

**Univerzita Hradec Králové**  
**Fakulta informatiky a managementu**  
**Katedra informačních technologií**

**Návrh informačního systému pro vymáhání dluhů**  
**DIPLOMOVÁ PRÁCE**

**Autor:** Filip Ustohal

**Studijní obor:** IM2-k

**Vedoucí práce:** Ing. Pavel Čech, Ph.D.

Hradec Králové

Listopad 2019

**Prohlášení:**

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 15.11.2019

---

Filip Ustohal

**Poděkování:**

Touto cestou bych rád poděkoval vedoucímu diplomové práce Ing. Pavlu Čechovi, Ph.D. za metodické vedení práce, za cenné připomínky a odborné rady k mé diplomové práci.

## **Anotace**

Tato diplomová práce je zaměřena na návrh informačního systému pro vymáhání dluhů, která zahrnuje vybrané části životního cyklu informačního systému. V práci jsou popsány obecné zásady pro vytváření diagramů a ostatní metodiky potřebné pro správnou definici návrhu systému, které vycházejí z odborné literatury. Práce je sestavena z modelů, které nejlépe reflektují požadavky diplomové práce. Na základě definovaného problému vymáhacích procesů je sestaven návrh informačního systému, který má za úkol vylepšit stávající situaci úvěrových firem z pohledu vymáhání dluhů.

**Klíčová slova:** Informační systém, UML, vymáhací procesy, diagram tříd, diagram případů užití, BPMN, diagram aktivit, sekvenční diagram, systémové požadavky, stavový diagram

## **Annotation**

This thesis is focused on the desing of an information system for debt collection, which includes selected parts of information system life cycle. The thesis describes general principles for diagramming and other methodologies necessary for proper definition of system design, which are based on scholar literature. The work consists of models that best reflect the requirements of the thesis. Based on a defined problem of recovery processes, an information system proposal is compiled, which aims to improve the current situation of credit companies in terms of debt recovery.

**Key words:** Information system, UML, debt collection process, class diagram, use case diagram, BPMN, activity diagram, sequence diagram, business requirements, state diagram

# OBSAH

ÚVOD.....	9
1 ZÁKLADNÍ VÝCHODISKA.....	10
1.1 Definice problému .....	10
1.2 Cíl práce .....	11
1.3 Metodika .....	12
1.3.1 CASE Nástroje .....	13
1.3.2 UML .....	13
1.3.3 Metoda FURPS.....	14
1.4 Nestrukturovaný rozhovor .....	15
1.5 Rešerše .....	16
1.6 Životní cyklus informačního systému.....	19
2 ANALÝZA FIREMNÍCH PROCESŮ.....	22
2.1 Business popis.....	22
2.2 BPMN model .....	24
3 SYSTÉMOVÉ POŽADAVKY .....	31
3.1 Funkční požadavky .....	32

3.2	Non-funkční požadavky.....	35
4	MODEL PŘÍPADŮ UŽITÍ.....	38
4.1	Případy užití.....	38
4.1.1	Interakce s funkčními požadavky.....	41
4.1.2	Aktéři.....	42
4.1.3	Vnitřní struktura případů užití.....	44
4.2	Use case diagram.....	51
5	STRUKTURA A DYNAMIKA SYSTÉMU.....	54
5.1	Class diagram.....	56
5.2	Popis tříd.....	58
5.3	Určení vztahů mezi třídami a návrhové vzory.....	60
5.4	Stavový diagram.....	62
5.5	Sekvenční diagram.....	65
6	SHRNUTÍ VÝSLEDKŮ.....	70
6.1	Technické zhodnocení.....	71
	ZÁVĚR.....	73
	SEZNAM ZKRATEK.....	74

SEZNAM OBRÁZKŮ .....	75
SEZNAM TABULEK .....	77
SEZNAM POUŽITÉ LITERATURY .....	78
PŘÍLOHY .....	82



# ÚVOD

Odvětví vymáhání dluhů neustále stoupá s rostoucím množstvím spotřebitelských úvěrů. V dnešní době je velice těžké určit správné vymáhací procesy tak, aby vyhověly jak zákonům dané země, tak aby byly spjaty se zákaznickým pohodlím. Stále přibývající nové úvěrové produkty na trhu lákají nové zákazníky k půjčce. Systémy pro vymáhání se proto musí neustále zlepšovat a musí být udržovány tak, aby firmy dokázaly maximalizovat svoji návratnost půjčených peněz zákazníkům. V dnešní době plné automatizace a digitalizace je pro firmy důležité mít správně fungující informační systém, který dokáže zaštitit všechny požadavky, které jsou na vymáhací proces kladeny. Pro firmy je také často důležité, aby byl systém flexibilní právě z důvodu často měnících se zákonů a tím pádem i změnou strategií vymáhacích procesů. Důležitým faktorem pro firmy při tvorbě vymáhacích systémů je také mít co nejnižší náklady na vymáhání. Skloubit všechny tyto vlastnosti informačního systému dohromady je často velmi obtížné.

Klíčovým bodem při tvorbě komplexního informačního systému, je správná analýza a návrh. Diplomová práce je zaměřena na návrh informačního systému pro vymáhání dluhů. Navrhnutý informační systém má za úkol připravit řádnou dokumentaci pro detailní analýzu a vývoj, která je navrhována na základě odborné literatury. Pro splnění výše uvedených faktů, navrhovaný systém dokáže flexibilně řídit všechny hlavní vymáhací procesy a je navrhnutý tak, aby nedocházelo k chybám, a zároveň aby firma nevykazovala příliš vysoké náklady na vymáhání spolu s cílem uspokojit požadavky firem i klientů.

# 1 ZÁKLADNÍ VÝCHODISKA

Následující kapitola je zaměřena na stanovení problému vymáhacích procesů. Na základě nestrukturovaného rozhovoru, dat a shromážděných informací představené dále v diplomové práci, jsou definovány problémy, které by měl navrhovaný systém pro vymáhání vyřešit. Dále je kapitola zaměřena na definování důležitých nástrojů a metodik, použitých při navrhování systému. Jsou zde specifikovány teoretické poznatky využití těchto nástrojů, metod a postupů, které jsou klíčové pro jejich aplikaci v praktické části při modelování vybraných diagramů.

## 1.1 Definice problému

Zhodnocení nestrukturovaného rozhovoru, které je blíže představeno v podkapitole 1.4 Nestrukturovaný rozhovor a jež klade důraz na zjištění stávajících vymáhacích procesů a situace vybrané společnosti na úvěrovém trhu, přispělo k nastínění problému vymáhacích procesů. Z vyhodnocení rozhovorů vyplývá, že Home Credit obecně využívá vymáhacích procesů zaměřené na kontrakty klienta. Dalším faktem je, že průměrně má firma Home Credit přibližně dvojnásobek kontraktů než klientů, proto při vymáhacích procesech (telefonní hovory nebo odesílání SMS), vytvoří mnohem více nákladů, než kdyby vymáhání bylo zaměřeno na klienta. Dále respondenti poukazovali, že náklady firmy na vymáhací procesy jsou poměrně vysoké a také, že cílem je jejich snižování. Z rozhovoru také vyplývá, že nejdůležitějšími vymáhacími procesy jsou telefonní hovory, fyzické vymáhání, posílání SMS a nejméně důležité posílání dopisů.

Řešením pro výše specifikované problémy je odpuštění od vymáhání na kontrakty. Po aplikaci klientského vymáhání by se snížily náklady. Vymáhání by již dále neprobíhalo odesíláním SMS zprávy nebo voláním na telefonní číslo, které je uvedeno u kontraktu, ale kontrakty by se spárovaly k jednomu klientovi, na kterém by tyto vymáhací procesy probíhaly. V tom případě by však musel být jeden z kontraktů nastavený jako hlavní kontrakt, aby vymáhací procesy mohly být stále jednoduše iniciovány. Díky klientskému

vymáhání by také došlo ke zlepšení kvality vymáhání na call centru. Operátor totiž bude mít možnost být informovaný o všech kontraktech klienta, a ne pouze o jednom kontraktu, kvůli kterému právě volá. Tento efekt by dále mohl zapříčinit rychlejší komunikaci s klientem a operátor by poté měl prostor obvolat denně více klientů. Stejně dobře informovaní potom budou i vymahači u fyzického vymáhání, tedy budou mít k dispozici všechny kontrakty klienta a tím pádem budou moci využít jiné, lepší vymáhací taktiky. Dále by se snížila i chybovost, kdy například při odpuštění pokut, vytváření reportů nebo datových manipulacích, které se většinou provádějí při masových exekucích (tedy je do balíku zahrnuto více kontraktů), bude zahrnut klient spolu se všemi jeho kontrakty automaticky. Snížila by se také náročnost segmentačního scriptu (určení rizikovosti), protože by nepočítal hodnotu pro každý kontrakt zvlášť, ale pouze pro klienta [1].

## 1.2 Cíl práce

Cílem práce je analýza a návrh informačního systému pro vymáhání dluhů. Systém bude spravovat kontrakty na klientské úrovni, údaje o klientovi a možnost automaticky spravovat všechny důležité vymáhací procesy dané firmy (telefonáty, posílání SMS, návštěvy vymahačů apod.). Systém je front-endový a bude nabývat funkce Core systému, pro ostatní interakční systémy, které jsou nezbytné pro vymáhací procesy (vymahačská aplikace, SMS brána, dialer atd.).

Další funkcionalitou bude evidence všech uskutečněných vymáhání na konkrétním klientovi v rámci veškerých vymáhacích procesů. V neposlední řadě, bude také systém nabízet možnost vytváření reportů, kterých budou moci využívat manažeři nebo datoví analytici k revizi výsledků.

Na základě uživatelských požadavků budou stanoveny klíčové funkcionality informačního systému. Všechny výstupy návrhu informačního systému se budou řídit metodikou jazyka UML.

### 1.3 Metodika

Celá práce je založená na rešerších a nestrukturovaného rozhovoru, které jsou blíže popsány v následujících podkapitolách. Z těchto zdrojů je koncipována praktická část, která je zaměřena především na počátek životního cyklu návrhu informačního systému. Samotný návrh informačního systému je prolínáný teoretickými a odbornými poznatky z odborných literárních zdrojů a studií.

Praktická část je dále zaměřena na analýzu a návrh informačního systému pro vymáhání dluhů, jehož úkolem je správa vymáhacích procesů daných klientů fiktivní firmy. Na základě BPMN návrhu, jehož hlavním podkladem je článek „Incident management system for debt collection in virtual Banking“ a rozhovor se systémovým specialistou vybrané firmy, je definován proces stávajícího systému. Samotný návrh je postaven na návrhových vzorech, které jsou v práci blíže specifikovány. Právě BPMN návrh je hlavní iniciativou pro vytvoření uživatelských požadavků, které definují cestu pro klíčové funkcionality. Funkcionalita systému je v práci reprezentována pomocí use case elementů a diagramů aktivit, jejichž tvorba se drží striktních pravidel pro vytváření těchto digramů, které jsou v diplomové práci blíže specifikovány. Tím je zachycen vnější pohled na modelovaný systém. Pro identifikaci datové a statické struktury systému je použit model tříd, jehož definice je založena především na návrhových vzorech, které zmiňuje ve své knize „Analytické modelování informačních systémů pomoc UML v praxi“ pan Ilja Kraval. Díky stavovým diagramům je dále identifikována dynamika modelu a blíže specifikovány některé vybrané třídy představené v modelu tříd. Poslední částí navrhovaného systému jsou sekvenční diagramy, které představují interakci mezi elementy v čase. Tyto interakce potvrzují správnou identifikaci vnitřní struktury modelu případů užití a modelu tříd.

### 1.3.1 CASE Nástroje

Pro návrh a analýzu kvalitního informačního systému je vhodné používat tzv. Case nástroje (Computer-Aided Software Engineering). Tyto nástroje podporují návrh systému a softwarové inženýrství téměř ve všech stádiích životního cyklu. Case nástroje podporují analýzy a návrh, samotnou implementaci a testování, dokonce i údržbu. Tyto nástroje jsou proto nezbytnou součástí návrhu informačního systému pro vymáhání dluhů.

Pro návrh informačního systému pro vymáhání dluhů je vybrán nástroj Enterprise architect od společnosti Sparx Systems ve verzi 13. Tento nástroj bude využit především pro vytváření veškerých digramů, které jsou nezbytné pro kvalitní návrh informačního systému. Tento nástroj dokáže nastítnit pohled na momentální organizační strategie a procesy. Enterprise Architect je zodpovědný za užívání těchto znalostí, aby zajistil IT a podnikové sladění. Tento nástroj je založený jazykem UML, díky kterému může být realizován celý životní cyklus návrhu informačního systému, který je popsán níže [2].

### 1.3.2 UML

Jazyk UML slouží k modelování softwaru a dalších systému a představuje kolekci spolupracujících objektů. Jak píše ve své knize Jim Arlow a Ila Neustadt: „Jazyk UML je otevřeným rozšiřitelným průmyslovým standardem pro vizuální modelování, schváleným sdružením OMG (Object Management Group).“ Je často používán v softwarovém inženýrství k návrhu informačních systémů a jejich dokumentaci. UML je známý pro modelování objektově orientovaných systémů, avšak má i širší využití. Pro práci je jazyk UML použit pro modelování dále představených diagramů, které blíže specifikují danou problematiku [3, 4].

### 1.3.3 Metoda FURPS

Množství vyvinutých softwarů v dnešní době rapidně roste a jsou využívány v téměř všech aktivitách lidského života. V důsledku toho se měření a hodnocení kvality softwarů stalo kritickým úkolem pro mnoho firem. Vývoj technik pro budování softwaru ovlivňuje vytváření modelů a jejich kvalit. Pro analýzu správného návrhu informačního systému, je vybrána metoda FURPS, která byla vyvinuta firmou Hewlett Packard, jako jedna z možných cest pro zjištění kvality informačního systému, řídící se podle normy ISO 9126. Metoda má dvě hlavní části – atributy interních a externích kvalit a atributy určující kvalitu při používání systému. Interní atributy kvality se zaměřují na ty části modelu, které mohou být evaluovány i bez exekuce systému, zatímco externí atributy jdou posoudit při spuštění systému. Metoda FURPS se dívá na kvalitu informačního systému z pěti základních hledisek, které jsou blíže představeny v kapitole 6.1 Technické zhodnocení, kde odráží technickou způsobilost navrhovaného systému na základě níže představeného odborného popisu. Základní hlediska kvality informačního systému:

- Funkčnost (Functionality)

Je zaměřena na hlavní funkčnosti a chování programu. Zkoumá, zda software splňuje podnikové požadavky a jsou podporované podnikové procesy.

- Použitelnost (Usability)

Hodnocení je prováděno především z pohledu koncového uživatele. Využití aplikace, celkový dojem aplikace. Je zde také zahrnuta kvalita dokumentace.

- Spolehlivost (Reliability)
- Výkon (Performance)

- Rozšiřitelnost (Supportability)

V této oblasti se hodnotí údržba a podpora aplikace a její testovatelnost. Kromě toho je zde hodnocena i přizpůsobitelnost a rozšiřitelnost dané aplikace [5, 6].

#### 1.4 Nestrukturovaný rozhovor

Pro zjištění a potvrzení stávajících procesů a situace úvěrových společností byl proveden nestrukturovaný rozhovor s vybranými firmami a vybranými zaměstnanci, kteří dokáží nejlépe reflektovat tyto skutečnosti. Home Credit je mezinárodní poskytovatel spotřebitelských úvěrů, který momentálně operuje v 9 zemích na světě. Firma se zaměřuje především na lidi, kteří mají malou nebo žádnou úvěrovou historii. Pro nestrukturovaný rozhovor byli vzhledem k povaze diplomové práce vybráni vždy pracovníci vymáhacího oddělení. Pro obecnější vyhodnocení výsledků byl vybrán pracovník Home Credit International a.s. Pro detailnější analýzu firmy Home Credit byli vybráni dva pracovníci dceřiných společností Home Credit China a Home Credit Indonesia.

Ve jménu společnosti Home Credit China byl rozhovor veden s Martinem Scholzem, manažerem systémových analytiků. Za Home Credit Indonesia byl dotazován Lukáš Charamza, manažer vývoje a analytiků. Specialista vymáhacích procesů, Monika Šustová odpovídala za Home credit International a.s. Tito respondenti byli vybráni, protože dokáží nejlépe identifikovat stávající situaci firmy, co se vymáhacích procesů týče. Všichni dotázaní respondenti mají totiž více než 5 let praxi s vymáhacími procesy.

Otázky v nestrukturovaném rozhovoru byly koncipovány tak, aby bylo možné blíže specifikovat nynější situaci firmy z pohledu procesů, zákazníků a jejich poskytnutých produktů. Pomocí rozhovoru by mělo být dále definováno, jakým směrem se firma ubírá, jinak řečeno, jakých ekonomických cílů chce firma dosáhnout. Otázky v rozhovoru byly kladeny spontánně a vybírány na základě předchozích odpovědí.

## 1.5 Rešerše

Hlavními akcemi vymáhacího procesu jsou především telefonáty, posílání SMS, dopisů apod., díky čemuž si firmy kladou za cíl dostat z klientů částku, kterou dluží, nejlépe ještě před datem splatnosti první splátky půjčky. Pokud klient nezplatí první splátku půjčky do devadesáti dnů, stává se z něj takzvaný „paid-off“ klient. To znamená, že klient již nesplácí půjčku ve splátkách, ale musí splatit již celou částku najednou. Pro firmy je tedy klíčové, aby klienti nebyli „paid-off“, protože firmám neúměrně rostou náklady s vymáhacím procesem a pravděpodobnost splacení celé půjčky klesá. Na základě výše uvedených důvodů je důležité, aby byl systém pro vymáhání byl navržen efektivně.

Dalším problémem vysokých nákladů firem na vymáhání je fakt, že vymáhací systém hodnotí klienta z hlediska kontraktů, a tudíž nesleduje kolik kontraktů má jeden klient současně. Z tohoto důvodu se stává, že byly některé ukončeny předčasně a některé později s ohledem na rizikovostní skóre daného kontraktu. Proto je důležité v tomto procesu ukončení najít řád a nastavit ho tak, aby byl ekonomicky dostupnější.

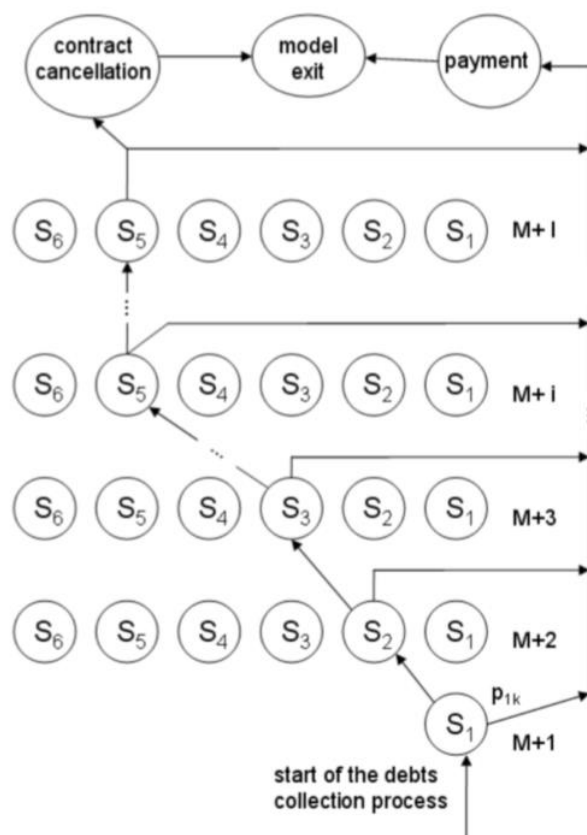
Autorem článku „Modelling of the debts collection process for service companies“ byla provedena studie, která se zabývá vytvořením modelu a určením, kdy je ekonomičtější daný kontrakt ukončit a kdy je lepší pokračovat v procesu vymáhání. Tato studie představuje dynamický model, který se vypořádává s problémem vzniku dluhu a snaží se mu předejít. Tento nástroj určuje, kdy zahájit některé konkrétní preventivní akce, aby se zabránilo tomu, že částka po splatnosti nebo částka náchylná k tomu, aby byla po splatnosti, přesáhla optimální částku vypočítanou pro daný den a pro danou rizikovost zákazníka.



Následující obrázek číslo 1 představuje dynamický model výše zmíněného problému. Díky tomuto modelu dokážeme odpovědět na následující dvě otázky, za předpokladu, že zákazník spadá do rizikovosti  $k$  (příklad low, medium, high) během měsíce  $M + i$  (ukončení kontraktu):

Známe-li momentální měsíční spotřebu. Který den by měla být zahájena akce  $M + i + 1$ , aby nebyla nad optimem?

Známe-li  $S_i$  (dlužná částka). Kdy by firma měla zastavit účtování zákazníka (nebo nastavení speciální k účtování s nižšími měsíčními poplatky, alespoň na pokrytí nákladů)?



Obrázek 1 – Dynamický model, Zdroj: [7]

Cílem bylo tedy pro každý kontrakt zákazníka optimalizovat jeho globální ekonomickou rizikovost  $Gk(I)$  (Global economical risk). Tedy pro jednotlivé rizikovostní úrovně zákazníků zavést novou účtovací metodu po nějaké konkrétní době a tím vytvořit více času před „zrušením“ kontraktu.

Aplikací výsledků této studie dochází k tomu, že dlužníci splácí pravidelněji a méně kontraktů je „zrušeno“ [7].

Další článek pojednává o jednom z hlavních nástrojů pro vymáhání dluhů ve Velké Británii, kterým je call centrum. Článek se zaměřuje především na to, jakým způsobem by měli lidé na call centru mluvit s klientem, což se mění v určitých fázích vymáhacího procesu a na základě rizikovosti klienta. Ve článku jsou popsány různé strategie a přístupy, které zlepšují kvalitu hovorů a zároveň vytváří určité pozitivní spojení mezi vymahačem a dlužníkem. Článek dále hovoří o velkém množství úkolů, které musí zaměstnanec na call centru znát a následovat tak, jak je striktně stanoveno. Příkladem mohou být přesně stanovené otázky a odpovědi v určitých situacích, být empatický, získat co nejvíce informací od klienta, odmítnout první nabídku (pokud se nejedná o zaplacení celé částky), apod.

Zefektivněním vymáhacího systému pro call centrum se sníží nátlak na pracovníky, poskytnete jim více času na jednoho klienta, práce bude efektivnější a tím firma vymůže více dluhů od klientů a zároveň sníží náklady spojené s množstvím telefonátů každého kontraktu u klienta zvláště [8].

Jedním z nejdůležitějších ukazatelů pro vymáhací systém je predikování, zda je klient schopný dluh splatit. To popisuje výzkum, který byl uskutečněn v roce 2018 v USA, který poukazuje právě na tuto problematiku a snaží se popsat matematické a data mining modely, které využívají historická data vymáhacích agentur o klientech. Výsledek výzkumu má určit nejlepší metodologii, která určí nejpřesnější výsledky predikce klienta splácet.

Jak již bylo zmíněno, tyto modely jsou použité přímo na klientovi a ne na každém kontraktu klienta zvlášť, proto při integraci systému na klientské vymáhání, místo vymáhání na kontraktech, by tento důležitý ukazatel nebyl nijak ovlivněn [9].

Další výzkum pojednává o aplikaci fuzzy logiky do Early fáze vymáhání, která má zapříčinit minimální komunikační náklady, na základě předchozích komunikačních zkušeností dlužníků. Článek poukazuje na problematiku nákladů při vymáhání, které často není zdaleka tak efektivní, jak by se předpokládalo. Výzkum představuje vývojové diagramy pro každý subsystém, které určí správnou cestu, jak komunikovat s dlužníkem. Všechny tři představené subsystémy představují dohromady 33 320 možných kombinací neboli fuzzy pravidel. Dále model může být vylepšen “adaptivním neuro fuzzy inference systémem”, který může generovat nová pravidla vylepšením již stávajících pravidel [10].

Ve článku „řízení a vymáhání pohledávek“ od Jiřího Součka, který porovnává různé přístupy vymáhání dluhů a pohledávek je uvedeno, že větší firmy používají pro vymáhací procesy výhradně automatizované informační systémy. Článek také popisuje, že informační systémy by měly sledovat bonitu odběratelů a tím předcházet vzniku pozdě splacených pohledávek. Dále je zde uvedeno, že menší firmy, které nevyužívají automatizovaný informační systém pro vymáhání, mají přibližně o 40% nižší úspěšnost vymožených pohledávek a stejně tak koreluje i doba a množství zpožděných plateb [11].

## 1.6 Životní cyklus informačního systému

Již zmíněný životní cyklus informačního systému může být popsán v následujících etapách:

- Etapa informační strategie organizace: specifikace zadání
- Etapa úvodní studie projektu: předběžná analýza
- Etapa globální analýzy a návrhu IS: detailní analýza
- Etapa detailní analýzy a návrhu IS: návrh
- Etapa implementace IS: implementace

- Etapa zavádění IS do provozu: zavedení do provozu
- Etapa provozu, údržby a dalšího: rozvoje IS provoz

Práce je zaměřena pouze na první 4 etapy životního cyklu informačního systému. Bude však koncipována tak, aby v budoucnu mohla být analýza a návrh použit pro samotnou implementaci a nasazení do konkrétní firmy.

Na následující straně budou detailněji popsány všechny části životního cyklu.

#### 1. Předběžná analýza neboli specifikace cílů

První část návrhu je zaměřena na shromáždění požadavků od uživatelů a specifikace od zainteresovaných členů firmy. Zde se sestaví časový plán projektu, zdroje nutné k řešení a odhad funkcí.

Pro tuto část budou použity nástroje pro specifikaci projektu, kterými jsou analýza současného stavu, získání požadavků uživatelů a seznam problémů, které mohou nastat. Na základě těchto dokumentů mohou být definovány jednotlivé části systému a jejich aplikace.

#### 2. Analýza systému neboli specifikace požadavků

Druhá část navazuje na část předchozí, a to především na analýzu a nalezení všech chyb. Pokud tyto chyby nenalezneme v této etapě, bude je v dalších fázích obtížné nalézt.

#### 3. Projektová studie neboli návrh

Část tři slouží jako výstupní dokument, který je výsledkem analýzy systému. Může být použit jako součást smlouvy s dodavatelem nebo jako podklad pro interní vývoj. Návrh by měl obsahovat několik prvků, jimiž jsou například základní informace o tvůrcích systému; či externí firmě; informace o organizaci, pro kterou je systém vyvíjen; popis současného stavu organizace; globální návrh IS (logický datový model);

detailní návrh IS, neboli fyzický datový model; detailní popis nasazení IS; celkový harmonogram práce.

4. Implementace
5. Testování
6. Zavádění systému
7. Zkušební provoz
8. Rutinní provoz a údržba
9. Reengineering

Následující kapitoly podrobně představí praktické zpracování a použití prvních třech bodů životního cyklu informačního systému [12].

## 2 ANALÝZA FIREMNÍCH PROCESŮ

V následující kapitole je představena analýza a popis vnitřních procesů firmy. Nejprve je představen obecný popis procesů vymáhání, který má za úkol nastínit danou problematiku. Na základě popisu procesů je definován BPMN model znázorňující interakce mezi podnikovými subjekty a jejich vzájemnými procesy. Veškeré praktické návrhy jsou v kapitole aplikovány na základě teoretických poznatků.

### 2.1 Business popis

Následující popis je definován na základě konzultace se specialistou vymáhacích procesů, který je již představen v kapitole 1.4 Nestrukturovaný rozhovor. Pracovník Home Credit international a.s. popsal stávající vymáhací proces, který je níže přepsán do srozumitelné podoby.

Klient podává žádost na pobočce nebo online o jeden z nabízených produktů (hypotéka, úvěr apod.). Na základě údajů, které byly poskytnuty, probíhá hodnocení klientovy situace pomocí „evaluátoru rizikovosti“. Ten díky vstupním informacím, naprogramovaným pravděpodobnostním modelům, dokáže klientovi přiřadit rizikovost „low“, „medium“, „high“ (čím vyšší rizikovost, tím je menší pravděpodobnost, že klient bude splácet). Následně jsou zaslány dotazy do registrů dlužníků, a to je finální krok vyhodnocení žádosti. Výsledkem je schválení nebo zamítnutí žádosti. V případě, že je klient vyhodnocen kladně, je mu poskytnut produkt, o který žádal. V opačném případě mu produkt není poskytnut, ale neznamená to, že si nemůže požádat o jiný produkt. Na základě údajů je klientovi vygenerována smlouva (kontrakt), kterou je možno podepsat elektronicky nebo na pobočce. Klient začíná produkt splácet od následujícího měsíce, v předem definované výši splátek. Informace o úvěru jsou odeslány do Base fáze procesu sledování úvěrů. V případě, že klient platí splátky pravidelně a bez odkladu, je v této fázi až do konce životního cyklu produktu (až do splacení). V opačném případě se daný

kontrakt dostává do fáze vymáhání. Jednotlivé fáze zároveň i korespondují s rizikovostí klienta, která je mu přiřazena „evaluátorem rizikovosti“.

### 1. Fáze „Base“

V Base fázi jsou evidovány pouze nedlužné kontrakty, nemusí přejít do vymáhacího procesu. Klient má při vstupu do této fáze vždy status „Valued“, tento status může jedině ztratit.

### 2. Fáze „PRECOLLECTION“

V „Precollection“ fázi se nachází **nedlužné kontrakty**, které sem vstupují nejčastěji 12 dní před splatností. Jsou dva možné stupy klienta do „Precollection“. Prvním je zařazení „Valued“ klienta na základě jeho žádosti, protože chce být informován před splatností. Druhou možností je zařazení na základě přiřazené rizikovosti. Na „low“ úrovni rizikovosti neprobíhají žádné vymáhací akce. V případě „medium“ úrovně je klient kontaktován pouze pomocí SMS. Na úrovni „high“ je klient kontaktován formou SMS a dvěma telefonáty během celé doby před splatností.

### 3. Fáze „Early“

Fáze Early začíná prvním dnem, následujícím po dni, kdy klient nezaplatil svoji splátku a trvá 90 dní. Celé toto období je rozděleno do 3 fází: „bucket 1“, „bucket 2“, „bucket 3“, každá fáze trvá 30 dní. V každém bucketu je klientovi naúčtována pokuta, která navyšuje dluh. Během Early fází je klient kontaktován na základě přiřazené rizikovosti. Na úrovni „low“, je klient telefonicky kontaktován každý druhý den a SMS jsou posílány každý desátý den. Na úrovni „medium“ a „high“ je klient telefonicky kontaktován každý den s výjimkou, kdy složí slib, o zaplacení. Na zaplacení je mu vyhrazeno období, kdy není vymáhán a pokuta je pozdržena. Toto období může trvat maximálně 7 dní. Úrovně „high“ je v třetím bucketu možné použít vymáhací proces osobní návštěvy („Fyzické vymáhání“). To znamená, že klient je na své adrese navštíven vyškoleným zaměstnancem firmy, který má za úkol vymoci peníze ve formě slibu o zaplacení nebo přijmout dlužnou splátku, ať v hotovosti nebo elektronicky.

Proces „Fyzického vymáhání“ se používá ve chvíli, kdy není možné více jak 10 dní po splatnosti navázat kontakt s klientem (ve všech možných formách). Pro nedohledatelné klienty funguje proces „Skip Tracing“, který má za úkol dohledat klienta nebo získat kontakt a pokusit se ho kontaktovat.

#### **4. Fáze „Late“**

V Late fázi se nachází kontrakty, které jsou více než 3 měsíce po splatnosti. Kontrakt je 91. den po splatnosti zesplatněn (terminován). Terminace znamená, že klientův úvěr je předčasně ukončen a prohlášen za splatný, zároveň je klient vyzván k uhrazení zbývajících dlužných částek včetně úroků a poplatků. Během celé fáze jsou každý druhý den uskutečňovány telefonické hovory a proces je na všech úrovních rizikovosti stejný (nebere se v potaz rizikovost daného kontraktu). V této fázi se více než kde jinde využívá „Fyzické vymáhání“ a v některých případech je využíván „Fyzické vymáhání“ externími agenturami.

#### **5. Fáze „Legal“**

Kontrakt se do fáze „Legal“, tedy poslední možné fáze vymáhání, dostane po uplynutí 360 dní po splatnosti nebo také v případě, je-li kontrakt nedohledatelný. Celý legal proces je z převážné části manuální, protože se jedná o individuální případy. V této fázi je kontrakt odeslán externí firmě (exekutoři, soudy, arbitráže apod.), kde je snaha o vymožení alespoň určitého procenta z dlužných částek.

## **2.2 BPMN model**

Modelování firemních procesů vytvořila Object Management Group, jako standardní Business Process Model Notation (BPMN). BPMN by mělo poskytnout business notaci, kterou dále využijí analytici, vývojáři a manažeři, kteří dále s touto notací pracují. BPMN tedy vytváří jakýsi rámec mezi implementací procesů a návrhem firemních procesů. BPMN popisuje „co organizace dělá“ a na základě toho je poté možné určit, jaké informace budou potřeba.



Hlavním důvodem modelování firemních procesů je nalezení nedostatků a bližší specifikace pro budoucí vylepšení a návrh těchto nových procesů. Návrh nového IS totiž znamená prvotní efektivní zlepšení firemních procesů [17].

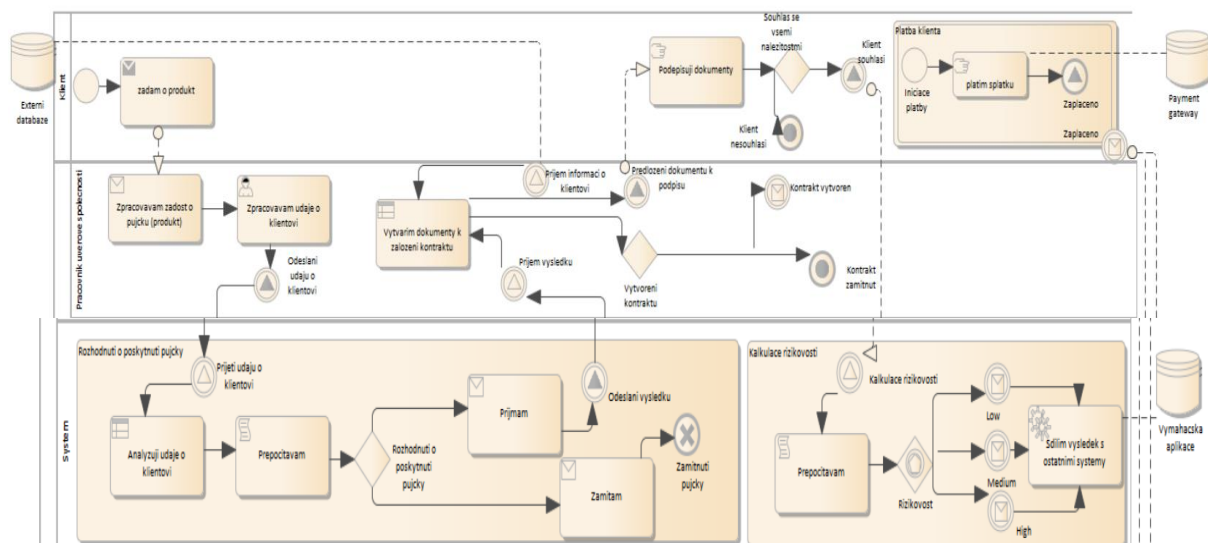
Pro BPMN návrh firemních procesů jsou použity návrhové vzory, které jsou popsány v knize „workflow patterns definitive guide“ od autorů Nick Russell, W.M.P.v.d.A. a Arthur H. M. ter Hofstede. Jejich iniciativa je založena na výzkumech, které slouží jako podpora pro analytiku při modelování firemních procesů. Na základě představených návrhových vzorů v knize jsou v následujícím BPMN návrhu použity návrhové vzory Sequence, Exclusive Choice, Synchronizing Merge a Arbitrary cycles.

Notace BPMN se skládá z několika klíčových elementů, které jsou kritické pro správný návrh firemních procesů dané organizace. Tyto elementy jsou dále blíže specifikovány v následujícím návrhu BPMN procesů.

Následující obrázky číslo 2 až 5 ukazují business model vytvořený pomocí BPMN notace. V modelu jsou použity všechny potřebné elementy pro správný popis stávajícího, standardního vymáhacího procesu. BPMN model je použit pro bližší specifikaci problematiky vymáhání. Model slouží pro lepší pochopení, jak tento proces funguje.

Hlavním podkladem pro modelování firemních procesů pomocí notace BPMN jsou články „Incident management system for debt collection in virtual Banking“, který popisuje a demonstruje na modelu tradiční vymáhání dluhů a pohledávek a dále také představený business popis.

Model je rozdělen do tří stěžejních částí, pro lepší přehlednost. V každé části jsou vyzdvihnuty důležité aspekty pro modelování konkrétního procesu. Celý model je možné nalézt v příloze číslo 1 – BPMN model.



Obrázek 2 - BPMN model 1. část, Zdroj: Autor

Obrázek číslo 2 představuje první část modelu - základní proces žádosti klienta o produkt (půjčku). V této části se mimo klienta objevuje také pracovník úvěrové společnosti (můžeme chápat jako pracovníka na přepážce) a také systém, se kterým pracovník pracuje.

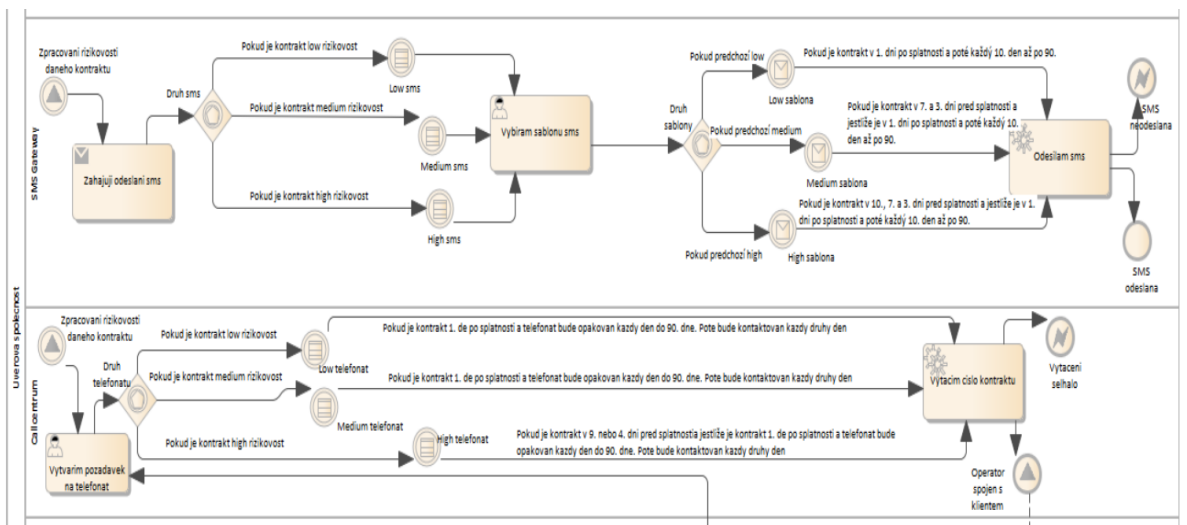
V první části jsou použity návrhové vzory Sequence a Exclusive choice. Pro účely demonstrace procesu jsou tyto dva návrhové vzory dostačující, protože proces představuje jednoduché rozhodování, kdy může pokračovat pouze jednou z více možných cest.

V následujícím procesu můžeme vidět návrhový vzor Sequence například u pracovníka úvěrové společnosti, když zpracovává žádost o produkt od klienta a následně zpracovává údaje o klientovi a odesílá je do systému.

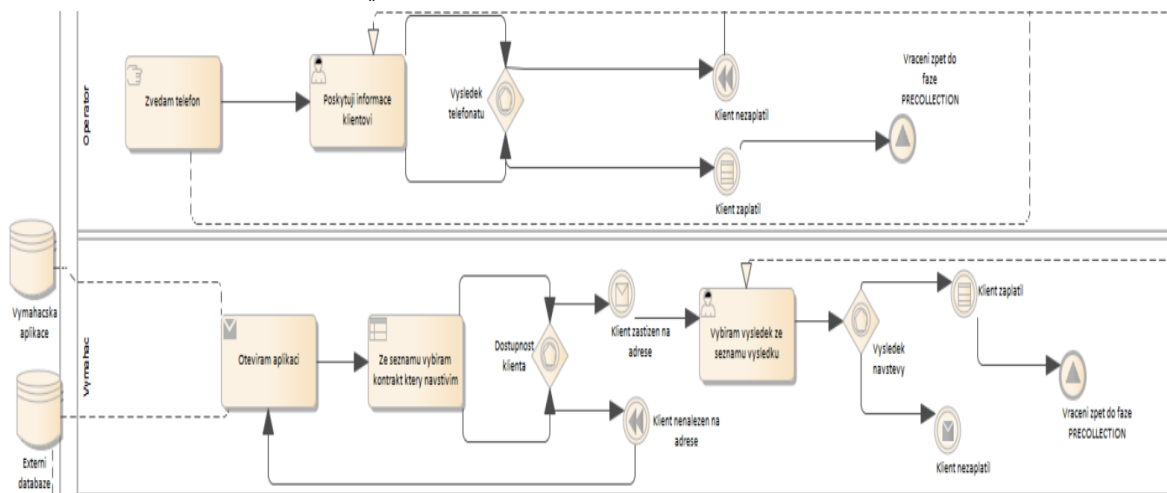
Návrhový vzor Exclusive choice je téměř v každém rozhodovacím procesu, který je v této části zprostředkován tzv. „Gateway“ elementem. Gateway rozděluje business procesy a aktivity na základě jejich druhu a značí se kosočtvercem. Gateway má několik možných

podob, proto ho můžeme v modelu vidět prázdný nebo s dalšími ornamenty uvnitř. Příkladem Gateway může být rozhodnutí o poskytnutí půjčky systémem, který může nastat u dvou stavů – přijímám nebo zamítám. Spolu s předchozí aktivitou „přepočítávám“, představuje tato sekvence návrhový vzor Exclusive choice. Jak již bylo zmíněno, v této části modelu můžeme vidět jeden rozdílný Gateway a to v sub-procesu „Kalkulace rizikovosti“. Tento typ Gateway je tzv. „Event-parallel“. Tento typ je použit z důvodu rozdělení cest do „event“ nikoliv do „aktivity“.

Již zmíněný sub-proces se zde objevuje ještě dvakrát, a to v části „Rozhodnutí o poskytnutí půjčky“ a „platba klienta“. Oba sub-procesy u aktéra „Systém“ jsou použity z důvodu detailnějšího popisu obou procesů v rámci jednoho aktéra. Pro lepší pochopení si můžeme představit, že by návrh obsahoval pouze aktivity „rozhodnutí o poskytnutí půjčky“ a „kalkulace rizikovosti“. Z takového popisu bychom však pravděpodobně nepochopili, že tento proces je automatizovaný a na základě vstupů vygeneruje výstupy pro pracovníka úvěrové společnosti a dalším systémům [13-15].



Obrázek 3 – BPMN model 2. část, Zdroj: Autor

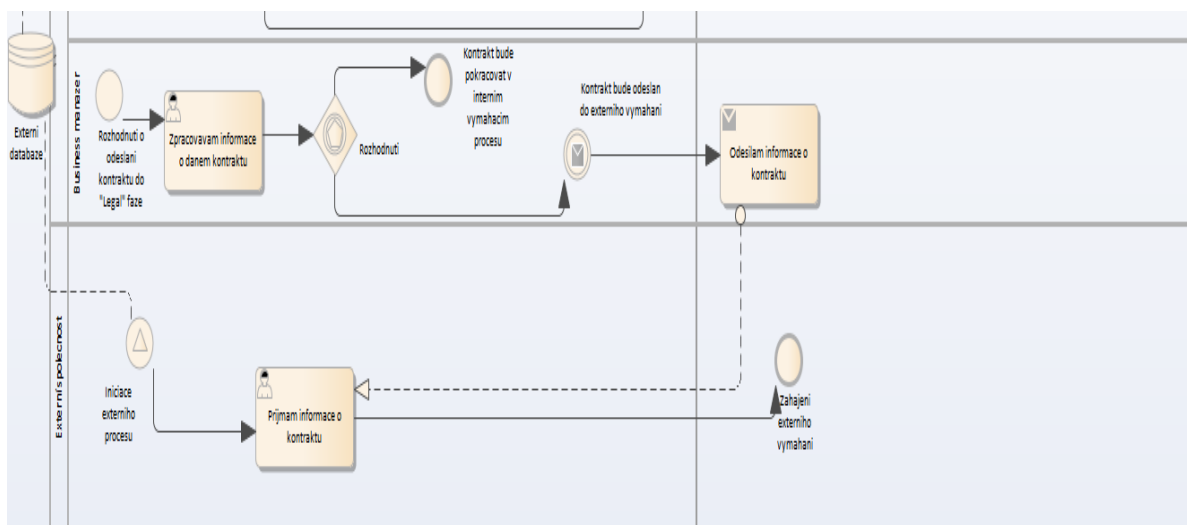


Obrázek 4 – BPMN model 2. část, Zdroj: Autor

Další část představená na obrázku číslo 3 a 4 demonstruje samotné procesy vymáhání. V tomto případě jde o procesy odesílání SMS klientovi pro připomenutí splátek, volání a fyzická návštěva vymahače u klienta. Hlavními procesy jsou zde interakce Systému „call centrum“ a „operátora“, dále zde hrají důležitou roli systém pro odesílání SMS – „SMS Gateway“ a „Vymahač“. V druhé části modelu jsou opět použity již zmíněné návrhové vzory Sequence a Exclusive choice a nově také Synchronizing Merge, Arbitrary Cycles a Implicit Termination.

Proces SMS Gateway je skoro celý navržený pomocí návrhového vzoru Synchronizing Merge. Tento návrhový vzor je specifický dvěma rozhodovacími procesy za sebou. V následujícím případě se jedná o situaci, kdy je potřeba rozhodnout na základě rizikovosti klienta, která z SMS šablon bude klientovi odeslána. Klíčovými body pro rozhodování jsou v tomto případě již v předchozí části zmíněné elementy „event“. Tento element spouští firemní proces nebo jednotlivou aktivitu a je důležitý pro řízení chování celého prostředí. Na konci procesu je použitý návrhový vzor Implicit Termination, jehož vizualizací dosáhneme jednoho ze dvou konců procesu, a to „SMS odeslána“ nebo „SMS neodeslána“.

Vymahač v této části plní roli člověka, který interaguje s klientem v procesu fyzického vymáhání. Především se zajímá o to, zda klient zaplatí splátku či nikoliv. Tento stav je v modelu vyobrazen v první části navrhovaného modelu na obrázku číslo 2. V procesu vymahače se nachází i poslední návrhový vzor „Arbitrary Cycles“, který je specifický zpětnou vazbou již běžícího procesu. To lze vidět v případě, že klient nebyl nalezen na adrese, a proto nemohl vymahač pokračovat v procesu a musel se vrátit zpět do své aplikace, kde vybere další kontrakt, který navštíví [13-15].



Obrázek 5- BPMN model 3. část, Zdroj: Autor

V poslední části modelu, který představuje obrázek číslo 5, je možné vidět proces „Business manažera“ a „Externí společnosti“. Zde dochází k interakci mezi těmito dvěma aktéry.

Mezi procesy můžeme vidět 3 různé typy interakcí. Jednotná černá čára s šipkou na konci se nazývá „Sequence flow“ a používá se ke spojení více elementů v rámci jednoho poolu. Pool představuje účastníky procesu. V naše případě se jedná o „Externí společnost“. Pool může být dále rozdělený do tzv. „Lanes“, které reprezentují konkrétní představitele daného procesu. „Business manažer“ je jeden ze sedmi představitelů konkrétního procesu v navrhovaném modelu.

Kromě Sequence flow je v poslední části možné vidět i „Message Flow“, jež se používá pro komunikaci mezi dvěma pooly a je značen přerušovanou čarou se šipkou na konci.

Poslední možností interakce je tzv. „Association“. Tento typ interakce se objevuje v situacích, kdy proces potřebuje ke správnému fungování další data. To je možné vidět ve spojení „Externí databáze“ a „Iniciace externího procesu“, kdy jsou oba elementy spojeny přerušovanou čarou bez šipky [13, 14, 16, 17].

Detailnější analýza a návrh vymáhacího procesu je představena v dalších kapitolách (především v kapitole 4.2 Use case diagram a v kapitole 5.1 Class diagram).

### 3 SYSTÉMOVÉ POŽADAVKY

Následující kapitola je zaměřena na definování systémových požadavků. Pro správnou analýzu a návrh systému je klíčové správně popsat služby, které zákazník požaduje od systému. Požadavky dále představují omezení, které definují správný vývoj informačního systému. Definice systémových požadavků může představovat obecný popis, ale i naopak velice detailní specifikaci.

Systémové požadavky jsou úzce spjaty s definováním případů užití a diagramu tříd. Stejně jako případy užití jsou klíčovou specifikací funkcionality systému pro uživatele a zákazníky, protože jsou psány v běžném jazyce.

Důležitost systémových požadavků popisuje článek „From user requirements to UML class diagram“, jehož hlavním cílem je analýza systémových požadavků s využitím přirozeného jazyka za účelem vybudování správného UML diagramu tříd s využitím doménové ontologie. Tato studie bere jako vstup textová data vyjádřená v přirozeném jazyce a následně reprezentování systémových požadavků. Na základě těchto dat se identifikují jména tříd, jejich atributy a asociace mezi nimi. K sestavení výsledků jsou použity techniky „NLP based tool“ a „Diagram class builder“. Výsledkem studie je zlepšení identifikace relevantních tříd a jejich atributů. Výsledky jsou detailně analyzovány porovnáváním výstupů s využitím systémových požadavků a bez nich [18].

Požadavky na systém jsou specifikovány na základě předchozí identifikace business procesů a navrhnutého BPMN modelu. Ten sloužil jako klíčové vodítko pro správný a jednodušší návrh systémových požadavků.

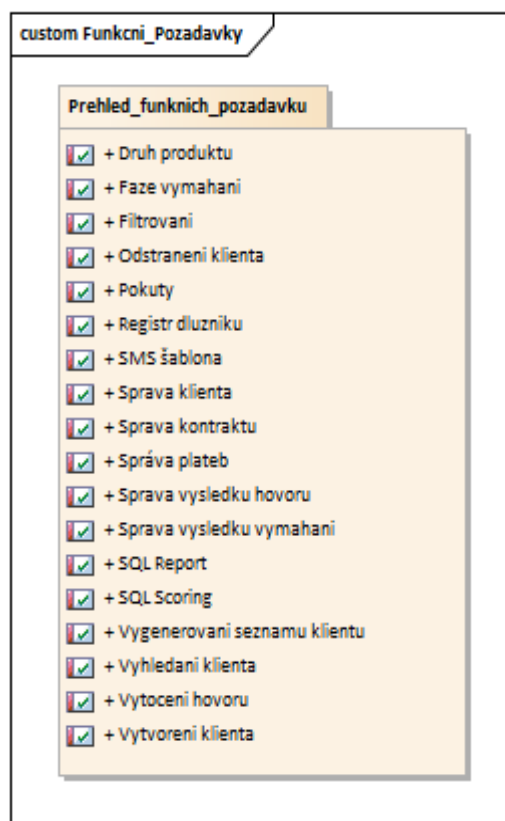
Primární rozdělení je na funkční a non-funkční požadavky. Popis jednotlivých požadavků je představen v následujících dvou podkapitolách. Pro správný popis a rozlišení každého z požadavků je doporučeno specifikovat jejich údaje.

Následující vlastnosti jsou povinnou součástí každého požadavku [19, 20]:

- Název – pojmenování vystihující požadavek
- ID – jednoznačný identifikátor konkrétního funkčního/non-funkčního požadavku
- Popis – stručný popis požadavku
- Priorita – význam přiřazený konkrétnímu požadavku

### 3.1 Funkční požadavky

Funkční požadavky by měly poskytnout náhled na to, co by měl systém nabízet. Jak bude systém reagovat na konkrétní vstupy a jak se bude chovat v určitých situacích. V některých případech mohou být ve funkčních požadavcích zahrnuty procesy, které by systém naopak neměl dělat [19, 20].



Obrázek 6 - Funkční požadavky, Zdroj: Autor



Obrázek číslo 6 představuje funkční požadavky informačního systému pro vymáhání. Tyto požadavky by měly blíže napovědět, jak bude systém přibližně fungovat, co můžeme od systému očekávat a jak bude uživatel systém využívat. Funkční požadavky jsou stanoveny na základě BPMN modelu, již zmíněných teoretických poznatků a business popisu.

custom Prehled_funkcnich_požadavku	
<p><b>Vyhledani klienta</b> <span style="float: right;">tags</span></p> <p>ID = 1            Popis = Vyhledání klienta a zobrazení všech jeho atributu            Priorita = Nezbytný</p>	<p><b>Filtrovani</b> <span style="float: right;">tags</span></p> <p>ID = 6            Popis = Systém automaticky zaradí klienta do konkrétní fáze vymáháčího procesu            Priorita = Nezbytný</p>
<p><b>Odstraneni klienta</b> <span style="float: right;">tags</span></p> <p>ID = 2            Popis = Odstranení veškerých dat o klientovi            Priorita = Nezbytný</p>	<p><b>Pokuty</b> <span style="float: right;">tags</span></p> <p>ID = 7            Popis = Klientovi bude naúčtována pokuta z prodlení            Priorita = Nezbytný</p>
<p><b>Vytvoreni klienta</b> <span style="float: right;">tags</span></p> <p>ID = 3            Popis = Systém vytvorí klienta na základe poskytnutých informaci            Priorita = Nezbytný</p>	<p><b>Druh produktu</b> <span style="float: right;">tags</span></p> <p>ID = 8            Popis = Systém priradí klientovi druh produktu            Priorita = Nezbytný</p>
<p><b>Sprava kontraktu</b> <span style="float: right;">tags</span></p> <p>ID = 4            Popis = Systém priradí kontrakt správnému klientovi            Priorita = Nezbytný</p>	<p><b>Vytoceni hovoru</b> <span style="float: right;">tags</span></p> <p>ID = 9            Popis = Systém automaticky vytocí jedno z klientových čísel            Priorita = Nezbytný</p>
<p><b>Faze vymahani</b> <span style="float: right;">tags</span></p> <p>ID = 5            Popis = Systém automaticky priradí klienta do určité fáze vymáhání            Priorita = Nezbytný</p>	<p><b>Sprava vysledku hovoru</b> <span style="float: right;">tags</span></p> <p>ID = 10            Popis = Vysledek hovoru se zaznamená a archivuje            Priorita = Nezbytný</p>
	<p><b>Správa plateb</b> <span style="float: right;">tags</span></p> <p>ID = 17            Popis = Systém zpracuje platby            Priorita = Nezbytný</p>

Obrázek 7 - Přehled funkčních požadavků 1. část, Zdroj: Autor

<b>Sprava vysledku vymahani</b>	tags
ID = 11 Popis = Výsledek fyzického vymáhání se zaznamená a archivuje Priorita = Nezbytný	
<b>SQL Report</b>	tags
ID = 12 Popis = Systém prepocítá na základe SQL scriptu vytvorí report v ruzných formátech Priorita = Nezbytný	
<b>Vygenerovani seznamu klientu</b>	tags
ID = 13 Popis = Systém na základe filtrace nebo SQL scriptu vytvorí seznam klientu Priorita = Nezbytný	
<b>SQL Scoring</b>	tags
ID = 14 Popis = Systém prepocítá na základe SQL scriptu rizikovost klienta Priorita = Nezbytný	
<b>Registr dlužniku</b>	tags
ID = 15 Popis = Systém pravidelne odesílá seznamy dlužniku Priorita = Nezbytný	
<b>SMS šablona</b>	tags
ID = 16 Popis = Systém automaticky zpracuje konkrétní šablona pro klienta a poskytne jí SMS bráne Priorita = Nezbytný	

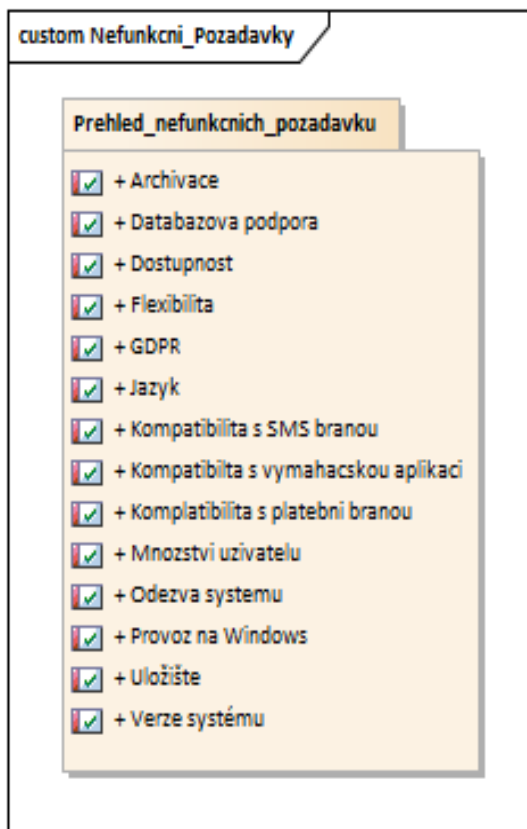
Obrázek 8 - Přehled funkčních požadavků 2. část, Zdroj: Autor

Na obrázku číslo 7 a 8 (příloha číslo 3 – přehled funkčních požadavků) je možné vidět bližší specifikaci jednotlivých funkčních požadavků. Zde je klíčový popis, který přiblíží danou funkcionalitu systému a lépe specifikuje případy užití, které jsou představeny v následující kapitole.

## 3.2 Non-funkční požadavky

Non-funkční požadavky představují sadu omezení, které systém bude nabízet. Obvykle se non-funkční požadavky aplikují na celý systém, a ne na jednotlivé funkce nebo služby systému.

Stanovené nefunkční požadavky na obrázku číslo 9 specifikují vnitřní strukturu systému. Představují omezení, které je od systému vyžadováno a kterých by se systém měl držet [19, 20].



Obrázek 9 - Nefunkční požadavky, Zdroj: Autor

custom Přehled_nefunkčních_požadavku	
<p><b>GDPR</b></p> <p style="text-align: center;">tags</p> <p>ID = 1            Popis = Systém se bude řídit zákony GDPR            Priorita = Nezbytný</p>	<p><b>Dostupnost</b></p> <p style="text-align: center;">tags</p> <p>ID = 5            Popis = Systém bude dostupný v nastavenou pracovní dobu            Priorita = Nezbytný</p>
<p><b>Jazyk</b></p> <p style="text-align: center;">tags</p> <p>ID = 2            Popis = Systém bude naprogramován v jazyce Java s využitím nejnovějších Frameworku            Priorita = Nezbytný</p>	<p><b>Uložiště</b></p> <p style="text-align: center;">tags</p> <p>ID = 6            Popis = Systém musí být schopný pracovat se 100 miliony klientu            Priorita = Nezbytný</p>
<p><b>Verze systému</b></p> <p style="text-align: center;">tags</p> <p>ID = 3            Popis = Systém bude fungovat na konkrétní verzi            Priorita = Nezbytný</p>	<p><b>Archivace</b></p> <p style="text-align: center;">tags</p> <p>ID = 7            Popis = Systém bude schopný archivovat veškeré informace o provedených aktivitách na klientovi            Priorita = Nezbytný</p>
<p><b>Odezva systému</b></p> <p style="text-align: center;">tags</p> <p>ID = 4            Popis = Systém bude umet zpracovat 10 tisíc klientu a 20 tisíc výsledku na jednou            Priorita = Nezbytný</p>	<p><b>Provoz na Windows</b></p> <p style="text-align: center;">tags</p> <p>ID = 8            Popis = Systém bude naprogramovaný pro využití na operacím systému Windows            Priorita = Nezbytný</p>

Obrázek 10 - Přehled nefunkčních požadavků 1. část, Zdroj: Autor

Na obrázku 10 a 11 (příloha číslo 4 – přehled nefunkčních požadavků) je specifikace výše zmíněných nefunkčních požadavků. Tyto požadavky budou hrát důležitou roli při samotném nastavování systému.

<b>Mnozství uživatelů</b>	<i>togs</i>
ID = 9 Popis = Systém bude schopný definovat uživatelské role Priorita = Nezbytný	
<b>Kompatibilita s SMS branou</b>	<i>togs</i>
ID = 10 Popis = Systém bude umět odesílat/přijímat informace do/z SMS brány Priorita = Nezbytný	
<b>Kompatibilita s platební branou</b>	<i>togs</i>
ID = 11 Popis = Systém bude umět přijímat informaci o zaplacení Priorita = Nezbytný	
<b>Kompatibilita s vymahacskou aplikací</b>	<i>togs</i>
ID = 12 Popis = Systém bude umět komunikovat s vymahacskou aplikací Priorita = Nezbytný	
<b>Databázová podpora</b>	<i>togs</i>
ID = 13 Popis = Systém bude předávat/získávat data do/z Databázového skladu Priorita = Nezbytný	
<b>Flexibilita</b>	<i>togs</i>
ID = 14 Popis = Systém bude umět lehce reagovat na změny strategií Priorita = Nezbytné	

Obrázek 11- Přehled nefunkčních požadavků 2. část, Zdroj: Autor

Zmíněné funkční i nefunkční systémové požadavky budou klíčovou předlohou pro následující kapitoly 4 „Model případů užití“ a 5 „Struktura a dynamika systému“.

## 4 MODEL PŘÍPADŮ UŽITÍ

Následující kapitola blíže představuje funkcionalitu a chování navrhovaného systému. V kapitole jsou definovány případy užití a jejich interakce mezi nimi. Dále také vnitřní struktura případů užití, která je představena ve dvou různých podobách (pomocí diagramů aktivit a klasických psaných scénářů). Všechny aplikované praktické návrhy jsou v kapitole podloženy teoretickými poznatky.

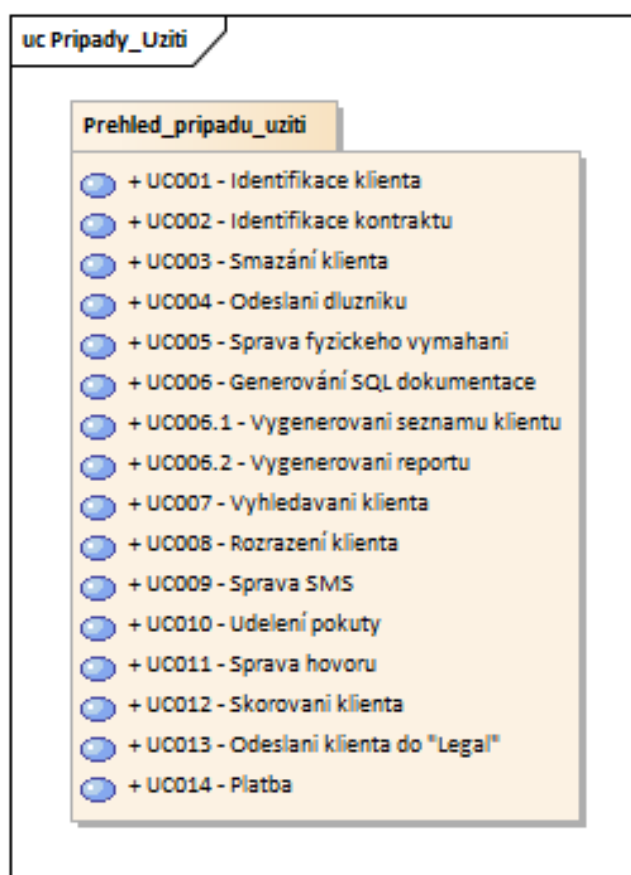
Při modelování informačního systému vždy existují kontroverzní pohledy na různé UML diagramy. Následující výzkum se ptá na otázky, zda jsou Use case diagram a class diagram užitečné při analýze a návrhu informačního systému. Hlavní otázkou je, zda existuje nějaký efekt mezi use case diagramem a class diagramem na výsledky a lepší pochopení problému. Výzkum je prováděn na základě stanovených hypotéz, testu ANOVA. Výsledek výzkumu představuje fakt, že use case diagram interpretuje model úplně a je lépe pochopitelný než class diagram. Naopak class diagram oproti use case diagramu je komplexnější. Výsledkem tohoto výzkumu je fakt, že je lepší se zákazníkem konzultovat systém, který bude popsán use case diagramem, než class diagramem. Naopak pro vývojáře je lepší variantou class diagram [21].

Z toho důvodu jsou v diplomové práci použity oba diagramy, pro detailnější popis návrhu informačního systému pro vymáhání. Tento návrh má být totiž podkladem pro podniky, které budou tento systém využívat jako koncoví uživatelé a zároveň pro vývojáře, kteří budou tento návrh používat pro vytvoření kódu.

### 4.1 Případy užití

Případem užití lze chápat jeden případ užití informačního systému. Je důležitý pohled na užitek, jaký pro informační systém přináší jednotlivé případy užití, jak analyticky, tak i z pohledu businessu. Jednotlivé případy užití představují funkcionality systému, reflektují to, co bude systém umět.

Při hledání případů užití mohou být aplikovány dvě různé metodiky. První metodika, je při hledání případů užití důležité nalézt nejprve prvky z okolí systému, což jsou v tomto případě aktéři. Mimo aktéry je také doporučeno najít ty prvky z okolí, které budou komunikovat se systémem nebo budou systému předávat informace. Další možná metodika spočívá v nalezení procesů podniku v okolí, které vyvolávají nějakou aktivitu systému. Pro účely diplomové práce byl vybrán druhý způsob, protože procesy již jsou definovány v kapitole 2.2 BPMN model. Tyto procesy, spolu se systémovými požadavky, tedy budou použity pro nalezení případů užití [22].

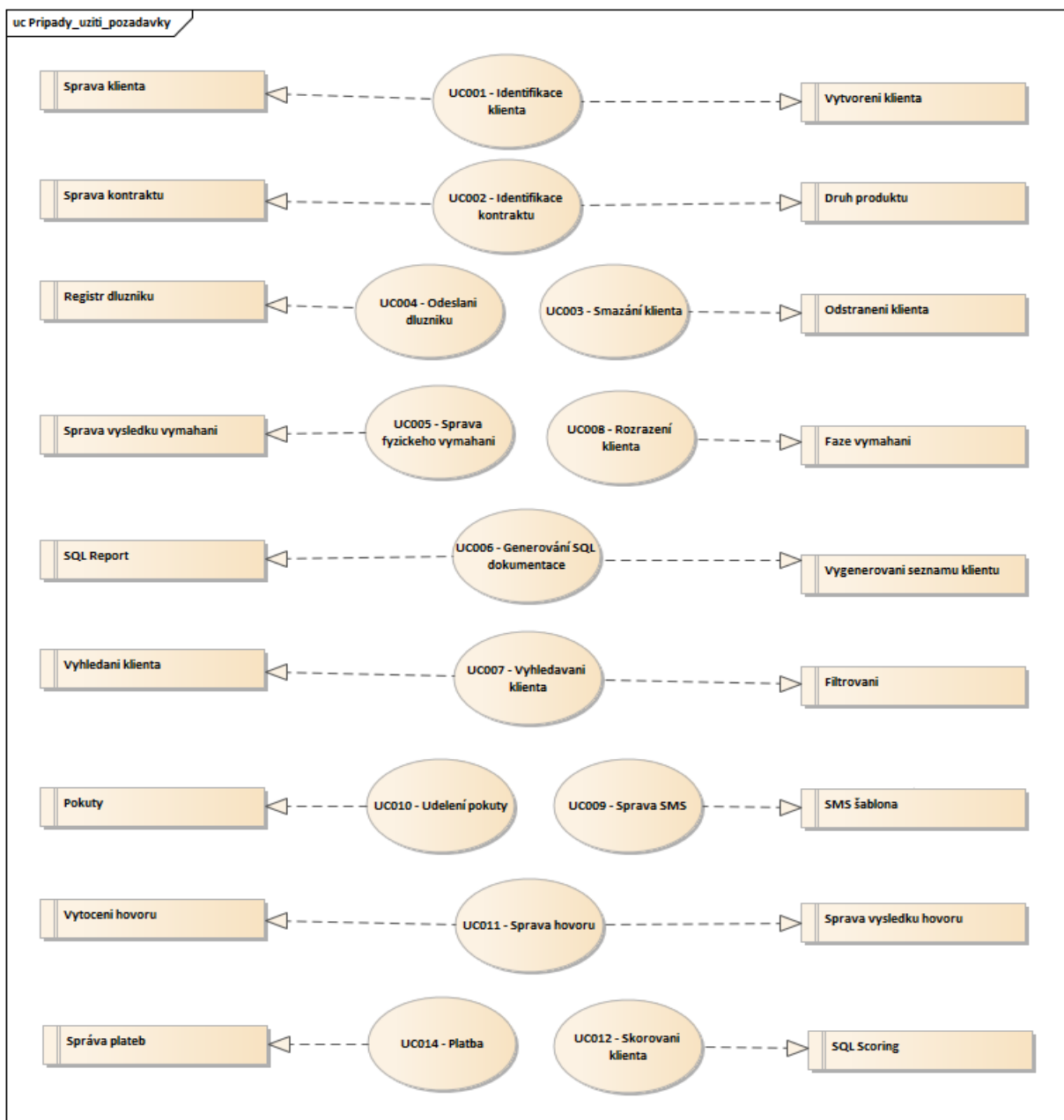


Obrázek 12 - Případy užití, Zdroj: Autor

Na obrázku číslo 12 je možné vidět nalezených 16 případů užití (14 případů užití plus 2 rozdělené generalizací – případy užití UC006.1 a UC006.2). Tyto případy užití představují hlavní funkcionality systému pro vymáhání. Systém bude umět pracovat s klienty a kontrakty a bude interagovat s externími systémy a databázemi kvůli přenosům dat a informací. Další důležité funkce jsou „Rozřazení klienta“ a „Skorování klienta“, které jsou úzce spjaty s vymáhacími procesy „Správa SMS“, „Správa hovorů“, „Správa fyzického vymáhání“ a „Udělení pokuty“. Vybrané případy užití a jejich bližší popis bude představen v následujících kapitolách.



#### 4.1.1 Interakce s funkčními požadavky



Obrázek 13 - Interakce případů užití s funkčními požadavky, Zdroj: Autor

Jak již bylo zmíněno, funkční požadavky jsou mimo vnější procesy důležitou částí pro správnou identifikaci případů užití, jejich vnitřní struktury a interakcí. Na obrázku číslo 13 je vyobrazen vztah mezi funkčními požadavky a případy užití. Díky tomuto modelu dokážeme zjistit, zda jsme nezapomněli na nějaký požadavek nebo naopak, jestli nemáme funkcionalitu navíc.

#### 4.1.2 Aktéři

Prvky typu aktér, interagují se systémem z jeho okolí. Mohou představovat uživatele systému nebo jiné externí systémy. Každý z aktéru by měl mít jméno a stručný popis, pro lepší identifikaci a představu, jak a proč bude právě tento aktér interagovat se systémem. Aktéři komunikují se systémem z různých důvodů: kvůli iniciaci případu užití; k získání určitých dat, které dále budou prezentovány aktérovi; kvůli nějaké speciální aktivitě mimo systém, o které by však systém měl vědět. Rozlišení aktérů by mělo záviset na odlišnostech v možnostech využití systému, často se jedná i o jiný interface pro každého z aktérů [23].



Obrázek 14 - Aktéři, Zdroj: Autor

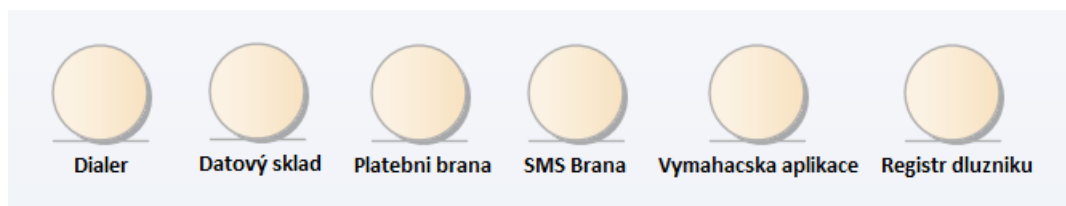
Na základě BPMN modelu, případů užití a jejich interakcemi s funkčními požadavky, jsou definovány 4 role typu aktér. Ti představují hlavní uživatele, kteří budou interagovat se systémem a využívat jeho funkcí.

Prvním Aktérem je **Čas** – slouží k iniciaci těch funkcionalit systému, které jsou závislé na konkrétním časovém okamžiku.

**Analytik** – představuje jednoho z uživatelů systému. Tento aktér se systémem interaguje především z pohledu analytických funkcionalit systému.

**Bankovní poradce** – tento aktér je dalším uživatelem systému, který zaštiťuje počáteční fázi vymáhacího procesu, tedy využívá funkcionalit při vytváření dokumentů pro vznik klienta a kontraktu.

Posledním uživatelem systému je **Supervisor** – ten představuje roli manažera s vyššími právy vůči systému.



Obrázek 15 - Aktéři (externí entity), Zdroj: Autor

Dále je definováno 6 externích entit, které jsou však se systémem úzce spjaté.

**Dialer** – systém pro vytáčení telefonních hovorů

**Datový sklad** – Databáze, která poskytuje data systému

**Platební brána** – Sdílení informací systému o platbě klientů

**SMS brána** – Portál pro odeslání zprávy

**Vymahačská aplikace** – Aplikace pro správu fyzického vymáhání

**Registr dlužníků** – Systém pro správu delikventních klientů

### 4.1.3 Vnitřní struktura případů užití

Pan Ilja Kraval ve své knize „Analytické modelování pomocí UML v praxi“ píše o třech školách, jak popisovat vnitřní strukturu případů užití. Popisuje, že je možné vnitřky případů užití nepopisovat vůbec. Tento přístup se však nedoporučuje, protože tak přicházíme o příležitost, jak představit algoritmy do technického zpracování programu a také hrozí nebezpečí, že nezjistíme špatně nadefinované případy užití a jejich relace. Další dva možné přístupy jsou použity v této diplomové práci, specifikované níže v této kapitole na obrázku číslo 16 až 18 a jako slovní popis pod těmito obrázky.

První přístup je popis vnitřků případů užití pomocí diagramu aktivit. Principem tohoto přístupu je využití vnější aktivity (aktivity v BPMN modelu) pro spuštění vnitřních aktivit, tedy aktivit systému (případů užití). V knize je uvedeno, že tento přístup popisu vnitřního chování případů užití, je z přístupu UML úplně čistší, avšak má praktickou nevýhodu v tom, že nemůže být použitý ve všech částech dokumentace na rozdíl od dalšího přístupu pomocí slovních scénářů.

Velmi preferovaným přístupem jsou slovní scénáře případů užití, který je představen za obrázky diagramů aktivit. Největší efekt popisu vnitřní struktury případů užití je kombinace slovních scénářů a diagramů aktivit [22].

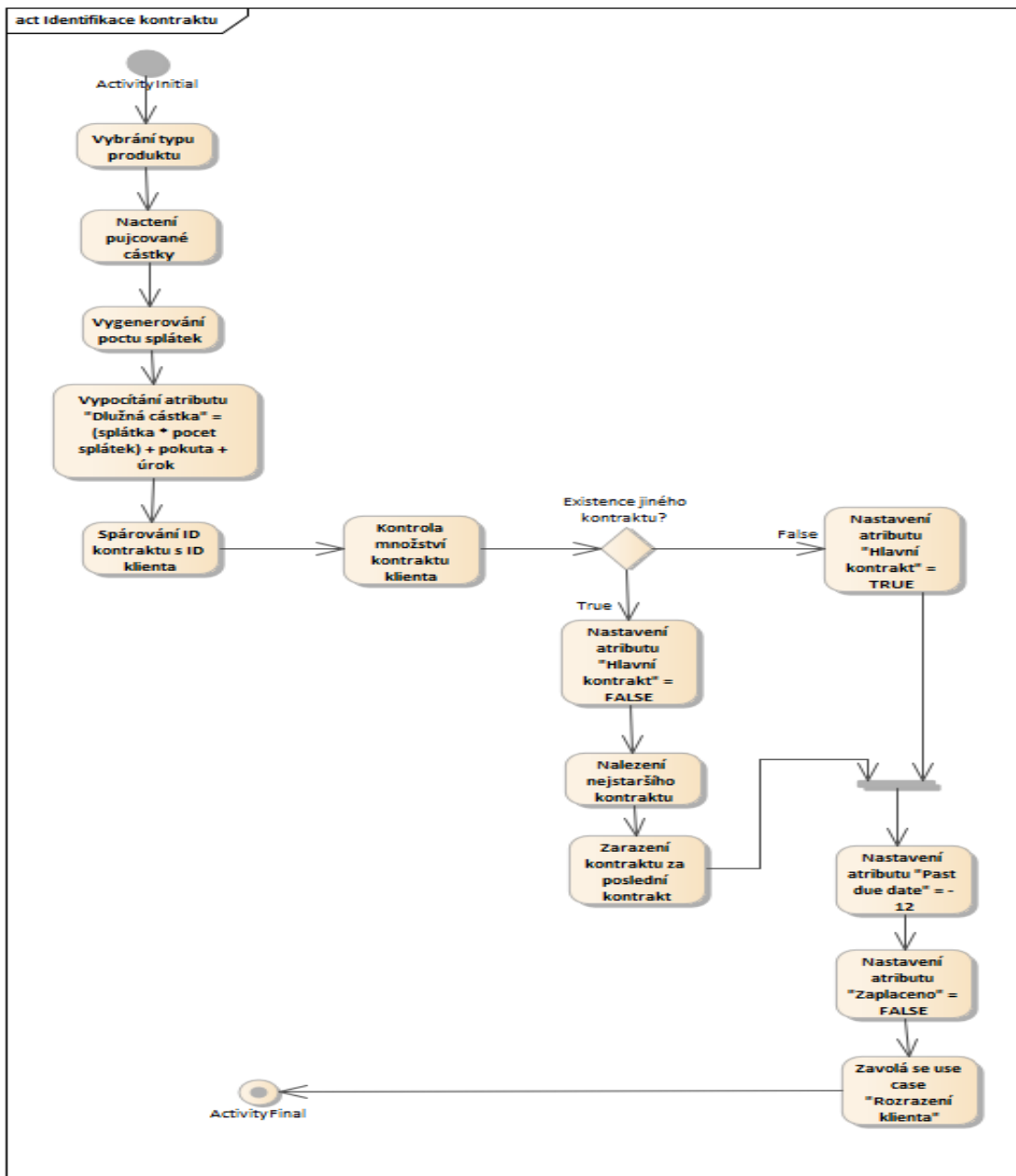
#### 4.1.3.1 Diagramy Aktivit

Následující tři obrázky číslo 16 až 18 ukazují vybrané diagramy aktivit popisující vnitřní strukturu případů užití „Identifikace kontraktu“, „Rozřazení klienta“ a „Skórování klienta“. Tyto případy užití jsou vybrány, protože nejlépe demonstrují hlavní funkcionalitu systému a jsou na sebe navázány. Pro lepší pochopení diagramů je však nutné si vysvětlit některé atributy, které jsou v diagramech aktivit použity:

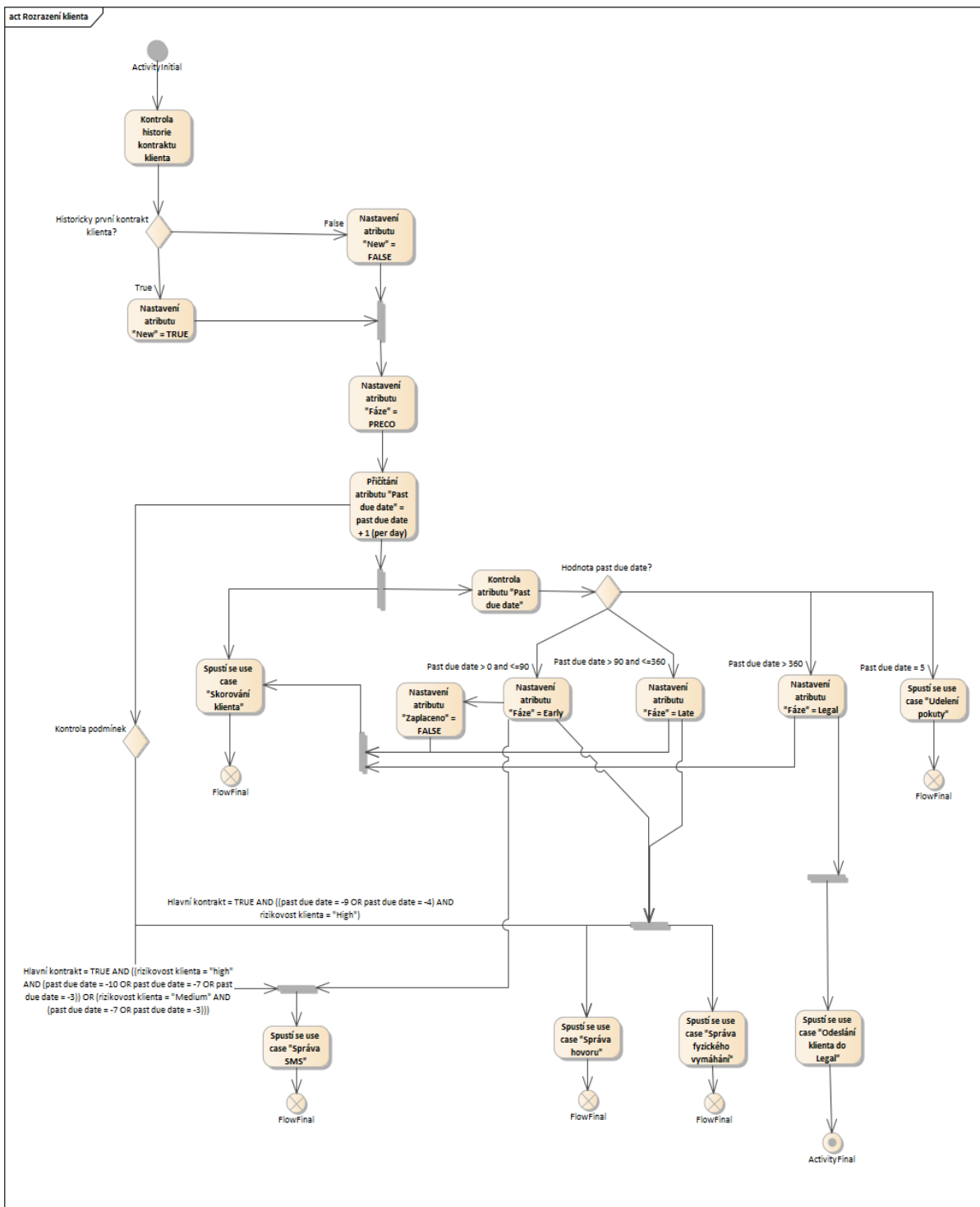
**Atribut „Hlavní kontrakt“** – Když bude tento atribut TRUE, automatické spuštění funkcionalit systému bude záležet právě na tomto kontraktu.

**Atribut „Past due date“** – Doba splatnosti kontraktu (Záporné číslo – počet dní před splatností)

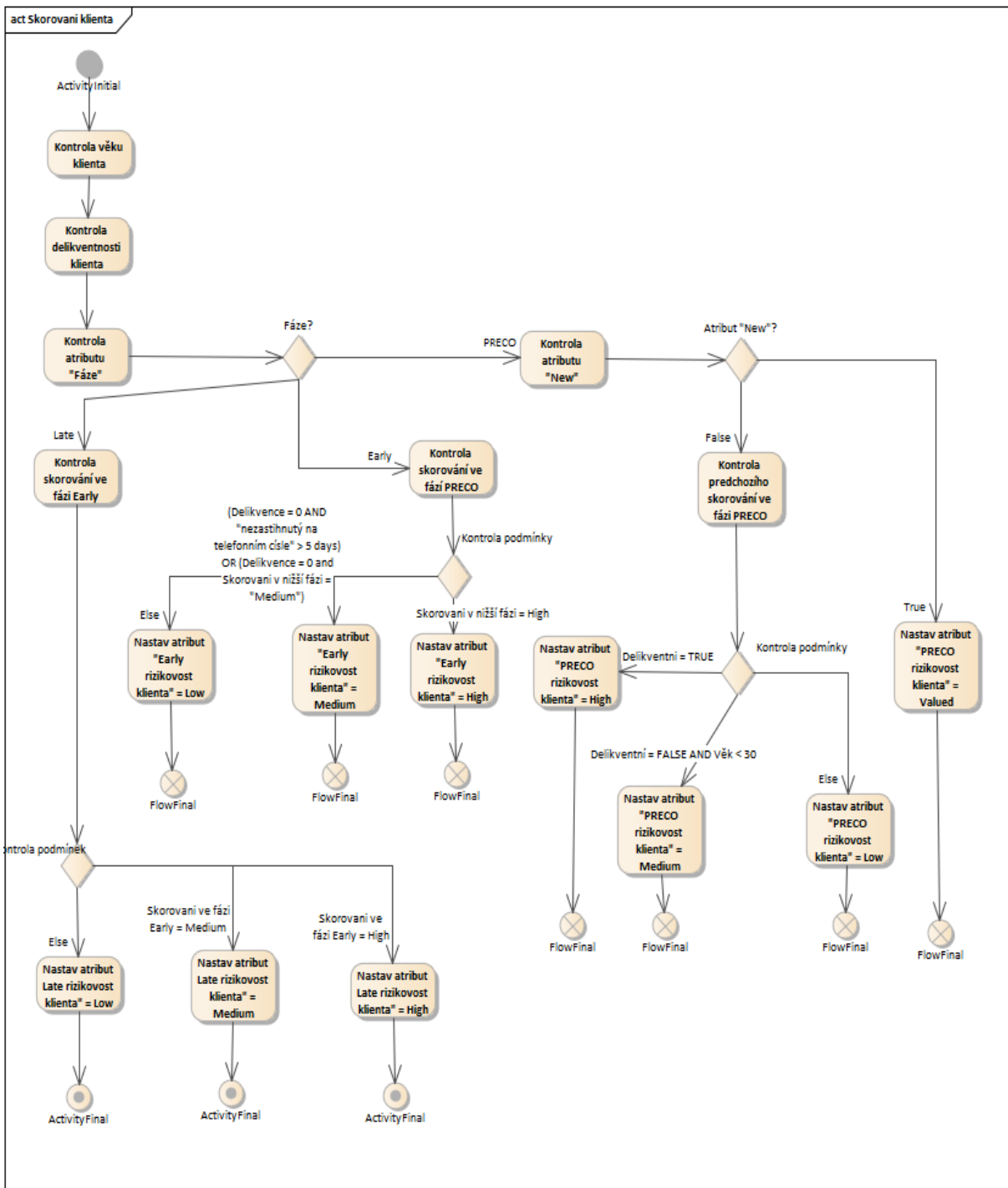
**Atribut „New“** – Sleduje, zda kontrakt je historicky klientův první, či nikoliv.



Obrázek 16 – Diagram aktivit: Identifikace kontraktu, Zdroj: Autor



Obrázek 17 – Diagram aktivít: Rozřazení klienta, Zdroj: Autor



Obrázek 18 - Diagram aktivít: Skórování klienta, Zdroj: Autor



#### 4.1.3.2 Scénáře případů užití

Již zmíněný druhý způsob popisu vnitřní struktury případů užití je představen v následujících třech scénářích případů užití „Správa SMS“, „Generování SQL dokumentace“ a „Vygenerování seznamu klientů“. Všechny scénáře navazují na případy užití představené diagramy aktivit.

#### **Případ užití „Správa SMS“**

1. Zpracuj informace o klientovi

2. Načti atribut "Fáze"

3. Načti atribut "Rizikovost klienta"

3.1. Pokud Rizikovost klienta = "high" AND Fáze = "PRECO", 3.1.1. Vystav šablonu "PRECO\_High"

3.2. Pokud rizikovost klienta = "Medium" AND Fáze = "PRECO", 3.2.1. Vystav šablonu "PRECO\_Medium"

3.3. Cyklus: For past due date = past due date + 10 End past due day = 91 OR Zaplaceno = TRUE

3.3.1. Pokud rizikovost klienta = "Low" AND Fáze = "Early", 3.3.1.1 Vystav šablonu

"Early\_Low" 3.3.2. Pokud rizikovost klienta = "Medium" AND Fáze = "Early", 3.3.2.1. Vystav

šablonu "Early\_Medium" 3.3.3. Pokud rizikovost klienta = "High" AND Fáze = "Early",

3.3.3.1. Vystav šablonu "Early\_High"

4. Vygeneruj všechny potřebné atributy klienta 4.1. Vlož informace do šablony
5. Zaregistruj telefonní číslo do SMS brány
6. Iniciuj odeslání SMS přes SMS bránu

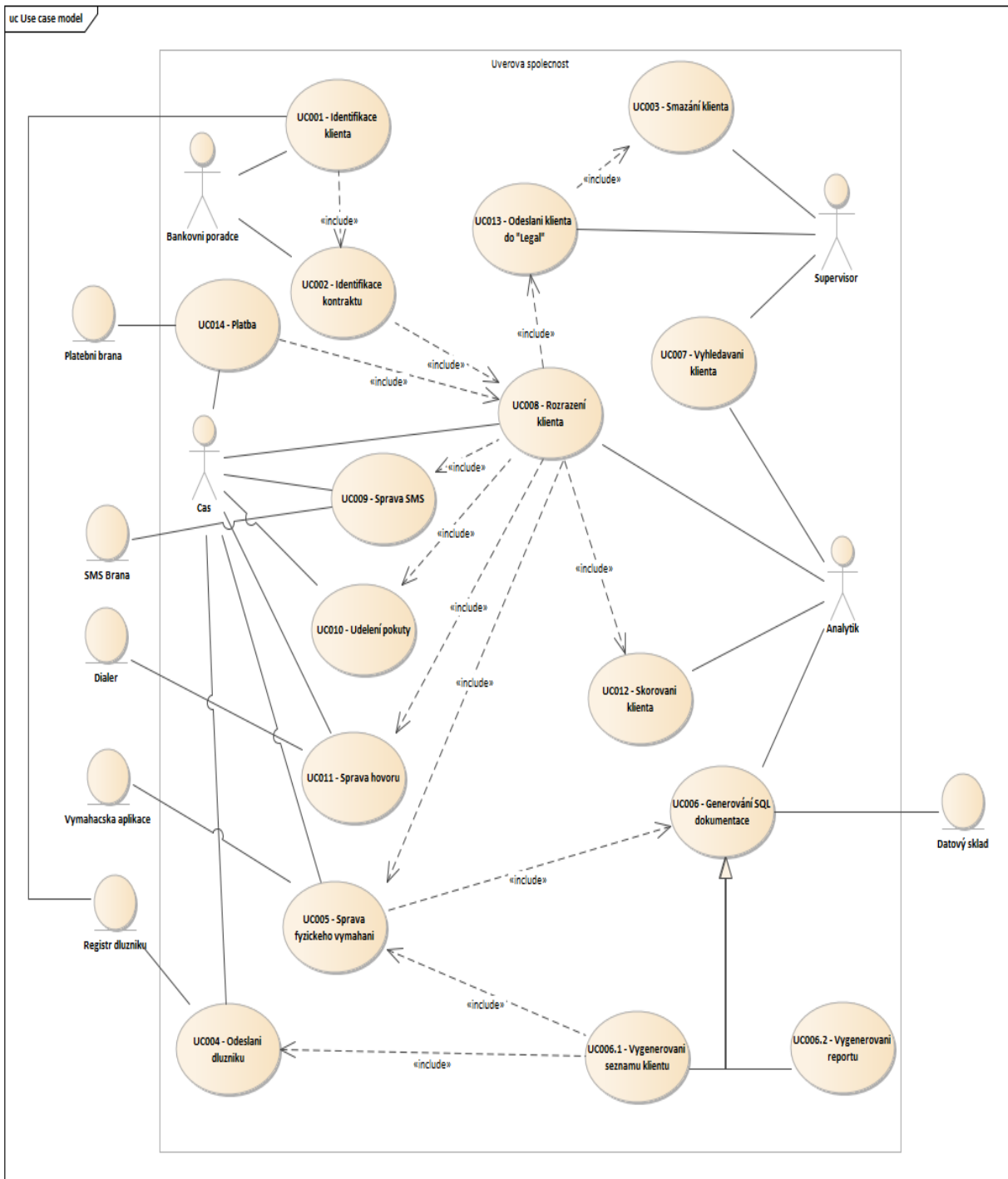
### **Případ užití „Generování SQL dokumentace“**

1. Zobraz možnosti
2. Pokud Analytik chce psát vlastní SQL dotaz, 2.1. Zobraz okno pro psaní SQL dotazu
3. Pokud Analytik chce vygenerovat dokument, 3.1. Načti typy dokumentu
4. Našepť chyby v SQL scriptu

### **Případ užití „Vygenerování seznamu klientů“**

1. Analytik iniciuje generování seznamu klientů
2. Zobraz typy vygenerovaných seznamů
3. Pokud typ vygenerovaného seznamu = "Fyzické vymáhání", 3.2. Zprostředkuj seznam spolu s potřebnými daty, 3.1. Spustí se use case "Správa fyzického vymáhání"
4. Pokud typ vygenerovaného seznamu = "Registr dlužníků", 4.1. Vygeneruj excel dokument s daty
5. Spustí se use case "Odeslání dlužníků"

## 4.2 Use case diagram



Obrázek 19 - Diagram případů užití, Zdroj: Autor

Poslední část definice funkcionality systému je vytvoření use case diagramu, který shrnuje všechny dříve představené podkapitoly do jednoho diagramu. Na obrázku číslo 19 je možné vidět všechny případy užití ve vzájemné interakci mezi sebou, s aktéry, externími systémy a databázemi. Tento diagram má za úkol shrnout, co bude systém dělat, tedy celkový přehled chování systému.

Specifikace relací mezi jednotlivými entitami v modelu je velice důležitá, protože definuje, jak jedna entita k druhé bude přistupovat. Příkladem tohoto vztahu může být typ <<include>>, který zahrnuje kroky z jednoho případu užití do jiného případu užití. Což je užitečné, když se zahrnuté kroky vyskytují jako sekvence v mnoha různých kontextech. Příkladem může být vztah případu užití „Rozřazení klienta“ a případů užití „Správa SMS“, „Správa hovorů“, „Správa fyzického vymáhání“ a „Udělení pokuty“. V tomto případě je „Rozřazení klienta“ iniciátorem ostatních případů užití a ostatní případy užití jsou v něm zahrnuty.

Zajímavou relací je i vztah generalizace, který se objevuje u UC006 a jeho potomků UC006.1 a UC006.2. Jde o vztah „Rodiče“ a „Potomků“, kdy rodičem je případ užití UC006, který předává svoji vnitřní strukturu, tedy již představené scénáře (Chování daného případu užití) dále svým potomkům. Jednoduše řečeno, když se spustí případ užití UC006.1, chování systému bude reflektovat scénáře případu užití UC006.1 a zároveň UC006. Výše, v podkapitole 4.1.3 Vnitřní struktura případů užití je představen scénář UC006 – Generování SQL dokumentace. Toto chování tedy budou oba potomci aplikovat při spuštění UC006.1 nebo UC006.2. Kromě tohoto chování však bude umět, například případ užití UC006.1 nabídnout na výběr typy generovaných seznamů, spolu s jejich samotným vygenerováním do vybraných formátů. Celý scénář je možné vidět ve stejné kapitole u případu užití UC006.1 - Vygenerování seznamu klientu [24].

Z use case diagramu je také možné vidět interakce s externími systémy a samotným Core systémem (případy užití uvnitř hranice „úvěrová společnost“). Odesílání dat a komunikace mezi externími systémy a navrhovaným systémem je uskutečněna pomocí API a Webové

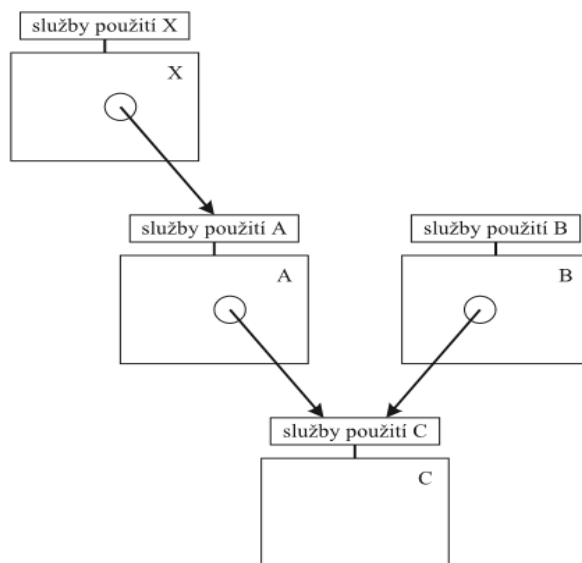
služby. Definice obou možností komunikace je velice podobná, liší se pouze ve zdroji odeslaných (respektive přijímaných) dat a ve faktu, že API může fungovat v určitých případech i bez připojení k internetu. Příkladem použití Webové služby může být přijímání dat z platební brány, kdy systém potřebuje tuto skutečnost zaregistrovat pro další možné fungování. Data budou v tomto případě přijímány jako SOAP formát. V use case diagramu je komunikace možná vidět ve spojení mezi externím systémem „Platební brána“ a případem užití UC014 „Platba“. API je v systému však použita častěji, a to například v situaci, kdy systém posílá informaci ve formátu JAR „Dialeru“, aby vytočil a spojil telefonní číslo klienta. V tomto případě je totiž důležité, aby systém nebyl závislý na připojení k internetu, protože jak již bylo v práci několikrát zmíněno, volání je jeden z nejdůležitějších vymáhacích procesů. Tato komunikace je v use case diagramu představena ve spojení mezi externím systémem „Dialer“ a případem užití UC011 „Správa hovoru“. Komunikace a sdílení dat probíhá přes cloud, aby se nemusel řešit problém s nedostatkem místa na hardveru [25 - 27].

## 5 STRUKTURA A DYNAMIKA SYSTÉMU

Následující kapitola představuje návrh tříd a jejich detailní popis. Mimo jiné jsou zde také specifikovány relace mezi těmito třídami a bližší specifikace funkcionalit a atributů daných tříd. Dále je v kapitole možné nalézt detailní popis vnitřního chování a vnitřní struktury navrhovaného systému. Je zde možné vidět navrhnuté stavové diagramy, které odrážejí dynamiku některých vybraných navrhnutých tříd. Sekvenční diagramy představují další popis vybraných definovaných případů užití z kapitoly model případů užití, které blíže specifikují chování systému v čase. Kapitola struktura a dynamika systému tedy vychází výhradně z předchozích kapitol, které jsou hlavním klíčem pro definici interakcí, struktury a dynamiky modelu. Všechny praktické návrhy jsou popsány pomocí teoretických poznatků, které jsou v kapitole blíže specifikovány.

Model tříd by má za úkol, určit představu nejen o datové struktuře, ale také o omezení a kritické části navrhovaného informačního systému. Správný model dokáže odhalit špatné kalkulace nebo chyby, které by mohly nastat.

Model tříd je hlavní reprezentací objektově orientovaného modelování (dále OOP). Základním principem OOP je znovu použitelnost, která hraje důležitou roli při návrhu tříd. Ke správnému nalezení znovu použitelných tříd je vhodné pochopit tzv. „objektové paradigma“ na základě kterého dokážeme určit, kde vlastně informace v IS leží, kde ji hledat, kam ji umístit apod. Obecný popis objektového paradigmatu představuje ve své knize Analytické modelování pomocí UML v praxi pan Ilja Kraval následovně: „Prvek v systému nabízí jiným prvkům svou službu možného použití. Jiný prvek přistupuje k tomuto prvku a použije tuto jeho nabízenou službu použití a tím tento prvek použije. Vzniká tak směrová interakce použití prvku prvkem. Klient má takto zpřístupněnou službu prvku, a přitom pro něj není viditelná vnitřní struktura použitého prvku nebo není viditelná implementace nabízené služby“. Tuto interakci a použití objektového paradigmatu je možné vidět na obrázku číslo 20.

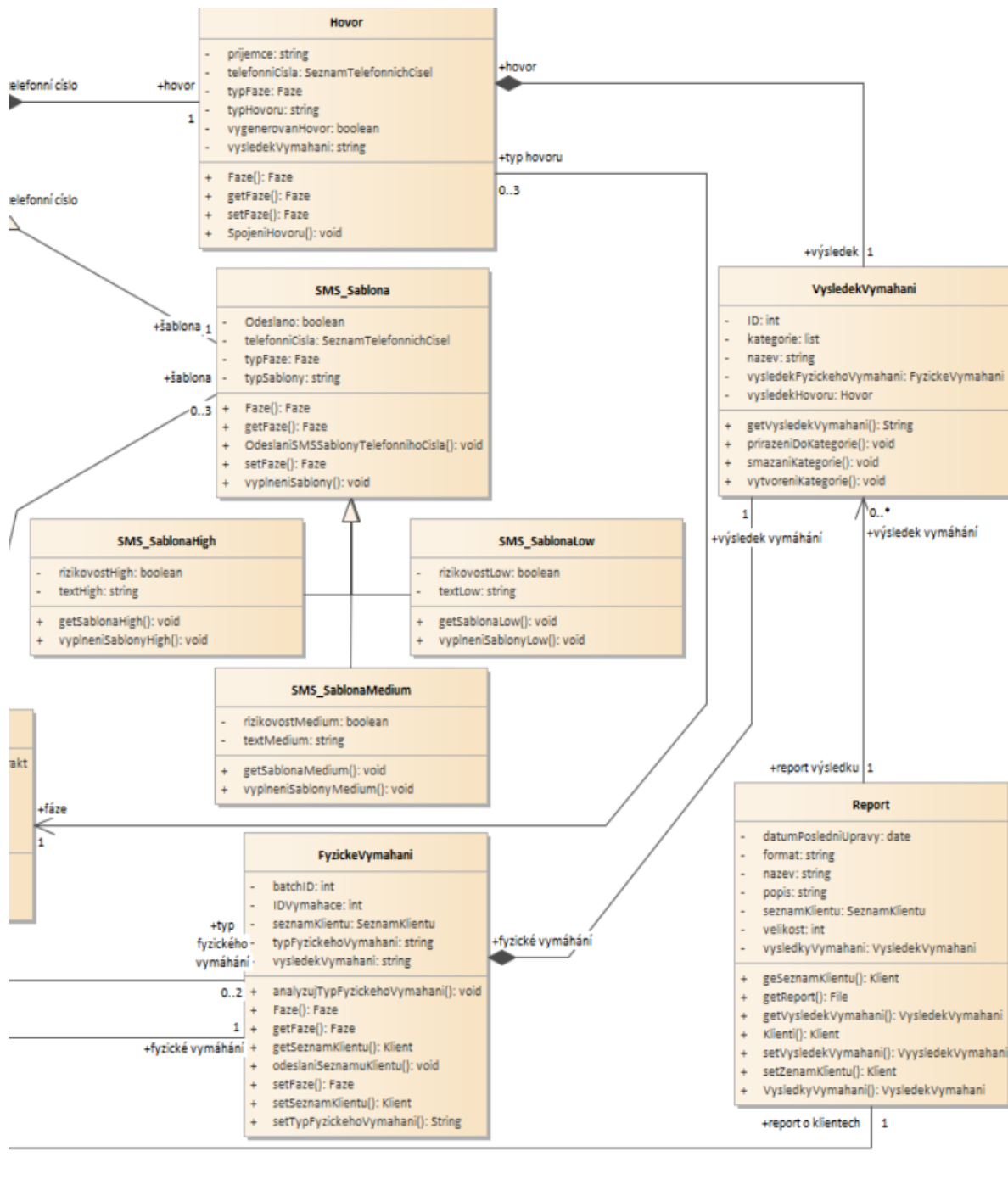


Obrázek 20 - Objektové paradigma, Zdroj: [22]

Častým problémem u nalézání tříd je tzv. „problém META“. Každá třída by měla být na úrovni Meta (tedy na nejvyšší úrovni znovupoužitelnosti) a měla by představovat flexibilní variantu. Často se totiž stává, že třídy jsou definovány na nižší úrovni (skoro na úrovni instancí) a poté je velmi obtížné nalézt instance tříd. Pro lepší pochopení je však nutné si vysvětlit, co instance třídy znamená. Jednoduše řečeno, instance třídy představuje konkrétní vizualizaci třídy. Jinými slovy, pokud existuje třída „Klient“ spolu s obecnými atributy, tak její instance bude „Osoba1“ s nadefinovanými konkrétními atributy té dané osoby [22, 28, 29].







Obrázek 22 - Class diagram 2. část, Zdroj: Autor

Pro lepší přehlednost je class diagram rozdělen do dvou částí. První část na obrázku číslo 21 představuje levou stranu modelu tříd a obrázek číslo 22 pravou stranu. Kompletní diagram tříd je možné nalézt v přílohách (příloha číslo 5 – class model). Class diagram je vytvořen na základě výše uvedených teoretických poznatků a aspektů pro správné modelování modelů tříd. Z diagramu je na první pohled zřejmé, že hlavním elementem je „Klient“, který interaguje s mnoha dalšími třídami. Celý koncept vymáhání je totiž přesunut na klienta, nikoliv na kontrakt, proto podstatně velké množství funkcionalit a tříd jsou provázány právě s klientem.

## 5.2 Popis tříd

Následující tabulka číslo 1 poslouží k lepšímu pochopení class diagramu. V tabulce je možné vidět jednotlivé třídy a jejich popis, který blíže specifikuje jejich využití v navrhovaném class diagramu.

<b>Třída</b>	<b>Popis</b>
Splatka	Třída Splatka představuje splátky klienta, které musí v určitý časový horizont zaplatit. Je možné zde nalézt i výše splátek
Kontrakt	Třída kontrakt určuje kontrakty klienta.
Pokuta	Třída pokuta udává množství a výši pokuty na konkrétním kontraktu klienta.
Adresa	Třída adresa představuje trvalé bydliště klienta.
Klient	Třída klient je osoba, která si bere půjčku v úvěrové společnosti.
SeznamKlientu	Třída seznamKlientu představuje skupinu klientů vygenerovaných v jedné dávce.
SeznamTelefonnichCisel	Třída seznamTelefonnichCisel představuje skupinu telefonních čísel vygenerovaných v jedné dávce.
Rizikovost	Třída rizikovost nastaví jednu z množných rizikovostí klienta.
Faze	Třída faze určuje do jaké fáze z časového hlediska klient patří.
SMS_Sablona	Třída SMS_Sablona určuje šablonu, která bude odeslána klientovi na jeho telefonní číslo.
Hovor	Třída hovor slouží k výběru konkrétního telefonního čísla klienta a k následnému spojení hovoru s dialerem přes JMS.

VysledekVymahani	Třída vysledekVymahani představuje výsledky hovoru a výsledek fyzického vymáhání.
FyzickeVymahani	Třída fyzickeVymahani zpracovává seznam klientů a přiřazuje jednotlivé klienty ke konkrétnímu vymahačovi a tyto informace jsou dále odeslány do vymahačské aplikace.
Report	Třída report slouží k vytvoření reportu, který může nabývat různých typů. Data pro report jsou poskytnuty z Výsledků vymáhání a ze seznamu klientů.

Tabulka 1 - Popis tříd, Zdroj: Autor

Každá třída obsahuje atributy, které představují „prvky“ dané třídy, tedy stanovují konkrétní vlastnosti tříd a jejich instancí. Kromě atributů, některé třídy obsahují i metody neboli funkce či chování, které daná třída, nebo konkrétněji její instance, budou vykonávat v průběhu jejich života. Každý atribut a metoda mají vlastní datový typ, kterým je definován jeho možný výstup (respektive vstup). Příkladem je možné uvést třídu „Pokuta“ s atributem „datumSplatnosti“, jehož datový typ je „Date“ a metoda „getID“ s datovým typem integer. Podle knihy *The rust programming language* od Steve Klabnik a Carol Nichols, by měl atribut „datumSplatnosti“ vrátit hodnotu, která může nabývat pouze formátu ve tvaru datumu (YY-MM-DD), zatímco metoda „getID“, by měla povolit zapsat pouze celočíselnou hodnotu [30].

### 5.3 Určení vztahů mezi třídami a návrhové vzory

Důležitým aspektem v modelování diagramu tříd je provázanost mezi třídami. Díky relacím mezi nimi je totiž možné určit jejich vztah, a to konkrétně, která třída je závislá na jiné a jak konkrétní třída může k jiné přistupovat.

- Určení vztahu „Osoba má adresu“.

Je specifický vztahem kompozice mezi třídami. Jde o případ vztahu 1:1, tedy multiplicita je pro obě třídy rovna přesně 1, ale stále splňuje definici kompozice. Vztah kompozice se vyskytuje v takových případech, kdy jedna třída (respektive instance), nemůže existovat bez té druhé. Jinak řečeno, pokud budeme mít třídu „Klient“ a „Adresa“, které je možné vidět na obrázku číslo 21, tak z toho vyplývá, že instance adresy může být vytvořena pouze a jen ve chvíli, kdy už existuje některá instance ze třídy Klient [22].

- Určení vztahu „Faktura má řádky faktury“

Tento vztah je velice podobný tomu předchozímu. Znovu jde o relaci typu kompozice, avšak je zde rozdíl v multiplicitě. Jde totiž o vztah 1:N. Příkladem z class diagramu mohou být třídy „Klient“ a „Kontrakt“. Takže opět třída Kontrakt z definice kompozice nemůže sama o sobě existovat bez Klienta, ale zde i navíc k jedné instanci třídy „Klient“, může existovat více instancí třídy „Kontrakt“. Čímž je zároveň i vysvětlena definice multiplicity neboli povolené množství vytvořených instancí z jedné třídy [22].

- Určení vztahu „Odkaz do seznamu“

Zde je použita relace asociace. V tomto případě je velice důležitá směrovost šipky, jejíž směr ukazuje, která ze tříd je seznam a která ze seznamu čerpá. Nejlepším příkladem pro pochopení je relace mezi třídami „VysledekVymahani“ a „Report“. Zde je možné vidět, že konkrétní report může obsahovat žádný nebo několik výsledků vymáhání, které je možné vybrat právě ze zmíněného seznamu (můžeme si představit například jako rolovací okýnko) [22].

- Určení vztahu „GSM telefon má SIM – kartu“

Jedná se o méně častou relaci, který je specifický vztahem agregace. Jedná se o takzvanou jednoduchou kompozici. Proti kompozici se liší v tom, že samy o sobě obě třídy (instance) mohou existovat, ale jedna z nich sama o sobě nebude fungovat, tedy její existence bez druhé je téměř nepoužitelná. Na obrázku číslo 21 a 22 (příloha číslo 5 – Class model) je toto určení vztahu použité mezi třídami „seznamTelefonnichCisel“ a „SMS\_Sablona“, kdy třída (instance) „SMS\_Sablona“, je „pasivní“ třída [22].

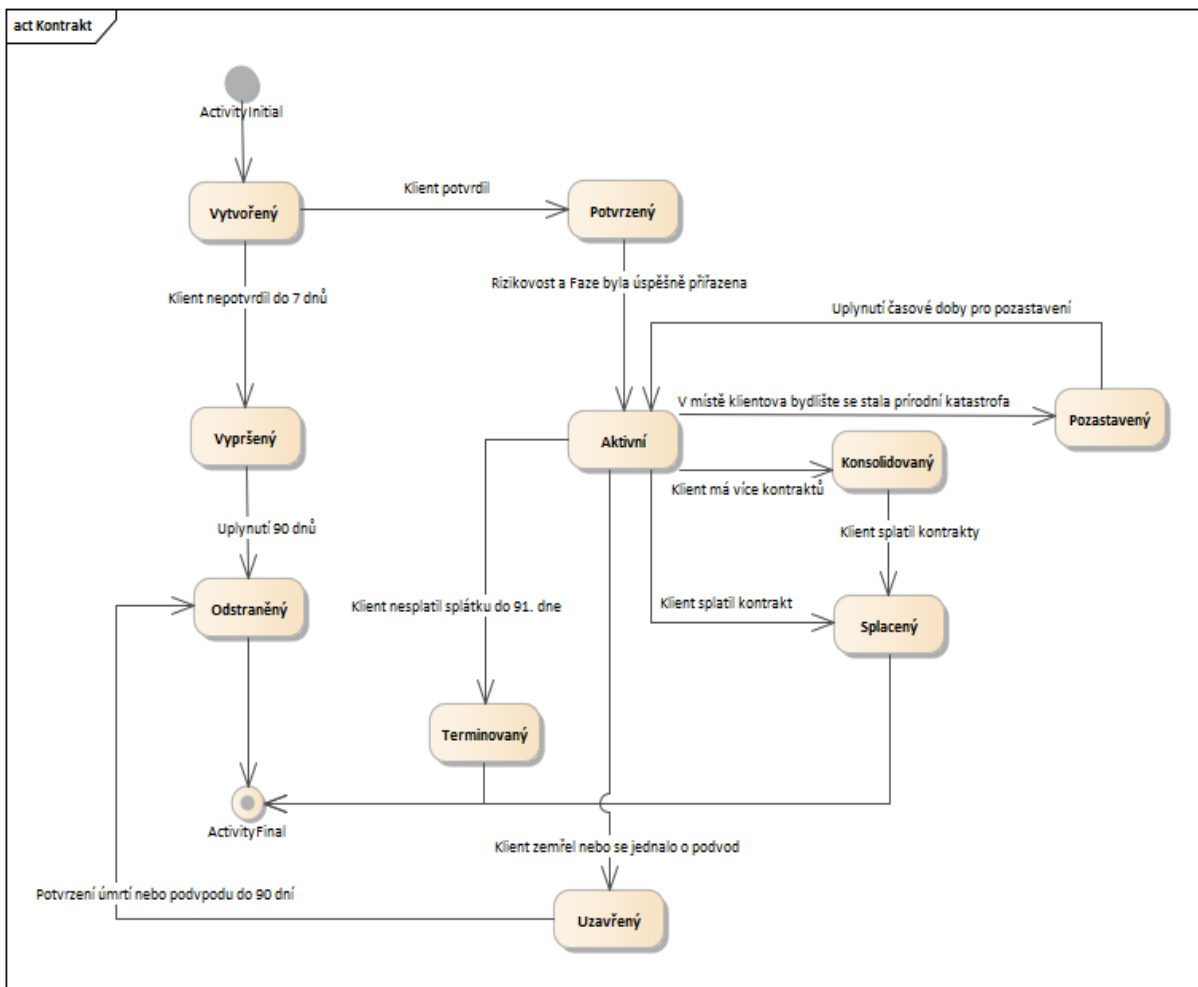
Pro správnou definici diagramu tříd jsou hlavním nástrojem návrhové vzory a s nimi důležitý pojem polymorfismus. Ten jednoduše slouží k využití jednotného rozhraní pro různé objekty. Příkladem může být metoda vyplneniSablony(): void u třídy SMS\_Sablona, kde samotná funkce vytvoří šablonu, avšak pro každou podtřídu třídy SMS\_Sablona, tedy SMS\_SablonaHigh, SMS\_SablonaMedium a SMS\_SablonaLow jiným způsobem. S předchozím příkladem třídy SMS\_Sablona je spjatý i další důležitý pojem, který je potřeba si vysvětlit, a to takzvaná abstraktní třída. Abstraktní třída představuje předlohu pro podtřídy v dědičnosti, jinak řečeno, ze třídy se nedají vytvořit žádné instance.

Z výše uvedených pojmů vyplývá návrhový vzor zvaný „Abstract factory“. Ten je založen na myšlence, že je možné vytvořit dvě různá rozhraní (GUI). Díky tomuto vzoru se stávají třídy a jejich objekty nezávislé na sobě. Poskytuje také rozhraní pro vytváření „rodin“ souvisejících nebo závislých objektů, bez určení jejich konkrétních tříd [31].

## 5.4 Stavový diagram

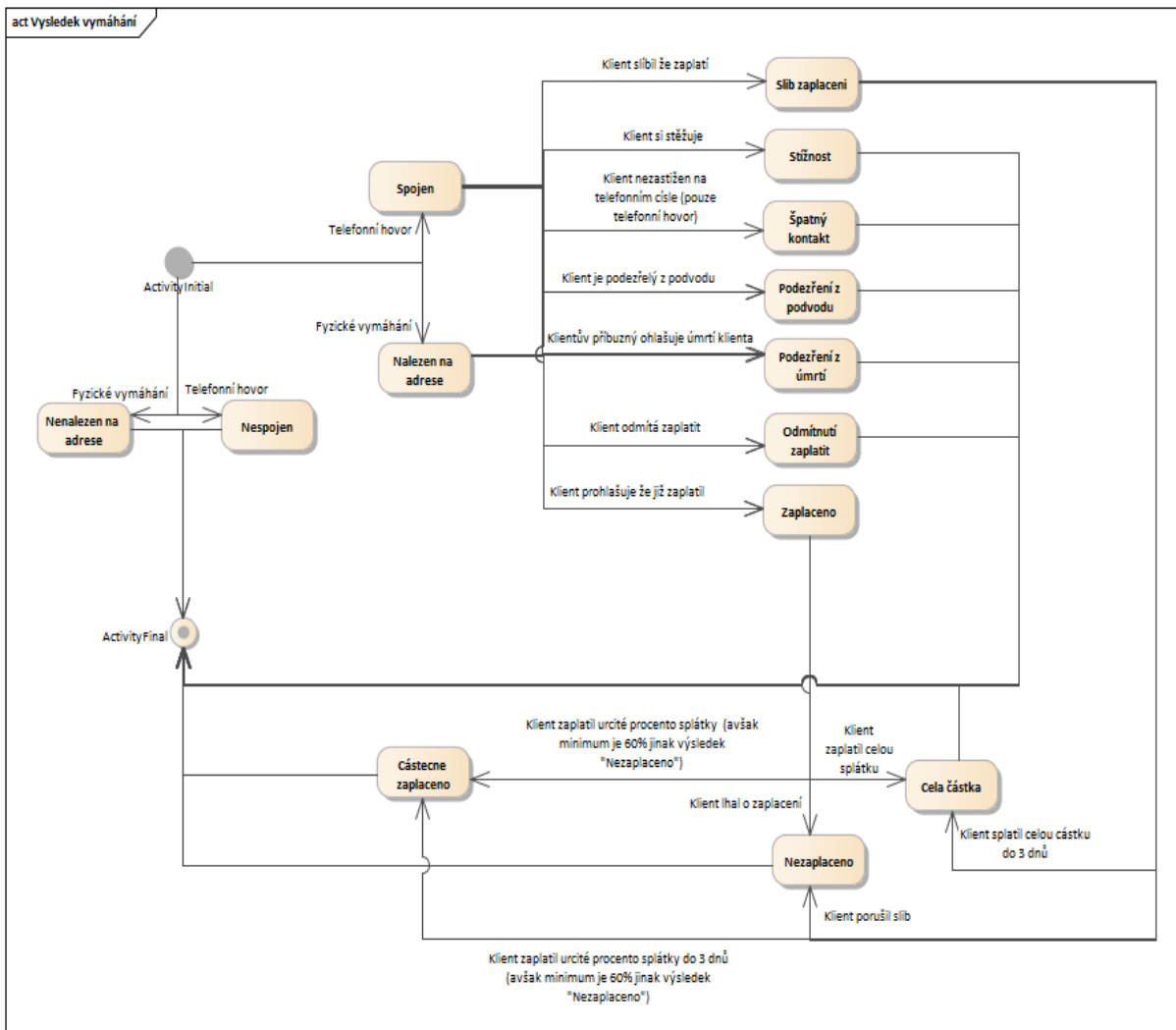
Stavový diagram je vzhledově velice podobný již použitému diagramu aktivit (kapitola 4 Model případů užití), avšak využití je znatelně jiné. Ve stavovém diagramu se totiž nezabýváme posloupností aktivit, ale definicí možných stavů a přechodů mezi nimi. Používá se pro specifikaci dynamiky objektů (tříd) či popis metod, z čehož vyplývá, že stavové diagramy vychází z modelu tříd (class diagramu). Stejně tak jako u diagramu aktivit, se skládá z počátečního a konečného bodu, dále také z přechodů s podmínkami a samotných stavů. Stavový diagram není v běžné praxi většinou využíván pro všechny navrhnuté objekty (třídy), ale pouze pro ty, u kterých má smysl tuto interakci modelovat. Jedná se tedy především o třídy, které často mění svůj stav za složitějších podmínek, které nejsou na první pohled na třídu úplně zřejmé. Příkladem může být třída „Kontrakt“, která je představena v předchozí podkapitole 5.1 Class diagram. Pomocí stavového diagramu ji však dokážeme ještě blíže specifikovat [32].

Na obrázku číslo 23 je možné vidět změny stavů třídy „Kontrakt“. Z příkladu je možné vidět, jak se v čase mění stav kontraktu klienta. V podstatě je v diagramu znázorněn celý životní cyklus kontraktu, respektive kontraktů (objektů třídy Kontrakt). Každý stav je definován sadou podmínek, které jsou v diagramu vyobrazeny u relací mezi stavy. Jak je možné vidět, kontrakt nabývá nejprve stavu „Vytvořený“. Tento stav (respektive iniciace celého stavového diagramu) nastává tehdy, když aktér „Bankovní poradce“ dokončí případ užití „Identifikace kontraktu“ (z kapitoly 4. Model případů užití). Tento kontrakt dále mění svůj stav, až nakonec dojde do jednoho z konečných stavů, který se liší na základě životního cyklu kontraktu a spuštěných případů užití.



Obrázek 23 - Stavový diagram: Kontrakt, Zdroj: Autor

Následující obrázek číslo 24 představuje další možnost využití stavového diagramu, pro detailnější popis třídy. V tomto případě se jedná o třídu „VysledekVymahani“, kterou je opět možné nalézt v předchozí kapitole. Třída zde může nabývat několika stavů (výsledků), které jsou iniciovány činnostmi klienta. Výsledky jsou možné získat ze dvou možných vymáhacích procesů, kterými jsou fyzické vymáhání a telefonní hovor. Konečné stavy opět vycházejí ze spuštěných případů užití.



Obrázek 24 - Stavový diagram: Výsledek vymáhání, Zdroj: Autor



## 5.5 Sekvenční diagram

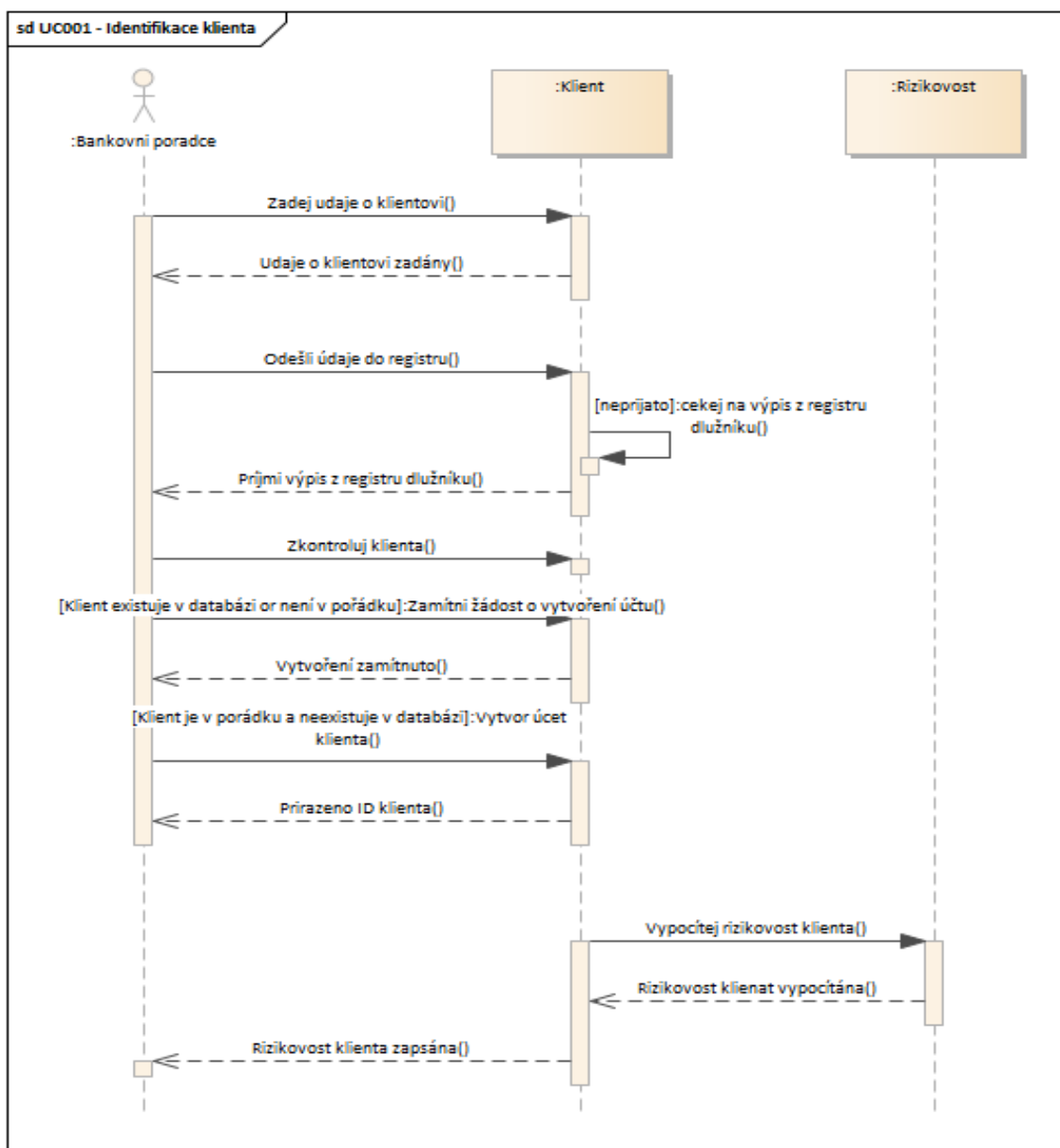
Pro správné definování vzájemných interakcí mezi objekty (třídami) je vybrán sekvenční diagram. Díky sekvenčnímu diagramu je možné určit, jaké operace jsou mezi objekty zasílány z pohledu času. Definice sekvenčního diagramu vychází z class diagramu a modelu případů užití, konkrétněji vnitřní struktury případů užití (scénářů), představené v podkapitole 4.1.3 Vnitřní struktura případů užití. Pokud předchozí kroky modelování byly detailní a správné, poté je také možné namodelovat úplný a plnohodnotný sekvenční diagram.

Struktura sekvenčních diagramů se skládá z aktéra, který interaguje s objekty poskládanými do sloupců. Interakce mezi elementy jsou iniciovány pomocí zasílaných zpráv mezi nimi. Tyto zprávy často představují strukturu jednotlivých tříd (objektů), tedy procedury či atributy. Z obrázku číslo 25 je možné vidět interakce mezi aktérem „Bankovní poradce“ a třídami „Klient“ a „Rizikovost“. Hluběji diagram představuje proces vytvoření klienta v čase. Většinou se v modelech objevují tzv. Synchronní zprávy, které jsou specifické tím, že při každém odeslání zprávy čekají na odezvu. V případě identifikace klienta to může být situace, kdy bankovní poradce posílá zprávu o zadání údajů o klientovi a přijímá zpět odezvu, že údaje o klientovi byly zadány. Tyto údaje jsou dále odeslány do registru dlužníků, který je definován v kapitole 4.1.2 Aktéři, jako externí systém. Tento proces musí být proveden před dalšími úkony, aby se zjistilo, zda klientovi úvěrová společnost vůbec může půjčit peníze.

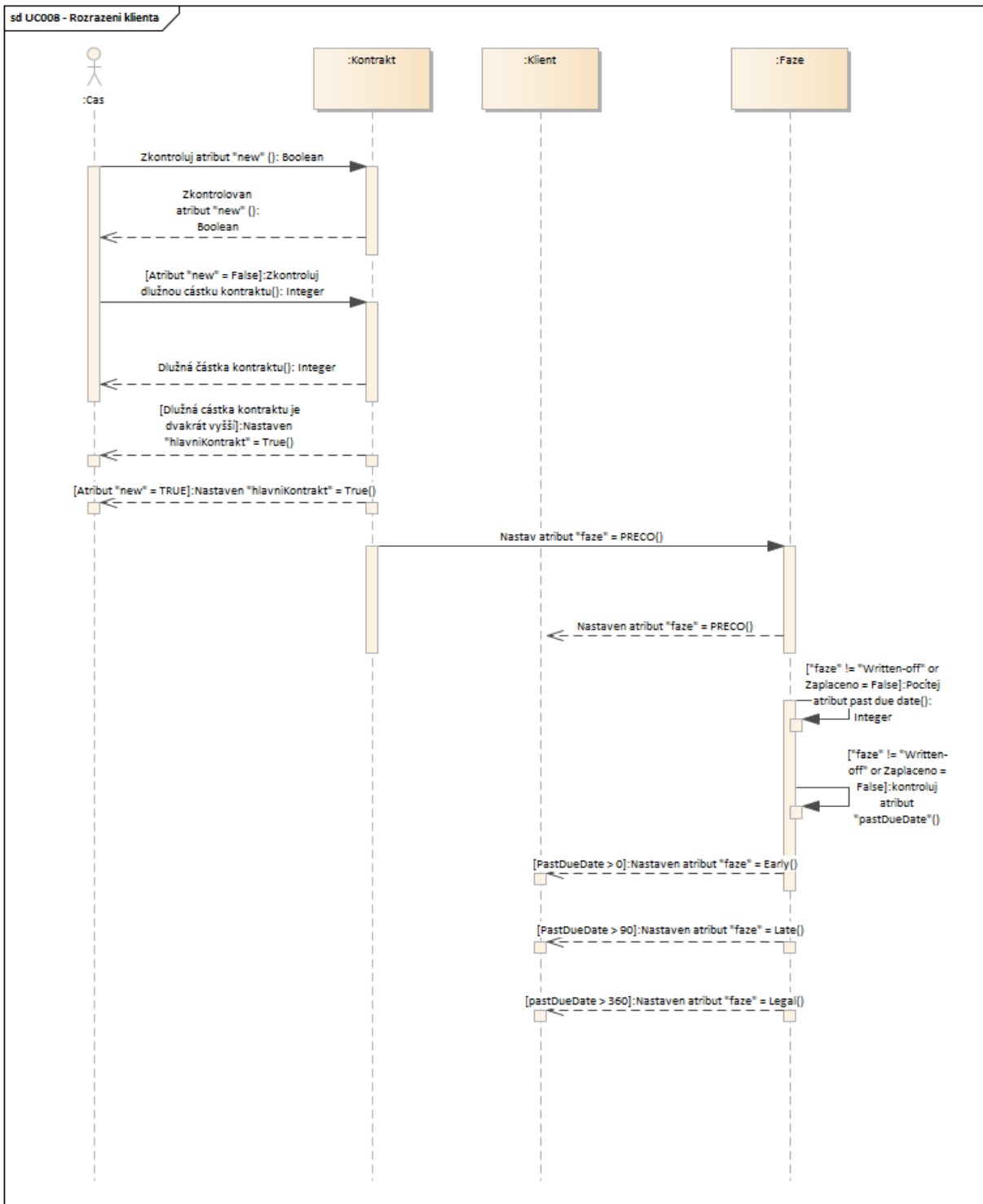
Další možnou zprávou v sekvenčních diagramech je tzv. Asynchronní zpráva. Bližší představení asynchronní zprávy je možné vidět na obrázku číslo 27, kde aktér čas iniciuje zaregistrování telefonního čísla klienta, avšak nedostává ihned zpětnou vazbu, ale pokračuje v odesílání zprávy k dalšímu objektu (třídě) „SeznamTelefonnichCisel“, který má za úkol vyhledat poptávané telefonní číslo. Celý diagram představuje proces odeslání SMS zprávy klientovi. V diagramu je spousta zpráv iniciováno na základě podmínek, které je možné vidět například ve zpětné zprávě odesílané od třídy „Klient“ k aktérovi čas následujícím způsobem: „[Atribut faze = PRECO] Poskytnut atribut „precoRizikovostKlienta““. Z příkladu vyplývá, že

první část v hranaté závorce je podmínka, při které zpráva o poskytnutí atributu bude odeslána zpět jako výstup. V sekvenčním diagramu je dále třída „SMS\_Sablona“ a její potomci, kteří hrají hlavní roli při interakcích. Před samotným určením, která šablona bude pro daného klienta vybrána a odeslána na telefonní číslo, je zjištění fáze a rizikovost klienta. Tato posloupnost vychází z BPMN modelu, který je představený v kapitole 2.2 BPMN model, konkrétně na obrázku číslo 3, kde vidíme proces a podmínky pro výběr a odeslání SMS šablony klientovi.

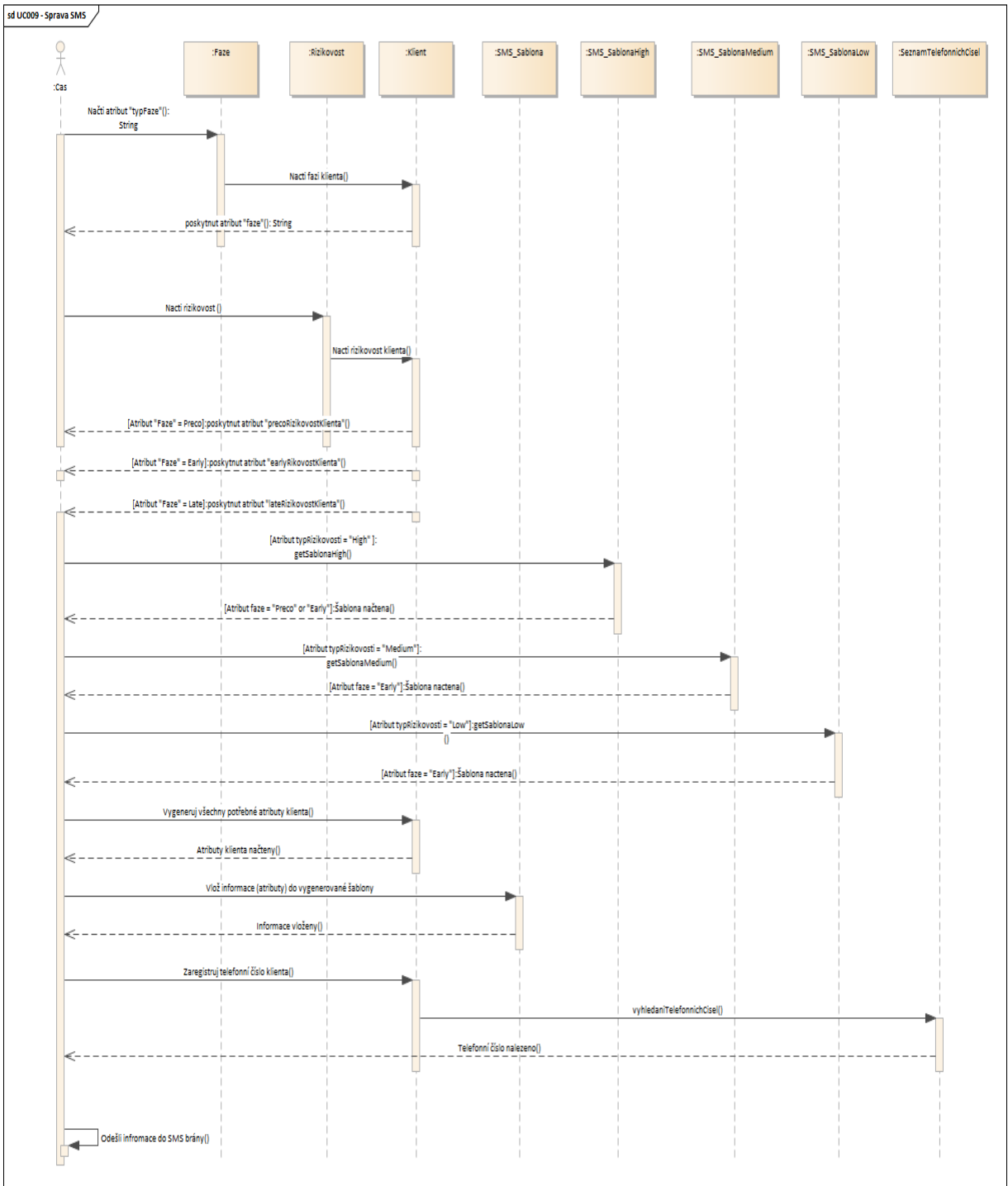
Diagram na obrázku číslo 26 určuje proces rozřazení klienta do určité fáze. Kromě podmínek, je v tomto diagramu také možné vidět cyklus, který bude prováděn do té doby, dokud nesplní (respektive neporuší) určitou stanovenou podmínku. V případě rozřazení klienta se jedná o počítání atributu `pastDueDate`, který je jeden z nejdůležitějších atributů při rozřazování klienta, protože, jak již bylo dříve řečeno, tento atribut počítá dny nesplacení splátky u klienta. Další cyklus v diagramu kontroluje hodnotu `pastDueDate` na základě které, může nabývat jednoho z několika možných výstupů. Tyto dva cykly jsou definovány před samotným určením fáze klienta, protože, jak vychází z class diagramu, třída „Faze“ poskytuje napočítanou fázi třídy „Klient“ [33].



Obrázek 25 - Sekvenční diagram: Identifikace klienta, Zdroj: Autor



Obrázek 26 - Sekvenční diagram: Rozřazení klienta, Zdroj: Autor



Obrázek 27 - Sekvenční diagram: Správa SMS, Zdroj: Autor

## 6 SHRNUÍ VÝSLEDKŮ

V kapitole 1.1 definice problému je stanoven cíl a problémy vymáhacích systémů, který tato diplomová práce má za úkol vyřešit. Práce je vedena v duchu návrhu informačního systému, který je jednoduchý a sníží náklady na vymáhací procesy. Navrhnutý systém se zaměřuje na návrh systému pro vymáhání na klientské úrovni, čímž dokáže snížit náklady na vymáhací procesy (díky omezení množství kontaktování jedné osoby během dne), oproti informačním systémům, které jsou orientovány na vymáhání na úrovni kontraktů. Díky tomuto návrhu je nejen snížena chybovost v systému, především při změnách procesů závislých na vymáhacích strategiích, ale také je zajištěna jednoduchost a uživatelská přívětivost. Zmíněné vylepšení lze potvrdit článkem od firmy Deloitte, který hovoří právě o problematice procesů dnešních informačních systémů pro vymáhání. Podotýká, že nejdůležitějšími aspekty pro snížení nákladů a vytvoření jednoduchosti informačního systému pro vymáhání je především flexibilita systému, rychlost a integrace systému s dalšími externími aplikacemi, čímž se právě tato diplomová práce zabývá. Článek také dále upozorňuje na problémy s několikanásobným kontaktováním klienta během jednoho dne, což je považováno za špatnou praxi a firmy se poté potýkají se zvýšenými náklady a možnými stížnostmi od klientů [34].

Práce je koncipována tak, že jsou zde implementovány všechny důležité náležitosti pro tvorbu správného a kvalitního informačního systému, počínaje stávající analýzou podnikových procesů na základě kterých, je možné navrhnout chování a strukturu systému. Analýza podnikových procesů je vytvořena pomocí BPMN diagramu a slouží jako hlavní podklad pro definici chování systému. Model případů užití, který reflektuje chování systému je mimo jiné také navrhnutý pomocí systémových požadavků, které vychází z business analýzy. Dále jsou detailně rozebrány dílčí funkcionality pomocí diagramu případů užití a diagramu aktivit, které jsou hlavním podkladem pro tvorbu struktury informačního systému. Díky diagramu tříd je návrh struktury připravený jako předloha pro detailnější návrh a vývoj. Poslední část práce je zaměřena na definování interakcí uvnitř systému pomocí sekvenčních diagramů, díky kterým je určen životní cyklus klienta a kontraktu v čase.

Dynamika systému na základě stavových diagramů zajišťuje možné stavy vybraných tříd, které lze v systému nabývat. Zde je definována funkčnost v čase a odráží návrhy chování a struktury systému dohromady. Technický návrh informačního systému je zhodnocen v následující podkapitole.

## 6.1 Technické zhodnocení

Vzhledem k povaze práce, je možné v tuto chvíli popsat pouze tři části metody FURPS, která je představena v kapitole 1.3.3 Metoda FURSP, a to Funkčnost, Použitelnost a Rozšiřitelnost. Ostatní dva body mohou být vyhodnoceny až po implementaci informačního systému.

### **F – funkčnost**

Správná funkčnost systému je stanovena především funkčními požadavky na systém. Pro bližší analýzu, zda kvalita funkčnosti systému byla naplněna, je možné vidět z podkapitoly 3.1 Funkční požadavky a podkapitoly 4.1.1 Interakce s funkčními požadavky. Právě nalezení případů užití systému pro každý definovaný funkční požadavek je odrazem toho, že základní funkčnosti systému jsou vyjasněny. Potvrzení toho, že byla funkčnost a požadavky splněny, je možné až ve fázi testování, která není součástí této práce.

### **U - Použitelnost**

Z pohledu dokumentace, je možné hodnotit, že tato práce je vedena úplně a jsou poskytnuty všechny potřebné informace a data pro další zpracování návrhu a poté vývoje informačního systému spolu s uživatelským rozhraním.

## **S - Rozšiřitelnost**

Jak již bylo zmíněno, pro informační systémy je velice důležitá flexibilita systému. Systém je navrhován tak, aby vymáhací procesy mohly být jednoduše změněny v rámci informačního systému, při jakékoliv změně vymáhacích strategií. Další potvrzení tohoto bodu je fakt, že systém je navrhnutý jako Core systém, který interaguje s ostatními externími aplikacemi a databázemi pomocí zmíněných API a Webové služby, proto není problém integrovat další, pro vylepšení dosavadních procesů. Tyto požadavky však budou záviset především na kvalitě a velikosti cloudu.



## ZÁVĚR

Hlavním cílem práce je návrh informačního systému pro vymáhání dluhů pomocí jazyka UML. Návrh a struktura systému jsou prováděny na základě stanovených cílů, které jsou definovány pomocí nalezených problémů vycházejících z nestrukturovaného rozhovoru a rešerší. Všechny části práce odráží životní cyklus informačního systému definovaného podle odborné literatury. Jednotlivé kapitoly se zaměřují na detailní vytvoření dokumentace pro další analýzu a vývoj softwaru. Veškeré metody a techniky, které jsou v práci aplikovány, jsou podloženy odbornou literaturou a články. Navržený systém konkrétně zajišťuje snížení nákladů na vymáhací procesy a automatizaci všech vymáhacích procesů. Zmíněné body jsou hlavním přínosem diplomové práce.

Navrhnuté modely by měly dále posloužit IT analytikům, kteří detailně prohloubí jednotlivé části na základě kterých, bude informační systém vyvíjen vývojáři a řádně otestován před samotnou implementací. Kromě dalšího rozšíření může práce pomoci novým analytikům nebo osobám pracujících v IT, kteří se doposud s danou problematikou neseťkali, ale chtěli by získat znalost a povědomí o této oblasti.

## SEZNAM ZKRATEK

<b>Zkratka</b>	<b>Celý název</b>
API	Rozhraní pro programování aplikace
GDPR	Obecné nařízení o ochraně osobních údajů
GUI	Grafické uživatelské rozhraní
ID	Identifikace
IS	Informační systém
IT	Informační technologie
JAR	Java archive
JMS	Java message service
OOP	Objektově orientované programování
SMS	Krátká textová zpráva
SOAP	Simple object access protocol
SQL	Standardizovaný strukturovaný dotazovací jazyk
UML	Unifikovaný modelovací jazyk

## SEZNAM OBRÁZKŮ

Obrázek 1 – Dynamický model, Zdroj: [7] .....	17
Obrázek 2 - BPMN model 1. část, Zdroj: Autor .....	26
Obrázek 3 – BPMN model 2. část, Zdroj: Autor .....	28
Obrázek 4 – BPMN model 2. část, Zdroj: Autor .....	28
Obrázek 5- BPMN model 3. část, Zdroj: Autor .....	29
Obrázek 6 - Funkční požadavky, Zdroj: Autor.....	32
Obrázek 7 - Přehled funkčních požadavků 1. část, Zdroj: Autor .....	33
Obrázek 8 - Přehled funkčních požadavků 2. část, Zdroj: Autor .....	34
Obrázek 9 - Nefunkční požadavky, Zdroj: Autor.....	35
Obrázek 10 - Přehled nefunkčních požadavků 1. část, Zdroj: Autor .....	36
Obrázek 11- Přehled nefunkčních požadavků 2. část, Zdroj: Autor .....	37
Obrázek 12 - Případy užití, Zdroj: Autor .....	39
Obrázek 13 - Interakce případů užití s funkčními požadavky, Zdroj: Autor .....	41
Obrázek 14 - Aktéři, Zdroj: Autor.....	42

Obrázek 15 - Aktéři (externí entity), Zdroj: Autor .....	43
Obrázek 16 – Diagram aktivit: Identifikace kontraktu, Zdroj: Autor.....	46
<i>Obrázek 17 – Diagram aktivit: Rozřazení klienta, Zdroj: Autor.....</i>	<i>47</i>
<i>Obrázek 18 - Diagram aktivit: Skórování klienta, Zdroj: Autor .....</i>	<i>48</i>
Obrázek 19 - Diagram případů užití, Zdroj: Autor.....	51
Obrázek 20 - Objektové paradigma, Zdroj: [22] .....	55
Obrázek 21 - Class diagram 1. část, Zdroj: Autor .....	56
Obrázek 22 - Class diagram 2. část, Zdroj: Autor .....	57
Obrázek 23 - Stavový diagram: Kontrakt, Zdroj: Autor .....	63
Obrázek 24 - Stavový diagram: Výsledek vymáhání, Zdroj: Autor.....	64
Obrázek 25 - Sekvenční diagram: Identifikace klienta, Zdroj: Autor .....	67
<i>Obrázek 26 - Sekvenční diagram: Rozřazení klienta, Zdroj: Autor .....</i>	<i>68</i>
Obrázek 27 - Sekvenční diagram: Správa SMS, Zdroj: Autor .....	69

## SEZNAM TABULEK

<i>Tabulka 1 - Popis tříd, Zdroj: Autor</i> .....	59
---	----

## SEZNAM POUŽITÉ LITERATURY

1. a.s., H.C.I.; Dostupné z: <http://www.homecredit.net/about-us/loans-tool.aspx>.
2. Techopedia. Enterprise Architect (EA). Dostupné z: <https://www.techopedia.com/definition/28052/enterprise-architect-ea>.
3. Jim Arlow, I.N., UML 2 a unifikovaný proces vývoje aplikací. 2014, Brno: Computer Press. 568.
4. Fowler, M., Destilované UML. 2009, Praha: Grada.
5. Rodríguez, J.P.M.D.M.G., A REVIEW OF SOFTWARE QUALITY MODELS FOR THE EVALUATION OF SOFTWARE PRODUCTS. International Journal of Software Engineering & Applications (IJSEA), 2014. 5(6).
6. Přehled metodiky vývoje GIS aplikací. Dostupné z: [https://is.muni.cz/el/1433/podzim2015/PV047/um/N\\_Metodika\\_vyvoje\\_GIS\\_aplikaci\\_lekce\\_2\\_Model\\_pozadavku.pdf](https://is.muni.cz/el/1433/podzim2015/PV047/um/N_Metodika_vyvoje_GIS_aplikaci_lekce_2_Model_pozadavku.pdf).
7. Antoine, D. and N.A. Rabie, MODELLING OF THE DEBTS COLLECTION PROCESS FOR SERVICE COMPANIES. IFAC Proceedings Volumes, 2006. 39(3): p. 349-354.
8. Harrington, L., "Helping you to pay us": Rapport management in debt collection call centre encounters, in Journal of Politeness Research 2018. p. 193.
9. Wijewardhana, U.A., A Mathematical Model for Predicting Debt Repayment: A Technical Note. Australasian Accounting, Business and Finance Journal, 2018. 12(3): p. 107-115.

10. Onar, S.C., B. Oztaysi, and C. Kahraman, A Fuzzy Rule Based Inference System for Early Debt Collection. *Technological and Economic Development of Economy*, 2018. **24**(5): p. 1845-1865.
11. SOUČEK, J., Řízení a vymáhání pohledávek - příklady z praxe, in *Hradecké ekonomické dny 2011*. 2011, Univerzita Hradec Králové: Hradec Králové.
12. Šmíd, V. *Management informačního systému*. 2005; Dostupné z: <https://www.fi.muni.cz/~smid/managis.html>.
13. Saberi, S. and S. Mohsen, An Incident Management System for Debt Collection in Virtual Banking. *International Journal of Advanced Computer Science and Applications*, 2016. 7.
14. Geary A. Rummler, A.P.B., *Improving Performance: How to Manage the White Space on the Organization Chart*, 3rd Edition. 2012, USA: San Francisco: Jossey-Bass.
15. Nick Russell, W.M.P.v.d.A., Arthur H. M. ter Hofstede, *Workflow Patterns: The Definitive Guide*. 2016: Information Systems.
16. slovníček, B. Procesy. Dostupné z: <http://bpm-slovník.blogspot.com/2007/09/proces.html>.
17. Weilkiens, T., et al., Preface. *OCEB 2 Certification Guide (Second Edition)*, ed. T. Weilkiens, et al. 2016: Morgan Kaufmann. xiii.
18. Herchi, H. and W. Ben Abdesslem, From user requirements to UML class diagram. 2012.

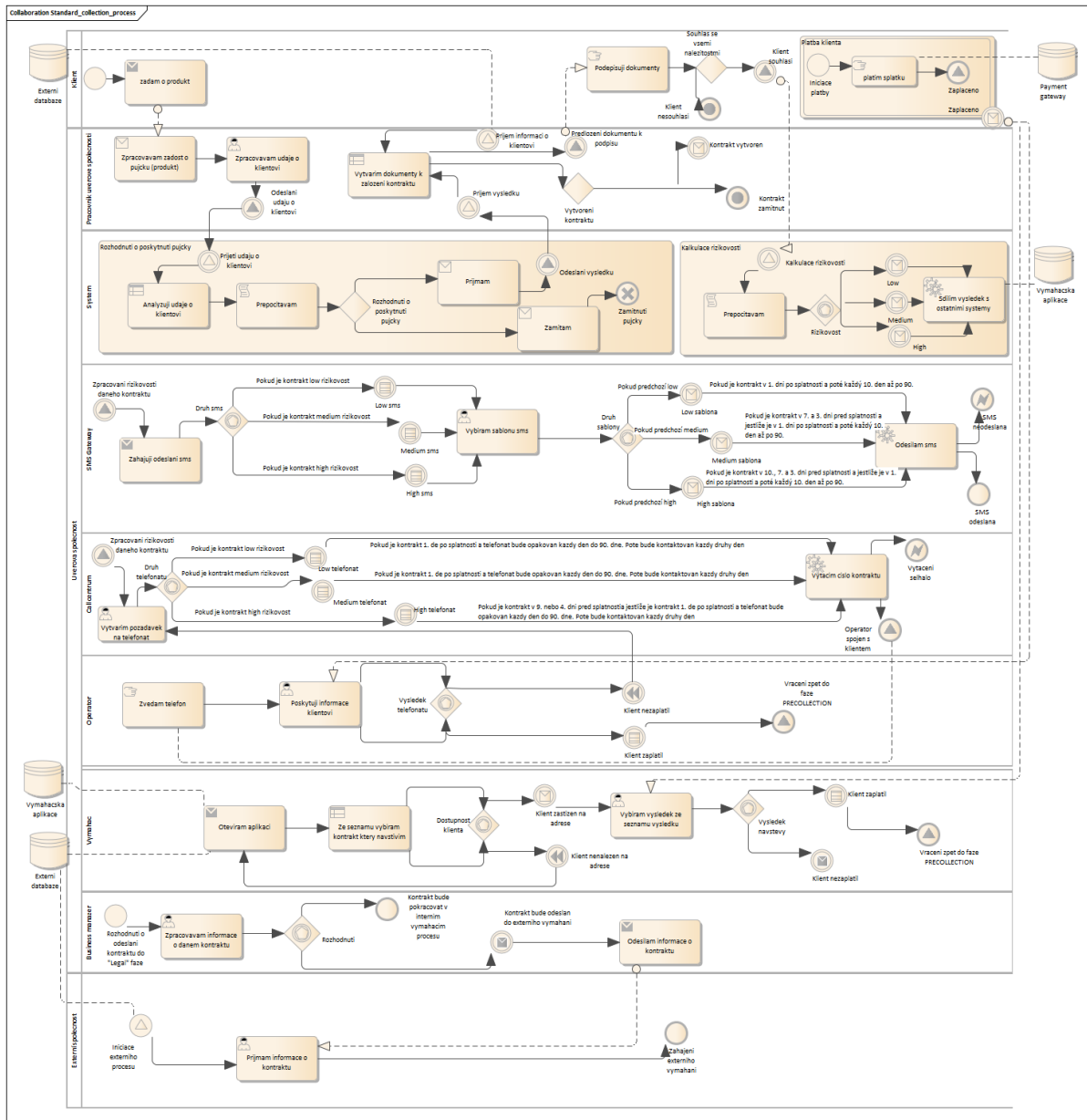
19. Malavolta, I. Requirements engineering with UML. 2016; Dostupné z: <https://www.slideshare.net/iivanoo/requirements-engineering-with-uml-software-modeling-computer-science-vrije-universiteit-amsterdam-20162017>.
20. Tomášková, H., Objektové modelování I - soubor přednášek: Univerzita Hradec Králové.
21. Siau, K. and L. Lee, Are use case and class diagrams complementary in requirements analysis? An experimental study on use case and class diagrams in UML. Requirements Engineering, 2004. 9(4): p. 229-237.
22. Kraval, I., Analytické modelování informačních systémů pomocí UML v praxi. 2010, Brno: Object Consulting.
23. Kurt Bittner, I.S., Use Case Modeling. 2003, Boston: Addison Wesley.
24. Moran, R.J.C. The <<include>> and <<extend>> Relationships in Use Case Models.
25. *System Integration Standards*. Dostupné z: [https://brocku.ca/policies/wp-content/uploads/sites/94/FINAL\\_System-Integration-Standards-2019Mar13.pdf](https://brocku.ca/policies/wp-content/uploads/sites/94/FINAL_System-Integration-Standards-2019Mar13.pdf).
26. Staff, R. *API vs Web Service: What is the Difference?* ; Dostupné z: [https://rapidapi.com/blog/api-vs-web-service/?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=DSA&utm\\_content=1t1&gclid=CjwKCAiAh5\\_uBRA5EiwASW3IaulQs0AvG6YIcsc1KpMaEt2zIvfy0X402PmL4rf8-eE2I8S9xgBHXBoCP-sQAvD\\_BwE](https://rapidapi.com/blog/api-vs-web-service/?utm_source=google&utm_medium=cpc&utm_campaign=DSA&utm_content=1t1&gclid=CjwKCAiAh5_uBRA5EiwASW3IaulQs0AvG6YIcsc1KpMaEt2zIvfy0X402PmL4rf8-eE2I8S9xgBHXBoCP-sQAvD_BwE).
27. Kumar, S. *9 Best API Management Platforms*.
28. Martina Seidl, M.S., Christian Huemer, Gerti Kappel, UML @ Classroom: An Introduction to Object-Oriented Modeling. 2015, Switzerland: Springer, Cham.



29. Starr, L. How to Build Articulate Class Models and get Real Benefits from UML.
30. Steve Klabnik, C.N., The Rust Programming Language. 2018, San Francisco: No Starch Press.
31. Kraval, I., Design Patterns v OOP se zaměřením na Javu, C# a Delphi, 2002.
32. Suchan, J. Stavové diagramy - tutoriál. 2007.
33. Vondrák, I., Úvod do softwarového inženýství, 2002: Ostrava.
34. Hales Damien, S.J., Wines Sarah, Demuth Eleanor, Singh Sudesh *A fresh perspective: Collections strategies for the digital age.* 2016.

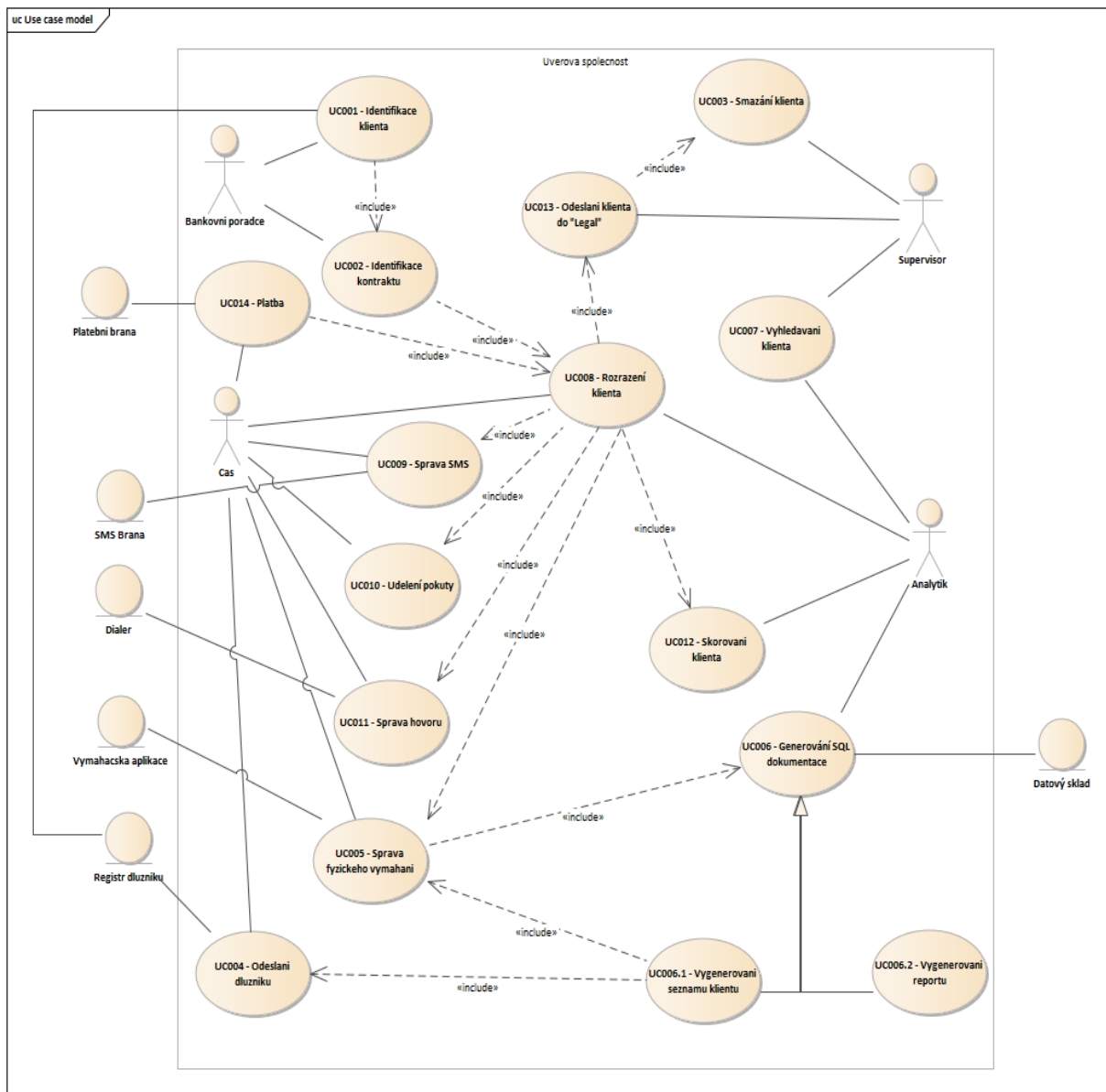
# PŘÍLOHY

## Příloha číslo 1 – BPMN model



Příloha číslo 1 - BPMN model

## Příloha číslo 2 – Use case model



Příloha číslo 2 - Use case model

## Příloha číslo 3 – Přehled funkčních požadavků

custom Přehled_funkcnich_požadavku		
<b>Vyhledání klienta</b> tags ID = 1 Popis = Vyhledání klienta a zobrazení všech jeho atributů Priorita = Nezbytný	<b>Filtrování</b> tags ID = 6 Popis = Systém automaticky zaradí klienta do konkrétní fáze vymáhácho procesu Priorita = Nezbytný	<b>Správa výsledku vymáhání</b> tags ID = 11 Popis = Výsledek fyzického vymáhání se zaznamená a archivuje Priorita = Nezbytný
<b>Odstranění klienta</b> tags ID = 2 Popis = Odstranění veškerých dat o klientovi Priorita = Nezbytný	<b>Pokuty</b> tags ID = 7 Popis = Klientovi bude naúčtována pokuta z prodlení Priorita = Nezbytný	<b>SQL Report</b> tags ID = 12 Popis = Systém přepočítá na základe SQL scriptu vytvoří report v různých formátech Priorita = Nezbytný
<b>Vytvoření klienta</b> tags ID = 3 Popis = Systém vytvoří klienta na základe poskytnutých informací Priorita = Nezbytný	<b>Druh produktu</b> tags ID = 8 Popis = Systém přiřadí klientovi druh produktu Priorita = Nezbytný	<b>Vygenerování seznamu klientu</b> tags ID = 13 Popis = Systém na základe filtrace nebo SQL scriptu vytvoří seznam klientu Priorita = Nezbytný
<b>Správa kontraktu</b> tags ID = 4 Popis = Systém přiřadí kontrakt správnému klientovi Priorita = Nezbytný	<b>Vytočení hovoru</b> tags ID = 9 Popis = Systém automaticky vytočí jedno z klientových čísel Priorita = Nezbytný	<b>SQL Scoring</b> tags ID = 14 Popis = Systém přepočítá na základe SQL scriptu rizikovost klienta Priorita = Nezbytný
<b>Fáze vymáhání</b> tags ID = 5 Popis = Systém automaticky přiřadí klienta do určité fáze vymáhání Priorita = Nezbytný	<b>Správa výsledku hovoru</b> tags ID = 10 Popis = Výsledek hovoru se zaznamená a archivuje Priorita = Nezbytný	<b>Registr dlužníku</b> tags ID = 15 Popis = Systém pravidelně odesílá seznamy dlužníků Priorita = Nezbytný
	<b>Správa plateb</b> tags ID = 17 Popis = Systém zpracuje platby Priorita = Nezbytný	<b>SMS šablona</b> tags ID = 16 Popis = Systém automaticky zpracuje konkrétní šablonu pro klienta a poskytne jí SMS bránu Priorita = Nezbytný

Příloha číslo 3 - Přehled funkčních požadavků

## Příloha číslo 4 – Přehled nefunkčních požadavků

custom Přehled_nefunkcnich_požadavku		
<b>GDPR</b> tags ID = 1 Popis = Systém se bude řídit zákony GDPR Priorita = Nezbytný	<b>Dostupnost</b> tags ID = 5 Popis = Systém bude dostupný v nastavenou pracovní dobu Priorita = Nezbytný	<b>Množství uživatelů</b> tags ID = 9 Popis = Systém bude schopný definovat uživatelské role Priorita = Nezbytný
<b>Jazyk</b> tags ID = 2 Popis = Systém bude naprogramován v jazyce Java s využitím nejnovějších Frameworků Priorita = Nezbytný	<b>Uložiště</b> tags ID = 6 Popis = Systém musí být schopný pracovat se 100 miliony klientů Priorita = Nezbytný	<b>Kompatibilita s SMS branou</b> tags ID = 10 Popis = Systém bude umět odesílat/přijímat informace do/z SMS brány Priorita = Nezbytný
<b>Verze systému</b> tags ID = 3 Popis = Systém bude fungovat na konkrétní verzi Priorita = Nezbytný	<b>Archivace</b> tags ID = 7 Popis = Systém bude schopný archivovat veškeré informace o provedených aktivitách na klientovi Priorita = Nezbytný	<b>Kompatibilita s platební branou</b> tags ID = 11 Popis = Systém bude umět přijímat informaci o zaplacení Priorita = Nezbytný
<b>Odezva systému</b> tags ID = 4 Popis = Systém bude umět zpracovat 10 tisíc klientů a 20 tisíc výsledků na jednu Priorita = Nezbytný	<b>Provoz na Windows</b> tags ID = 8 Popis = Systém bude naprogramován pro využití na operačním systému Windows Priorita = Nezbytný	<b>Kompatibilita s vymáhacskou aplikací</b> tags ID = 12 Popis = Systém bude umět komunikovat s vymáhacskou aplikací Priorita = Nezbytný
		<b>Databazová podpora</b> tags ID = 13 Popis = Systém bude předávat/získávat data do/z Databázového skladu Priorita = Nezbytný

Příloha číslo 4 - Přehled nefunkčních požadavků



## Příloha číslo 6 – Podklad pro zadání diplomové práce

Univerzita Hradec Králové  
Fakulta informatiky a managementu  
Akademický rok: 2019/2020

Studijní program: Systémové inženýrství a informatika  
Forma: Kombinovaná  
Obor/komb.: Informační management (im2-k)

### Podklad pro zadání DIPLOMOVÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Bc. Ustohal Filip	Mezírka 8, Brno - Brno-město	I1800714

#### TÉMA ČESKY:

Návrh informačního systému pro vymáhání dluhů

#### TÉMA ANGLICKY:

Design of information system for debt collection

#### VEDOUcí PRÁCE:

Ing. Pavel Čech, Ph.D. - KIT

#### ZÁSADY PRO VYPRACOVÁNÍ:

Úvod, Anotace, Cíl práce,  
Rešerše, Interview, Business popis, BPMN, Požadavky,  
Use case diagram, Class diagram,  
Diskuze, Závěr a shrnutí výsledků,  
Literatura, Obrázky

#### SEZNAM DOPORUČENÉ LITERATURY:

1. Analytické modelování informačních systémů pomocí UML v praxi, IJla Kraval
2. UML 2 a unifikovaný proces vývoje aplikací, Jim Arlow, Ila Neustadt
3. Beginning Databases with MySQL, Richard Stones, Neil Matthew
4. BPMN 2.0, Books on Demand
5. Příklady modelů analýzy a návrhu aplikace v UML, vysoká škola ekonomická v Praze

Podpis studenta:



Datum:

23.07.2019

Podpis vedoucího práce:



Datum:

2.9.2019