde k vytvoření rezervace, ve které bude nabídka figurovat, přejde do stavu *Rezervovaná*. V případě, že dojde k připojení smlouvy k obchodnímu případu, čímž mění obchodní případ svůj stav na *Podepsaný*, přechází nabídka do stavu *Prodaná*. Z obou těchto stavů může přejít zpět do stavu *Aktivní* smazáním rezervace nebo stornováním obchodního případu. Nabídka může přejít do stavu *Odstraněná* pouze ze stavu *Neaktivní*, a to buď prostřednictvím požadavku administrátora nebo pokud je neaktivní více než rok.



Obrázek 20: Stavový diagram pro třídu Nabídka

Stavový diagram třídy ObchodníPřípad

Na obrázku č. 21 je zobrazen stavový diagram třídy *ObchodníPřípad*. Jak je z obrázku patrné, třída *ObchodníPřípad* prochází v systému celkem devíti stavy – *Nový*, *V přípravě*, *Nekompletní*, *Aktivní*, *Podepsaný*, *Zaplacený*, *Uzavřený*, *Stornovaný* a *Odstraněný*.



Obrázek 21: Stavový diagram pro třídu ObchodníPřípad

Výchozím stavem pro Obchodní případ je stav *Nový*. Po načtení všech polí formuláře pro založení obchodního případu přejde do stavu *V přípravě*. V tomto stavu dochází k vyplnění povinných polí formuláře, následně je formulář uložen a je provedena kontrola povinných polí. Pokud je při kontrole nalezena chyba, přejde obchodní případ do stavu *Nekompletní*. Systém podá zprávu o chybě a znovu načte uložený formulář. Obchodní případ přejde opět do

stavu *V Přípravě*. V případě úspěšně provedené kontroly formuláře přejde obchodní případ do stavu *Aktivní*. V tomto stavu zůstává do té doby, než je k obchodnímu případu připojena podepsaná smlouva. Po potvrzení podpisu smlouvy přejde obchodní případ do stavu *Podepsaný*. V tomto stavu opět zůstává do doby, než je evidováno přijetí platby od klienta, následně přechází obchodní případ do stavu *Zaplacený*. Po úspěšném dokončení obchodu uživatel uzavře obchodní případ, který tím přejde do stavu *Uzavřený*.

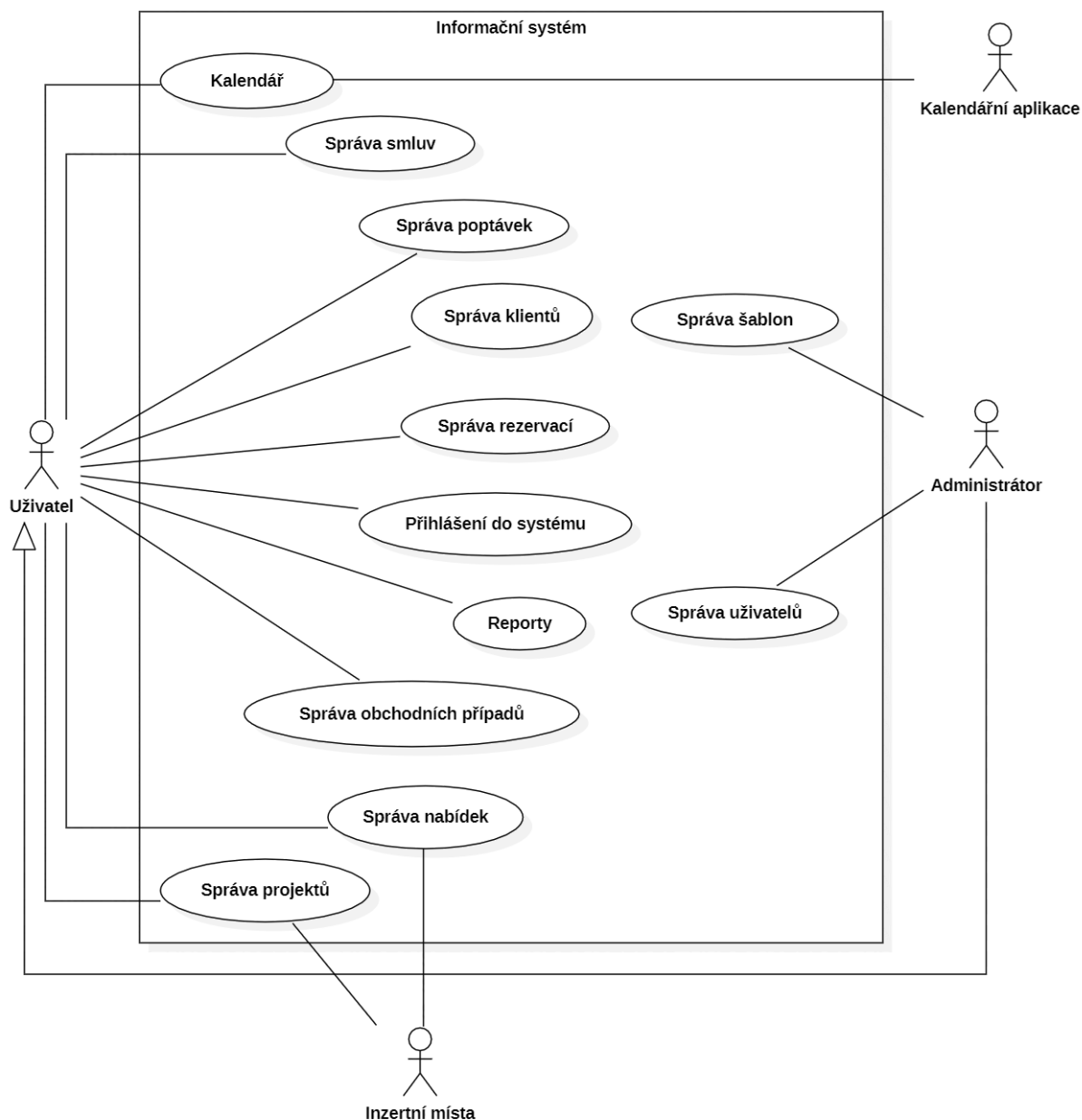
V případě, že byl obchodní případ z nějakého důvodu chybně založen nebo skončil neúspěšně, může jej uživatel smazat ze systému pouze, pokud se obchodní případ nachází ve stavu *Aktivní*. Pokud je obchodní případ ve stavu *Podepsaný* nebo *Zaplacený*, může jej uživatel pouze stornovat. Obchodní případ pak přechází do stavu *Stornovaný*. Pokud administrátor *Stornovaný* obchodní případ ze systému odstraní, přechází do stavu *Odstraněný*, čímž končí životní cyklus obchodního případu v systému.

4.4.3 Model interakcí

Model interakcí popisuje spolupráci objektů v systému a zároveň doplňuje dva předchozí modely. Model interakcí zobrazuje objekty v systému v několika úrovních abstrakce. Diagramy případů užití zachycují funkčnost systému na nejvyšší úrovni abstrakce. Podrobnější pohled přinášejí sekvenční diagramy, které popisují posloupnost chování z pohledu uživatele systému. Nejdetailnější pohled na systém pak poskytuje diagram aktivit, který podrobně modeluje kroky v rámci složitého procesu či algoritmu.

Modelování případů užití

Nejobecnějším diagramem modelu interakcí je diagram případů užití, který lze vidět na obrázku č. 22. Tento diagram zachycuje celkový pohled na systém. Případy užití vychází z definovaných funkčních požadavků.

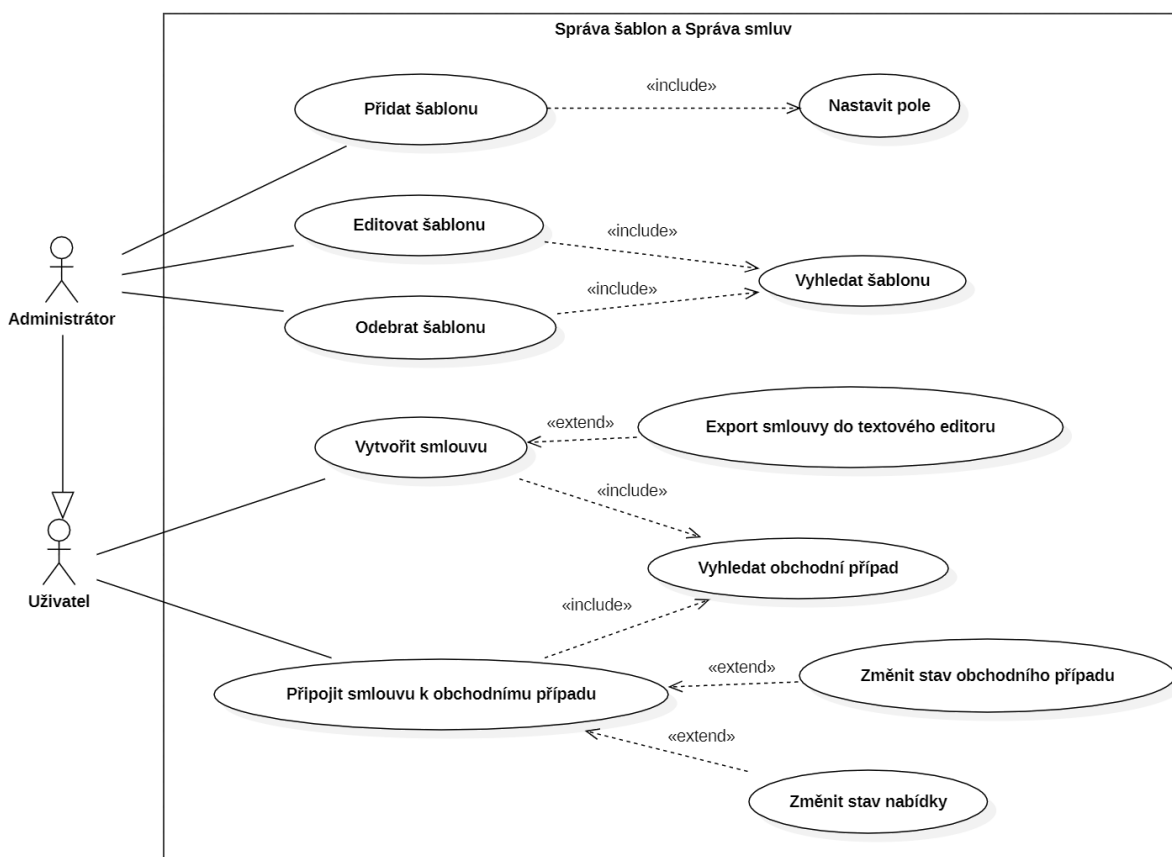


Obrázek 22: Diagram případů užití – Celkový pohled na systém

Jako aktéři jsou v diagramu identifikováni Uživatel a Administrátor. Uživatel představuje fyzickou osobu, která má v systému založen uživatelský účet a pracuje s funkcionalitami systému na základě svých oprávnění. Uživatelem v systému může být kterákoliv osoba, již byl založen účet, např. realitní makléř, asistentka, majitel nebo účetní. Administrátor je pak osobou, která spravuje všechny uživatele v systému a má na starosti bezproblémový chod celého systému. Jak je patrné z diagramu, mezi těmito dvěma aktéry existuje vztah generalizace. Administrátor tedy dědí všechny případy užití od Uživatele a zároveň figuruje

ve svých vlastních případech užití. Dalšími identifikovanými aktéry, se kterými systém komunikuje, jsou Kalendářní aplikace a Inzertní místa. Kalendářní aplikace představují osobní kalendáře uživatelů, do kterých jsou sdílena data z kalendáře v systému. Inzertní místa pak představují webové stránky nebo realitní servery, kde jsou publikovány nabídky a projekty.

U případů užití *Správa smluv* a *Správa šablon* z předchozího diagramu byla provedena dekompozice, která je zachycena na obrázku č. 23. Tyto případy užití zachycují funkcionalitu, která je hlavní výhodou navrhovaného systému oproti popsaným konkurenčním řešením. Proto jsou tyto případy užití dále podrobněji rozpracovány v následující části práce. Naopak dekompozice ostatních případů užití z předchozího diagramu jsou zařazeny v přílohách této práce, kde je uveden i seznam všech identifikovaných případů užití.



Obrázek 23: Dekompozice případů užití Správa šablon a Správa smluv

Sekvenční model

Sekvenční model se skládá ze scénářů a sekvenčních diagramů. Scénář představuje tok událostí v rámci jednoho případu užití. Sekvenční diagram typicky zachycuje chování jednoho scénáře. Zobrazuje objekty a zprávy, které jsou předávány mezi těmito objekty, v rámci daného případu užití.

V sekvenčním modelu v této práci je modelován pouze hlavní tok událostí neboli hlavní scénář popisující ideální kroky případu užití. V rámci hlavních scénářů jsou uvedeny pouze názvy alternativních scénářů a v rámci vytvořených diagramů aktivit je naznačeno větvení hlavního scénáře.

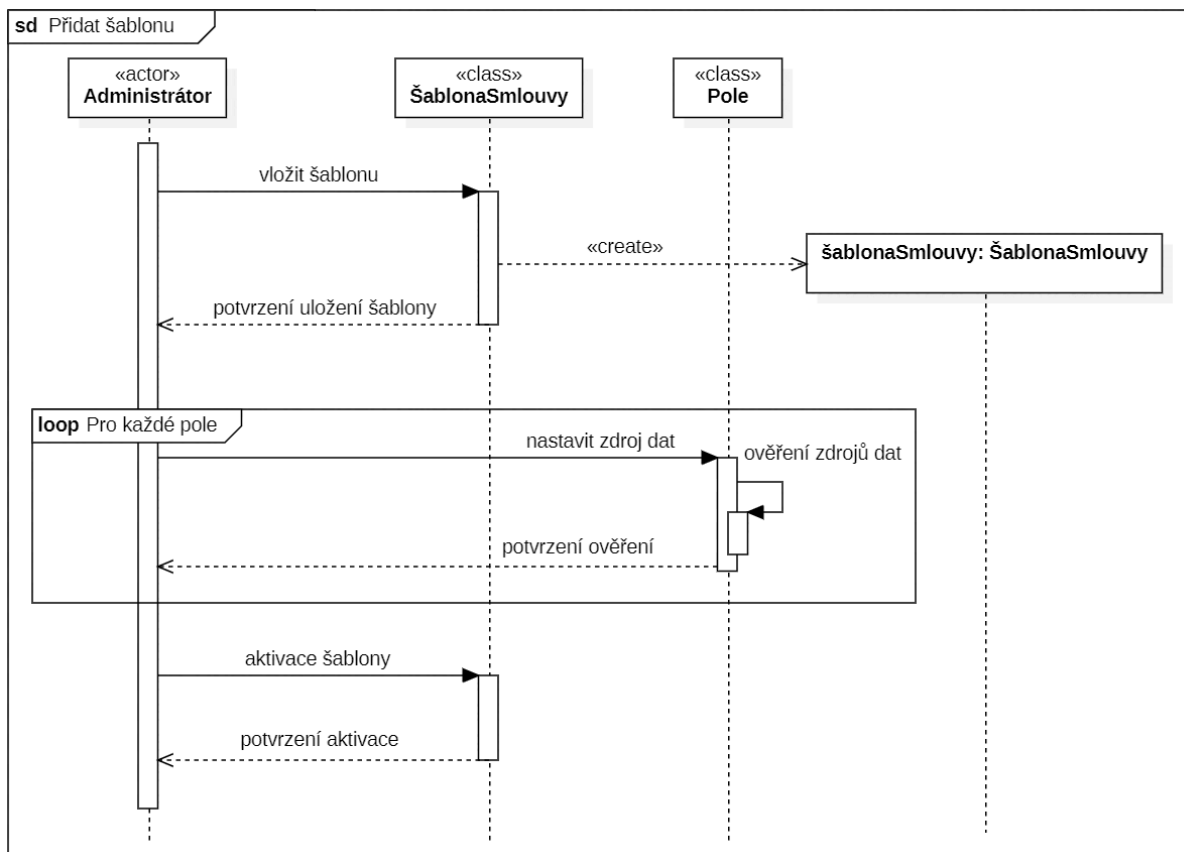
Scénář případu užití *Přidat šablonu* popisuje vložení nové šablony smlouvy do systému a její aktivaci.

Případ užití	Přidat šablonu
ID	UC72
Stručný popis	Administrátor vkládá novou šablonu smlouvy do systému.
Hlavní aktéři	Administrátor
Vstupní podmínky	Administrátor je přihlášen.
Hlavní scénář	<ol style="list-style-type: none">1. Zahrnout (Nastavit pole).2. Scénář začíná v okamžiku, kdy systém otevře formulář pro vložení šablony.3. Administrátor vyplní požadovaná pole, nahraje šablonu a potvrdí formulář.4. Systém uloží šablonu.5. Administrátor definuje zdroje dat pro pole v šabloně.6. Systém ověří zdroje dat.7. Administrátor potvrdí aktivaci šablony.8. Systém aktivuje šablonu.
Výstupní podmínky	Šablona je vložena do systému.
Alternativní scénáře	Chyba formuláře. Chyba v ověření zdrojů dat.

Tabulka 7: Scénář případu užití – Přidat šablonu

Sekvenční diagram na obrázku č. 24 znázorňuje proces vložení šablony do systému, jehož se účastní *Administrátor* jako hlavní aktér a třídy *ŠablonaSmlouvy* a *Pole*. *Administrátor* nejprve vloží šablonu smlouvy do systému, čímž je vytvořena nová instance třídy *ŠablonaSmlouvy*.

Administrátor následně definuje zdroje dat pro pole v šabloně, po úspěšném ověření zdrojů dat těchto polí může šablonu aktivovat. Odebrání šablony ze systému probíhá analogicky, tzn. odesláním zprávy se stereotypem `<<destroy>>` odstraňovanému objektu.



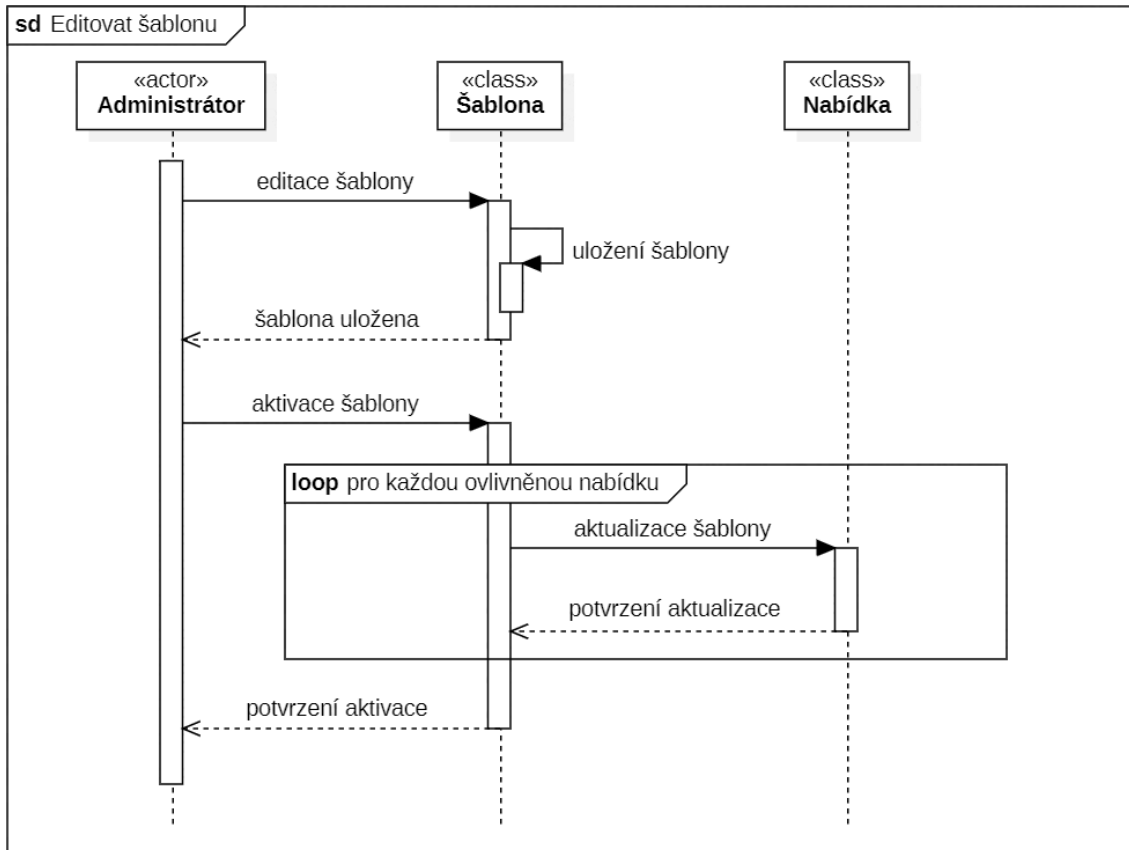
Obrázek 24: Sekvenční diagram – Přidat šablonu

Potom, co je šablona v systému aktivována, je možné ji přiřadit k nabídkám v systému. Pokud dojde ke změně v textaci vzoru smlouvy, lze šablonu v systému editovat. Tento proces popisuje scénář případu užití *Editovat šablonu*.

Případ užití	Editovat šablonu
ID	UC74
Stručný popis	Administrátor edituje šablonu, která je již vložena do systému.
Hlavní aktéři	Administrátor
Vstupní podmínky	Administrátor je přihlášen. Šablona je vložena do systému.
Hlavní scénář	<ol style="list-style-type: none"> 1. Zahrnout (Vyhledat šablonu). 2. Scénář začíná v okamžiku, kdy systém otevře formulář pro editaci šablony. 3. Administrátor vyplní požadovaná pole, nahraje šablonu a potvrdí formulář. 4. Systém uloží šablonu do systému a odešle zprávu o výsledku akce. 5. Uživatel potvrdí aktivaci šablony. 6. Systém aktivuje šablonu a provede aktualizaci u všech nabídek, ke kterým je šablona přiřazena. 7. Systém odešle zprávu o výsledku akce.
Výstupní podmínky	Šablona je editována. Aktualizují se všechny nabídky, ke kterým je šablona přiřazena.
Alternativní scénáře	Chyba formuláře. Šablona není přiřazena k žádné nabídce. Chyba aktualizace šablony u nabídek.

Tabulka 8: Scénář případu užití – Editovat šablonu

Proces editace šablony popisuje sekvenční diagram na obrázku č. 25. Editaci šablony provádí administrátor. Po uložení a aktivaci editované šablony provede systém aktualizaci šablony u všech ovlivněných nabídek. Editaci šablony lze provést, jen pokud nedochází ke změnám polí v šabloně. Při změně polí je nutné vložit šablonu do systému jako novou šablonu.



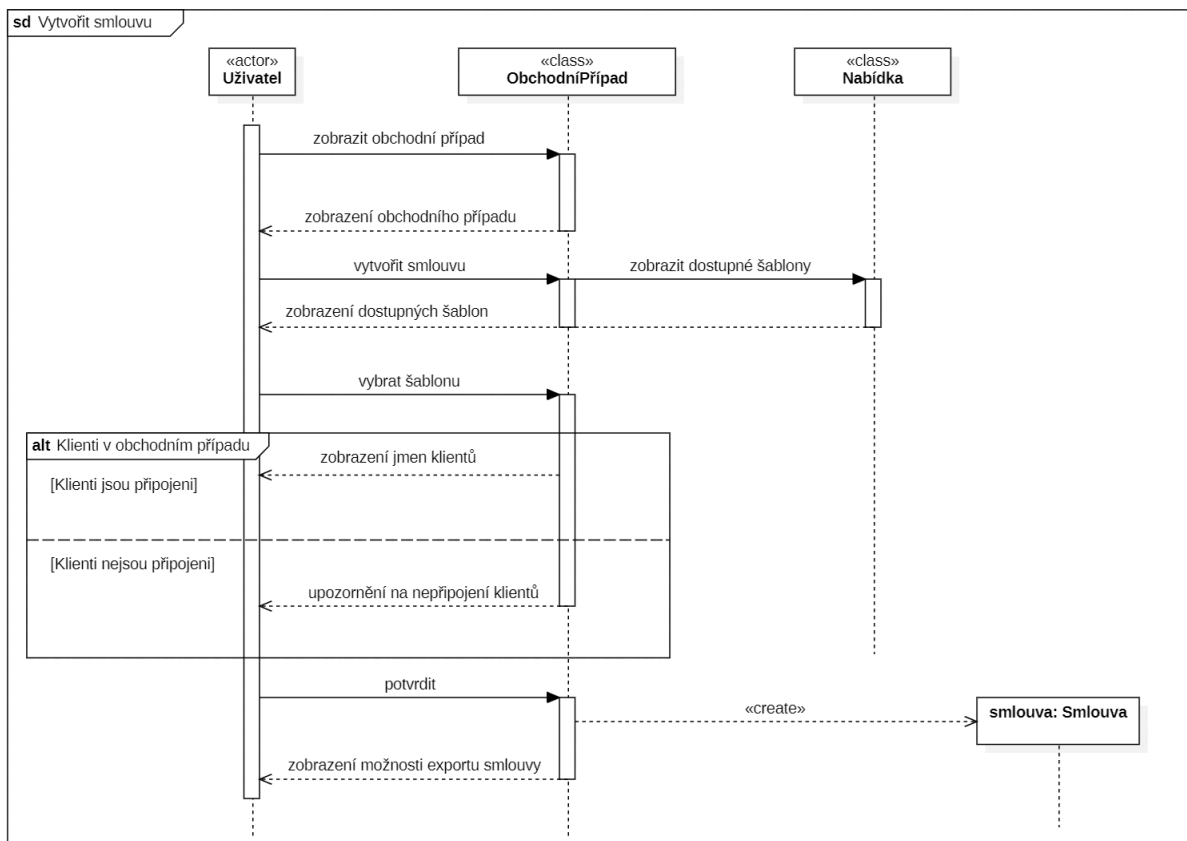
Obrázek 25: Sekvenční diagram – Editovat šablonu

Pokud je šablona přiřazena k nabídce a tatáž nabídka je zároveň součástí obchodního případu, lze vygenerovat ze šablony hotovou smlouvu. Tento proces popisuje scénář případu užití *Vytvořit smlouvu*.

Případ užití	Vytvořit smlouvu
ID	UC77
Stručný popis	Uživatel vytváří novou smlouvu ze šablony v systému.
Hlavní aktéři	Uživatel
Vstupní podmínky	Uživatel je přihlášen. Uživatel má přidělena požadovaná oprávnění. K nabídce je přiřazená šablona. Je založen obchodní případ.
Hlavní scénář	<ol style="list-style-type: none"> 1. Zahrnout (Vyhledat obchodní případ). 2. Scénář začíná v okamžiku, kdy systém zobrazí obchodní případ. 3. Uživatel pošle požadavek o vytvoření nové smlouvy. 4. Systém zobrazí dostupné šablony pro nabídku v obchodním případě. 5. Uživatel vybere šablonu a potvrdí výběr. 6. KDYŽ jsou k obchodnímu případu připojeni klienti. <ol style="list-style-type: none"> 6.1. Systém zobrazí jména klienta(ů) přiřazených k obchodnímu případu. 7. JINAK k obchodnímu případu nejsou připojeni klienti. <ol style="list-style-type: none"> 7.1. Systém upozorní uživatele. 8. Uživatel potvrdí vytvoření smlouvy. 9. Systém vytvoří smlouvu. 10. Systém nabídne export smlouvy do textového editoru.
Výstupní podmínky	Smlouva je vytvořená. Smlouvu je možné exportovat do textového editoru.
Alternativní scénáře	Chyba vytvoření smlouvy.

Tabulka 9: Scénář případu užití – Vytvořit smlouvu

Sekvenční diagram na obrázku č. 26 popisuje proces vytvoření nové smlouvy, jehož se účastní *Uživatel* jako hlavní aktér, který komunikuje s třídami *ObchodníPřípad* a *Nabídka*. *Uživatel* nejprve zašle zprávu pro vytvoření smlouvy obchodnímu případu, ten následně vyžádá seznam dostupných šablon od nabídky. Uživatel vybere šablonu a následně potvrdí vytvoření nové smlouvy. Nová smlouva je vytvořena zasláním zprávy se stereotypem <<create>>, tedy obdobně jako u případu užití *Vložit šablonu*. Po vytvoření smlouvy může *Uživatel* smlouvu exportovat do textového editoru, což zachycuje rozšiřující případ užití *Export smlouvy do textového editoru*.



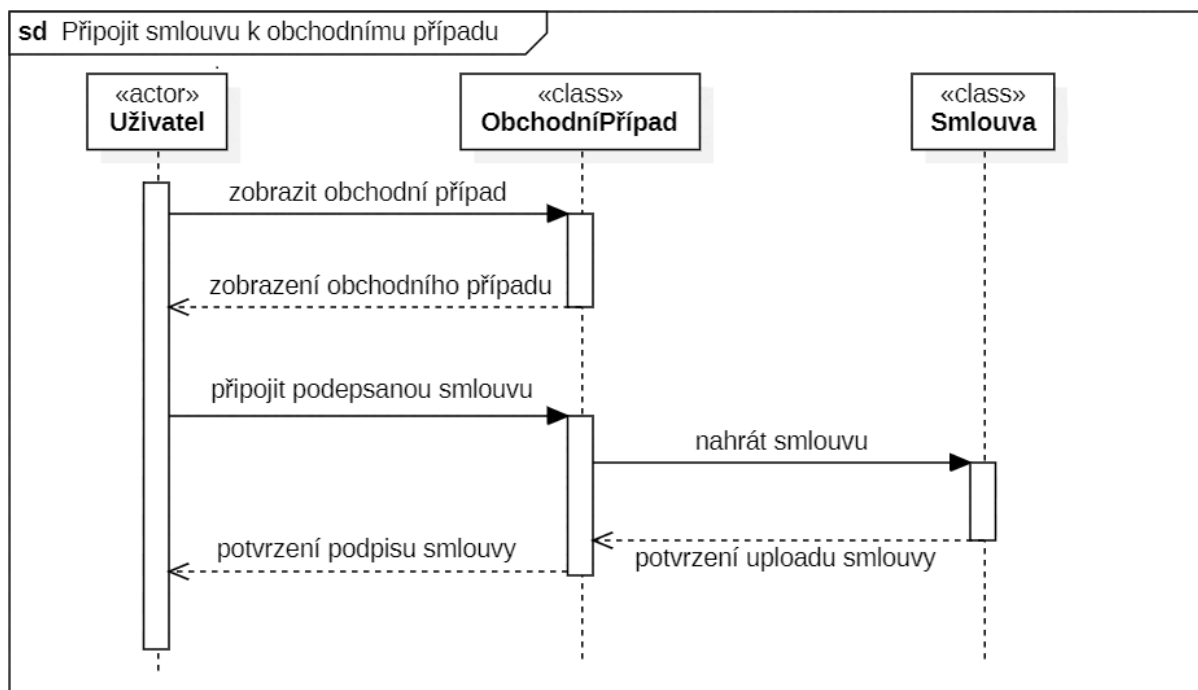
Obrázek 26: Sekvenční diagram – Vytvořit smlouvu

Podpisem smlouvy je realizován obchod. Podepsanou smlouvu je možné připojit k obchodnímu případu a zároveň tak evidovat podpis smlouvy. Tento proces popisuje scénář případu užití *Připojit smlouvu k obchodnímu případu*.

Případ užití	Připojit smlouvu k obchodnímu případu
ID	UC79
Stručný popis	Uživatel připojuje podepsanou smlouvu k obchodnímu případu.
Hlavní aktéři	Uživatel
Vstupní podmínky	Uživatel je přihlášen. Uživatel má přidělená požadovaná oprávnění.
Hlavní scénář	<ol style="list-style-type: none"> 1. Zahrnout (Vyhledat obchodní případ). 2. Scénář začíná v okamžiku, kdy systém zobrazí obchodní případ. 3. Uživatel pošle požadavek o připojení podepsané smlouvy. 4. Systém otevře formulář pro připojení smlouvy. 5. Uživatel vyplní požadovaná pole, nahraje kopii smlouvy a potvrdí formulář. 6. Systém informuje uživatele o úspěšném provedení akce.
Výstupní podmínky	Smlouva je připojena k obchodnímu případu. Podpis smlouvy je evidován v obchodním případě.
Alternativní scénáře	Chyba formuláře.

Tabulka 10: Scénář případ užití – Připojit smlouvu k obchodnímu případu

Proces připojení smlouvy k obchodnímu případu je zobrazen sekvenčním diagramem na obrázku č. 27. Po připojení smlouvy do obchodního případu nahrazuje podepsaná smlouva původní vygenerovanou smlouvu, zároveň se mění stav obchodního případu na podepsaný a stav nabídky na prodaná, což popisují rozšiřující případy užití *Změnit stav obchodního případu* a *Změnit stav nabídky*.

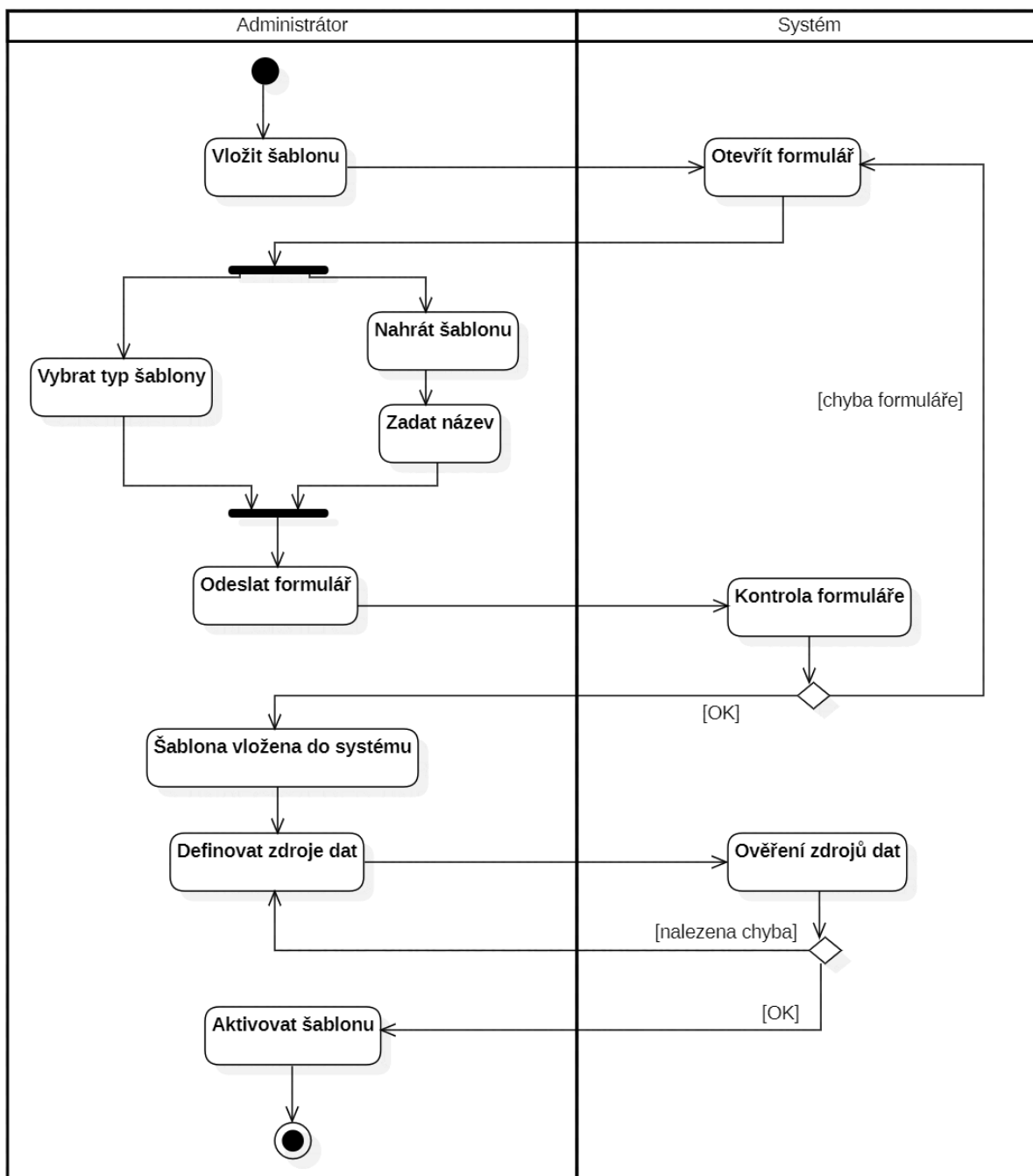


Obrázek 27: Sekvenční diagram – Připojit smlouvu k obchodnímu případu

Model aktivit

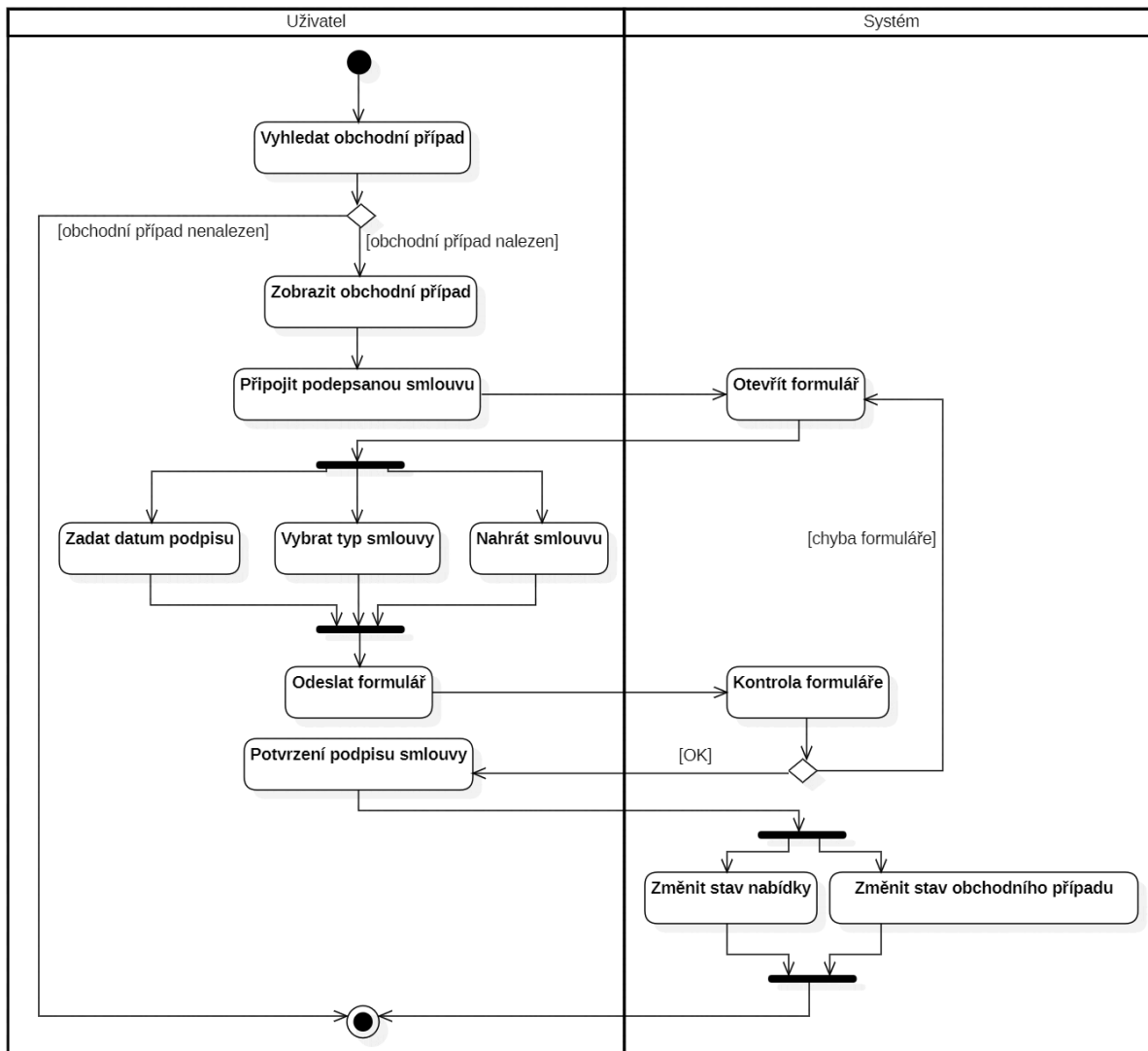
Nejdetailnějším modelem, zachycujícím interakce v systému, je model aktivit, který popisuje kroky složitých procesů v systému.

Diagramy aktivit jsou vytvořeny pro procesy *Vložení šablony* a *Připojení smlouvy k obchodnímu případu*. Oba tyto procesy byly již částečně zachyceny v sekvenčních diagramech. Diagramy aktivit však zahrnují více případů užití současně, popisují detailně sled kroků, zachycují konkurenční toky a naznačují také případné alternativní scénáře.



Obrázek 28: Diagram aktivit – Vložení šablony do systému

Na obrázku č. 28 je zobrazen diagram aktivit popisující proces vložení šablony smlouvy do systému. Oproti sekvenčnímu diagramu zobrazuje detailněji postup nahrání šablony. Větvení diagramu pak naznačuje možné alternativní scénáře, které by mohly při vkládání šablony nastat.



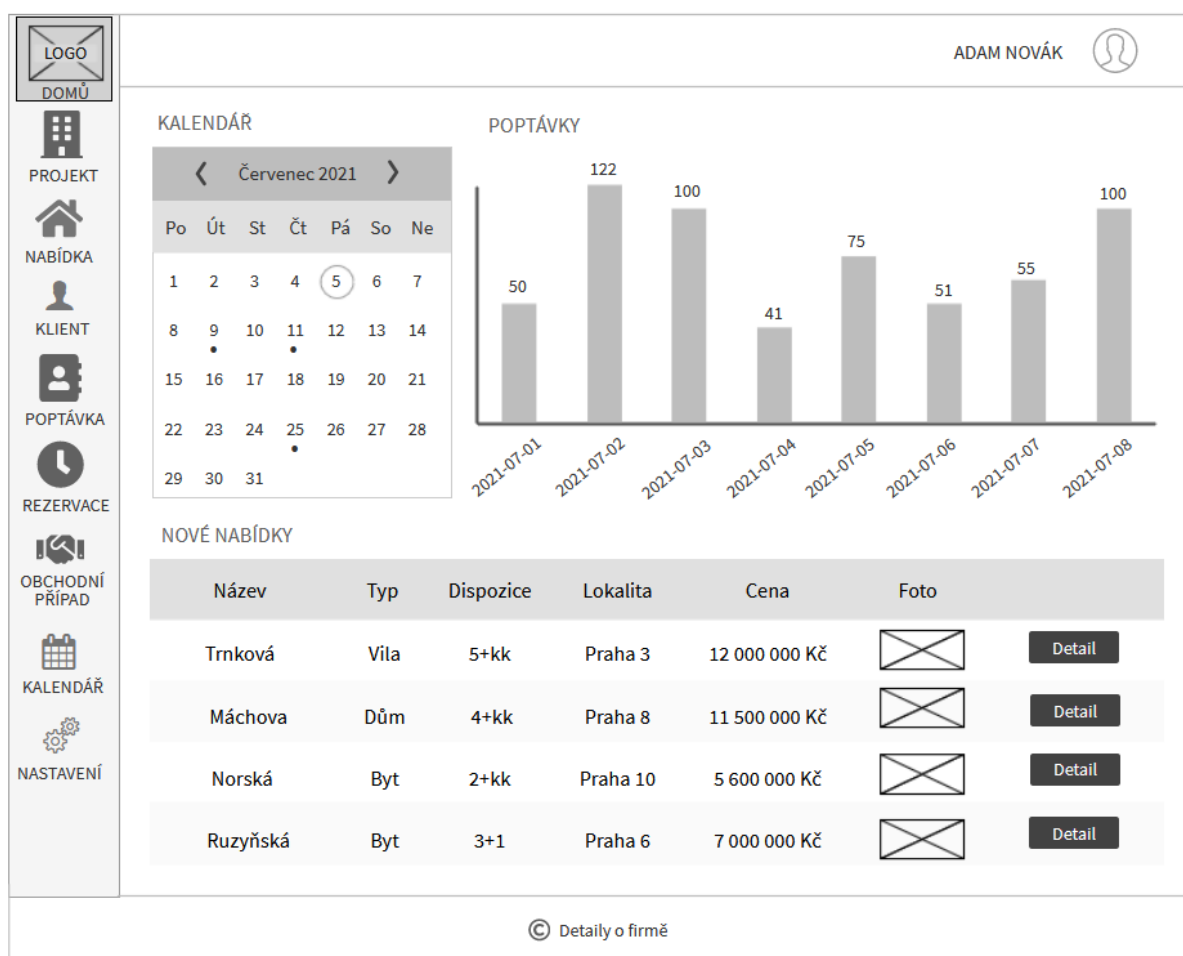
Obrázek 29: Diagram aktivit – Připojení smlouvy k obchodnímu případu

Diagram aktivit na obrázku č. 29 popisuje proces připojení smlouvy k obchodnímu případu. Tento diagram zachycuje případy užití *Vyhledat obchodní případ*, *Připojit smlouvu k obchodnímu případu*, *Změnit stav obchodního případu* a *Změnit stav nabídky*. Stejně jako předchozí diagram naznačuje možné větvení hlavního scénáře. Zároveň je zde zásadní také zobrazení konkurenčních aktivit.

4.4.4 Wireframes

Cílem wireframů je rozvrhnout obsah jednotlivých stránek na webu, tak aby bylo jasné, jaký obsah bude na dané stránce, jakou bude mít vizuální prioritu a jaké budou vztahy mezi částmi obsahu. Wireframy ve své podstatě představují jakousi kostru webu (Řezáč, 2016).

Typicky jsou wireframy tvořeny jen čarami a textem, je však možné vytvořit wireframy s větší mírou detailu, díky kterým lze získat lepší představu o podobě finální aplikace. Tyto typy wireframů byly využity i v této práci.



Obrázek 30: Wireframe – Domovská stránka

Obrázek č. 30 zachycuje domovskou stránku navrhovaného systému, která se uživateli zobrazí po úspěšném přihlášení. Domovská stránka obsahuje vybrané aktuality, tj. kalendář uživatele, grafické zobrazení počtu nových poptávek v průběhu posledních dní a seznam nově

přidaných nabídek. Tento výběr aktualit je pouze výchozí a lze ho modifikovat dle preferencí konkrétního zákazníka. Na pravé straně horní lišty jsou uvedeny iniciály přihlášeného uživatele. V patičce jsou pak uvedeny podrobné informace o firmě zákazníka. Procházení jednotlivých stránek je umožněno prostřednictvím vertikálního menu v levé části obrazovky.

Po kliknutí na položku menu „Obchodní případ“ je uživatel přesměrován na stránku pro obchodní případy, jejíž wireframe je zobrazen na obrázku č. 31. Tato stránka zachycuje kompletní správu obchodních případů v systému. Horní část stránky je vyhrazena pro vyhledávání obchodních případů podle různých kritérií, dolní část stránky obsahuje konkrétní vyhledané obchodní případy, které je možné exportovat do tabulkového editoru.

LOGO ADAM NOVÁK

DOMŮ
PROJEKT
NABÍDKA
KLIENT
POPTÁVKA
REZERVACE
OBCHODNÍ PŘÍPAD
KALENDÁŘ
NASTAVENÍ

Hledat Přidat nový

Stav: aktivní podepsaný zaplacený stornovaný uzavřený

Založen od: do:

Cena bez DPH od: do:

Cena s DPH od: do:

Prodejce:

Klient:

OBCHODNÍ PŘÍPAD Označit vše Exportovat seznam

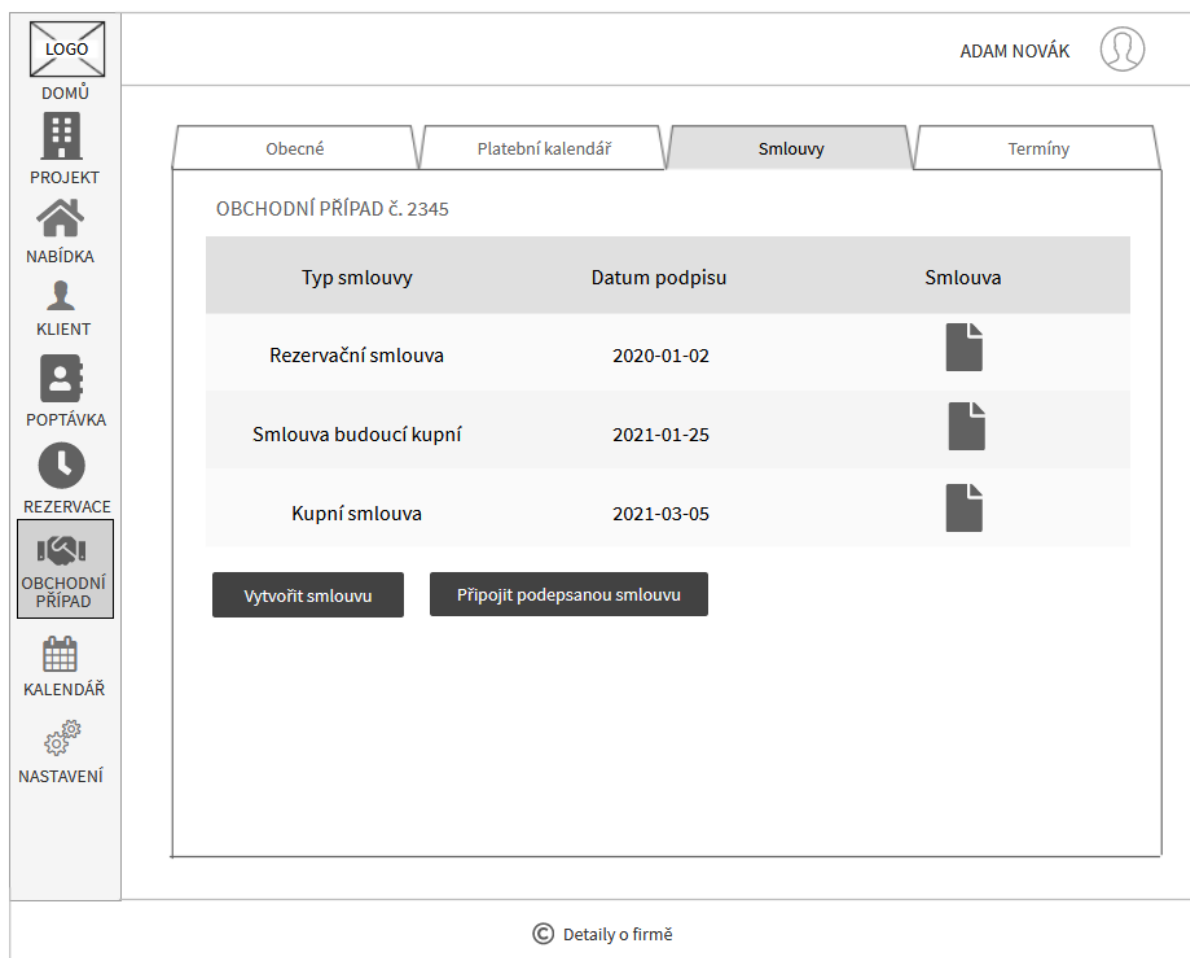
	ID	Založen	Cena bez DPH	Cena s DPH	Stav	Prodejce	Klient	
<input checked="" type="checkbox"/>	1245	2020-01-02	3 455 678 Kč	5 000 000 Kč	zaplacený	Jan Dvořák	Tomáš Pánek	Detail
<input checked="" type="checkbox"/>	3456	2021-01-25	5 667 899 Kč	7 000 000 Kč	stornovaný	Jan Dvořák	Jan Nový	Detail
<input checked="" type="checkbox"/>	6786	2021-03-05	3 356 788 Kč	5 600 000 Kč	aktivní	Adéla Malá	Jiří Velký	Detail
<input checked="" type="checkbox"/>	3457	2021-03-05	6 878 990 Kč	8 000 000 Kč	podepsaný	Jana Bohatá	Sylva Krátká	Detail

© Detaily o firmě

Obrázek 31: Wireframe – Správa obchodních případů

Poslední wireframe, který lze vidět na obrázku č. 32, zobrazuje detail obchodního případu, konkrétně část týkající se smluv, již se zabýval také model interakcí. Na wireframu lze vidět

způsob zaznamenání podepsaných smluv a obsahuje také tlačítka pro vytvoření a připojení nové smlouvy.

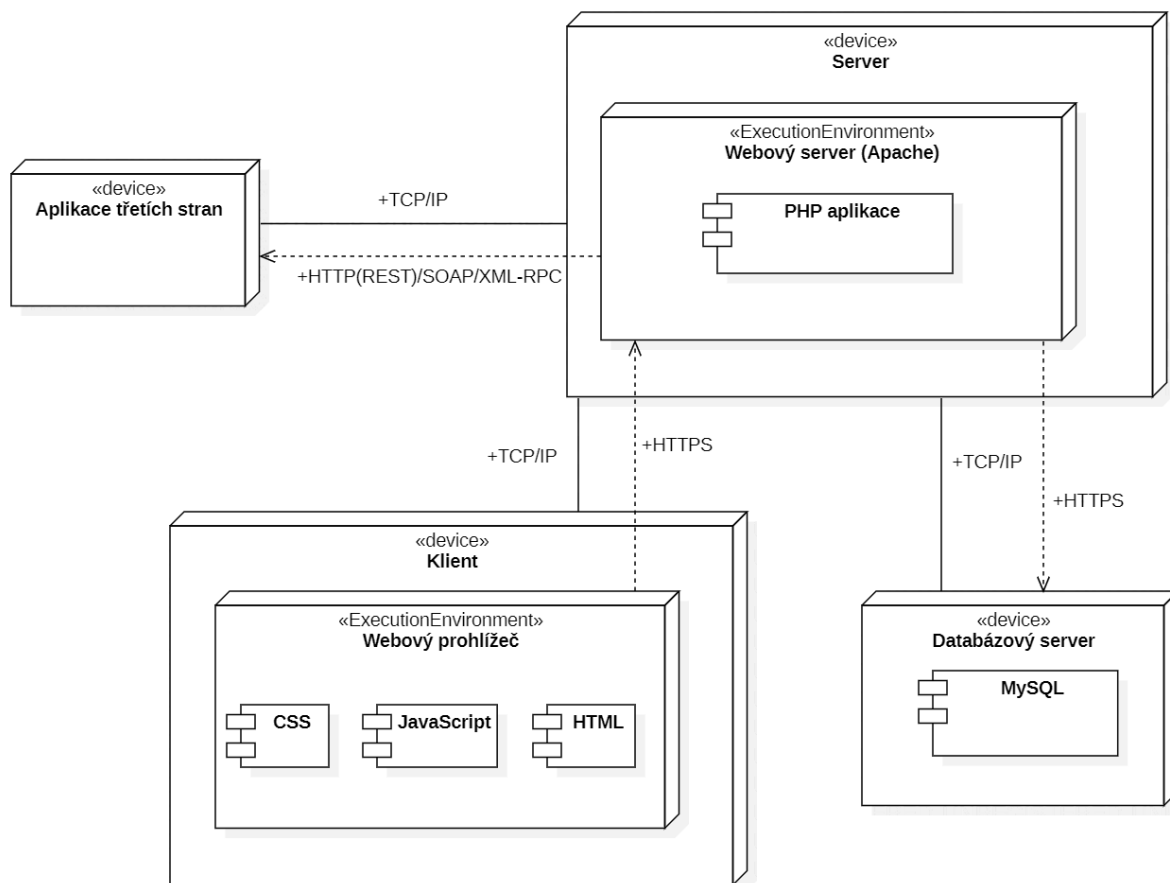


Obrázek 32: Wireframe – Detail obchodního případu

Jelikož obsahem požadavku *N05 – Respektování zásad použitelnosti* bylo, aby výsledná aplikace byla responzivní, byly vytvořeny stejné wireframes také pro zobrazení v mobilním telefonu. Tyto wireframes lze vidět v přílohách této práce. Wireframes pro ostatní části systému nebyly vytvářeny, neboť jejich podoba by byla obdobná.

4.4.5 Návrh realizace

Součástí návrhu systému je i UML diagram nasazení, který zachycuje fyzickou architekturu systému. Tento diagram zároveň představuje návrh realizace systému splňující stanovené nefunkční požadavky na systém.



Obrázek 33: Diagram nasazení

Jak je patrné z diagramu, autorka doporučuje realizovat systém na principu třívrstvé architektury. Prezenční vrstva zajišťuje vykreslení grafického rozhraní uživateli, což je realizováno přes webový prohlížeč, který používá technologie HTML, CSS a JavaScript. Prezenční vrstva komunikuje pomocí protokolu HTTPS s webovým serverem. Navrhovaný systém využívá webový server Apache a je naprogramován v jazyce PHP. Webový server komunikuje s databází MySQL, která je umístěná na samostatném databázovém serveru. Pro komunikaci s aplikacemi třetích stran (např. s kalendářními aplikacemi nebo realitními servery) jsou využívány webové služby. Komunikace probíhá prostřednictvím protokolů XML-RPC, SOAP nebo prostřednictvím rozhraní REST API. Výběr konkrétního způsobu komunikace závisí na konkrétní aplikaci.

5 Výsledky a diskuse

V rámci této kapitoly bude provedeno zhodnocení návrhu systému a následně zde budou popsány některé problémy, se kterými se autorka potýkala při zpracování návrhu.

5.1 Zhodnocení návrhu

Vzhledem k faktu, že v rámci diplomové práce byl vytvořen návrh systému, který nebyl v rámci této práce implementován, nelze systém hodnotit např. pomocí ISO norem jakosti nebo pomocí jiných metodik hodnocení kvality informačních systémů, neboť by nemohly být zhodnoceny všechny aspekty systému, jako např. výkon, latence apod. Zhodnocení bude tedy provedeno bez využití standardizované metodiky.

Návrh systému byl vytvořen na základě stanovených funkčních a nefunkčních požadavků. Ověření splnění všech definovaných funkčních požadavků lze vyhodnotit poměrně jednoduše. Každý funkční požadavek by měl být realizován nějakým případem užití. Z ověření vyplynulo, že každý funkční požadavek je realizován případem užití, tím pádem je systém plně funkční. Ověření nefunkčních požadavků je komplikovanější, neboť většinu nefunkčních požadavků lze vyhodnotit až po implementaci systému. Díky navrženým wireframes lze částečně zhodnotit použitelnost systému. Při návrhu byl brán maximální zřetel na přehlednost a jednoduchost systému. Wireframes zobrazují přehledné menu systému, díky němuž bude orientace v systému pro uživatele snadná. Wireframes jsou navrženy také pro mobilní zařízení, čímž je splněna dílčí část požadavků *N05 – Respektování zásad použitelnosti – responzivní design*. V rámci nefunkčních požadavků je zároveň počítáno také s budoucím rozšířením systému, konkrétně o funkce týkající se podpory procesu pronájmu nemovitostí. Díky tomu, že je systém navržen jako objektově orientovaný, není problém systém v budoucnu rozšířit o další objekty a funkce.

V rámci návrhu systému byly vytvořeny tři modely – model tříd, stavový model a model interakcí. Byly vybrány nejdůležitější části systému, které jsou popsány vybranými diagramy UML. Modely jsou zároveň částečně zjednodušené, což ale může být naopak výhodou, protože finální návrh systému je ve výsledku plně univerzální a jde zde ponechán prostor pro detailní rozpracování a volnost pro implementaci. Výsledný návrh tak může být velmi dobrým základem pro implementaci obdobných systémů. Největší přednost navrženého systému vidí

autorka v detailním popisu automatizovaného procesu přípravy smluv, neboť tuto funkci žádné z konkurenčních řešení nenabízí.

5.2 Diskuse

V rámci zpracování praktické části práce byl řešen problém, který zde bude podrobněji popsán. Tato problematická část se týká správy šablon smluv a správy smluv, konkrétně vytvoření smlouvy v systému. V rámci návrhu bylo zvažováno několik možných řešení, jak tuto funkci v systému realizovat.

Jedním z možných řešení bylo, že v systému bude umístěn samostatný modul smlouvy, kde budou uloženy všechny šablony smluv a uživatel bude moct po výběru šablony, klienta a nabídky smlouvu vygenerovat. Toto řešení je nejjednodušší z hlediska návrhu, ale zároveň není příliš uživatelsky přívětivé, jelikož pro vygenerování smlouvy je uživatel nucen vyhledat šablonu smlouvy a k ní přiřadit klienta a nabídku. Následně pak opět provádí přiřazení klienta a nabídky v rámci přidání obchodního případu.

Druhé řešení, které bylo nakonec realizováno, spočívá v tom, že uživatel bude generovat smlouvy až po vytvoření obchodního případu přímo v detailu obchodního případu. Toto řešení bude výrazně efektivnější především v případě, kdy v rámci obchodního případu bude vytvářeno více typů smluv. Úskalí tohoto řešení je, že nabídka bude vázána v obchodním případě, aniž by byl obchodní případ realizován. Ačkoliv při vytváření smlouvy se již částečně předpokládá úspěšné uzavření obchodu, může se stát, že obchod nakonec realizován nebude. Tato situace byla v návrhu vyřešena tak, že obchodní případ, ke kterému není dosud připojena smlouva, lze ze systému jednoduše smazat i bez účasti administrátora.

6 Závěr

Hlavním cílem této diplomové práce byla analýza a návrh informačního systému v UML, který bude poskytovat podporu při zprostředkování služeb v oblasti prodeje nemovitostí.

V teoretické části práce byla na základě studia odborných informačních zdrojů zpracována teoretická východiska. Byla přiblížena problematika týkající se druhů informačních systémů, životního cyklu informačních systémů, analýzy uživatelských požadavků a přístupů a metodik k tvorbě informačních systémů. Na závěr této části byly popsány standardy jazyka UML. Zpracování teoretických východisek bylo nezbytné pro správné uchopení praktické části práce, při jejíž tvorbě byly využity získané znalosti. Zároveň byl tímto splněn jeden z dílčích cílů práce.

V praktické části práce byla provedena analýza a návrh informačního systému, při kterých autorka částečně vycházela také ze svých profesních zkušeností. Nejprve byl navrhovaný systém stručně popsán. Bylo stanoveno, že klíčovou podstatou systému je efektivní podpora obchodních procesů při prodeji nemovitostí a zároveň snížení administrativní náročnosti nezbytných činností při uzavírání obchodů. Jelikož je navrhovaný systém určen pro volný trh a není řešením na míru pro jednoho zadavatele, byl systém porovnán s dostupnými konkurenčními řešeními na současném trhu. Tato předběžná analýza byla zpracována především za účelem ověření, zda má navrhovaný systém vůbec šanci uspět na trhu a má tedy smysl ho navrhovat. Byla provedena vícekritériální analýza popsaných řešení, ze které vyplynulo, že navrhovaný systém je ze všech popsaných řešení nejlepší. Navíc bylo zjištěno, že žádné z konkurenčních řešení oproti navrhovanému systému nenabízí automatizaci procesu přípravy smluv, která je přitom velmi časově náročná. Pozdní odhalení chyb ve smlouvách může mít navíc za následek neúspěch obchodu. Automatizace procesu přípravy smluv tedy představuje hlavní výhodu navrhovaného systému oproti konkurenčním řešením.

V rámci analytické fáze byla zpracována analýza uživatelských požadavků, která byla zároveň dílčím cílem práce. V rámci této analýzy byly na základě pohovorů s potenciálními uživateli systému zjištěny uživatelské požadavky. Takto zjištěné požadavky byly následně upřesněny autorkou práce. Finální požadavky byly rozděleny do dvou kategorií – na funkční požadavky a nefunkční požadavky. Na základě stanovených požadavků byl vytvořen návrh systému v modelovacím jazyce UML.

V rámci návrhu byl vytvořen model tříd, stavový model a model interakcí navrhovaného systému. Vytvoření těchto tří modelů bylo zároveň dílčím cílem práce. Model tříd je zachycen diagramem tříd, který je doplněn datovým slovníkem. Datový slovník obsahuje popis jednotlivých tříd a jejich rozsahu v rámci systému. Stavový model je zachycen stavovými diagramy. Stavové diagramy byly vytvořeny pouze pro třídy, které vykazují zajímavé chování a jejich tvorba je pro model přínosem. Posledním vytvořeným modelem systému je model interakcí, který zobrazuje spolupráci objektů v systému. Model interakcí je zachycen diagramy případů užití, sekvenčními diagramy a diagramy aktivit. Každý z těchto diagramů zobrazuje objekty systému v jiné úrovni abstrakce. Model interakcí se zaměřuje na případy užití Správy smluv a Správy šablon, jež zachycují funkcionalitu, která představuje hlavní výhodu navrhovaného systému oproti konkurenčním řešením.

V závěru praktické části práce byly vytvořeny wireframes některých částí systému pro lepší představu o výsledné podobě systému. Jelikož jedním z nefunkčních požadavků bylo, aby měl systém responzivní design, byly navrženy wireframes jak pro desktopovou, tak pro mobilní verzi systému. Nakonec byly formulovány doporučení pro implementaci systému. Autorka doporučila implementovat systém jako webovou aplikaci založenou na třívrstvé architektuře. Fyzickou architekturu systému zobrazuje diagram nasazení.

Výsledný návrh systému je s přihlédnutím k rozsahu diplomové práce dle názoru autorky dostatečně detailní. Model sice nepopisuje všechny případy užití, na druhou stranu klíčové případy užití Správa smluv a Správa šablon jsou v práci detailně popsány. Pokud by byly jednotlivé části modelu detailněji rozpracovány a model by byl doplněn o algoritmy hlavních funkcí, bylo by možné systém implementovat. Hlavní cíl práce lze dle názoru autorky považovat za splněný.

Výsledný model systému je univerzální a lze jej použít pro návrh jakéhokoliv obdobného systému. Hlavní předností modelu je pak detailní rozpracování procesů správy smluv a přípravy smluv. Model systému v této podobě lze považovat za kvalitní základ pro detailní zpracování návrhu a následnou implementaci.

7 Seznam použitých zdrojů

AMBLER, Scott W. *The elements of UML 2.0 style*. Cambridg: Cambridge University Press, 2005. ISBN 0-521-61678-6.

ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.

BASL, Josef a Roman BLAŽÍČEK. *Podnikové informační systémy: podnik v informační společnosti*. 3., aktualiz. a dopl. vyd. Praha: Grada, 2012. Management v informační společnosti. ISBN 978-80-247-4307-3.

BOOCH, G., J. RUMBAUGH a I. JACOBSON. *Unified modeling language user guide*. Massachusetts: Addison-Wesley, 1999. ISBN 0-201-57168-4.

BROŽOVÁ, Helena a Milan HOUŠKA. *Základní metody operační analýzy*. Praha: Credit, 2002. ISBN 80-213-0951-2.

BRUCKNER, Tomáš. *Tvorba informačních systémů: principy, metodiky, architektury*. Praha: Grada, 2012. Management v informační společnosti. ISBN 978-80-247-4153-6.

BUCHALCEVOVÁ, Alena a Iva STANOVSKÁ. *Příklady modelů analýzy a návrhu aplikace v UML*. Praha: Oeconomica, 2013. ISBN 978-80-245-1922-7.

BUCHALCEVOVÁ, Alena. *Metodiky vývoje a údržby informačních systémů: kategorizace, agilní metodiky, vzory pro návrh metodiky*. Praha: Grada, 2005. Management v informační společnosti. ISBN 80-247-1075-7.

COAD, P. *Object oriented analysis*. New Jersey: Prentice-Hall, 1991. ISBN 0-13-629981-4.

FIALA, Petr a Miroslav MAŇAS. *Vícekritériální rozhodování: Určeno pro stud. všech fak.* Praha: Vysoká škola ekonomická, 1994. ISBN 80-7079-748-7.

FOWLER, Martin. *Destilované UML*. Praha: Grada, 2009. Knihovna programátora (Grada). ISBN 978-80-247-2062-3.

GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. Podniková informatika. 2., přeprac. a aktualiz. vyd. Praha: Grada, 2009. Expert (Grada). ISBN 978-80-247-2615-1.

GDPR. *GDPR: Obecné nařízení o ochraně osobních údajů prakticky* [online]. [cit. 2021-03-01]. Dostupné z: <https://www.gdpr.cz/gdpr/osobni-udaje/>

CHLAPEK, Dušan, Václav ŘEPA a Iva STANOVSKÁ. Analýza a návrh informačních systémů. Praha: Oeconomica, 2011. ISBN 978-80-245-1782-7.

JABLONSKÝ, Josef. *Operační výzkum: kvantitativní modely pro ekonomické rozhodování*. 3. vyd. Praha: Professional Publishing, 2007. ISBN 978-80-86946-44-3.

KANISOVÁ, Hana a Miroslav MÜLLER. UML srozumitelně. Brno: Computer Press, 2004. ISBN 80-251-0231-9.

KENDALL, K.E. *System Analysis and Design*. Prentice Hall, 1991. ISBN 01-31454-55-2

KRÁL, Jaroslav. *Informační systémy: specifikace : realizace : provoz*. Veletiny: Science, c1998. ISBN 80-86083-00-4.

MELLOR, Stephen. *Agile MDA*, 2004. MDA Journal

MERUNKA, Vojtěch. *Datové modelování*. Praha: Alfa Publishing, 2006. Informatika studium. ISBN 80-86851-54-0.

MOLNÁR, Zdeněk. *Podnikové informační systémy*. Vyd. 2., přeprac. V Praze: České vysoké učení technické, 2009. ISBN 978-80-01-04380-6.

MYSLÍN, Josef. *Scrum: průvodce agilním vývojem softwaru*. Brno: Computer Press, 2016. ISBN 978-80-251-4650-7.

PAGE-JONES, Meilir. *Základy objektově orientovaného návrhu v UML*. Praha: Grada, 2001. *Moderní programování*. ISBN 80-247-0210-X.

POLÁK, Jiří, Antonín CARDA a Vojtěch MERUNKA. *Umění systémového návrhu: objektově orientovaná tvorba informačních systémů pomocí původní metody BORM*. Praha: Grada, 2003. *Management v informační společnosti*. ISBN 80-247-0424-2.

Poski REAL. *Poski REAL úspěšný realitní software* [online]. [cit. 2021-01-16]. Dostupné z: <https://www.realitni-system.com/>

POŽÁR, Josef. *Manažerská informatika*. Plzeň: Vydavatelství a nakladatelství Aleš Čeněk, 2010. ISBN 978-80-7380-276-9.

RÁČEK, Jaroslav. *Strukturovaná analýza systémů*. Brno: Masarykova univerzita, 2006. ISBN 80-210-4190-0.

Realko. *Realitní software* [online]. [cit. 2021-01-16]. Dostupné z: <https://www.realko.cz/>

ŘEPA, Václav. *Analýza a návrh informačních systémů*. Praha: Ekopress, 1999. ISBN 80-86119-13-0.

ŘEZÁČ, Jan. *Web ostrý jako břitva: návrh fungujícího webu pro webdesignery a zadavatele projektů*. Vydání druhé. [Brno]: House of Řezáč, 2016. ISBN 978-80-270-0644-1.

SCHMULLER, Joseph. *Myslíme v jazyku UML*. Praha: Grada, 2001. Knihovna programátora (Grada). ISBN 80-247-0029-8.

SKLENÁK, Vilém. *Data, informace, znalosti a Internet*. Praha: C.H. Beck, 2001. C.H. Beck pro praxi. ISBN 80-7179-409-0.

SODOMKA, Petr. *Informační systémy v podnikové praxi*. Brno: Computer Press, 2006. ISBN 80-251-1200-4.

Softreal. *Realitní program Softreal* [online]. [cit. 2021-01-16]. Dostupné z: <https://www.softreal.cz/>

SOMMERVILLE, Ian. *Softwarové inženýrství*. Brno: Computer Press, 2013. ISBN 978-80-251-3826-7.

ŠMÍD, Vladimír. *Management informačního systému* [online]. Brno, 2005 [cit. 2020-10-10]. Dostupné z: <http://www.fi.muni.cz/~smid/managis.html>. Výukový materiál. Masarykova Univerzita

VRANA, Ivan. Projektování informačních systémů s UML. V Praze: Česká zemědělská univerzita, Provozně ekonomická fakulta, 2008. ISBN 978-80-213-1817-5.

VRANA, Ivan a Karel RICHTA. Zásady a postupy zavádění podnikových informačních systémů: praktická příručka pro podnikové manažery. Praha: Grada, 2005. Management v informační společnosti. ISBN 80-247-1103-6.

VYMĚTAL, Dominik. Informační systémy v podnicích: teorie a praxe projektování. Praha: Grada, 2009. Průvodce (Grada). ISBN 978-80-247-3046-2.

WIEGERS, Karl Eugene. *Požadavky na software*. Brno: Computer Press, 2008. ISBN 978-80-251-1877-1.

YOURDON, E.: Modern structured analysis. Prentice-Hall, Englewood Cliffs, 1989.

8 Přílohy

Příloha A – Vícekriteriální analýza

Kritérium	Body	Váha
Správa účtů	10	0,081967
Správa klientů	10	0,081967
Správa poptávek	10	0,081967
Správa nabídek	10	0,081967
Správa projektů	5	0,040984
Správa obchodních případů	9	0,07377
Evidence plateb	6	0,04918
Provize	6	0,04918
Publikace nabídek na realitních serverech	8	0,065574
Publikace nabídek na webových stránkách	9	0,07377
Topování	5	0,040984
Kalendář	5	0,040984
Kalendář propojený s aplikacemi třetích stran	4	0,032787
Reporty	9	0,07377
Sledování nákladů na inzerci	6	0,04918
Přehled prohlídek	3	0,02459
Automatická příprava smluv	7	0,057377
Suma	122	1

Bodovací metoda

Bodovací škála – 1 až 10 (1 = nejméně, 10 = nejvíce)

	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX
	Správa účtů	Správa klientů	Správa poptávek	Správa nabídek	Správa projektů	Obchodní případy	Evidence plateb	Provize	Publikace na web	Publikace na r. servery	Topování	Kalendář	Kalendář (propojení)	Reporty	Náklady na inzerci	Přehled prohlídek	Příprava smluv
Realko	1	1	1	1	0,8	0,8	0	0,8	1	0,8	0,8	1	1	1	0	1	0
Poski Real	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0
Softreal	1	1	1	1	0,8	0	0	0,8	1	1	0	1	0	1	0,8	0	0
Navrhovaný IS	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1
Váhy	0,08197	0,08197	0,08197	0,08197	0,04098	0,07377	0,04918	0,04918	0,06557	0,07377	0,04098	0,04098	0,03279	0,07377	0,04918	0,02459	0,05738

Metoda TOPSIS – vstupní data

	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX	MAX
	Správa účtů	Správa klientů	Správa poptávek	Správa nabídek	Správa projektů	Obchodní případy	Evidence plateb	Provize	Publikace na web	Publikace na r. servery	Topování	Kalendář	Kalendář (propojení)	Reporty	Náklady na inzerci	Přehled prohlídek	Příprava smluv
Realko	0,50000	0,50000	0,50000	0,50000	0,44173	0,49237	0,00000	0,44173	0,50000	0,41931	0,49237	0,50000	0,57735	0,50000	0,00000	0,70711	0,00000
Poski Real	0,50000	0,50000	0,50000	0,50000	0,55216	0,61546	0,70711	0,55216	0,50000	0,52414	0,61546	0,50000	0,57735	0,50000	0,00000	0,70711	0,00000
Softreal	0,50000	0,50000	0,50000	0,50000	0,44173	0,00000	0,00000	0,44173	0,50000	0,52414	0,00000	0,50000	0,00000	0,50000	1,00000	0,00000	0,00000
Navrhovaný IS	0,50000	0,50000	0,50000	0,50000	0,55216	0,61546	0,70711	0,55216	0,50000	0,52414	0,61546	0,50000	0,57735	0,50000	0,00000	0,00000	1,00000
Váhy	0,08197	0,08197	0,08197	0,08197	0,04098	0,07377	0,04918	0,04918	0,06557	0,07377	0,04098	0,04098	0,03279	0,07377	0,04918	0,02459	0,05738

Metoda TOPSIS – normalizovaná kritériální matice R

	MAX Správa účtů	MAX Správa klientů	MAX Správa poptávek	MAX Správa nabídek	MAX Správa projektů	MAX Obchodní případy	MAX Evidence plateb	MAX Provize	MAX Publikace na web	MAX Publikace na r. servery	MAX Topování	MAX Kalendář	MAX Kalendář (propojení)	MAX Reporty	MAX Náklady na inzerci
Realko	0,04098	0,04098	0,04098	0,04098	0,01810	0,03632	0,00000	0,02172	0,03279	0,03093	0,02018	0,02049	0,01893	0,03689	0,00000
Poski Real	0,04098	0,04098	0,04098	0,04098	0,02263	0,04540	0,03478	0,02716	0,03279	0,03867	0,02522	0,02049	0,01893	0,03689	0,00000
Softreal	0,04098	0,04098	0,04098	0,04098	0,01810	0,00000	0,00000	0,02172	0,03279	0,03867	0,00000	0,02049	0,00000	0,03689	0,04918
Navrhovaný IS	0,04098	0,04098	0,04098	0,04098	0,02263	0,04540	0,03478	0,02716	0,03279	0,03867	0,02522	0,02049	0,01893	0,03689	0,00000
Váhy	0,08197	0,08197	0,08197	0,08197	0,04098	0,07377	0,04918	0,04918	0,06557	0,07377	0,04098	0,04098	0,03279	0,07377	0,04918
Ideální	0,04098	0,04098	0,04098	0,04098	0,02263	0,04540	0,03478	0,02716	0,03279	0,03867	0,02522	0,02049	0,01893	0,03689	0,04918
Bazální	0,04098	0,04098	0,04098	0,04098	0,01810	0,00000	0,00000	0,02172	0,03279	0,03093	0,00000	0,02049	0,00000	0,03689	0,00000

MAX Přehled prohlídek	MAX Příprava smluv	di+	di-	ci
0,01739	0,00000	0,08449	0,04886	0,36641
0,01739	0,00000	0,07557	0,06839	0,47507
0,00000	0,00000	0,08894	0,04978	0,35888
0,00000	0,05738	0,05216	0,08756	0,62667
0,02459	0,05738			
0,01739	0,05738			

Metoda TOPSIS – vážená kritériální matice W

Příloha B – Případy užití

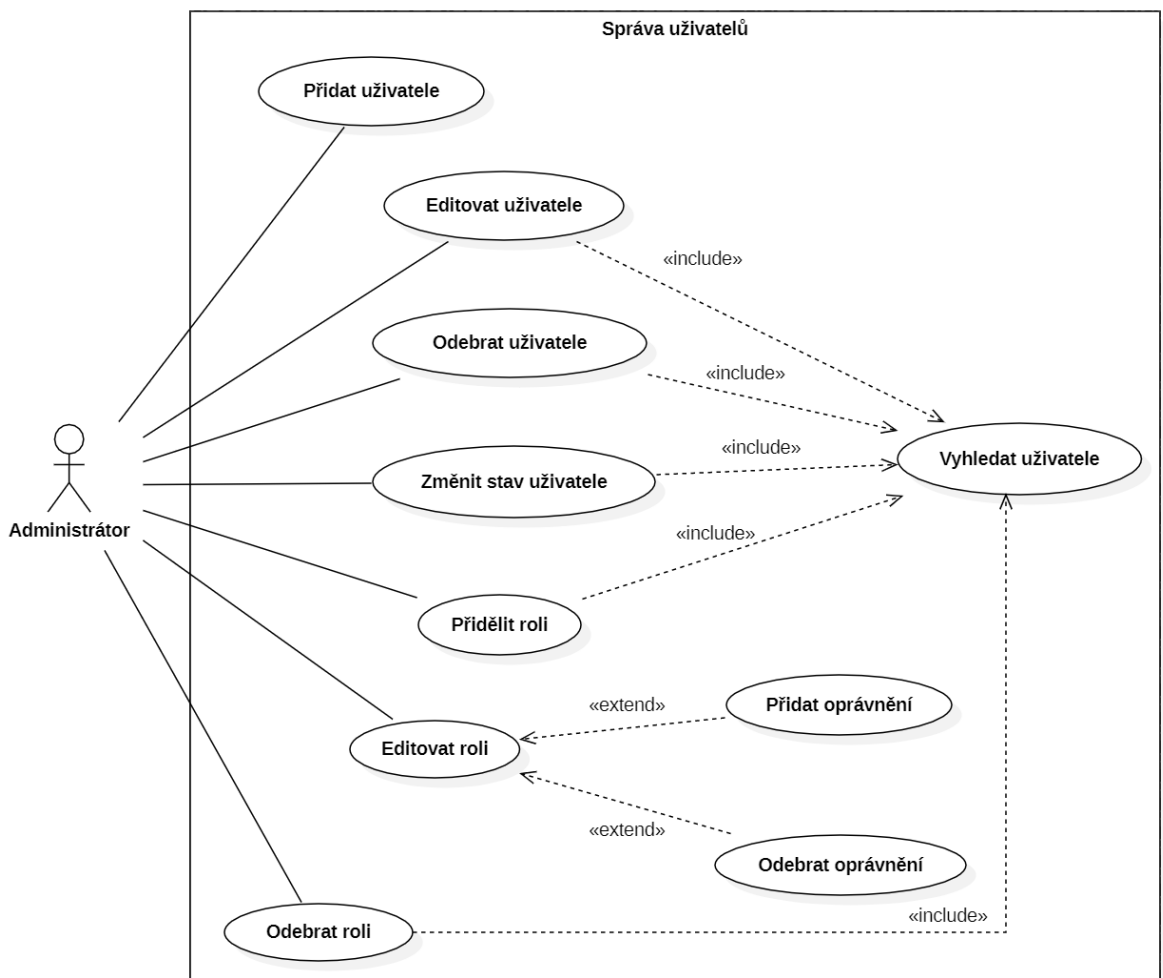
ID	Název případu užití
UC01	Přihlášení do systému
UC02	Správa uživatelů
UC03	Kalendář
UC04	Správa nabídek
UC05	Správa projektů
UC06	Správa klientů
UC07	Správa rezervací
UC08	Správa poptávek
UC09	Správa obchodních případů
UC10	Správa smluv
UC11	Správa šablon
UC12	Reporty
UC13	Přidat uživatele
UC14	Vyhledat uživatele
UC15	Editovat uživatele
UC16	Odebrat uživatele
UC17	Změnit stav uživatele
UC18	Přidělit roli
UC19	Editovat roli
UC20	Odebrat roli
UC21	Přidat oprávnění
UC22	Odebrat oprávnění
UC23	Naplánovat událost
UC24	Vyhledat událost
UC25	Editovat událost
UC26	Odebrat událost
UC27	Odeslat data do osobního kalendáře
UC28	Přidat nabídku
UC29	Přiřadit prodejce
UC30	Vyhledat nabídku
UC31	Publikovat nabídku
UC32	Editovat nabídku
UC33	Změnit stav nabídky
UC34	Topovat nabídku
UC35	Odebrat nabídku
UC36	Přidat projekt

UC37	Vyhledat projekt
UC38	Publikovat projekt
UC39	Editovat projekt
UC40	Změnit stav projektu
UC41	Topovat projekt
UC42	Přiřadit nabídku k projektu
UC43	Odebrat projekt
UC44	Přidat klienta
UC45	Vyhledat klienta
UC46	Editovat klienta
UC47	Odebrat klienta
UC48	Vytvořit rezervaci
UC49	Vyhledat rezervaci
UC50	Prodloužit rezervaci
UC51	Smazat rezervaci
UC52	Přidat poptávku
UC53	Vyhledat poptávku
UC54	Editovat poptávku
UC55	Odebrat poptávku
UC56	Přidat obchodní případ
UC57	Vyhledat obchodní případ
UC58	Editovat obchodní případ
UC59	Stornovat obchodní případ
UC60	Uzavřít obchodní případ
UC61	Odstranit obchodní případ
UC62	Změnit stav obchodního případu
UC63	Evidovat platbu
UC64	Zadat výši DPH
UC65	Vypočítat cenu
UC66	Nastavit platební kalendář
UC67	Nastavit termíny
UC68	Odeslat data do kalendáře
UC69	Zadat výši provize
UC70	Vypočítat provizi
UC71	Nastavit pole
UC72	Přidat šablonu
UC73	Vyhledat šablonu

UC74	Editovat šablonu
UC75	Odebrat šablonu
UC76	Přiřadit šablonu k nabídce
UC77	Vytvořit smlouvu
UC78	Export smlouvy do textového editoru
UC79	Připojit smlouvu k obchodnímu případu
UC80	Zobrazit seznam uživatelů
UC81	Zobrazit seznam nabídek

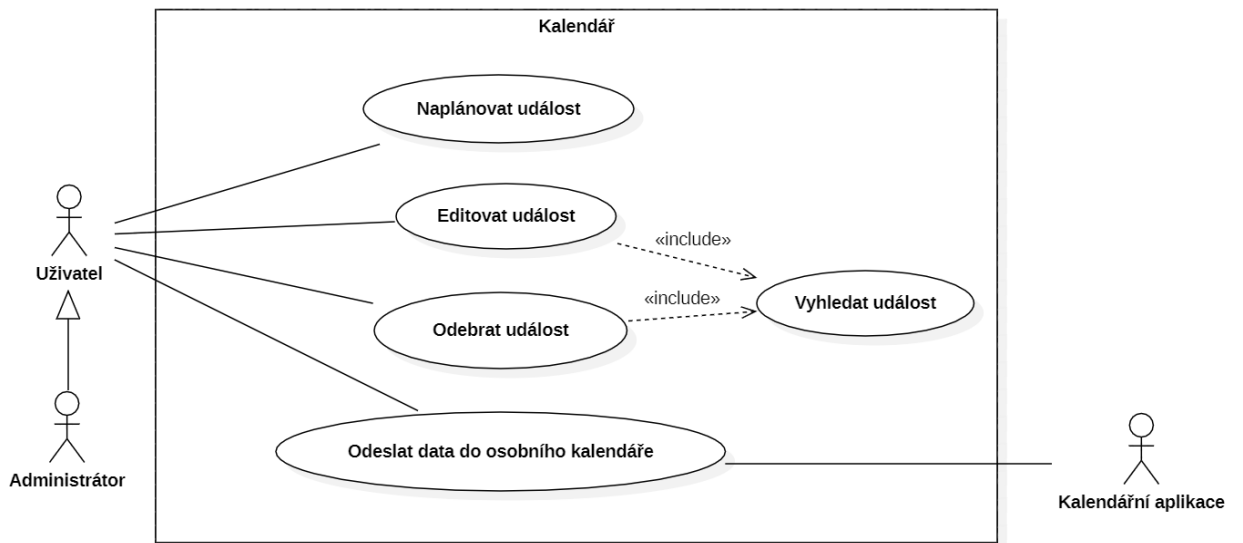
UC82	Zobrazit seznam projektů
UC83	Zobrazit seznam klientů
UC84	Zobrazit seznam poptávek
UC85	Zobrazit seznam rezervací
UC86	Zobrazit seznam obchodních případů
UC87	Export do tabulkového editoru
UC88	Změna hesla
UC89	Odhlášení ze systému

Seznam případů užití

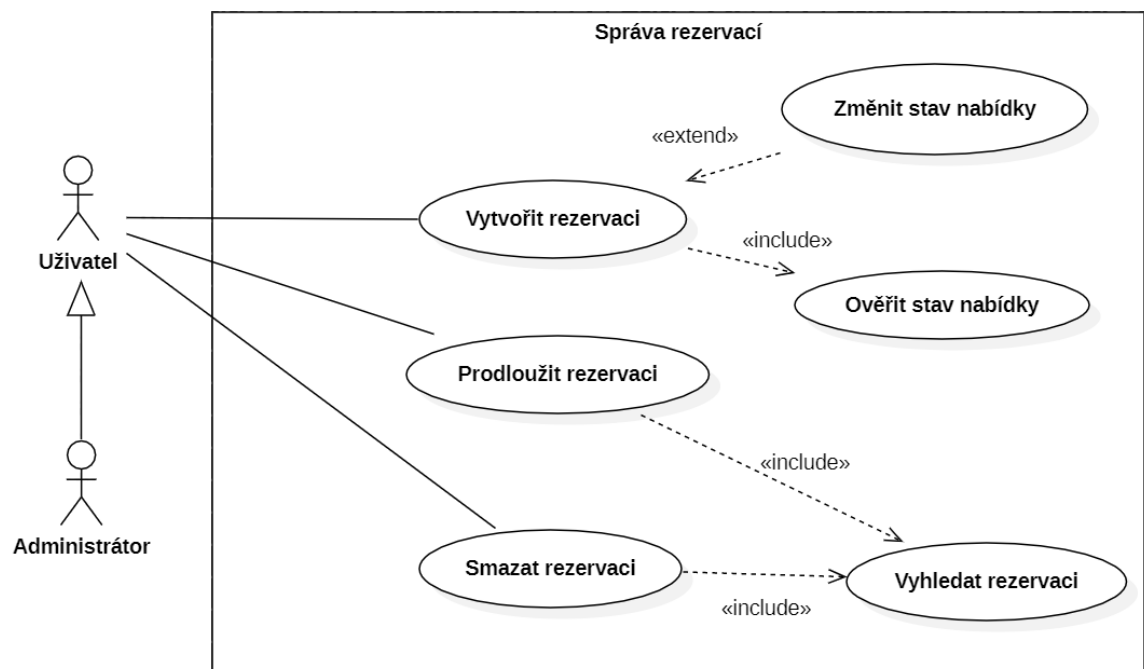


Množina případů užití Správa uživatelů

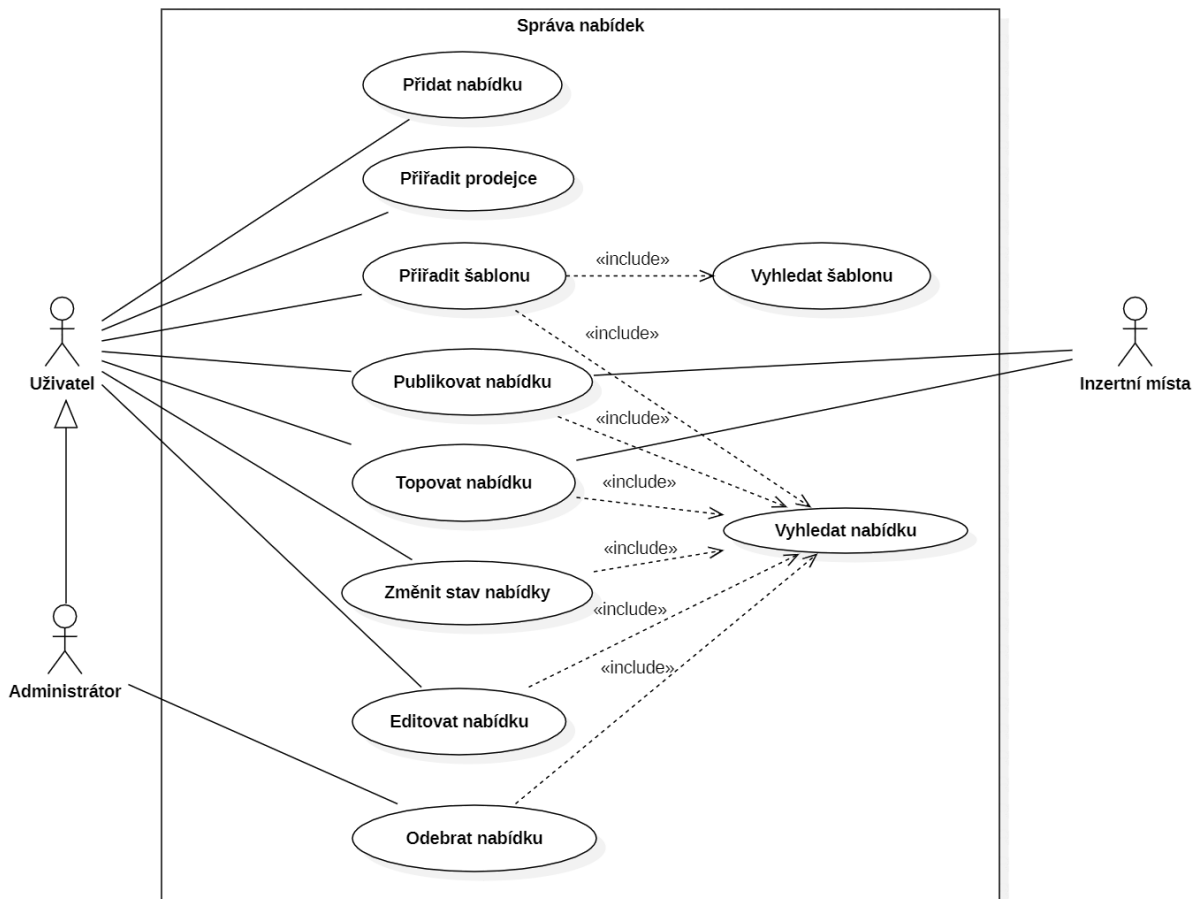
Množina případů užití Kalendář



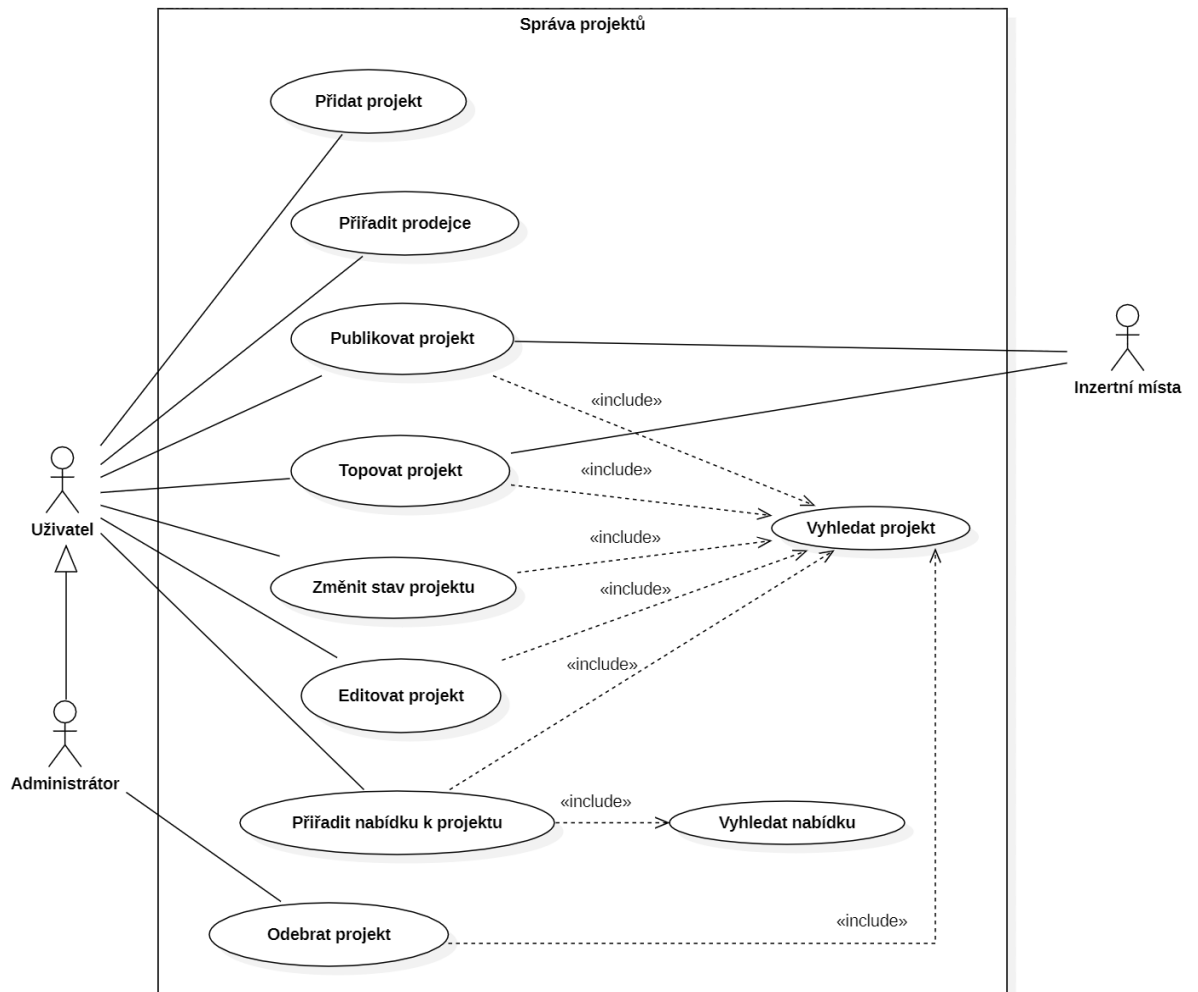
Množina případů užití Kalendář



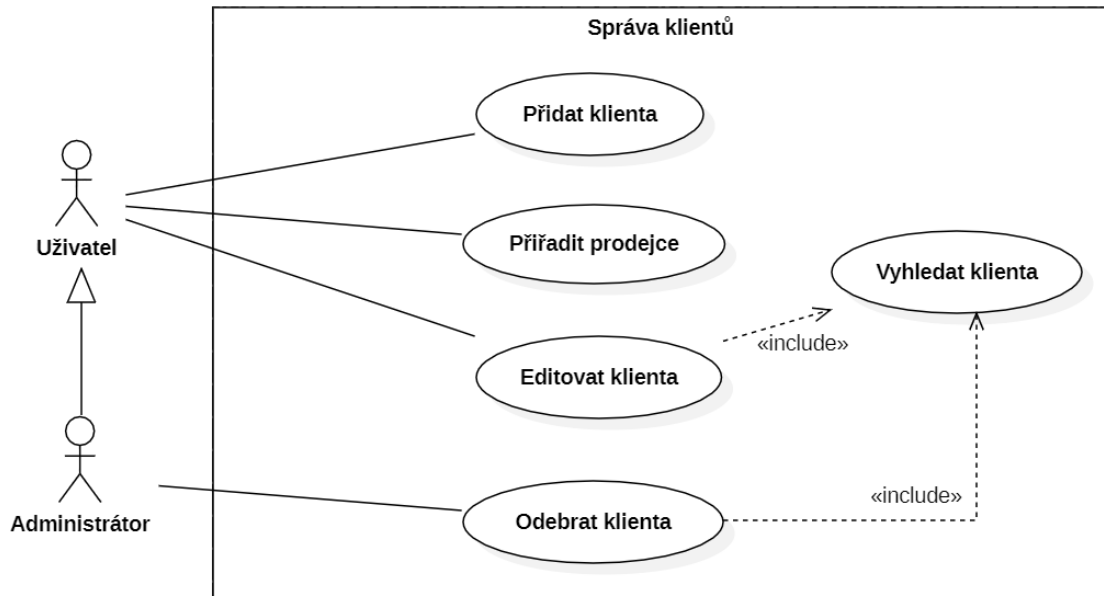
Množina případů užití Správa rezervací



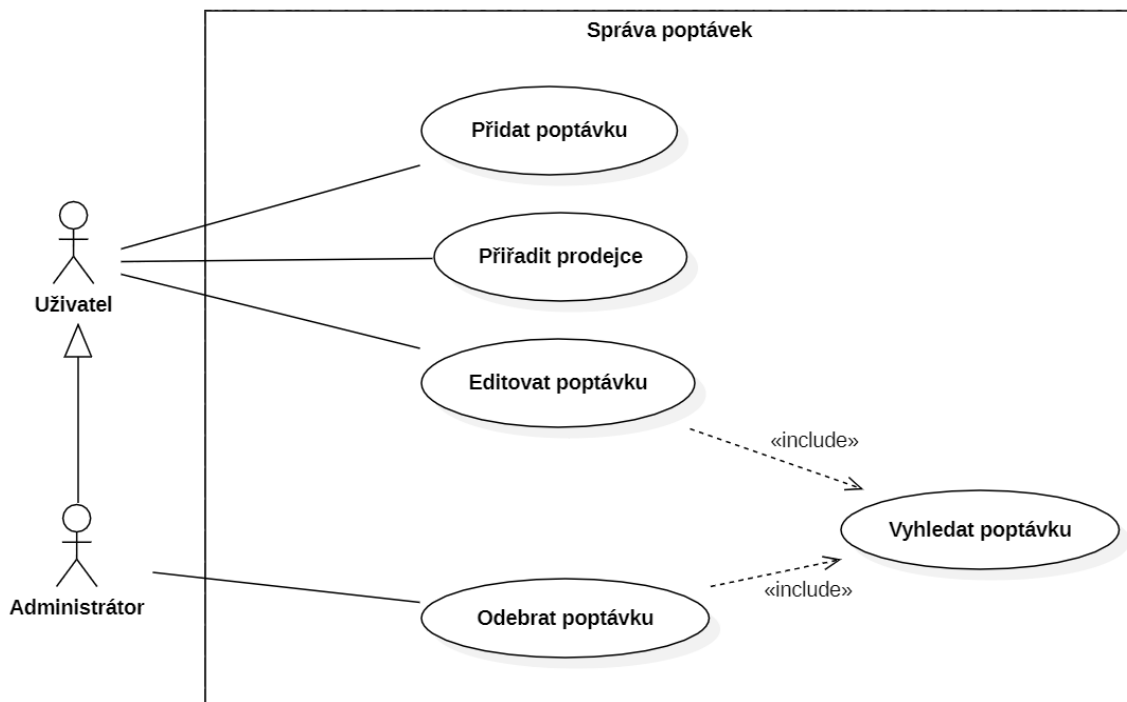
Množina případů užití Správa nabídek



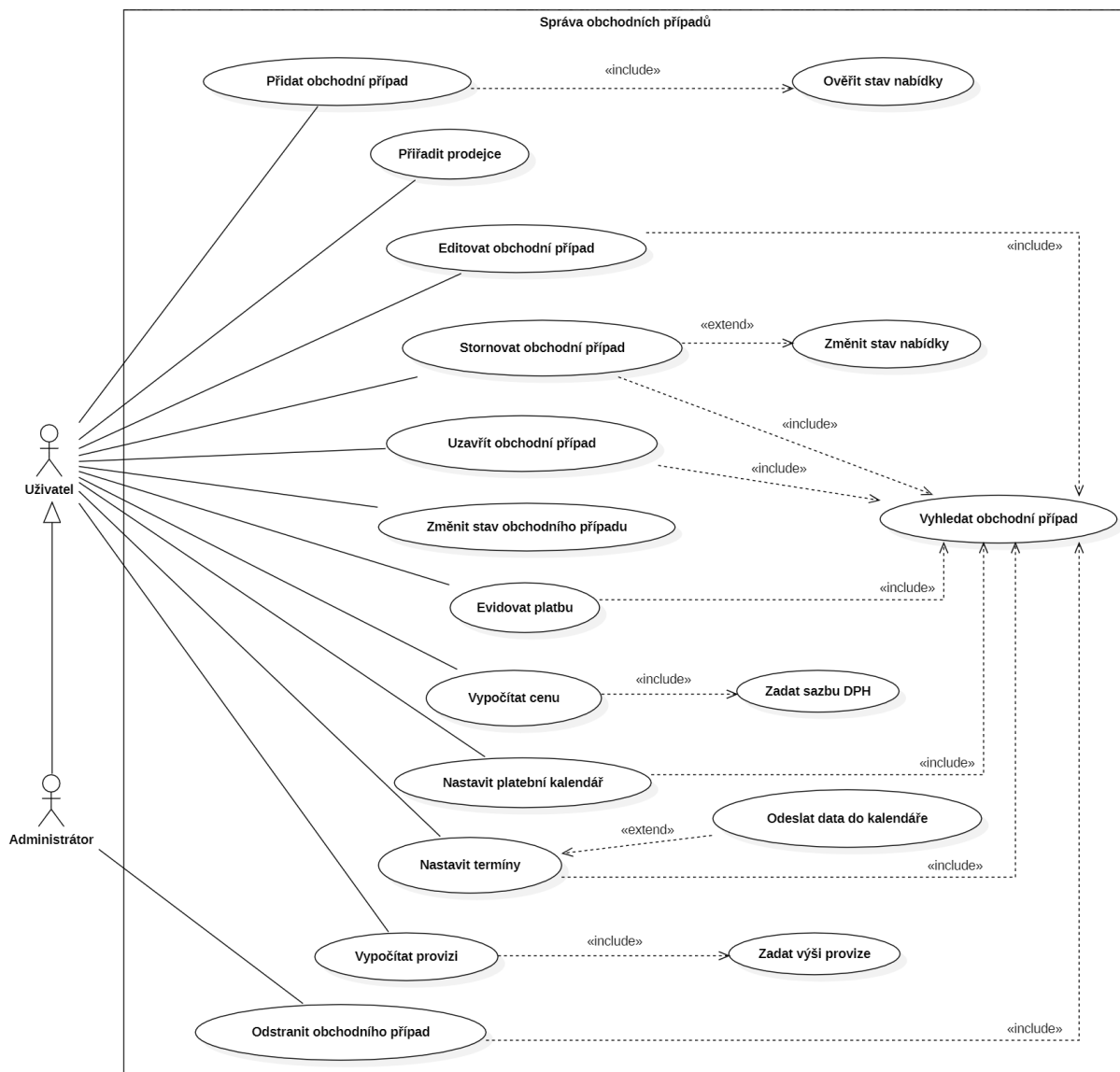
Množina případů užití Správa projektů



Množina případů užití Správa klientů



Množina případů užití Správa poptávek

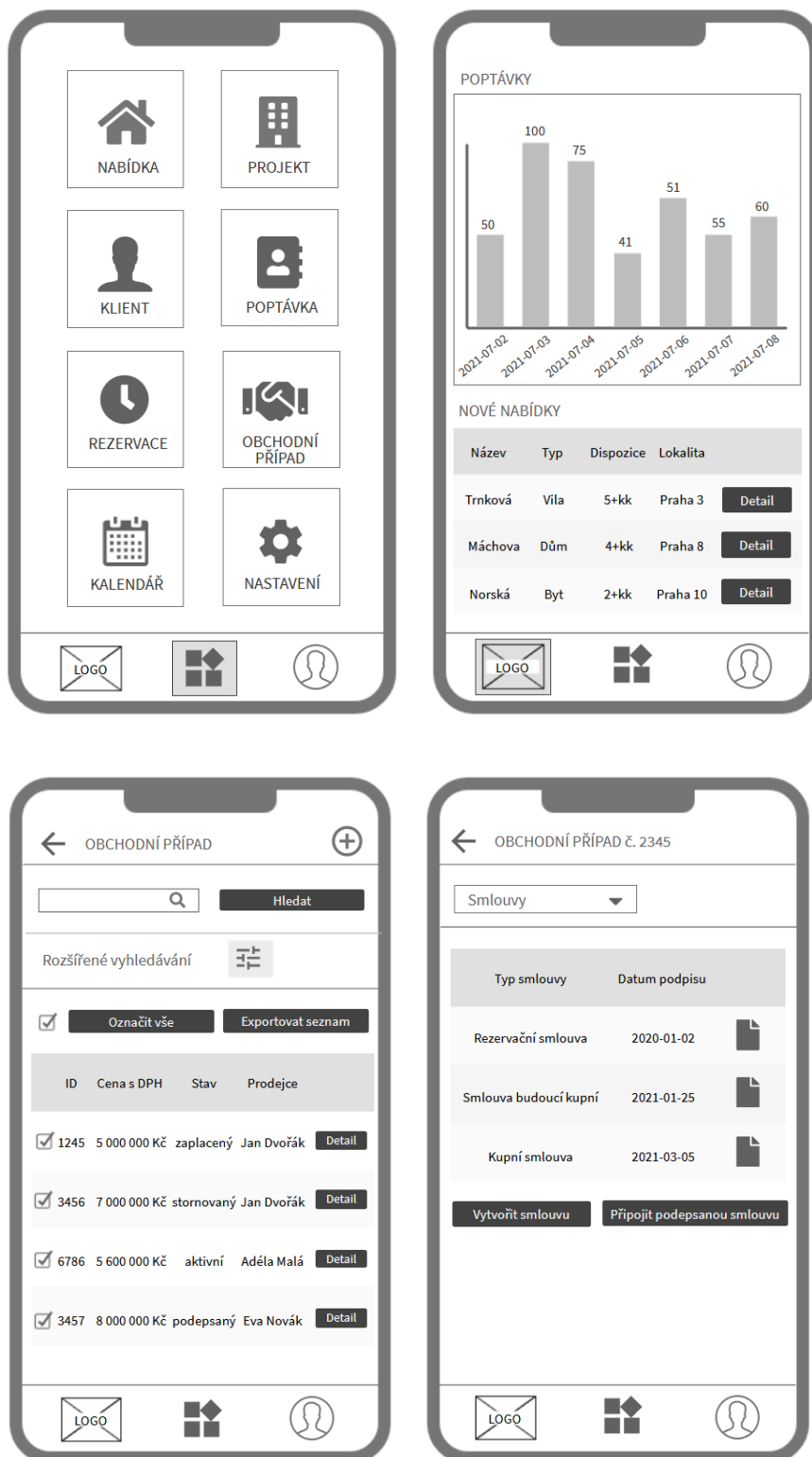


Množina případů užití Správa obchodních případů



Množina případů užití Reporty

Příloha C – Wireframes



Wireframes pro mobilní zařízení