

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Diplomová práce

Návrh informačního systému v UML

Bc. Kamil Kováč

© 2018 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Kamil Kováč

Podnikání a administrativa

Název práce

Návrh informačního systému v UML

Název anglicky

Design of Information System in UML

Cíle práce

Cílem práce je návrh informačního systému pomocí jazyka UML. Informační systém je navržen pro bankovní sektor k prodeji finančních produktů a služeb tak, aby respektoval specifické nároky elektronického kontraktingu. Informační systém umožňuje uživatelům v rámci samoobslužného kanálu požádat o vybrané produkty banky. Systém provede uživatele prodejním procesem, ověří rizikovost žadatele a zajistí zprostředkování produktu včetně online identifikace a autorizace uživatele.

Metodika

Metodika práce spočívá ve studiu odborných literárních zdrojů, které se zabývají tématy vývoje informačních systémů, metodikami vývoje software, jazykem UML a bankovními systémy. Součástí diplomové práce je analýza současného stavu informačních systémů finanční instituce, sběr požadavků, jejich rozbor a syntéza funkčních požadavků.

S využitím takto získaných poznatků je proveden návrh informačního systému v souladu s definovanými požadavky.

Doporučený rozsah práce

60 – 80

Klíčová slova

UML, XML, XSD, wireframe, objektově orientovaný přístup, informační systém, funkční analýza, technický design, webová služba

Doporučené zdroje informací

BLAHA, M. – RUMBAUGH, J. *Object-oriented modeling and design with UML*. Upper Saddle River, NJ: Pearson Education, 2005. ISBN 0130159204.

FOWLER, M. *UML distilled : a brief guide to the standard object modeling language*. Boston: Addison-Wesley, 2004. ISBN 0321193687.

KANISOVÁ, Hana a Miroslav MÜLLER. *UML srozumitelně. 2., aktualiz. vyd.* Brno: Computer Press, 2006. ISBN 80-251-1083-4.

NEUSTADT, I. – ARLOW, J. *UML 2 a unifikovaný proces vývoje aplikací : objektově orientovaná analýza a návrh prakticky*. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.

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

SOMMERVILLE, Ian. *Softwarové inženýrství*. Brno: Computer Press, 2013. ISBN 978-802-5138-267.

Předběžný termín obhajoby

2017/18 LS – PEF

Vedoucí práce

doc. RNDr. Dana Klimešová, CSc.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 23. 2. 2018

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 23. 2. 2018

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 24. 02. 2018

Čestné prohlášení

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

V Praze dne:

Poděkování

Rád bych touto cestou poděkoval doc. RNDr. Daně Klimešové, CSc. za její ochotu, cenné rady a odborný dohled při vedení mé diplomové práce.

Návrh informačního systému v UML

Abstrakt

Diplomová práce se zabývá návrhem informačního systému, který slouží k online prodeji bankovních produktů. Systém je navrhován pomocí sjednoceného modelovacího jazyka (UML). Teoretická část práce přibližuje problematiku vývoje informačních systémů, objektově orientovaný přístup a vývoj v jazyce UML.

Praktická část definuje soubor systémových požadavků, analýzu současného stavu a analýzu systémů, na které je nutné nově vyvíjený systém integrovat. Navržený systém je doplněn technickým návrhem obrazovek, logickým datovým modelem a specifikací webových služeb v programu Enterprise Architect, které slouží jako podklady pro vývoj aplikace.

Klíčová slova: UML, wireframe, objektově orientovaný přístup, informační systém, funkční analýza, technický design, webová služba

Design of Information System in UML

Abstract

The diploma thesis deals with the design of information system, which serves for the online sale of banking products. The information system is designed using a unified modelling language (UML). The theoretical part approaches the problems of information systems, object-oriented approach and development in UML.

The practical part defines a set of requirements for the system, analysis of the current state and analysis of the systems to which the newly developed system needs to be integrated. The proposed system is complemented by a wireframe design, a logical data model, and a Web service specification in Enterprise Architect, which serve as the assignment for application development.

Keywords: UML, wireframe, object-oriented approach, information system, functional analysis, technical design, web service

Obsah

1 Úvod.....	13
2 Cíl práce a metodika	15
2.1 Cíl práce	15
2.2 Metodika.....	15
3 Teoretická východiska	17
3.1 Metodiky vývoje informačních systémů	17
3.1.1 Životní cyklus informačního systému.....	17
3.1.2 Modely životního cyklu systému – klasické	19
3.1.3 Modely životního cyklu systému – agilní	24
3.2 Přístupy k analýze informačních systémů	26
3.2.1 Strukturovaný přístup.....	27
3.2.2 Objektově orientovaný přístup.....	27
3.3 UML	29
3.3.1 Specifikace UML	29
3.3.2 Využití UML.....	31
3.3.3 Vybrané digramy.....	32
3.4 Core Banking systémy a podpůrné bankovní informační systémy	41
3.4.1 Klientské portály	41
3.4.2 Obslužné portály	41
3.4.3 Integrovaná vrstva.....	42
3.4.4 ERP – Plánování podnikových zdrojů	44
3.4.5 Bankovní systémy	45
3.4.6 Reportingové systémy.....	45
3.4.7 Systémy pro asset management	46
3.4.8 Tiskové systémy.....	46
3.4.9 Bankovní a nebankovní registry	47
3.4.10 Mezibankovní komunikační protokoly	49
3.4.11 Systémy monitorující funkčnost systémů	50
3.5 Vybrané legislativní opatření	50
3.5.1 AML – Opatření proti legalizaci výnosů z trestné činnosti	50
3.5.2 GDPR – Obecné nařízení o ochraně osobních údajů.....	51
3.5.3 PEP – Politicky exponované osoby	52
4 Vlastní práce	53
4.1 Současný stav a popis systémů.....	53
4.2 Výchozí předpoklady	58

4.2.1	Nabízené produkty	58
4.2.2	Specifikace prodejního procesu a některých interních pravidel	59
4.3	Specifikace požadavků	63
4.3.1	Funkční požadavky	63
4.3.2	Nefunkční požadavky	64
4.3.3	Identifikace aktérů	65
4.3.4	Případy užití	66
4.3.5	Vztah funkčních požadavků a případů užití	74
4.3.6	Slovníček pojmů	74
4.4	Analýza informačního systému	75
4.4.1	Analytický model tříd	76
4.4.2	Datový slovník	79
4.4.3	Realizace případů užití	80
4.4.4	Procesní logika	87
4.5	Návrh informačního systému	91
4.5.1	Návrhový model tříd	91
4.5.2	Stavové diagramy	94
4.5.3	Digram komponent a diagram rozmístění	96
4.5.4	Wireframes	99
5	Výsledky a diskuse	105
5.1	Zhodnocení návrhu aplikace	105
5.2	Diskuse	106
6	Závěr	108
7	Seznam použitých zdrojů	111
8	Přílohy	114
8.1	Systémové požadavky	114
8.2	Kompletní datový slovník	117
8.3	Kompletní slovníček pojmů	119
8.4	Scénáře případů užití	121
8.5	Sekvenční diagramy realizace případů užití a metod	124
8.6	Omni kanálový prodejní proces	130
8.7	Návrhy obrazovek	131

Seznam obrázků

Obrázek 1: Vodopádový model; zdroj: https://www.fi.muni.cz/~smid/mis-zivcyk.htm	20
Obrázek 2: Prototypový model; zdroj: https://www.fi.muni.cz/~smid/mis-zivcyk.htm	21
Obrázek 3: Spirálový model; zdroj: https://www.fi.muni.cz/~smid/mis-zivcyk.htm	22
Obrázek 4: Pracovní postupy v rámci RUP; zdroj: http://swi.cs.vsb.cz/swi	24
Obrázek 5: Schéma metodiky SCRUM; zdroj: scrum.org	25
Obrázek 6: UML 2.5 diagramy; zdroj: https://commons.wikimedia.org	32
Obrázek 7: Ilustrativní příklad use case diagramu; zdroj: [autor]	33
Obrázek 8: Detail případu užití, zdroj: Arow&Neustadt 2005	34
Obrázek 9: Ilustrativní příklad sekvenčního diagramu; zdroj: [autor].....	36
Obrázek 10: Ilustrativní příklad aktivity diagramu; zdroj: [autor]	37
Obrázek 11: Notace tříd; zdroj: [autor].....	38
Obrázek 12: Vztahy mezi třídami; zdroj: [autor].....	39
Obrázek 13: Ilustrativní příklad stavového diagramu; zdroj: [autor]	40
Obrázek 14: Integrace napřímo; zdroj: [autor]	42
Obrázek 15: Integrační vrstva; zdroj: [autor].....	43
Obrázek 16: Servisně orientovaná architektura; zdroj: volně dle [25]	44
Obrázek 17: Přehled systémů; zdroj: [autor]	54
Obrázek 18: Aktivity diagram procesu prodeje kreditních produktů; zdroj: [autor]	60
Obrázek 19: Přehled funkčních systémových požadavků; zdroj: [autor]	63
Obrázek 20: Vybrané důležité funkční požadavky; zdroj: [autor].....	64
Obrázek 21: Přehled nefunkčních požadavků; zdroj: [autor]	65
Obrázek 22: Aktéři případů užití; zdroj: [autor]	66
Obrázek 23: Přehled případů užití; zdroj: [autor]	67
Obrázek 24: Diagram případů užití; zdroj: [autor]	68
Obrázek 25: Vztah požadavků a případu užití ZaložitŽádost; zdroj: [autor]	74
Obrázek 26: Analytické balíčky; zdroj: [autor]	76
Obrázek 27: Class Diagram analytických tříd 1/2; zdroj: [autor].....	77
Obrázek 28: Class Diagram analytických tříd 2/2; zdroj: [autor].....	78
Obrázek 29: Sekvenční diagram ZaložitŽádost; zdroj: [autor].....	82
Obrázek 30: Sekvenční diagram IdentifikaceUživatele; zdroj: [autor]	83

Obrázek 31: Sekvenční diagram PřihlásitKlienta; zdroj: [autor].....	84
Obrázek 32: Sekvenční diagram NahrátDokument; zdroj [autor]	85
Obrázek 33: Sekvenční diagram ověřTransakci; zdroj: [autor]	86
Obrázek 34: Aktivita diagram případu užití ZaložitŽádost; zdroj: [autor]	88
Obrázek 35: Aktivita diagram případu užití PosouditŽádost; zdroj: [autor].....	89
Obrázek 36: Diagram návaznosti případů užití; zdroj: [autor]	90
Obrázek 37: Class diagram návrhových tříd 1/2; zdroj [autor]	92
Obrázek 38: Class diagram návrhových tříd 2/2; zdroj: [autor]	93
Obrázek 39: Stavový diagram pro třídu Žádost; zdroj: [autor]	95
Obrázek 40: Stavový diagram pro třídu Uživatel, zdroj: [autor]	96
Obrázek 41: Diagram komponent systému e-shop; zdroj: [autor].....	97
Obrázek 42: Poskytované rozhraní systémem e-shop; zdroj: [autor]	98
Obrázek 43: Diagram komponent systému e-shop a návazných systémů; zdroj: [autor]....	98
Obrázek 44: Diagram rozmístění; zdroj: [autor].....	99
Obrázek 45: Využití označených hodnot při tvorbě wireframes; zdroj: [autor].....	100
Obrázek 46: Wireframe domovské obrazovky; zdroj: [autor].....	102
Obrázek 47: Wireframe formuláře specifikace produktu půjčka; zdroj: [autor]	103
Obrázek 48: Wireframe formuláře zadání osobních údajů 2/3; zdroj: [autor]	104
Obrázek 49: Seznam funkčních požadavků 1/2; zdroj: [autor]	114
Obrázek 50: Seznam funkčních požadavků 2/2; zdroj: [autor]	115
Obrázek 51: Seznam nefunkčních požadavků; zdroj: [autor].....	116
Obrázek 52: Sekvenční digram ZaložitŽádost; zdroj: [autor]	124
Obrázek 53: Sekvenční diagram metody získejPředschválenéLimity; zdroj: [autor]	125
Obrázek 54: Sekvenční diagram PosouditŽádost; zdroj: [autor].....	126
Obrázek 55: Sekvenční diagram Contracting; zdroj: [autor].....	127
Obrázek 56: Sekvenční diagram pro metodu autorizujDokument; zdroj: [autor]	128
Obrázek 57: Sekvenční diagram ZaložitProdukt; zdroj: [autor].....	129
Obrázek 58: Omni kanálový prodejní proces; zdroj: [autor]	130
Obrázek 59: Wireframe formuláře produktové specifikace – půjčka; zdroj: [autor]	131
Obrázek 60: Wireframe formuláře osobní údaje – kontakt; zdroj: [autor]	132
Obrázek 61: Wireframe formuláře osobní údaje – osobní údaje; zdroj: [autor].....	134

Obrázek 62: Wireframe formuláře osobní údaje – finance 1/2; zdroj: [autor]	134
Obrázek 63: Wireframe formuláře osobní údaje – finance 2/2; zdroj: [autor]	135
Obrázek 64: Wireframe formuláře vyhodnocení nabídky; zdroj: [autor]	136
Obrázek 65: Wireframe formuláře nabídka – schváleno; zdroj: [autor]	137
Obrázek 66: Wireframe formuláře identifikace; zdroj: [autor]	138
Obrázek 67: Wireframe formuláře generace smluv; zdroj: [autor]	139
Obrázek 68: Wireframe formuláře podpis smluv; zdroj: [autor]	140
Obrázek 69: Wireframe formuláře poděkování; zdroj: [autor]	141

Seznam tabulek

Tabulka 1: Nabízené produkty na e-shopu; zdroj: [autor]	58
Tabulka 2: Detail případu užití ZaložitŽádost; zdroj: [autor]	70
Tabulka 3: Detail případu užití PřihlásitKlienta; zdroj: [autor]	71
Tabulka 4: Detail případu užití NahrátDokument; zdroj: [autor]	72
Tabulka 5: Detail případu užití IdentifikaceUživatele; zdroj: [autor]	73
Tabulka 6: Výňatek ze slovníčku pojmů projektu; zdroj: [autor]	75
Tabulka 7: Kompletní datový slovník; zdroj: [autor]	117
Tabulka 8: Kompletní slovníček pojmů; zdroj: [autor]	119
Tabulka 9: Detail případu užití ZaložitŽádost; zdroj: [autor]	121
Tabulka 10: Tabulka 10: Detail případu užití PosouditŽádost; zdroj: [autor]	122
Tabulka 11: Detail případu užití Contracting; zdroj: [autor]	123

Seznam použitých zkratk

CASE – Computer-aided software engineering

HW – hardware

IS – informační systém

SOAP – Simple Object Access Protocol

SW – Software

UML – Unified Modeling Language

1 Úvod

Tak jako celá řada jiných oborů – jako je např. průmysl, obchod nebo zdravotnictví, tak i bankovníctví je přímo ovlivněno rozvojem informačních technologií – především pak nástupem mobilních zařízení, jejichž rozvoj masivně probíhá v posledních letech.

Na současném retailovém bankovním trhu probíhá konkurenční boj, který je příchodem nových internetových bank stále tvrdší. Klasické bankovní domy se musí na rozdíl od nových subjektů vyrovnat mimo jiné také se stále provozovanými „historickými“ IT řešeními, jejichž vznik v mnohých případech spadá až do devadesátých let minulého století. Právě tento fakt zavedeným společnostem (nejen v oboru bankovníctví) často prodražuje investice do nové infrastruktury – zejména pak náklady na analýzu současného stavu, která v mnohých případech přesahuje obecně akceptovatelnou hranici 30 % kapacity analytiků. I přes tyto skutečnosti banky a další finanční i nefinanční instituce investují do IT řešení, částečně v rámci konkurenčního boje, částečně pak kvůli některým legislativním opatřením na zabezpečení dat (např. úprava legislativy pro elektronický podpis) nebo ochranu dat spotřebitele. Aktuálně mezi tato legislativní opatření patří směrnice GDPR – neboli obecné nařízení o ochraně osobních údajů, které významně zvyšuje ochranu a způsob zpracování osobních dat občanů. Zejména banky a pojišťovací společnosti v posledních letech vynakládají nemalé prostředky, aby zajistily, že budou schopny dostát novým pravidlům, což často dává prostor pro vývoj nových řešení na místo úpravy stávajících.

Tato diplomová práce se zabývá návrhem informačního systému ve *fiktivní* zavedené bance, která provozuje řadu technologických řešení pro každodenní provoz. Společnost se rozhodla investovat do nového systému, který bude umožňovat online prodej vybraných bankovních produktů, a tím nahradí současné již nevyhovující řešení. Podstatným požadavkem společnosti je specifikace omni-kanálového prodejního procesu tak, aby jakýkoli další nově implementovaný prodejní frontend/aplikace (např. sekce smartbankingu sloužící pro prodej) mohl využívat služby již existujícího univerzálního subsystému. Navrhovaný systém bude, pokud možno, co nejvíce využívat stávající řešení. Například data z CRM systému, informace ze systému, který zajišťuje rizikový scoring osoby nebo nástroj, který provádí úvěrový scoring. Podstatným přínosem nového informačního systému – e-shopu je možnost

online autentizace osoby a plně automatizované zřízení produktu v bankovních systémech. E-shop bude sloužit klientům i neklentům banky na jednom sjednoceném prostředí. Vyvíjený frontend částečně sdílí backend s paralelně vyvíjeným pobočkovým systémem. Na samoobslužném kanále, na rozdíl od kanálu sloužícímu pro asistovaný prodej, budou pokryty pouze hlavní případy užití. Datové úložiště žádostí bude sdílené pro všechny prodejní kanály.

Návrh systému je proveden pomocí unifikovaného modelovacího jazyka (UML). Jazyk UML je souborem grafických notací, které se používají při vývoji softwaru. V oblasti analýzy a návrhu informačních systémů se stal standardem, který slouží zároveň jako dokumentace systému a specifikace pro vývoj. Jedním z důvodů rozšíření jazyka je možnost vytvoření odlišných pohledů na systém, jak statických, tak dynamických, které mohou nabývat různých úrovní abstrakce. Další nespornou výhodou popisu systému pomocí grafických notací je také jazyková nezávislost, která je významná pro velké korporace, které často mívají outsourcovaný vývoj do zemí třetího světa, ale analýza je prováděna interními zaměstnanci nebo externí společnostmi se sídlem v dané zemi.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem práce je analýza a návrh informačního systému – e-shopu, který umožňuje online prodej bankovních produktů. Díky specifickým produktům, které banky nabízejí a legislativním opatřením, které upravují jejich způsob prodeje, není v současné době dostupné žádné vhodné krabicové řešení. Navrhovaný systém musí mimo jiné podporovat omni-kanálový¹ prodej (tzn. respektovat sjednocený prodejní proces) a umožnit prodej a nákup produktu klientovi i neklientovi společnosti. Dílčím cílem práce je návrh řešení online autentizace uživatele informačního systému, který dosud není klientem banky a ověření identity osoby tak nelze jednoduše provést přihlášením do systému s využitím již přiděleného identifikátoru a hesla.

Základem pro návrh systému je soubor požadavků uživatelů – tzv. seznam systémových požadavků, na jehož základě jsou dále specifikovány hlavní případy užití, datový model a v neposlední řadě samotný prodejní proces.

2.2 Metodika

Teoretická východiska práce jsou založena na studiu odborných literárních zdrojů zabývajících se tématy vývoje informačních systémů, metodikami vývoje software, jazykem UML, a dále pak bankovními systémy. Výše zmíněná problematika je rozsáhlým tématem, a proto je v teoretické části nastíněn vždy jen nezbytný kontext, který je východiskem pro praktickou část práce.

Praktická část se zabývá návrhem informačního systému umožňující prodej online bankovních produktů stávajících i novým klientům *fiktivní* zavedené banky. Z výčtu požadavků, který byl získán strukturovanými rozhovory s koncovými uživateli a z následné analýzy těchto požadavků byl identifikován soubor funkčních a nefunkčních požadavků na

¹ Všechny prodejní kanály jsou chápány jako „jedno“ prostředí. Zákazník tam může např. započít svůj nákup na pobočce (asistovaný prodej) v pobočkovém systému a dokončit doma v prostředí e-shopu.

system, který slouží jako vstup pro další fáze návrhu systému. Za pomoci pozorování byl ve fázi analýzy popsán současný stav infrastruktury a zevrubný popis subsystémů, na které bude nově implementovaný systém integrován. Následuje tvorba datového modelu, realizace případů užití a návrhy obrazovek a některých služeb.

Návrh systému je tvořen na základě jazyka UML. Model navrhovaného systému byl vytvořen v prostředí CASE nástroje Enterprise Architect verze 13.5.

3 Teoretická východiska

3.1 Metodiky vývoje informačních systémů

Metodika vývoje informačního systému je souborem doporučených přístupů, pravidel, metod, technik a nástrojů, které jsou vedou ke snížení pravděpodobnosti chyb a opomenutí. Metodika je jen šablonou, je zpravidla nutné ji přizpůsobit konkrétnímu projektu.

Metodika předepisuje postup, jeho kroky a artefakty získané v těchto krocích [5].

- *Metoda* určuje, co je nutné v určité etapě vývoje informačního systému vykonat. Zpravidla je metoda navázaná na určitý přístup (viz kapitola 3.2).
- *Technika* udává způsob, jak dosáhnout požadovaného výsledku. Určuje např. sled kroků, užití nástrojů aj.
- *Nástroj* je prostředkem k realizování a formalizování dané činnosti. Obvykle bývá svázán s technikou – CASE² nástroje [6].

3.1.1 Životní cyklus informačního systému

Životní cyklus informačního systému definuje vývojové fáze systému, které jsou dle různých autorů a metodik dělené do jiného počtu etap na základě různých kritérií. Z této struktury se dále odvíjí další obsah metodiky – cíle, metody, artefakty, techniky atp.

Např. dle metodiky MDIS³ je životní cyklus informačního systému členěn na následující etapy [7]:

1. etapa informační strategie organizace: specifikace zadání
2. etapa úvodní studie projektu: předběžná analýza
3. etapa globální analýzy a návrhu IS: detailní analýza
4. etapa detailní analýzy a návrhu IS: návrh
5. etapa implementace IS: implementace

² CASE: Computer Aided Software Engineering. Integrované nástroje pro podporu analýzy a návrhu aplikací.

³ MDIS: Multidimensional Development of Information System, česky Metodický základ systémové integrace. Metodika MDIS je vyvíjena od počátku 90 let na katedře informačních technologií VŠE [7].

6. etapa zavádění IS do provozu: zavedení do provozu
7. etapa provozu, údržby a dalšího: rozvoje IS provoz

Šmíd uvádí následující obecné životní cykly informačního systému [10]:

1. *Předběžná analýza neboli specifikace cílů*

Jedná se o elementární fáze vývoje, ve které jsou definovány požadavky organizace vzhledem k systému. Součástí této fáze je sběr těchto požadavků, jejich hrubý rozbor a odhad času implementace a nákladů založený na výše zmíněném rozboru.

2. *Analýza systému neboli specifikace požadavků*

Navazuje na předchozí etapu, jedná se o detailní rozbor struktury a dat systému.

3. *Projektová studie neboli návrh*

Výsledkem této etapy je dokument, obsahující časový harmonogram, cenu vývoje informačního systému a konkrétní návrh implementace systému. Dokument musí být zpracován v dostatečném detailu, ale zároveň s přihlédnutím k faktu, že se jedná o klíčový dokument, na jehož základě se management společnosti rozhoduje o realizaci projektu – musí být tedy srozumitelný všem.

4. *Implementace*

V rámci této etapy probíhá programování informačního systému na základě vstupů vzniklých v přechodných etapách. V závěru etapy jsou připravena testovací data.

5. *Testování*

Testování má za cíl ověřit správnou funkcionalitu systému dle připravených testovacích scénářů. Zároveň v této etapě probíhá oprava chyb.

6. *Zavádění systému*

Etapa zahrnuje instalaci, zavedení do provozu, případnou migraci dat a poskytnutí manuálů a školení.

Zavedení systému může být provedeno jedním z následujících způsobů.

- a. Souběžné zavádění: nově zaváděný systém pracuje určitou dobu současně s nahrazovaným systémem, dokud není otestován.
- b. Pilotní zavádění: systém se zavádí jedné části (oddělení) společnosti a po jeho otestování je zveden do celé organizace.

- c. Postupné zavádění: vhodné pro složité systémy. Nejprve je zaveden primární systém a následně na něm závislé moduly.
- d. Nárazová strategie: původní systém je nárazově odstraněn a nahrazen novým.

7. *Zkušební provoz*

Zkušební provoz je realizací projektu. V této fázi má dodavatel povinnost odstranit chyby nalezené během provozu a zároveň je zde prostor pro řešení dodatečných požadavků uživatelů na funkcionality systému.

8. *Rutinní provoz a údržba*

Závěrečná fáze projektu, kdy je poskytována podpora provozování systému na základě SLA⁴.

9. *Reengineering*

Zhodnocení naplnění požadavků na systém a jeho funkcionality. Pokud je systém nevyhovující a nelze provést pouze úpravu, vede krok do počáteční fáze životního cyklu.

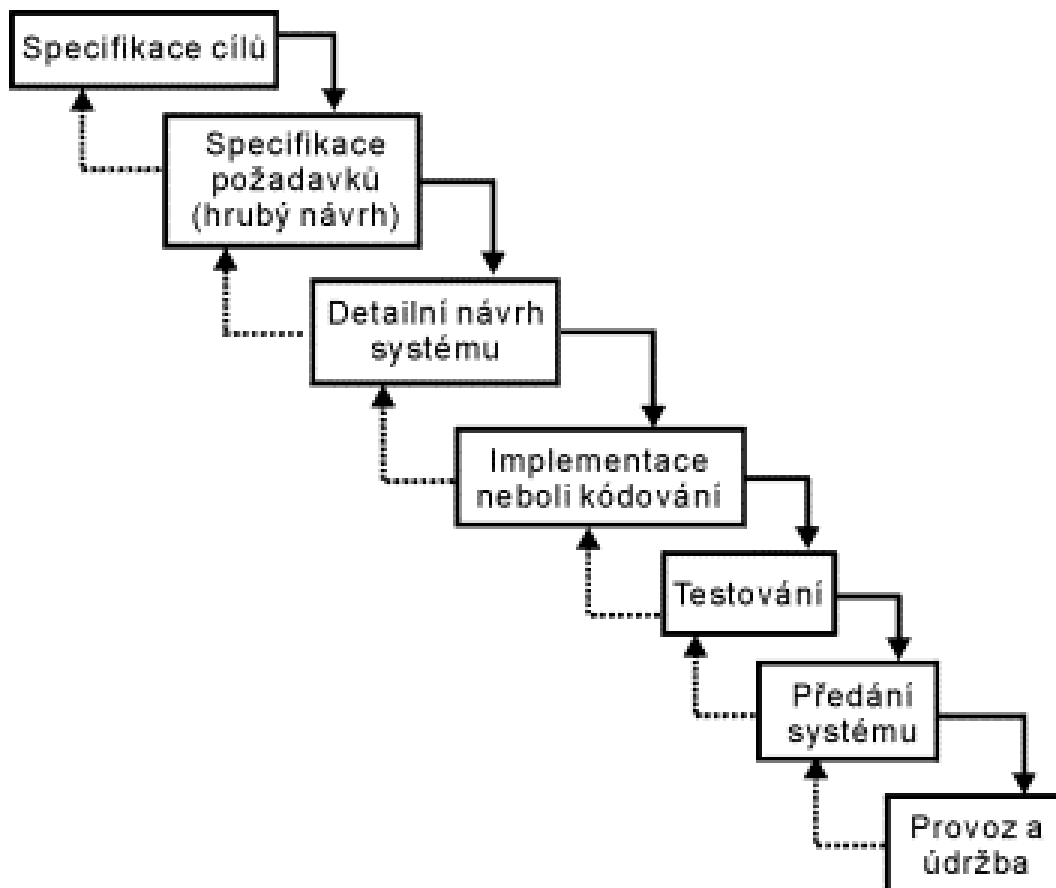
3.1.2 **Modely životního cyklu systému – klasické**

Vodopádový model

Je sekvenční vývojový proces, jehož základní charakteristikou jsou na sebe navazující, neprotínající se etapy životního cyklu informačního systému. Systém patří mezi tradiční metody vývoje již od 70. let. Jeho cílem je vnést do vývoje systémů jednotný řád [9, 10].

⁴ SLA: Service Level Agreement [12], dohoda o úrovni poskytovaných služeb. SLA definuje rozsah, úroveň a kvalitu služby, například:

- Garantovanou časovou dostupnost (např. 24/7/365–24 hodin, 7 dnů v týdnu a 365 dní v roce)
- Garantovanou cenu
- Garantovanou rychlost řešení potíží se službou (např. do 30 minut po oznámení problému)

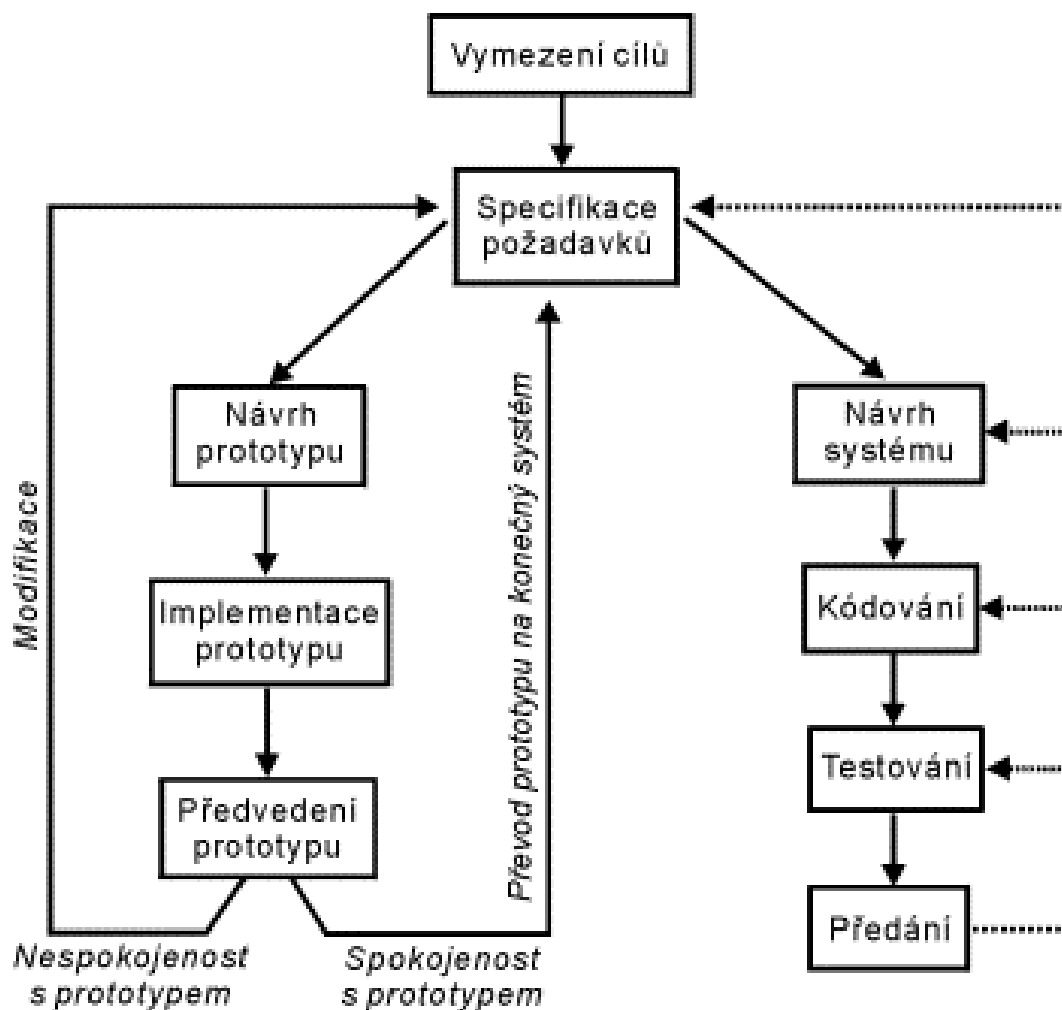


Obrázek 1: Vodopádový model; zdroj: <https://www.fi.muni.cz/~smid/mis-zivcyk.htm>

- *Výhody:* Rychlý a levný přístup, v případě, kdy je přesně známý problém a způsob řešení.
- *Nevýhody:* Zejména u rozsáhlejších projektů prakticky nelze dokončit jednu fázi a zahájit další, bez toho, aniž bych se k předchozí fázi v budoucnu opět vrátil – požadavky klientů se často mění v průběhu vývoje, popř. ve fázi testování.

Prototypový model

Podstatou modelu je předpoklad změn výchozích požadavků na funkcionalitu systému již během vývoje a umožnění reakce na tyto změny. Zákazník je seznámen s první verzí systému v co nejkratší době v takové formě, aby bylo možné reagovat na výsledek. Následně je prototyp modifikován dle požadavků klienta, dokud není zákazník spokojen [9, 11].



Obrázek 2: Prototypový model; zdroj: <https://www.fi.muni.cz/~smid/mis-zivcyk.htm>

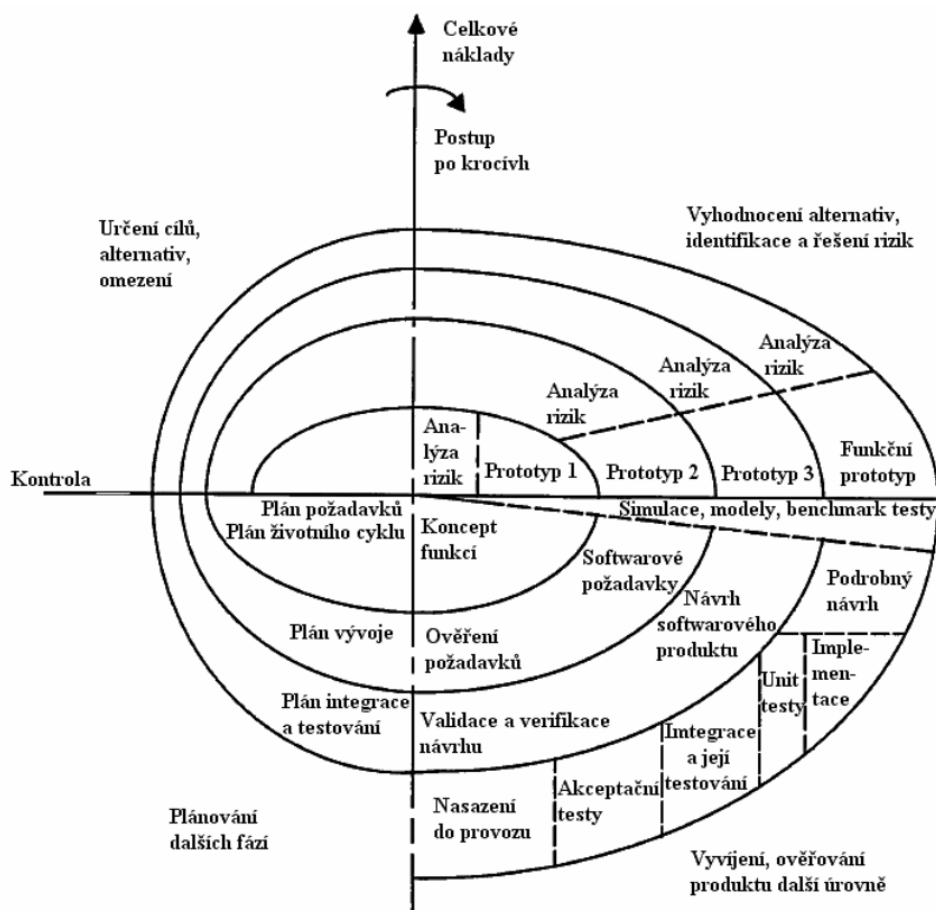
- *Výhody:* Umožňuje velmi přesně obsáhnout požadavky budoucích uživatelů a reagovat na změny.
- *Nevýhody:* U rozsáhlých projektů je tato metoda poměrně náročná, proto se většinou předem určuje množství opakování prototypů a každé z nich musí být provedeno do stanoveného termínu.

Spirálový model

Model je kombinací prototypového modelu a analýzy rizik⁵. Podstatou spirálového modelu je opakování vývojových kroků tak, že je v každé další fázi na již uzavřenou část navazována

⁵ Rizika lze v kontextu spirálového modelu chápat v nejobecnějším smyslu (mohou tak zahrnovat např. legislativní omezení nebo marketingové požadavky). Model je založen na iterativním přístupu, a zavádí opakovanou analýzu všech rizik. Lépe se tak vyrovnává s pozdější úpravou požadavků.

část na vyšší úrovni [1, 10]. Úhlová dimenze udává časovou náročnost a radiální úroveň udává rostoucí náklady.



Obrázek 3: Spirálový model; zdroj: <https://www.fi.muni.cz/~smid/mis-zivcyk.htm>

- *Výhody:* První verze systému je možné představit a hodnotit při jejich vzniku pomocí prototypů. Přístup dovoluje konzultovat požadavky zákazníků v jednotlivých krocích a modifikovat systém podle nových požadavků.
- *Nevýhody:* Vyžaduje úzkou spolupráci zákazníků s analytiky, není vhodný pro systémy vyvíjené na zakázku bez účasti uživatelů a neumožňuje přesné naplánování termínů.

RUP

RUP neboli Rational Unified Process je jednou z rigózních metodik, podobně jako např. OPEN⁶. RUP náleží do skupiny tzv. přístupů řízených případy použití (*use-case driven approach*). Jako základní element je chápán případ použití definovaný jako posloupnost akcí prováděných systémem či uvnitř systému, která poskytuje určitou hodnotu uživateli systému. Pro modelování procesů se využívá prostředků jazyka UML [13, 14, 1].

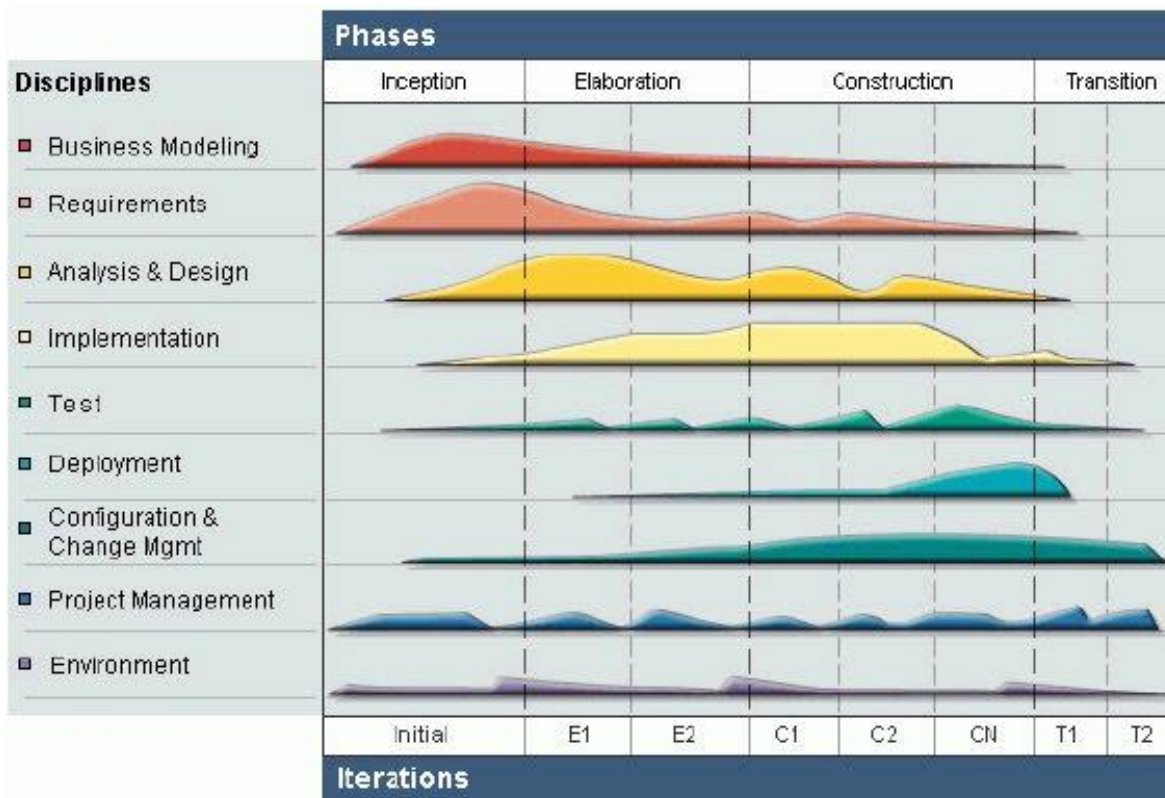
Projekty vyvíjené metodou RUP by měli dodržovat následující fáze [2]:

- *Fáze zahájení (inception)*: definuje účel, rozsah projektu a alokaci zdrojů
- *Fáze projektování (elaboration)*: v této fázi je potřeba analyzovat požadavky zákazníka, celého projektu a definovat základy architektury. V této fázi bychom měli identifikovat a vyřešit hlavní rizika projektu.
- *Realizační fáze (construction)*: v této fázi probíhá vývoj funkcí systému
- *Fáze předání (transition)*: v této fázi je projekt předán zákazníkovi nebo do dalšího cyklu. Součástí je např. nasazení software na určitý hardware nebo školení uživatelů.

Činnosti, které provázejí vývoj produktu se nazývají *disciplíny* a metodika RUP je definuje následovně:

- Tvorba modelu
- Správa požadavků
- Analýza a návrh
- Implementace
- Testování
- Nasazení
- Konfigurace a změny
- Řízení projektu
- Správa prostředí

⁶ Object-oriented Process, Environment and Notation: Metodika založená na objektově orientovaném vývoji. Metodika je charakteristická přizpůsobivým řešením dle schopností vývojového týmu, požadavků zákazníka a velikosti projektu.



Obrázek 4: Pracovní postupy v rámci RUP; zdroj: <http://swi.cs.vsb.cz/swi>

Každá fáze obsahuje několik iterací. Před započítím nové iterace musí být splněna definovaná kritéria předchozí iterace.

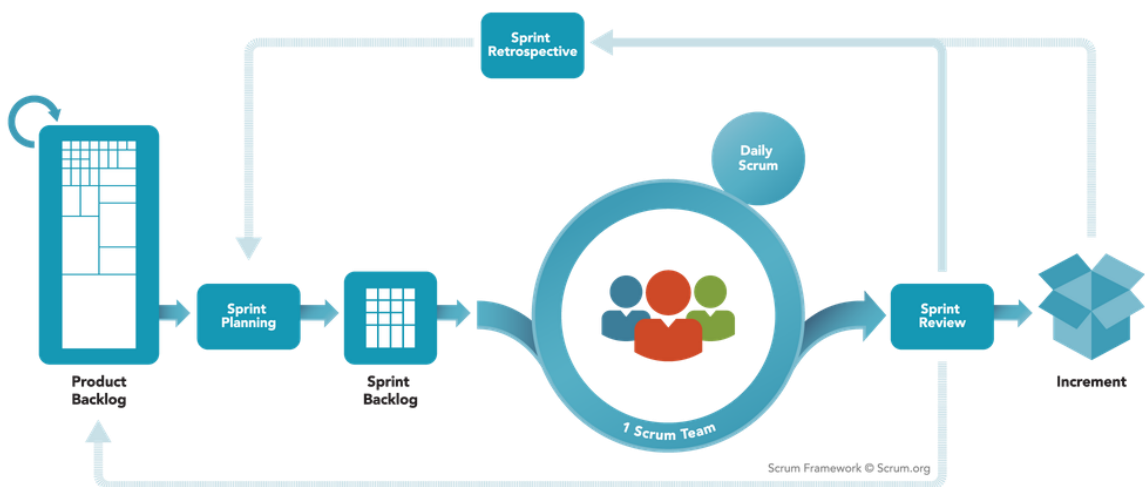
3.1.3 Modely životního cyklu systému – agilní

SCRUM

Metodika je založená na tzv. iteracích (*sprints*), jež obvykle trvají okolo dvou až čtyř týdnů. V rámci každé iterace je řešeno několik předem definovaných úkolů (*backlog*), které si řešitelský tým (zpravidla 5-10 členů) rozdělí. Jednotlivé iterace jsou zakončeny schůzí (*sprint review meeting*), kterého se účastní všichni členové řešitelského týmu včetně zákazníků, jejich účast ale není vyžadována [8].

Výstupem každého sprintu je funkční prototyp a určení náplně dalšího sprintu. Důležitou praktikou používanou v celém průběhu sprintu je každodenní krátká porada celého vývojového týmu, která slouží k určení dalšího postupu na základě dosavadních výsledků.

Velkou výhodou metodiky SCRUM je schopnost rychle reagovat na změny vznikající v průběhu projektu. Prostřednictvím úpravy plánu sprintu lze dobře reflektovat přicházející požadavky a rizika z nich plynoucí. Takto koncipovaný způsob vývoje lze charakterizovat jako proces empirický, na rozdíl od předem stanoveného, který je charakteristický pro tradiční přístup k vývoji.



Obrázek 5: Schéma metodiky SCRUM; zdroj: scrum.org

Charakteristické znaky metodiky SCRUM:

- vývoj v krátkých iteracích 2-4 týdny
- fixní náklady a fixní termíny
- flexibilní určení výstupů předměty dodávky
- malé týmy obvykle 5-10 lidí
- časté analýzy rizik a revize
- důraz na komunikaci a spolupráci mezi všemi zúčastněnými včetně zákazníka

3.2 Přístupy k analýze informačních systémů

Elementárními postupy analýzy a návrhu informačních systémů je abstrakce a konkretizace.

- *Abstrakce*: zobecnění, vyloučení odlišnosti objektů
- *Konkretizace*: vyčlenění specifických vlastností objektů

Na těchto postupech jsou založeny principy:

1. *Princip rozlišovacích úrovní*

Vyvíjený systém je vždy zobrazován na určité úrovni detailu. Nejhrubším detailem je zobrazení systému jako jednoho prvku s vazbami na okolí. Na podrobnějších úrovních systému jsou zobrazeny detaily rozpracované do potřebné úrovně pro řešení problému.

2. *Princip tří architektur*

Souvisí s výše popsaným principem rozlišovacích úrovní. Zobrazení vyvíjeného informačního systému vždy dělíme do tří rozlišovacích úrovní, jedná se o tzv. vrstvenou abstrakci systému.

- Vrstva konceptuální: zobrazení systému a jeho vazeb na okolní systémy, subsystémy a vazbami mezi nimi. Z tohoto zobrazení je patrné, které subsystémy fungují jako rozhraní mezi vyvíjeným systémem a okolím. Účelem zobrazení je zjistit CO je obsahem systému.
- Vrstva technologická: zobrazení systému, která vypovídá o tom, JAK je systém realizován. Je zde zohledněna organizace dat a architektura systému,
- Vrstva implementační: detaily konkrétní implementace. Z této vrstvy je patrné, ČÍM je systém realizován.

3. *Princip modelování*

účelové zjednodušené zobrazení systému jako abstraktního obrazu reality za pomoci vhodných prostředků. Model je formalizovaným prostředkem pro znázornění, který zobrazuje strukturu systému na dané rozlišovací úrovni.

Základní přístupy k analýze a návrhu IS jsou popsány níže v kapitolách 3.2.1 a 3.2.2.

3.2.1 Strukturovaný přístup

Strukturovaný přístup vývoje informačního systému byl zformulován v 70. letech 20. století a odráží myšlenkový postup „strukturování“ problematiky, zkoumání předmětu.

Tento prvek strukturalizace je popsán jako prostředek pro boj se složitostí. Strukturovaný přístup rozkládá celek hierarchickou formou na menší části. Tyto části jsou dále děleny na koncové části nazývané listy stromu hierarchického rozkladu [15].

Principy strukturovaného přístupu vycházejí ze základních myšlenek strukturovaného programování. Základním principem strukturovaného přístupu je hierarchická dekompozice systému a v průběhu návrhu informačního systému jsou rozpracovány dva aspekty vyvíjeného systému, a to:

- *Procesy*
- *Data*

Problematice toto přístupu se věnovala řada autorů, např. Tom DeMarco (*Structured Analysis and System Specification*, 1979), Gane, Sarson, Warnier, Ken Orr nebo Edward Yourdon (*Modern Structured Analysis*, 1989).

Strukturované techniky pro návrh systému zahrnují především nástroje a postupy pro:

- *funkční model* – vytvořený pomocí Data Flow diagramů (DFD) a mini specifikací elementárních funkcí
- *datový model* – vytvořený pomocí ERD diagramů a konceptuálního schématu dat
- *model přechodu stavů* – stavů vytvořený pomocí State Transition Diagramů – STD

3.2.2 Objektově orientovaný přístup

Objektově orientovaný přístup k analýze a návrhu IS je založen na objektech, jakožto strukturách, které mají definované své vlastnosti (atributy) a své chování (operace, které daný objekt může provádět). Vlastnosti i operace jsou zapouzdřené v jednotlivých objektech. V pojetí objektově orientovaných metod analýzy a návrhu IS je pak systém chápán jako množina spolupracujících objektů. Každý objekt je přitom schopen reagovat na události, které na něj působí jako impuls [1].

Pro každý objekt zavádíme čtyři charakteristické aspekty (OOM):

- *Jedinečnost*: každý objekt je rozlišitelnou entitou, i když má jinak stejné kvalitativní i kvantitativní charakteristiky. K rozlišení se používá identifikační číslo.
- *Zatříditelnost*: objekty se stejnou datovou strukturou (atributy) a chováním (operacemi) jsou seskupovány do tříd.
- *Mnohotvárnost*: (polymorfismus) stejně označená operace se může chovat rozdílně a dávat rozdílné výsledky pro různé třídy objektů.
- *Dědičnost*: sdílení atributů a operací mezi třídami založenými na hierarchických vztazích. Třída může být definována poměrně široce a potom ji lze následně zjemňovat do podtříd. Každá podtřída pak zahrnuje vlastnosti své nadřazené třídy.

Objektově orientované programování je založeno na 4 principech, které vyjadřují podstatu objektového přístupu [10].

1. *Zobecnění* – představuje zaměření na podstatné vnitřní aspekty objektů a ignoruje jeho nepodstatné vedlejší vlastnosti. Zaměřuje pozornost na to, co objekt je
2. a co by měl dělat, spíše než na to, jak by to měl dělat.
3. *Zapouzdření* – znamená oddělení externích aspektů objektu, které jsou dostupné ostatním objektům, od interních implementačních detailů objektu, zakrytých ostatním objektům. To znamená, že implementace objektů může být změněna, aniž by se to dotklo aplikací, ze kterých je objekt použit.
4. *Sdílení* – je umožněno dědičností datových struktur a chováním mezi několika podobnými třídami. To umožňuje opakované použití již navržených a ověřených programů v dalších aplikacích.
5. *Spolupůsobení* – jednoznačnost, zařaditelnost, mnohotvárnost a dědičnost charakterizují hlavní proud objektově orientovaných jazyků.

3.3 UML

Do roku 1994 se pro vizuální modelování používalo několik soupeřících jazyků a pro vývoj softwaru vznikaly různé metodiky. Mezi první pokusy o sjednocení byla v roce 1994 metodika Fusion, která se dlouhodobě neuchytila. Kolem roku 1995 začínají práce na jazyku UML, který se o dva roky později stal prvním průmyslově orientovaným standardem pro vizuální modelování [1, s. 29, 30].

3.3.1 Specifikace UML

Unified Modeling Language je objektově orientovaný jazyk využívaný v softwarovém inženýrství pro navrhování a dokumentaci informačních systémů. UML je grafický nástroj, který byl vytvořen pro vizuální modelování systémů. Žádný z diagramů v rámci UML se nesnaží zachytit navrhovaný systém jako celek, ale vyjadřuje vždy právě jeden pohled na vyvíjený systém [6, s. 11, 11].

UML není metodikou ani programovacím jazykem⁷. Jazyk UML má vlastní grafickou syntaxi, tzn. svá pravidla pro sestavování jednotlivých elementů do větších objektů a sémantiku, která určuje jednotlivým výrazům jejich význam. Není to programovací jazyk. UML má přesné specifikace, co by měl každý diagram obsahovat, a tím ulehčuje práci celému týmu. Protože všechny diagramy jsou mezi sebou propojené a sdílejí potřebné informace mezi analytiky a vývojáři [1, 11].

Stavební bloky UML

Jazyk UML se skládá z tří hlavních stavebních bloků [1, 16]:

- *Předměty*: Samotné prvky modelu.
- *Relace*: Vztahy (vazby) mezi předměty (prvky modelu), umožňují zachytit významový vztah mezi předměty.
- *Diagramy*: Jednotlivé pohledy na model, který je tvořen předměty a relacemi.

⁷ Přestože při podrobném zápisu některých diagramů lze přímo generovat spustitelný kód. této souvislosti se často používá výraz Model Driven Architecture (MDA) – standard skupiny OMG, který definuje použití UML jako programovacího jazyka.

Obecná mechanika UML

UML obsahuje čtyři mechanismy, které jsou používány v rámci jazyka konzistentně. Jedná se o tuto strukturu:

- *Specifikace*

Každý element je specifikován textem, který element popisuje, tato specifikace podrobněji popisuje – udává smysl modelovanému elementu, protože popisuje business pravidla.

- *Ornament*

- Každý prvek modelu lze obohatit ornamenty – dodatečnými informacemi, pokud je nutné poukázat na specifický obsah nebo význam.

- *Podskupina*

Rozlišujeme dvě hlavní podskupiny:

- *Klasifikátory a instance*: Abstraktní pohled na předmět je klasifikátorem, konkrétní předměty jsou instance. Třída je tedy klasifikátor (např. bankovní účet) a objekt je instancí (např. účet číslo 2100564298).

- *Rozhraní a implementace*: V jazyce UML je striktně odděleno rozhraní – tedy „to“ co předmět vykonává a implementace tedy způsobu „jak“ to vykonává.

- *Rozšiřitelnost*

Vzhledem ke specifickým požadavkům v některých odvětvích, ale není možné vytvořit jeden univerzální jazyk. Díky nástrojům rozšiřitelnosti je ale UML velmi flexibilní a umožňuje tak pokrýt široké spektrum potřeb uživatelů.

- *Omezení*: umožňuje přidávat k prvku nová pravidla

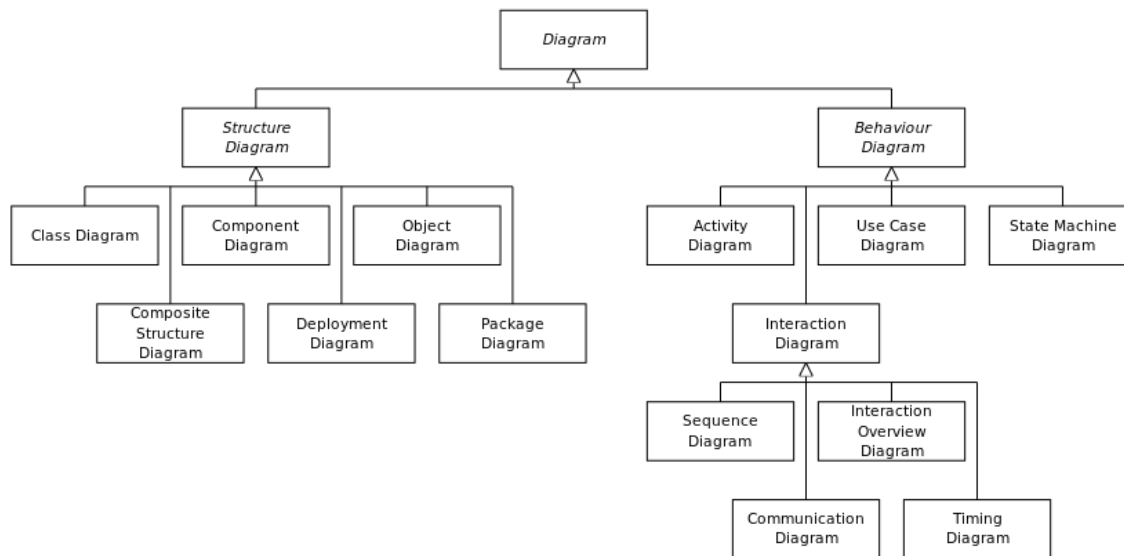
- *Stereotypy*: umožňuje definovat nový prvek UML

- *Označené hodnoty*: umožňuje rozšířit specifikaci prvku informací sestavenou jen k tomuto účelu

3.3.2 Využití UML

Aktuální verze jazyka UML 2.5 definuje 14 oficiálních typů diagramů zobrazených na Obr. 6, dělených do dvou hlavních kategorií:

- a. Diagramy struktur
 - i. Diagram tříd
 - ii. Diagram složených struktur
 - iii. Diagram objektů
 - iv. Diagram komponent
 - v. Diagram nasazení
 - vi. Diagram balíčků
 - vii. Profilový diagram
- b. Diagramy chování
 - i. Diagram aktivit
 - ii. Diagram případů užití
 - iii. Stavový diagram
 - iv. Diagram interakcí
 - 1. Sekvenční diagram
 - 2. Diagram komunikace
 - 3. Diagram přehledu interakcí
 - 4. Diagram časování



Obrázek 6: UML 2.5 diagramy; zdroj: <https://commons.wikimedia.org>

3.3.3 Vybrané digramy

Níže jsou blíže popsány vybrané diagramy jazyka UML, které jsou z pohledu praktické části této práce významné:

Diagram případů užití

Případ užití (use case) je metodou pro zachycení funkčních požadavků na systém. Případy popisují typické operace mezi uživateli systému a samotným systémem a předkládají nám příběh o tom, jak je systém používán [2].

Obsah případu užití:

- Hlavní scénář
- Primární/sekundární účastníci
- Krok
- Rozšíření

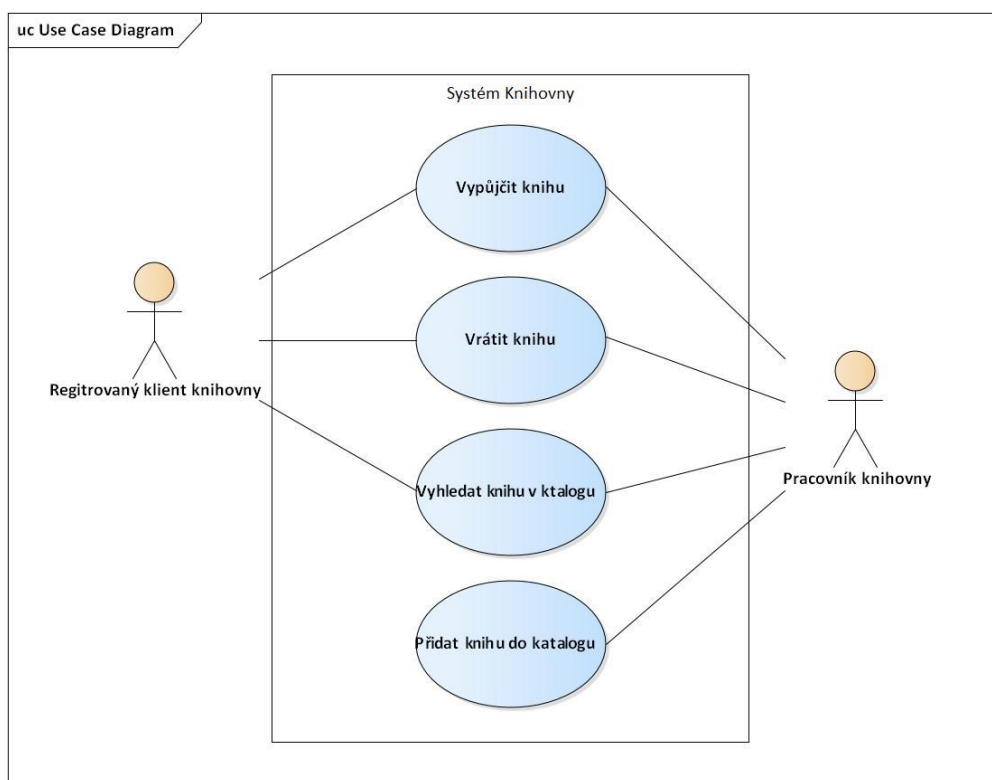
Vazby případů užití

- *Include*: vazba zahrnutí, zahrnutý případ užití je spuštěn pokaždé kdy je spuštěn zahrnující případ užití

- *Exclude*: vazba rozšíření, jedná se o samostatný případ užití, který může, ale nemusí být spuštěn z definovaného bodu rozšíření
- *Dědičnost*: mechanismus pro snížení složitosti modelu

V rámci modelování diagramu případů užití identifikujeme následující komponenty:

- *Ohraničení*: Vymezuje hranici systému nebo subsystému.
- *Aktéry*: Rozlišujeme primárního aktéra – jehož cílem je naplnění případu užití a sekundární aktéry – další aktéři, se kterými systém komunikuje při vykonávání případu užití.
- *Scénáře*: Posloupnost kroků popisujících interakci mezi systémem a uživatelem, vedoucí k naplnění⁸ cíle případu užití.
- *Relace*: Vztahy mezi účastníky případu užití



Obrázek 7: Ilustrativní příklad use case diagramu; zdroj: [autor]

⁸ Ne vždy je scénář vede k naplnění cíle – např. pokud v rámci procesu selže některá z autorizací, pak není případ užití naplněn, jedná se poté o alternativní scénář nebo chybový scénář případu užití.

K případu užití je možné také přidávat další informace např.:

- Vstupní podmínky a výstupní podmínky
- Garance
- Spouštěč

Jazyk UML nedefinuje jednotný standard pro popis případů užití, existuje ale několik šablon vzniklých na základě best practices⁹, např. Jim Arlow ve své knize *UML 2 a unifikovaný proces vývoje aplikací* doporučuje následující strukturu:

- Název případu užití
- Jedinečný identifikátor
- Stručný popis
- Aktéři
- Vstupní a výstupní podmínky
- Tok událostí

Use case: FindEmployeeDetails
ID: 4
Brief description: The Manager finds the employee details.
Primary actors: Manager
Secondary actors: None.
Preconditions: 1. The Manager is logged on to the system.
Main flow: 1. The Manager enters the employee's ID. 2. The system finds the employee details.
Postconditions: 1. The system has found the employee details.
Alternative flows: None.

Obrázek 8: Detail případu užití, zdroj: Arow&Neustadt 2005

⁹ Best Practice: překládá se jako nejlepší praxe, osvědčená praxe, je pojem pro osvědčené postupy, procesy či osvědčené metody řízení, pomocí kterých se ve více organizacích dosáhlo dobrých výsledků a používají se proto jako doporučení pro ostatní [17].

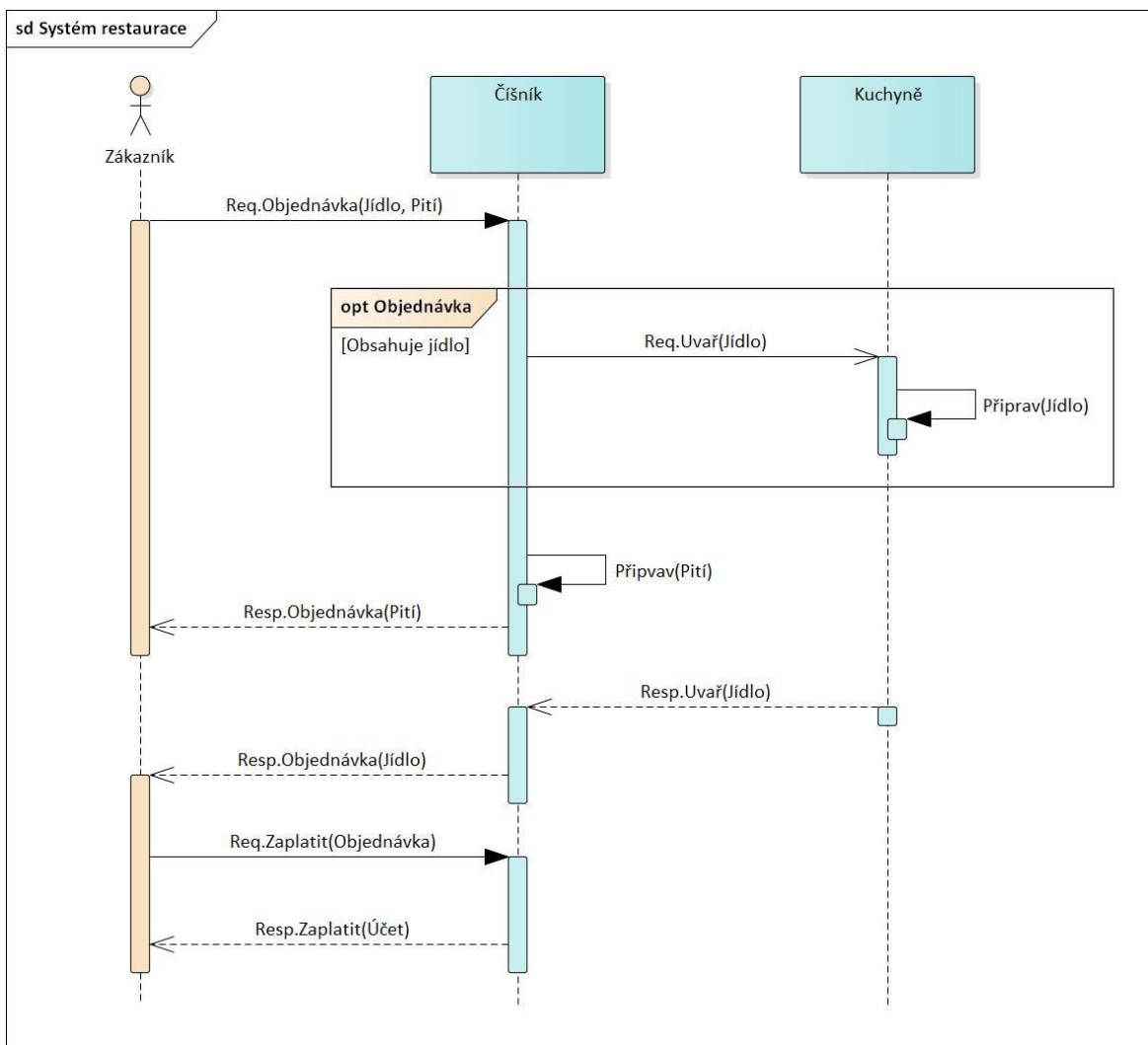
Sekvenční diagram

Sekvenční diagram typicky zachycuje chování jednoho scénáře. Ukazuje několik vzorových objektů a zpráv, které jsou předávány mezi těmito objekty v rámci daného případu užití a jejich časovou souslednost.

Sekvenční diagramy znázorňují klasifikátory na horní hraně diagramu. Na horní liště vlevo je umístěn aktér, který je zodpovědný za spuštění případu užití. Z klasifikátorů vede svislá čára reprezentující život objektu v průběhu daného případu užití. V terminologii UML je tako čára nazýváána jako čára života (*lifeline*). Každá zpráva mezi objekty je znázorněna šipkou mezi čarami života. Zprávy jsou požadavky objektu o vyvolání operace druhého objektu. Objekty mohou zasílat zprávy i sami sobě (tzv. *selfcall*) Pořadí (*sekvence*) zasilání zpráv je reprezentována osou Y, čas plyne od shora dolů [6].

Prvky diagramu [1]:

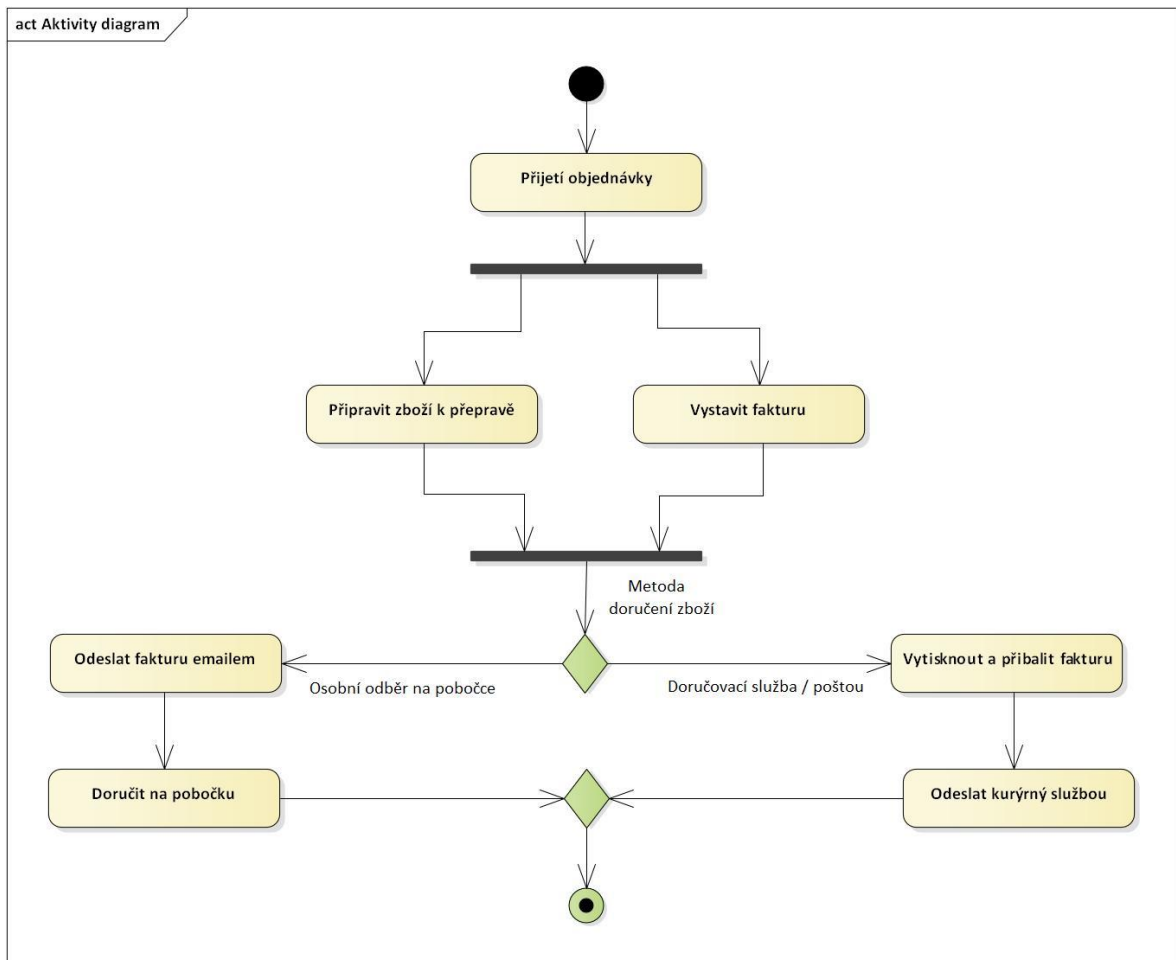
- *Lifeline*: Reprezentuje účastníka interakce.
- *Message*: Představuje komunikaci mezi dvěma účastníky a definuje druh zprávy
- *Fragmenty*: Definuje oblasti sekvenčního diagramu s odlišným chováním nebo podmínkami volání (reprezentuje např. podmínky IF, LOOP, CASE atp.).



Obrázek 9: Ilustrativní příklad sekvenčního diagramu; zdroj: [autor]

Diagram aktivit

Diagramy aktivit jsou technikou určenou k popisu procedurální logiky, business procesů a toku práce (*workflow*). V mnoha ohledech hrají podobnou roli jako vývojové diagramy (*flowcharts*), ale hlavní rozdíl mezi těmito dvěma notacemi je v tom, že diagramy aktivit jsou schopny zachytit paralelní chování [11 s. 117]. Proces zde modelujeme jako aktivitu složenou z uzlů a hran. Tyto diagramy se jako samostatná kategorie objevily až s příchodem UML 2, dříve byly součástí stavových modelů [1 s.286].



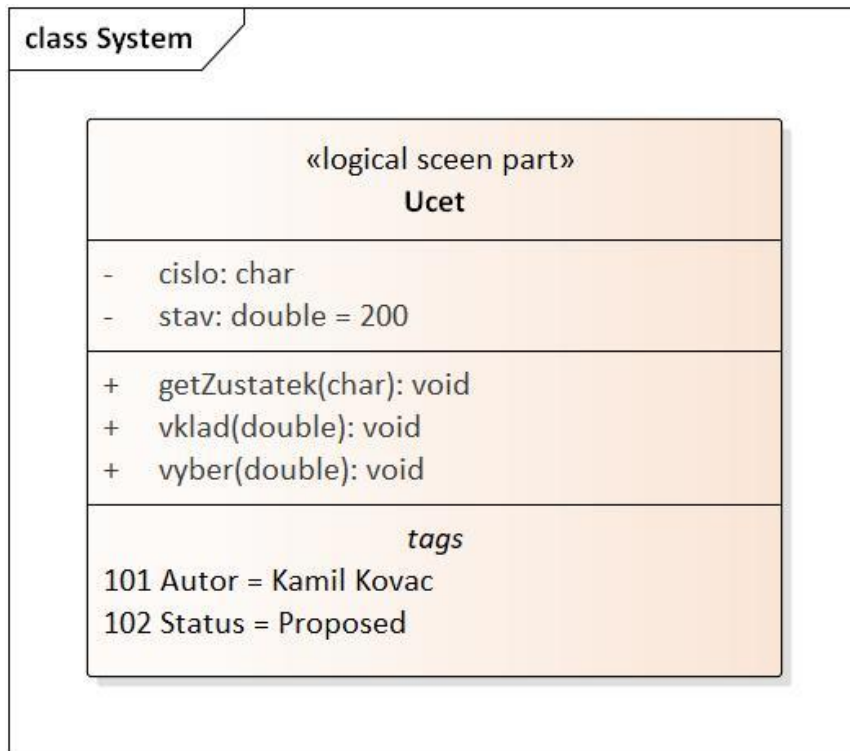
Obrázek 10: Ilustrativní příklad aktivity diagramu; zdroj: [autor]

Diagram tříd

Diagram tříd popisuje typy objektů v systému a různé druhy statických vztahů, které mezi nimi existují. Diagramy tříd ukazují i vlastnosti a operace třídy, a také omezení týkající se způsobu jakým jsou objekty spojovány [2].

Notace tříd

Úplná notace třídy je poměrně rozsáhlá, jak je vidět na Obr. 8 nicméně jediným povinným atributem je název třídy. Nepovinné atributy definují např. chování tříd (oddíl operací) nebo data (oddíl atributů).



Obrázek 11: Notace tříd; zdroj: [autor]

- *Oddíl názvu:* UML nedefinuje konvenci pro pojmenování tříd. Nejčastěji se pro názvy tříd používá konvence Upper Camel Case¹⁰ (někdy označováno jako Pascal Case).
- *Oddíl atributů:* Atribut je nositelem informací (dat) o objektu. Je definován jménem, datovým typem a viditelností.
- *Oddíl operací:* Operace třídy je funkce, která je navázaná na tuto třídu. Každá funkce má definovaný název, argumenty a návratové hodnoty.
- *Viditelnost:* Definuje přístup k členům třídy.
- *Typ:* Specifikuje datový typ atributu (např. string, integer, boolean nebo také double, char atp.)
- *Násobnost:* Podává informaci o tom, kolik objektů může být do vlastnosti umístěno.

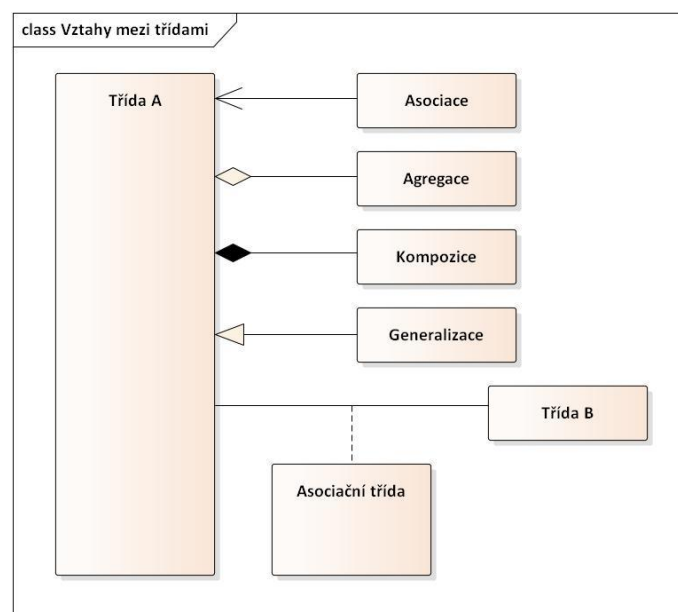
¹⁰ CamelCase označuje způsob psaní víceslovných spojení a nadpisů, kdy každé slovo začíná velkým písmenem a mezery jsou vynechány. Používá se často při programování pro názvy proměnných a funkcí.

Existují tyto možnosti – podle velikosti prvního písmene:

- lower-CamelCase – první písmeno je malé. Například funkce s názvem getName()
- upper-CamelCase (často také nazývaný PascalCase) – první písmeno je velké. Například BorderColor nebo ErrorLevel [32].

Vztahy mezi třídami:

- *Asociace*: Vyjadřuje přímý vztah mezi třídami.
- *Agregace*: Vyjadřuje vztah součást – celek
- *Kompozice*: Speciální případ agregace, který vyjadřuje silnější typ závislosti, kdy podřízený objekt nemůže existovat bez objektu nadřízeného.
- *Generalizace*: Vyjadřuje vztah mezi třídou a potomky, kdy podřízené třídy dědí po svém předkovi atributy a operace.
- *Asociační třídy*: Asociační třídy dovolují přiřadit atributy, operace a další vlastnosti k asociační vazbě.



Obrázek 12: Vztahy mezi třídami; zdroj: [autor]

Tvorbu diagramu tříd lze rozdělit na určité úrovně dle následujících modelů:

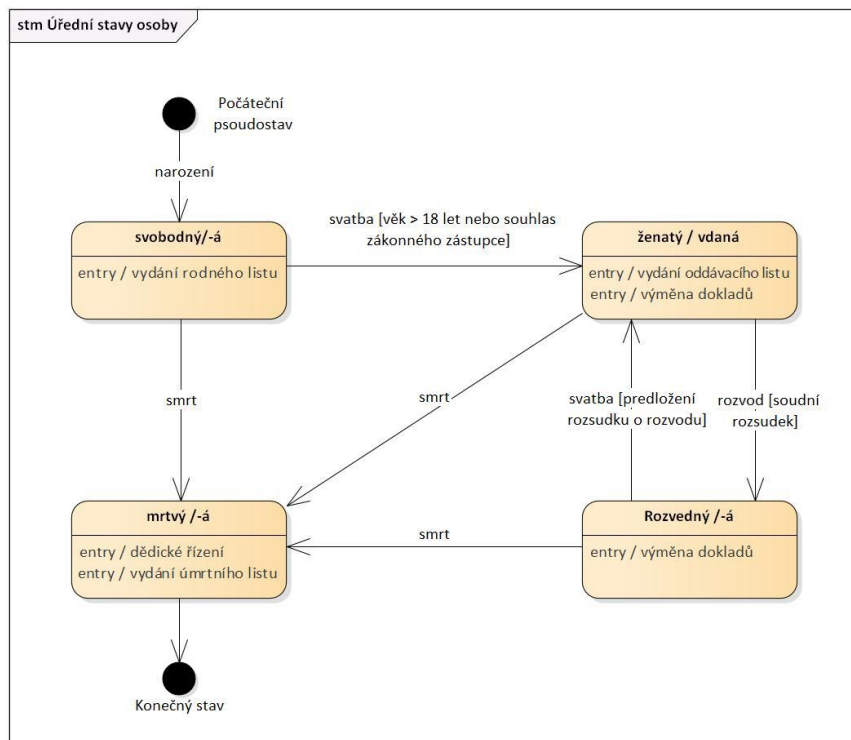
- *Doménový model* zobrazuje logickou strukturu navrhovaného systému. Je nezávislý na platformě, struktura obsahuje pouze třídy objektů, atributy a vztahy mezi třídami objektů [2, 15].
- *Návrhový model* doplňuje doménový model o datové typy atributů, metody a další doplňující třídy objektů (například výčetové třídy). Dále model obsahuje upřesnění vazeb mezi třídami objektů [7, 22].
- *Implementační model* se zaměřuje na podrobný popis struktury modelu. Tento model bývá určen koncovým vývojářům systému [17].

Stavový diagram

Stavový diagram je technikou k popisu chování systému, poskytuje pohled na dynamické chování objektu v průběhu času pomocí modelování životního cyklu dané třídy, jinými slovy modelují chování objektu napříč případy užití. Jednotlivé objekty reagují na události v systému a podle toho mění své stavy. Stav je množina hodnot atributů třídy [2, 16]. Dalšími prvky například jsou počáteční a koncové stavy (initial a final state) či rozhodovací uzly.

Prvky diagramu:

- *Stav (State)*: Reprezentace konkrétní situace, která nastala během životního cyklu objektu. Ke stavu mohou být přiřazeny události, které proběhnou během zpracování stavu. [16]
- *Přechod (Transition)*: Vyjadřuje změnu stavu u jednoho do jiného na základě určitých podmínek. [2]
- *Událost (Event)*: Specifikace něčeho významného, co nastane v určitém čase a prostoru. [1]



Obrázek 13: Ilustrativní příklad stavového diagramu; zdroj: [autor]

3.4 Core Banking systémy a podpůrné bankovní informační systémy

V následující kapitole jsou identifikovány a v teoretické rovině stručně popsány obecné moduly systémů, které jsou dnes běžně součástí banky. Kapitola je zpracována na základě autorovy zkušenosti z podnikové praxe, a na základě literatury uvedené v seznamu zdrojů.

3.4.1 Klientské portály

Do této kategorie spadají jak veřejně dostupné, tak privátní portály, primárně sloužící klientům k využívání poskytovaných služeb, ke správě jejich produktů nebo k získání (zakoupení) produktů nových. V prostředí banky tedy jde především o:

- Internetové bankovníctví
- Mobilní bankovníctví (*Smartbanking*)
- Veřejně dostupné webové stránky banky a další prezentace produktů např. prostřednictvím *microsites*¹¹.

3.4.2 Obslužné portály

Obslužné portály představují pobočkové systémy sloužící pro asistovaný prodej produktů, správu produktů nebo osobních údajů klienta nebo pracovníka banky. Zpravidla nejsou sjednoceny s klientskými portály s ohledem na fakt, že vznik pobočkových systémů výrazně předchází vzniku systémů klientských a vzhledem k odlišnostem procesů nebo např. možnostem identifikace klienta. Obslužné portály na rozdíl od klientských portálů musí být schopny pokrýt všechny případy užití které v bance vznikají při komplexní obsluze klienta.

Další příklady obslužných portálů:

- Intranet
- Systémy pro asistovaný prodej (pobočkový systém)

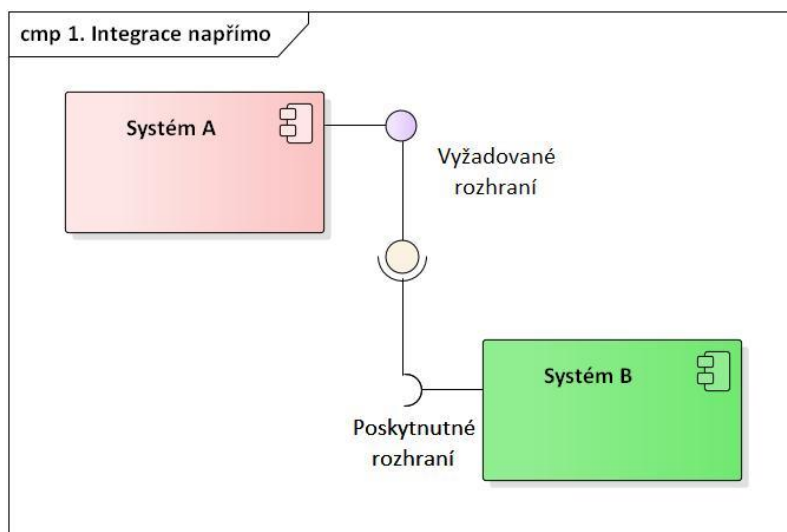
¹¹ Microsite, rovněž nazývaná minisite či weblet, je webdesignerským názvem pro speciální malý web, fungující jako doplněk hlavní webové prezentace. Rovněž ji lze označit za prostředek marketingu [23].

3.4.3 Integrovaná vrstva

V dnešní době jsou informační systémy většiny korporátních společností rozděleny na více subsystémů, kdy každý z nich řeší určitou množinu funkcí – např. sklad, finance, lidské zdroje atp. Integrace systémů je založena na principu sdílení dat mezi různými subsystémy jednoho informačního celku. Systémová integrace je rozsáhlé téma, které přesahuje rozsah této diplomové práce. Podrobně se oblasti integrace věnuje např. prof. Ing. Jiří Voříšek, CSc. v knize *Strategické řízení informačního systému a systémová integrace* (2006). Na systémovou integraci lze nahlížet z více úhlů, jedním z nich je např. stupeň rozvinutosti integrace, která často kopíruje vývoj informačních systémů ve společnosti [24].

Integrace napřímo

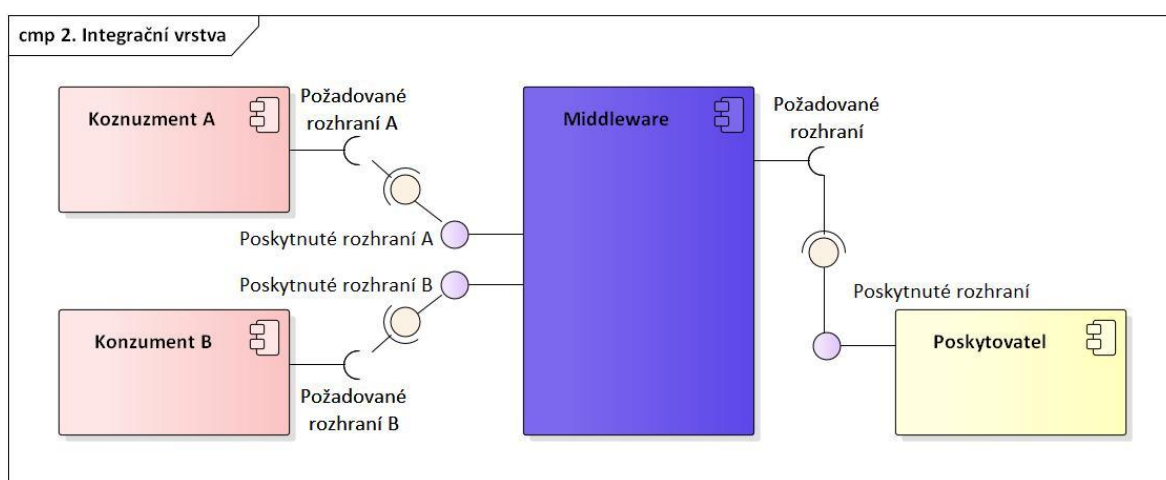
Často také označována jako „spaghetti model“. Jde o velmi jednoduchou architekturu, kdy máme dva a více systémů. Jeden ze systémů vystavuje rozhraní pro druhý systém, který se prvního dotazuje předem definovaným způsobem. Výhodou je počáteční jednoduchost na analýzu a implantaci. Nevýhody tohoto přístupu vznikají s rostoucím počtem integrovaných systémů. Tím, jak v čase narůstá počet systémů, které si mezi sebou vyměňují data, může se lišit způsob komunikace nebo formát dat. Je také problematické vyměnit jeden systém za druhý pro vyšší počet rozhraní.



Obrázek 14: Integrace napřímo; zdroj: [autor]

Integrační vrstva

Je technologicko-architektonická vrstva (často také nazývána jako middleware nebo servisní vrstva), která je vložena mezi konzumenty a poskytovatele dat. Konzument (systém, který požaduje data) je od poskytovatelů dat odstíněn – jinými slovy konzumenta nezajímá v jakých a jakých systémech, ať už interních nebo externích, jsou data uložena. Konzument data požaduje po integrační vrstvě. Je určen formát dat a způsob doručení, získání a případnou transformaci dat má na starosti právě integrační vrstva v rámci tzv. orchestrace.



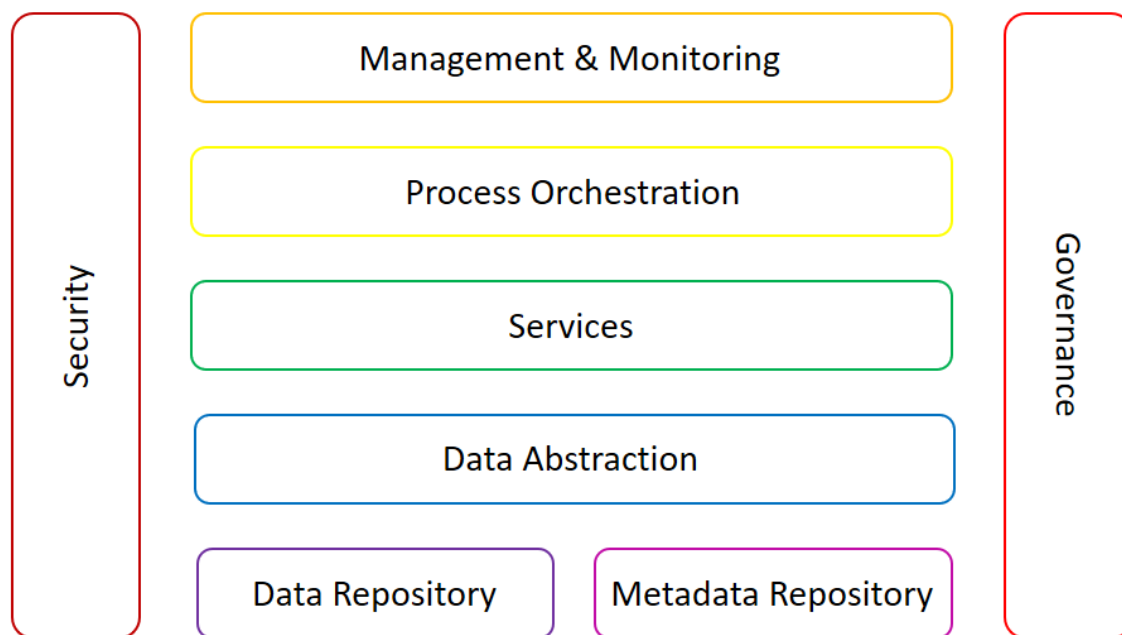
Obrázek 15: Integrační vrstva; zdroj: [autor]

Servisně orientovaná architektura

SOA je architektonický styl a způsob analýzy, návrhu, integrace a údržby informačních systémů založených na službách. Dle architektury SOA je služba autonomní softwarovou jednotkou, která pomocí svých poskytovaných rozhraní nabízí jednu nebo více implementovaných funkcí. Rozhraní služby popisuje, jaké typy požadavků svého okolí je služba schopna zpracovat, a jak vypadají odpovědi služby na tyto požadavky, tj. jaké funkce nabízí služba svému okolí, jaká je struktura vstupních a výstupních parametrů či návratových hodnot těchto funkcí, jaký je způsob komunikace se službou atp.

Konkrétní implementace SOA pak realizuje konkrétní informační systém jako specifickou kompozici služeb. Při specifikaci rozhraní a následné kompozici služeb SOA by mělo platit několik principů, např. abstrakce (rozhraní je kompletním popisem služby), volná vazba

(okolí využívá službu pouze prostřednictvím zveřejněného rozhraní), znouvopoužitelnost (služba je navržena tak, aby mohla být použita opakovaně v různém kontextu), bezstavovost (stejně volání služby by mělo vracet vždy stejné výsledky, tj. služba by neměla udržovat stavovou informaci) [25].



Obrázek 16: Servisně orientovaná architektura; zdroj: volně dle [25]

3.4.4 ERP – Plánování podnikových zdrojů

Za obchodní a podpůrné systémy – neboli ERP je označován komplexní modulární informační systém, který zajišťuje širokou škálu firemní agendy a procesů. ERP je účinný nástroj, který je schopen efektivního plánování a řízení všech podnikových zdrojů ve výrobním nebo distribučním podniku či v podniku zaměřeném na služby. Tyto zdroje jsou nezbytné k přijetí a realizaci objednávky zákazníka včetně následného dodání a fakturace [20]. Významné ERP systémy:

- CRM,
- SCM
- Software pro řízení rizik
- Účetní systém
- Call center software

3.4.5 Bankovní systémy

Za hlavní bankovní systémy lze označovat ty systémy, bez kterých banka jako taková nemůže fungovat. Ačkoli systémy jsou v dnešní době informačních technologií plně elektronizovány a v praxi se jedná o jednotlivé moduly komplexního řešení, lze na problematiku nahlížet s jistou mírou abstrakce, která nám pomůže vystihnout jádro myšlenky. Pak na hlavní bankovní systémy můžeme nahlížet takto:

- Systém pro správu účtů a operací na účtech¹²
- Systém pro správu dispozičních oprávnění k účtům

V případě, kdy banka poskytuje také úvěrové služby, pak lze to do hlavních bankovních systémů také zařadit systémy které prověřují bonitu klientů a rizikovost poskytnutí úvěru, od čehož se dále odvíjí úroková míra. Kompletní krabicové řešení pro řízení úvěrových rizik poskytuje například firma CRIF, která mimo jiné provozuje také nebankovní registry.

3.4.6 Reportingové systémy

V širším pojetí lze reporting vymezit jako komplexní systém zpravodajství poskytující mimo jiné také externím zainteresovaným orgánům, skupinám i jednotlivcům, informace o všech aktivitách podniku, které se jich mohou dotýkat [19]. Reporting je obvykle chápán jako samostatná součást informačního systému podniku, která zahrnuje výběr, zpracování, formální úpravu a distribuci informací o podniku určených pro nejrůznější skupiny adresátů (uživatelů). Protože uživatelů může být mnoho, a navíc s velmi různými požadavky, klade to na obsahovou i formální stránku reportingu značné nároky. Uživatele reportů lze v podstatě členit do dvou skupin:

- *Interní uživatelé:* Vlastníci a management na různých řídicích pozicích, představenstvo a dozorčí rada. V širším pojetí jde o adresáty, kteří mají rozhodovací pravomoci a jsou odpovědní za výsledky podnikových činností.
- *Externí adresáti:* Široký okruh oprávněných kontrolních orgánů, zájmových skupin i jednotlivců.

¹² Příkladem může být např. systém PROFILE od společnosti Profile Software, který nabízí v balíkovém řešení software obsluhu a kompletní řízení bankovních účtů.

3.4.7 Systémy pro asset management

V rámci světa IT je pojem asset management (neboli systém pro správu aktiv) řeší dva důležité aspekty provozu IT oddělení – používání softwaru v souladu s licenční politikou a nákladovou stránku. Obecně systém pro správu aktiv rozumíme systémy pro řízení a správu provozních zdrojů organizace, které umožňují firmám s rozsáhlými investičními celky a kontinuálními provozními procesy hospodařit s vlastním hmotným majetkem tak, aby vytvářely a zajišťovaly svůj ekonomický růst. V konečném důsledku lze za asset management systém považovat i systémy pro rezervaci zasedacích místností nebo správu vozového parku.

3.4.8 Tiskové systémy

Banka, tak jako každá společnost, komunikuje s řadou dalších subjektů, mohou to být např. zákazníci, obchodní partneři, dodavatelé nebo státní správa. Ať už tato komunikace probíhá elektronicky nebo v tištěné podobě, jsou dokumenty zpravidla standardizované – založené na příslušné šabloně. Tyto šablony nedefinují jen strukturu dokumentu (vizuální podobu), ale také přístup k dokumentům – např. proces schvalování nebo archivace dokumentu. ISTS systémy slouží jako komplexní řešení centrální správy těchto šablon. Na žádost dalších systému obstarávají generaci dokumentu, jeho tisk a v závislosti na implementaci IS také např. jeho archivaci v systému pro správu dokumentů¹³.

Tisk dokumentu tedy může např. probíhat následujícím způsobem: systém vyžadující tisk dokumentu zašle požadavek na tisk dokumentu ISTS s konkrétními daty, identifikátorem šablony a klíčem určujícím způsob tisku. ISTS naplní příslušnou šablonu daty, dokument vygeneruje a v závislosti na požadavku způsobu tisku např. dokument uloží do DMS systému a vrátí zpět identifikátor vytištěného dokumentu.

¹³ Tzv. DMS (*Document Management System*) systémy – systémy pro správu oběhu dokumentů.

3.4.9 Bankovní a nebankovní registry

Úvěrové registry, jsou využívány jako jeden z nástrojů pro řízení rizik souvisejících s poskytováním bankovních úvěrů. V zájmu každé společnosti je snižování úvěrových rizik, a tím možných ztrát. Na území české republiky působí celá řada bankovních i nebankovních registrů, které jsou koncipované jako pozitivní i negativní. V pozitivních registrech může být vedena i osoba, která svým závazkům dostává včas, klient si tím buduje kvalitní úvěrovou historii, na základě, níž je poté z pohledu bank považován za důvěryhodného partnera. Pokud jméno klienta figuruje v negativním registru, může tato skutečnost klientovi zkomplikovat získání úvěrů.

K nejčastěji využívaným registrům patří:

Centrální registr úvěrů (CRÚ)

Centrální registr úvěrů je informační systém, který soustřeďuje informace o úvěrových závazcích fyzických osob podnikatelů a právnických osob a umožňuje operativní výměnu těchto informací mezi účastníky CRÚ. Vzhledem k uvedenému zaměření nejsou v databázi CRÚ evidovány spotřebitelské úvěry fyzických osob, hypoteční úvěry fyzických osob, ručitelé závazky klientů, údaje o depozitních účtech (běžné účty bez povoleného debetu, spořicí, termínové účty). Účastníkem CRÚ jsou všechny banky a pobočky zahraničních bank působících na území České republiky a další osoby, stanoví-li tak zvláštní zákon. Povinností účastníka CRÚ je provádění pravidelné měsíční aktualizace databáze CRÚ. Přístup k informacím je umožněn jednotlivým účastníkům a České národní bance v rozsahu potřebném pro zajištění provozu CRÚ [22]. V CRÚ jsou registrovány negativní i pozitivní úvěrové informace o dlužnících a je koncipován jako registr úplný.

Bankovní registr klientských informací (BRKI)

Bankovní registr klientských informací provozuje databázi sdílených klientských informací. Prostřednictvím BRKI mají banky možnost prověřit schopnost potenciálních klientů dostát svým budoucím závazkům. Bankovní registr klientských informací se mohou účastnit na základě smlouvy organizace které nejsou v nucené správě nebo v likvidaci. Informace obsažené v BRKI jsou souborem informací, které si banky navzájem poskytují o smluvních

vztazích mezi obchodními bankami a jejich klienty. Obsahuje negativní i pozitivní informace o klientech, poskytuje informace o současném zadlužení klienta, ale i historické informace. Získání informací z registru je podmíněno souhlasem žadatele o úvěr. V současné době BRKI používají téměř všechny obchodní banky v České republice. Informace obsažené v BRKI vypovídají o bonitě a důvěryhodnosti klienta. Dále pak například osobní údaje klienta. Podle současně platné právní úpravy (zákon č. 21/1992 Sb., zákon č. 101/2000 Sb.) je možné rodná čísla klientů zpracovávat bez jejich souhlasu. Rodné číslo je používáno k přesné identifikaci klientů, tzn., nemůže dojít k záměně s jinou osobou uvedenou v databázi BRKI [21].

Nebankovní registr clientských informací (NRKI)

Nebankovní registr clientských informací obsahuje data o fyzických osobách, fyzických osobách podnikatelích a právnických osobách a napomáhá ke zjednodušení a zprůhlednění vztahů mezi věřitelem a dlužníkem. Zprostředkovává komunikaci mezi věřitelskými společnostmi k vzájemné výměně určitých údajů o svých klientech (především o bonitě), aby úvěrové společnosti lépe předcházely podvodům ze strany svých klientů a aby se chránily proti riziku spojeným s nesolventností klientů. NRKI obsahuje ve své databázi následující údaje: identifikační údaje klientů (např. jméno, příjmení, rodné číslo atd.), informace o smluvních vztazích mezi věřiteli a jejich klienty, o bonitě, důvěryhodnosti a platební morálce klientů.

SOLUS

Sdružení Solus je zájmové sdružení právnických osob na ochranu leasingu, úvěrů a dalších vybraných služeb. Tyto služby jsou poskytovány spotřebitelům, podnikatelům a právnickým osobám se sídlem v Praze. Registr obsahuje negativní informace o fyzických osobách, fyzických osobách podnikatelích a právnických osobách, které se dostaly do problémů s platební disciplínou nebo nemají zájem splácet své pohledávky, alespoň vůči jednomu členovi sdružení. U fyzických osob podnikatelích a právnických osob registr poskytuje informace o druhu a rozsahu porušení smluvních povinností, tj. existence 2 pohledávek po splatnosti, anebo pohledávky, která je déle jak 30 dnů po splatnosti. Registr tímto umožňuje členům efektivněji řídit svá rizika u služeb, které poskytují.

TelcoScore

Služba TelcoScore poskytuje predikce chování zákazníka ve finančních službách na základě telekomunikačních dat sebraných za 3 měsíce. Dodavateli score jsou tři největší mobilní operátoři: O2 Czech Republic a.s., T-Mobile Czech Republic a.s. a Vodafone Czech Republic a.s. Výpis z registru je poskytnut na základě telefonního čísla nebo rodného čísla (v případě hodnocení zákazníka – fyzické osoby). Výpočet je realizován na základě smluvních, provozních a kreditních dat zákazníka. Používání TelcoScore je vždy podmíněno souhlasem zákazníka. Registr je každý den aktualizován a je dostupný 7/24. Pro využití služby je vždy nutné získat prokazatelný souhlas zákazníka s ověřením score, který je vyžadován u všech subjektů a bude kontrolován provozovatelem platformy, vše v souladu s GDPR compliance [27].

Aktuálně je využívání registru považováno za kontroverzní z důvodu nezveřejněné metodiky výpočtu skóre a dat, které jsou zohledněny ve výpočtu. Například společnost Equa Bank je za podmiňování poskytnutí úvěru udělením souhlasu s dotazem do TelcoScore registru dlouhodobě kritizována a v roce 2018 za tuto skutečnost získala anticenu Velkého bratra [29].

3.4.10 Mezibankovní komunikační protokoly

Tyto protokoly umožňují přijímat a odesílat informace týkající se finančních transakcí v mezinárodně akceptovatelné standardizované a zabezpečené formě. Bankovní instituce jsou také napojeny na další organizace a clearingová centra národních bank pomocí API¹⁴.

Příklady mezibankovních protokolů:

- SWIFT
- EBICS
- SEPA

¹⁴ API (zkratka pro Application Programming Interface) označuje v informatice rozhraní pro programování aplikací. Tento termín používá softwarové inženýrství. Jde o sbírku procedur, funkcí, tříd či protokolů nějaké knihovny (ale třeba i jiného programu nebo jádra operačního systému), které může programátor využívat. API určuje, jakým způsobem jsou funkce knihovny volány ze zdrojového kódu programu. [26]

3.4.11 Systémy monitorující funkčnost systémů

Tyto systémy mají za úkol odhalit případný výpadek činnosti dalších informačních systémů. V předem definovaných intervalech provolají jednotlivé systémy, a pokud není obdržena předem definovaná odpověď, pak je kontaktován správce systému.

3.5 Vybrané legislativní opatření

3.5.1 AML – Opatření proti legalizaci výnosů z trestné činnosti

Banky jsou při své činnosti povinné řídit se mimo jiné právními předpisy které regulují oblast AML¹⁵ – zejména zákon 253/2008 Sb., o některých opatřeních proti legalizaci výnosů z trestné činnosti a financování terorismu. Tyto opatření lze rozdělit na preventivní – postupy, které vedou k minimalizaci rizika legalizace výnosů pocházejících z trestné činnosti a reaktivní – postupy bank následující po zachycení podezřelého obchodu.

Mezi preventivní opatření patří např. kontrola transakcí a identifikace klienta, archivace dokumentace související s obchodním vztahem a údajů o provedených obchodech, disponentech účtů, údajů doprovázejících platby, identifikace doručitelů platebních příkazů, školení zaměstnanců zaměřená na rozpoznání podezřelých obchodů ve smyslu zákona č. 253/2008 a další. Kontroly jsou zpravidla v gesci specializovaného oddělení společnosti, které je k této činnosti vybaveno příslušnými nástroji a pravomocemi. Z nástrojů lze jmenovat například software vyhodnocující transakce dle zadaných parametrů. Tyto parametry definuje pracovník oddělení – specialista, který je na základě svých zkušeností schopen posoudit potenciál rizikovosti transakce. Dále pak například hlášení zaměstnanců přicházejících do styku s klienty, kteří jsou školeni tak, aby byli schopni identifikovat podezřelé chování klientů nebo falsifikáty identifikačních dokladů.

¹⁵ AML (*Anti Money Laundering*) opatření proti praní špinavých peněz

3.5.2 GDPR – Obecné nařízení o ochraně osobních údajů

Obecné nařízení představuje nový právní rámec ochrany osobních údajů v evropském prostoru, které bude od 25. května 2018 přímo stanovovat pravidla pro zpracování osobních údajů, včetně práv subjektu údajů (fyzické osoby). V českém právním prostředí tak obecné nařízení od 25. května 2018 nahradí zákon č. 101/2000 Sb., o ochraně osobních údajů a o změně některých zákonů, resp. zákon o ochraně osobních údajů po jeho novele bude již upravovat jen některé aspekty týkající se Úřadu pro ochranu osobních údajů (např. jeho ustavení, organizaci atd.) a některé dílčí záležitosti nutné k dotvoření celého rámce ochrany osobních údajů, které nejsou obecným nařízením upraveny či je jím umožněno je upravit na vnitrostátní úrovni (nebo dokonce i stanoveno, že mají být upraveny) [30].

Nařízení nově zavádí princip tzv. zodpovědnosti, který spočívá v povinnosti správců a zpracovatelů údajů bez ohledu na jejich velikost nebo počet zaměstnanců zavést technická, organizační a procesní opatření za účelem prokázání souladu s principy GDPR. Uplatnění principu zodpovědnosti bude představovat pro podnikatele nemalé časové a finanční investice. Ty se budou týkat zejména těchto oblastí [31]:

- implementace záměrné a nezbytné ochrany dat
- vypracování posouzení vlivu na ochranu osobních údajů
- jmenování pověřence pro ochranu osobních údajů
- zavedení tzv. pseudo anonymizace osobních údajů
- vedení záznamů o činnostech zpracování
- konzultace s dozorovým orgánem před samotným zpracováním osobních údajů

3.5.3 PEP – Politicky exponované osoby

Jako politicky exponovaná osoba (někdy také PEP) se v bankovní terminologii označuje fyzická osoba ve významné veřejné funkci s celostátním nebo regionálním významem, případně blízký příbuzný či obchodní partner takového funkcionáře. Termín je používán také v legislativě zaměřené proti praní špinavých peněz (viz kapitola 3.5.1) na úrovni mezinárodní organizace FATF a jejích členských zemích včetně zemí Evropské unie.

Přesná definice dle zák. 253/2008 Sb., o některých opatřeních proti legalizaci z trestné činnosti a financování terorismu, ve znění pozdějších zákonů, § 4, odst. 5, politicky definované osoby [28]:

- a) fyzická osoba, která je nebo byla ve významné veřejné funkci s celostátním nebo regionálním významem

- b) fyzická osoba, která je
 - 1. osobou blízkou k osobě uvedené v písmeně a),
 - 2. společníkem nebo skutečným majitelem stejné právnické osoby, popřípadě svěřenského fondu nebo jiného právního uspořádání bez právní osobnosti, jako osoba uvedená v písmenu a) nebo je v jakémkoli jiném blízkém podnikatelském vztahu s osobou uvedenou v písmenu a) nebo
 - 3. skutečným majitelem právnické osoby, popřípadě svěřenského fondu nebo jiného právního uspořádání bez právní osobnosti, které byly vytvořeny ve prospěch osoby uvedené v písmenu a).

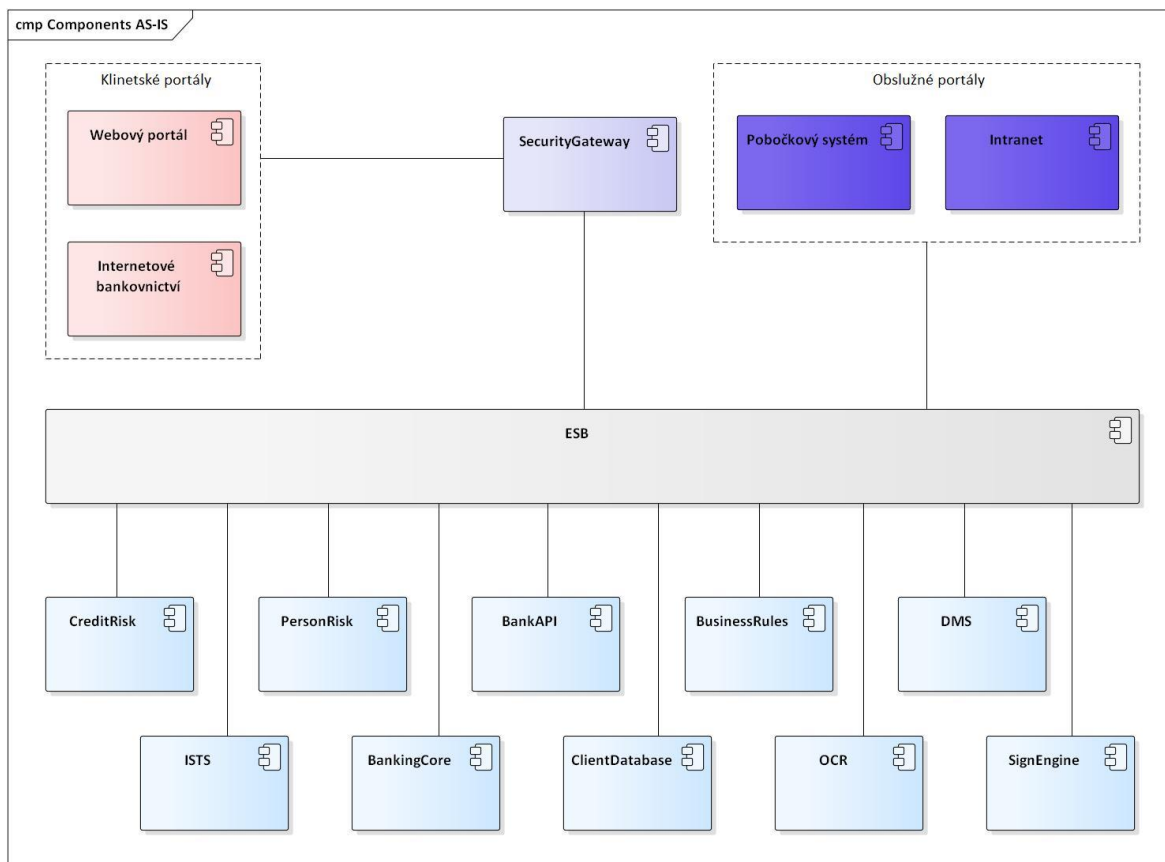
4 Vlastní práce

Obsahem praktické části diplomové práce je návrh IS sloužícího jako e-shop s produkty a službami poskytovanými bankou. Projekt vývoje informačního systému umožňujícího prodej bankovních produktů včetně banko-pojištění svým rozsahem a komplexitou překračuje doporučený rozsah této diplomové práce. Vzhledem k této skutečnosti bude dále uvažován pouze jeden vybraný produkt – neúčelový úvěr, z důvodu schopnosti generovat nejvyšší zisk z produktů, které budou na tomto kanále nabízeny.

V následujících kapitolách je navrhován systém pro *fiktivní* bankovní společnost. Specifikace požadavku a datový model jsou definovány na základě autorovy osobní zkušenosti s účastí na obdobném projektu v obchodní bance poskytující služby na českém trhu.

4.1 Současný stav a popis systémů

V této kapitole je zevrubně charakterizováno prostředí informačních systémů společnosti, je zde uveden výčet vybraných stávajících komponent (systémů) provozovaných ve společnosti, se kterými bude navrhovaný e-shop komunikovat pomocí webových služeb založených na SOAP. SOAP využívá značkovací jazyk XML s XSD notací pro definici datového toku mezi aplikacemi. Existuje vícero integračních patternů popisujících komunikaci mezi systémy. Dvě hlavní skupiny jsou asynchronní a synchronní. Pro zajištění runtime nezávislosti systému se často využívá asynchronní integrační pattern. V případě synchronního způsobu komunikace jedna aplikace pošle v XML zprávě požadavek druhé aplikaci, ta požadavek zpracuje a výsledek zašle zpět aplikaci, která zaslala požadavek. Asynchronní komunikace probíhá tak, že systém zašle zprávu do fronty a dále se již o zprávu nezajímá. Druhý systém (systémy) mohou být na tuto frontu napojeny a mohou využívat zpracovávat zprávy, které jsou pro ně určené.



Obrázek 17: Přehled systémů; zdroj: [autor]

CreditRisk

Systém slouží k získání úvěrovému scoringu. Skládá se z více subsystémů, zajišťuje orchestraci úvěrového scoringu a iniciuje generaci smluvní dokumentace k úvěrovým produktům. CreditRisk komunikuje s vícero systémy a využívá data z bankovních a nebankovních registrů. V případě uživatele, který není klientem, systém spolupracuje s OCR, které vytěžuje dokumenty dokládající příjem pro účely úvěrového scoringu.

Využívaná poskytovaná rozhraní:

- *vrátPředschválenéLimity*: vrací 0..n předdefinovaných variant pro daný produkt a klienta, které jsou schvalovány ve zjednodušeném procesu
- *vrátSocring*: zakládá požadavek v systému CreditRisk, vrací chybu, pokud už má uživatel otevřenou jinou žádost, nebo vrací závaznou nabídku banky
- *vrátDokumentaci*: spouští proces tvorby smluvní dokumentace
- *uzavřiProdukt*: spouští proces zřízení produktů nebo ruší žádost

PersonRisk

Systém provádí rizikový scoring osoby na základě interních i externích dat, které jsou o dané osobě k dispozici. Tyto údaje slouží jako vstup do subsystému vyhodnocujícího rizikový profil potenciálního klienta na základě interně definovaných pravidel.

Využívaná poskytovaná rozhraní:

- *vrat'RizikovýProfil*: služba vrací rizikový profil klienta (odpovědí je hodnota z číselníku)

BusinessRules

Systém slouží pro správu produktů a obsahuje interní pravidla na ně navázaná, v rámci e-shopu je systém využíván k posouzení, zdali klient splňuje požadavky pro vlastnění produktu. Např. pokud je podmínkou získání slevy na studentském účtu věk klienta nižší nebo roven 26, systém toto pravidlo realizuje.

Využívaná poskytovaná rozhraní:

- *vyhodnot'Zpusobilost*: na základě klientského čísla nebo zaslaných údajů osoby a identifikátoru produktu kontroluje způsobilost klienta/uživatele vlastnit produkt, vrací booleovskou hodnotu

DMS

Systém pro správu dokumentů. Spravuje veškeré elektronické dokumenty existující v rámci společnosti.

Využívaná poskytovaná rozhraní:

- *vrat'Dokument*: vrací dokument na základě reference
- *modifikujDokument*: provádí změny atributů dokumentu
- *uloz'Dokument*: uloží dokument a vrací jeho referenci

ISTS

Tiskový systém. Obsahuje šablony všech používaných dokumentů. Na základě identifikátoru šablony a zaslaných dat vyžadovaných příslušnou šablonou vrací vytvořený dokument.

Využívaná poskytovaná rozhraní:

- *vytiskniDokument*: vrací referenci dokumentu v DMS

OCR

Systém pro optické rozpoznávání znaků založený na neurální síti. Systém umožňuje rozpoznat různé druhy elektronických dokumentů obsahující znaky psaného jazyka, rozpoznat kontext dat a uložit je do databáze – ve zkratce systém pozná, že naskenovaný dokument je občanský průkaz, protože odpovídá definované šabloně (např. se v určité oblasti se vyskytuje slovní spojení „občanský průkaz“), vytěží požadovaná data a uloží je do databáze.

Využívaná poskytovaná rozhraní:

- *vytežDokument*: vrací identifikátor vytěženého záznamu v databázi nebo chybu

BankAPI

Rozhraní, které banky poskytují jiným subjektům, sloužící k různým účelům. V kontextu e-shopu je využíváno rozhraní umožňující vyhledání určité transakce na účtu a získat její detailní informace.

Využívané poskytované rozhraní:

- *vratDetailTransakce*: na základě variabilního symbolu provede dotaz a vrátí detail transakce – jméno a příjmení vlastníka účtu

ClientDatabase

Klientská databáze. Obsahuje záznamy osobních dat týkající se klientů.

Využívané poskytované rozhraní:

- *vraťKlientskáData*: na základě klientského id vrací všechny dostupné záznamy osoby
- *vytvořKlienta*: zakládá osobu
- *modifikujKlienta*: modifikuje osobu
- *vraťSouhlasy*: na základě klientského id vrací souhlasy udělené klientem
- *modifikujSouhlas*: nastavuje status souhlasu pro klienta
- *modifikujStatus*: modifikuje status klienta
- *ověřKlientskéÚdaje*: na základě klientského čísla a hesla ověřuje identitu klienta, vrací klientské číslo při úspěšném ověření

SignEngine

System generující podpisové SMS a umožňují elektronický podpis dokumentu. V první fázi je dokument podepsán certifikátem banky a následně SMS podpis klientem.

Využívané poskytované rozhraní:

- *podepišDokument*: startuje proces podpisu dokumentu, na základě zaslání linku na dokument v DMS
- *odešliSMS*: systém vygeneruje a odešle SMS kód
- *ověřSMS*: systém ověří zadanou SMS a v případě shody uloží referenci k dokumentu

BankingCore

System zakládá produkty – vytváří instance účtů a obsluhuje transakce prováděné s účty – odchozí platby, příchozí platby, trvalé příkazy atp.

Využívaná poskytovaná rozhraní:

- *založÚčet*: na základě zaslání parametrů zřizuje příslušný účet (např. spořicí, nebo technický účet pro splácené úvěru).
- *založKartu*: iniciuje proces založení karty, včetně jejího vydání
- *založKontokorent*: iniciuje proces založení kontokorentu k příslušnému účtu
- *načerpejÚvěr*: převede finanční prostředky na příslušný účet

4.2 Výchozí předpoklady

Podstatné systémy, se kterými musí navrhovaný systém spolupracovat, již byly stručně charakterizovány v kapitole 4.1, nicméně pro lepší pochopení komplexnosti řešené domény (online prodeje bankovních produktů, v tomto případě konkrétně úvěru) považuji za přínosné detailněji představit důležité pojmy z této oblasti a nastínit řešení online autentizace uživatele, která byla stanovena jako jeden z cílů praktické části.

4.2.1 Nabízené produkty

E-shop bude sloužit k obsluze pouze retailové klientely. Pojem retailové bankovníctví můžeme vymezit jako bankovníctví maloobchodní. Jinými slovy můžeme říci, že se jedná o bankovníctví pro občany – nepodnikatele. Na rozdíl od korporátního bankovníctví, které je zaměřeno na podnikatele a firmy. Bankovní služby můžeme dělit dle několika kritérií. Jedním z možných přístupů je např. třídění bankovních produktů dle jejich odrazu v bilanci banky [12]:

- a) aktivní: banka zde vystupuje v pozici věřitele, bance vznikají pohledávky
- b) pasivní: banka zde vystupuje v pozici dlužníka, bance vznikají závazky,
- c) neutrální: banka nevystupuje ani v pozici dlužníka ani v pozici věřitele, bance nevznikají závazky ani pohledávka

Na nově vznikajícím online kanále e-shop budou prodávány následující primární produkty banky:

Tabulka 1: Nabízené produkty na e-shopu; zdroj: [autor]

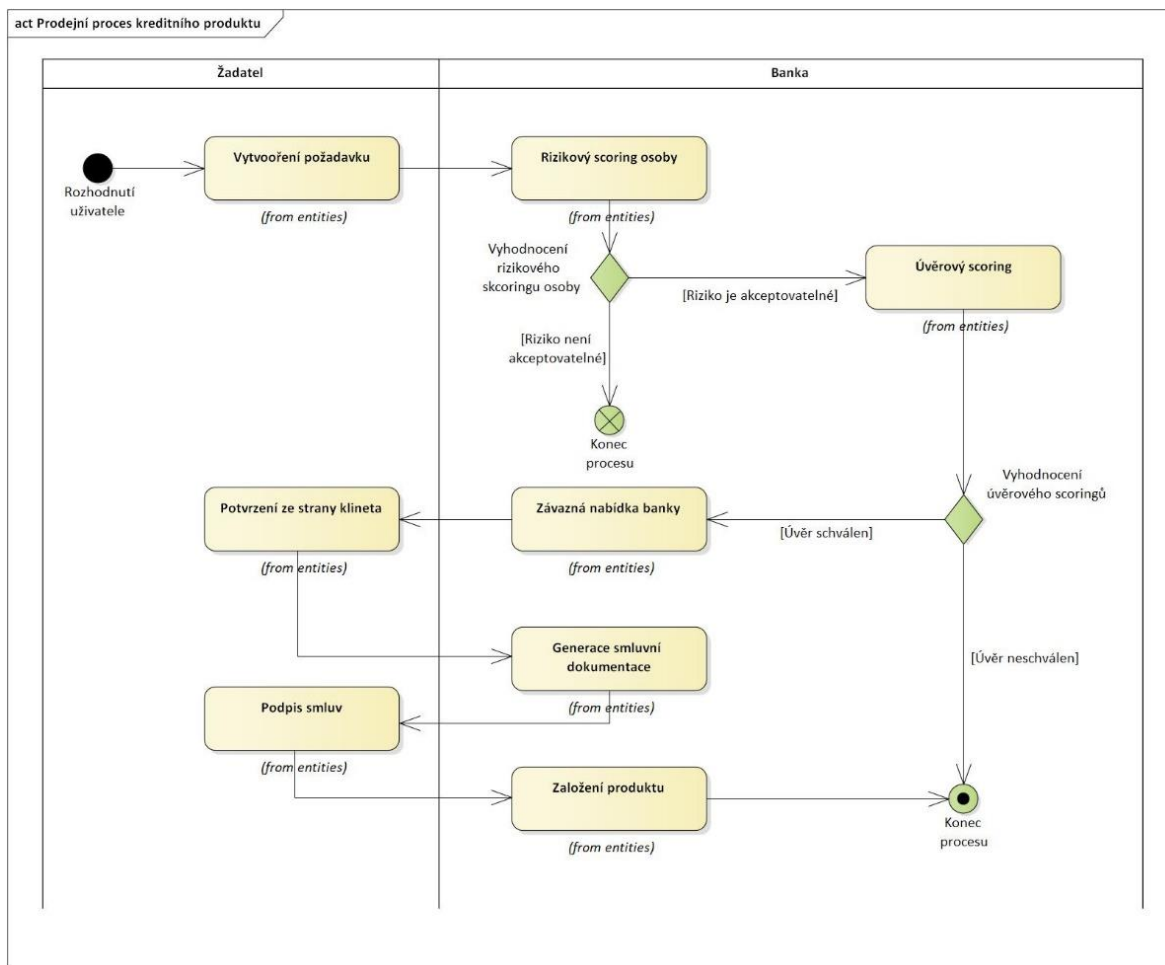
Název produktu:	Marketingový název	Skupina:	Produktový kód
Spotřebitelský úvěr neúčelový	<i>Půjčka bez starostí</i>	Kreditní produkty	C01
Spotřebitelský úvěr účelový	<i>Půjčka na váš projekt</i>	Kreditní produkty	C02
Kreditní karta	<i>Kreditka</i>	Kreditní produkty	C03
Kontokorent	<i>Kontokorent</i>	Kreditní produkty	C03
Běžný účet	<i>Konto plus</i>	Depozitní produkty	D01
Spořicí účet	<i>Chytré spoření</i>	Depozitní produkty	D02

Ke kreditním produktům bude dále nabízeno jako doplňkový cross-sale produkt banko-pojištění schopnosti splácet a ke kreditní kartě cestovní pojištění.

4.2.2 Specifikace prodejního procesu a některých interních pravidel

Proces prodeje kreditních produktů

Prodejní proces kreditních produktů je rozdělen do 5 fází. V první fázi probíhá *konfigurace produktu*, uživatel zvolí limit, délku splácení a případné další parametry. Následuje fáze *osobní údaje*, kde klient doplní nezbytné osobní údaje a případně udělí souhlasy vztahující se k úvěrovým registrům. Potvrzením této fáze vzniká požadavek – žádost. Následuje vyhodnocení objednávky, které zahrnuje rizikový a úvěrový scoring, a na základě těchto údajů je žádost posouzena. Ve fázi *nabídky* je uživateli prezentován výsledek, uživatel může nabídku akceptovat nebo ji zrušit. V případě akceptace provede doplnění konfigurace produktu a zvolí metodu podpisu. Následuje fáze *identifikace*, která je v případě přihlášeného klienta vynechána z důvodu nadbytečnosti. V této fázi neautentizovaný uživatel nahraje fotokopii identifikačního průkazu. V závěrečné páté fázi *podpis dokumentace* je uživateli umožněn náhled na smluvní dokumenty a jiné přílohy. V případě podpisu na pobočce nebo pomocí kurýra tato stránka obsahuje poučení o následných krocích a pro online podpis je umožněna autorizace dokumentů.



Obrázek 18: Aktivita diagram procesu prodeje kreditních produktů; zdroj: [autor]

Rizikový profil osoby

Rizikový scoring osoby provádí systém *PersonRisk*. Systém vyhodnocuje pravidla AML (viz kap. 3.5.1), ověřuje, zdali klient není na seznamu hledaných osob, osob, na něž jsou uvaleny sankce nebo nefiguruje na bankovních černých listinách. Na prodejním kanálu e-shop dále banka neumožňuje zřizovat produkty osobám se zvláštním vztahem k bankám a politicky exponovaným osobám (viz kap. 3.5.3).

Vyhodnocení rizikového profilu má následující varianty výsledku:

- Osoba je přípustná
- Osoba je nepřípustná
- Osoba je riziková

Úvěrový scoring osoby

Úvěrový scoring provádí systém *CreditRisk*, který se skládá z dalších subsystémů. Samotný systém slouží pro orchestraci procesu. Subsystémy, které stojí za povšimnutí jsou následující:

- *INTEREST konektor*: Subsystém, který na základě vnitřních pravidel definovaných risk oddělením banky zpracovává vstupní parametry klienta. Výstupem je výše úrokové míry, za kterou je banka ochotna poskytnout úvěr, popř. další podmínky poskytnutí úvěru včetně možnosti podpisu.
- *RPSN konektor*: Subsystém, který vypočítá na základě vstupních parametrů (úroková míra, počet splátek úvěrů, výše splátky, poplatky, pojištění atp.) RPSN.
- Systém je dále napojen na nebankovní registry (viz kap. 3.4.9), jejichž informace slouží jako vstupy pro *INTEREST* konektor.

Vyhodnocení úvěrového scoringu má následující varianty:

- Schváleno bez dodatečných nabídek
- Schváleno s dodatečnými nabídkami
- Zamítnuto s dodatečnými nabídkami
- Zamítnuto

Podpisové metody:

Přípustné podpisové metody jsou závislé na produktu, interních pravidlech banky a legislativních nárocích na autentizaci klienta. Společnost pracuje s následujícími metodami uzavírání smluvní dokumentace:

- Online SMS
- Popis na pobočce
- Podpis kurýrem

Preferovaná je metoda pomocí autorizační SMS. V případě, kdy klient žádá o půjčku vyšší než určitou částku (definováno BUS pravidlem) a zároveň existuje společné jmění manželů, pak je určena metoda podpis kurýrem, protože je vyžadován spolupodpis manžela/-ky. Pravidlo vyhodnocuje *CreditRisk* ve spolupráci se systémem *BusinessRules*, kde jsou

definována obchodní pravidla. Pro nepřihlášeného klienta nebo neklienta banky, nyní neexistuje online resp. pseudo online proces pro identifikaci a autorizaci. Navrhovaný e-shop má tento problém vyřešit.

Autentizace uživatele

Jedním z podstatných systémových požadavků na nově navrhovaný systém je implementace online autentizace klienta (zachyceno v požadavku F06 viz 8.1). V současné době lze autentizaci nových klientů banky provést pouze pracovník banky nebo kurýr smluvní společnosti. Pokud má tedy klient zájem o některý z jejich produktů, pro podpis smluvní dokumentace musí navštívit některou z poboček banky nebo smlouvy podepsat prostřednictvím kurýra.

Navrhované řešení umožňuje online autentizaci klienta pomocí převzetí ověřené identity osoby od třetí společnosti – banky. Neautentizovaný uživatel společnosti nahraje do systému kopii svého dokladu a provede identifikační transakci – provede platbu z účtu, na kterém figuruje jako vlastník, na účet banky s předem definovanými parametry (částka = 1 Kč, variabilní symbol). Systém vytěží identifikační dokument, uloží data do databáze a poté vyhledá identifikační transakci na účtu banky, získá její detail. Následně systém odešle platbu zpět. Pokud jsou data získaná identifikačního dokladu s údaji z identifikačního dokladu shodná, pak je uživatel označen jako autentizovaný.

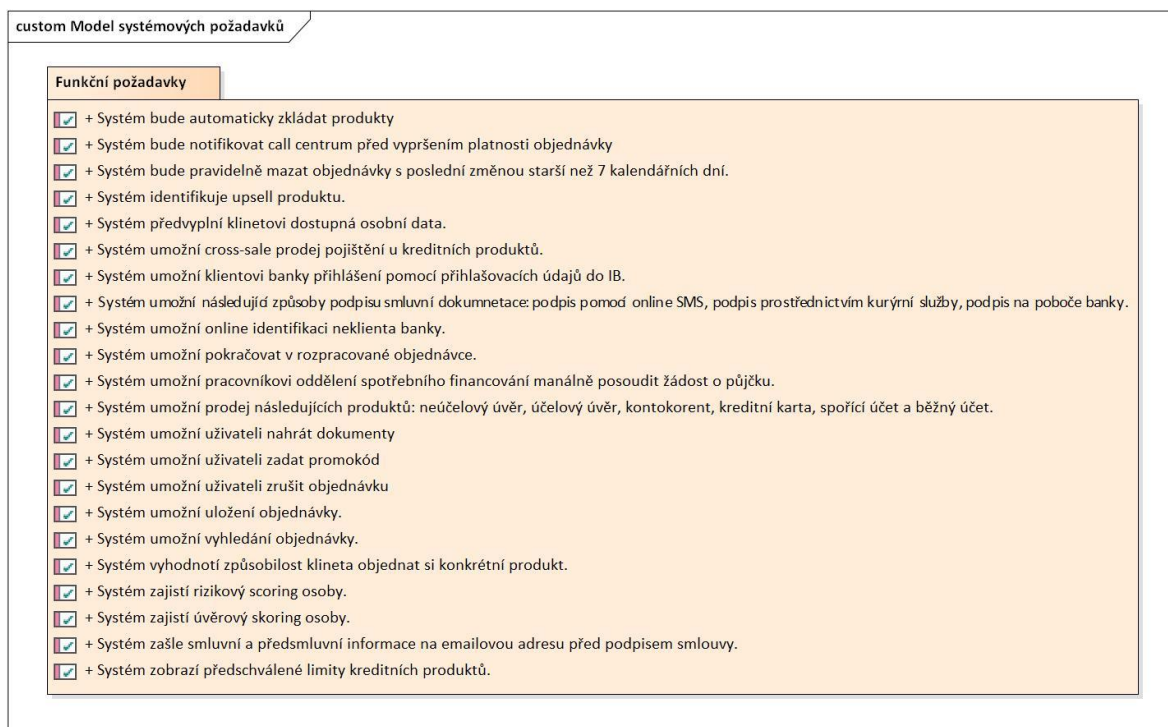
Protože clearingové centrum ČNB CERIS má omezenou provozní dobu, je nutné, aby si společnost zřídila účty ve všech společnostech poskytujících bankovní služby na českém trhu, aby bylo docíleno toho, že uživatel bude ověřen v reálném čase. Pokud systém nedohledá transakci příslušném účtu ve stanovené době, bude proces pozastaven, uživatel bude na tuto skutečnost upozorněn, a bude mu nabídnuta možnost podpisu produktu na pobočce nebo kurýrem nebo vyčkání na ověření. Až proběhne úspěšné ověření, může v procesu online nákupu pokračovat.

4.3 Specifikace požadavků

Požadavky byly specifikovány na základě autorovy zkušenosti z obdobného projektu v reálné společnosti. Zde byl soubor požadavků získán pomocí konzultací s pracovníky banky ze všech zúčastněných oddělení. Cílem těchto konzultací bylo získání představy o potřebách a očekávání vlastníků systému. Konečný soubor požadavků byl analyzován a jednotlivé požadavky byly roztrženy dle typu do následujících skupin:

4.3.1 Funkční požadavky

Funkční požadavky popisují požadovanou službu, kterou musí systém podporovat. Každému funkčnímu požadavku je přiřazeno identifikační číslo, určena priorita a zanesena informace o aktérovi. Na obrázku níže jsou je zobrazen souhrn funkčních systémových požadavků. Kompletní seznam požadavků i s podrobnostmi je k dispozici v příloze 8.1.



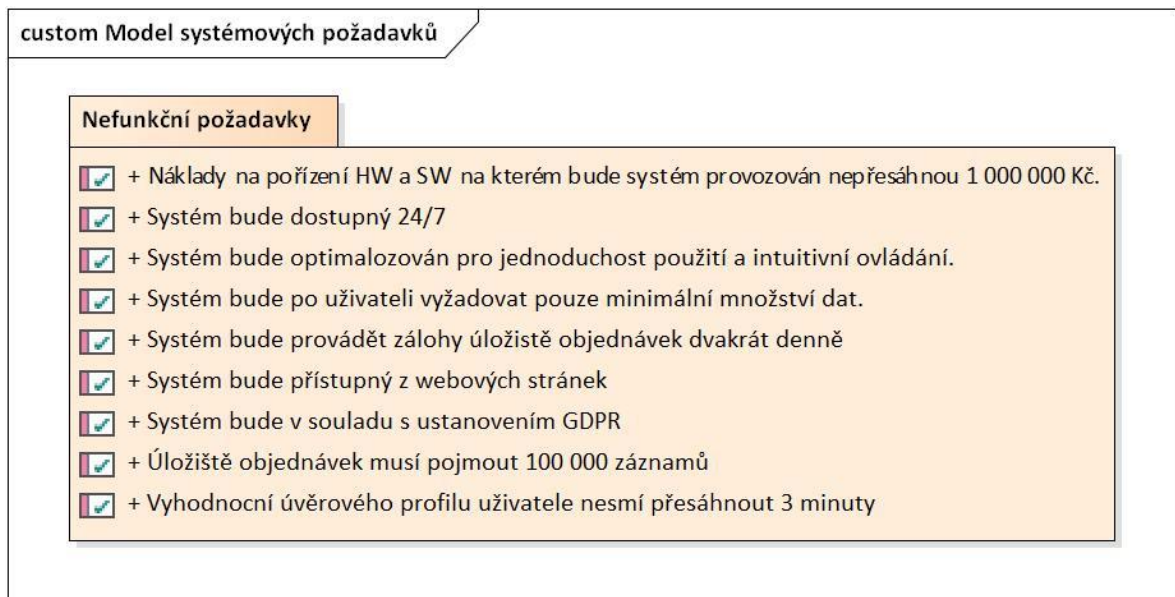
Obrázek 19: Přehled funkčních systémových požadavků; zdroj: [autor]

custom Vybrané funkční požadavky	
<p>Systém umožní online autentizaci neklienta banky.</p> <p style="text-align: right;"><i>tags</i></p> <p>Aktér = Uživatel ID = F06 Priorita = Nezbytný</p>	
<i>(from Funkční požadavky)</i>	
<p>Systém zajistí rizikový scoring osoby.</p> <p style="text-align: right;"><i>tags</i></p> <p>Aktér = Systém ID = F16 Priorita = Nezbytný</p>	
<i>(from Funkční požadavky)</i>	
<p>Systém zajistí úvěrový scoring osoby.</p> <p style="text-align: right;"><i>tags</i></p> <p>Aktér = Systém ID = F17 Priorita = Nezbytný</p>	
<i>(from Funkční požadavky)</i>	
<p>Systém bude automaticky zkládat produkty</p> <p style="text-align: right;"><i>tags</i></p> <p>Aktér = Systém ID = F20 Priorita = Nezbytný</p>	
<i>(from Funkční požadavky)</i>	

Obrázek 20: Vybrané důležité funkční požadavky; zdroj: [autor]

4.3.2 Nefunkční požadavky

Nefunkční požadavky podávají informace o technických požadavcích na systém. Jedná se například o nároky na specifický hardware, software, implementační jazyk, dostupnost systému, zabezpečení nebo legislativní opatření, které musí systém respektovat. Na obrázku níže jsou je zobrazen souhrn nefunkčních požadavků na systém, kompletní seznam požadavků i s podrobnostmi je k dispozici v příloze 8.1.

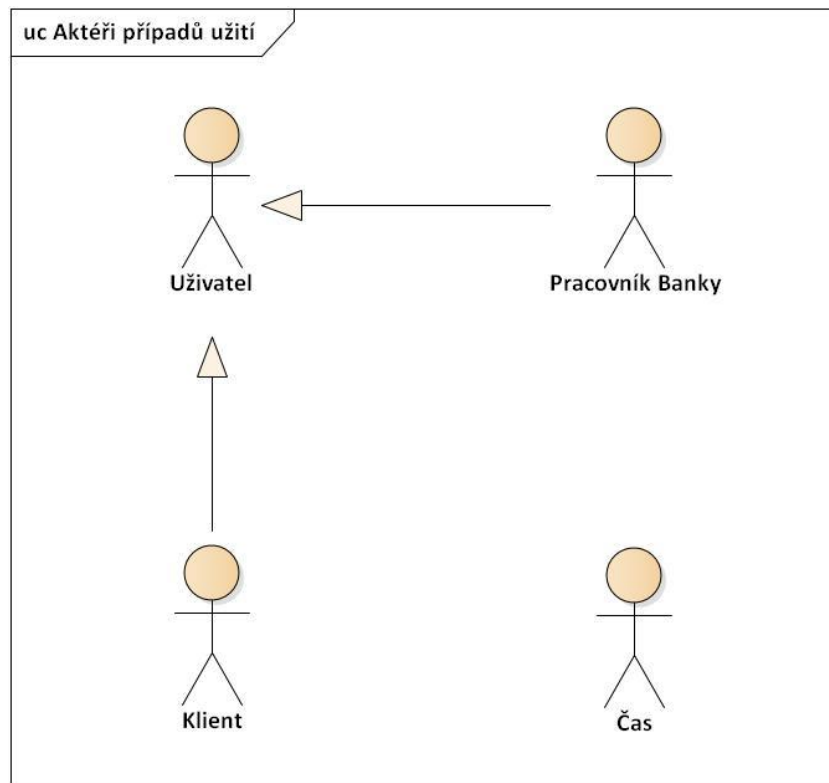


Obrázek 21: Přehled nefunkčních požadavků; zdroj: [autor]

4.3.3 Identifikace aktérů

Z analýzy systémových požadavků vyplývá, že systém bude interagovat s následujícími aktéry:

- *Uživatel* systému je z neautentizovaná fyzická osoba, která obsluhuje systém a podává žádost o zřízení produktu. Neidentifikovaná osoba nemá přístup do sekce uložených objednávek. K zobrazení své uložené objednávky využívá URL zaslaný na emailovou adresu.
- *Klient* je fyzická osoba, která vlastní alespoň jeden produkt banky a má přístup do elektronického bankovníctví prostřednictvím klientského čísla a hesla. Klient je ve vztahu generalizace s aktérem uživatel.
- *Pracovník banky* je fyzická osoba, která je se subjektem banky v pracovně právním vztahu. Náplní práce aktéra je posouzení potencionálního klienta v případech, kdy systém vyhodnotí osobu jako rizikovou. Přihlášení do systému probíhá pomocí osobního čísla zaměstnance a hesla.
- *Čas* – plynutí času může být plnohodnotným aktérem. V tomto případě iniciuje případy užití, které mají být provedeny v určitý časový okamžik.



Obrázek 22: Aktéři případů užití; zdroj: [autor]

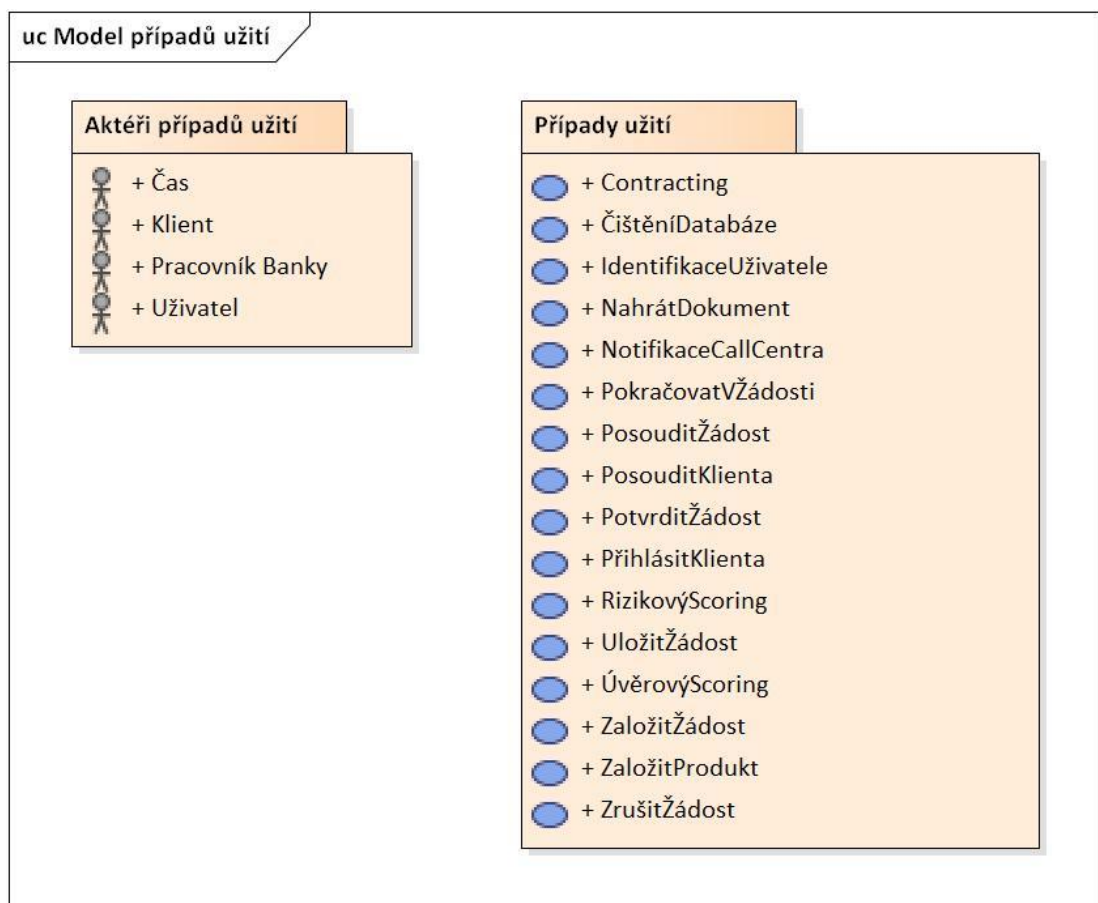
4.3.4 Případy užití

Případ užití je funkcionalita nebo soubor funkcionalit, které uživatel od systému očekává – jsou tedy vždy iniciovány uživatelem. Model případů užití byl vytvořen na základě rozboru sady systémových požadavků a aktérů s ohledem na způsob jakým budou uživatelé tento systém používat.

Model případů užití

Diagram případů užití slouží k přesnému zachycení funkcionality, kterou na základě specifikovaných požadavků bude informační systém poskytovat. Model zachycuje hlavní případy užití systému. Následující případy užití jsou považovány za primární:

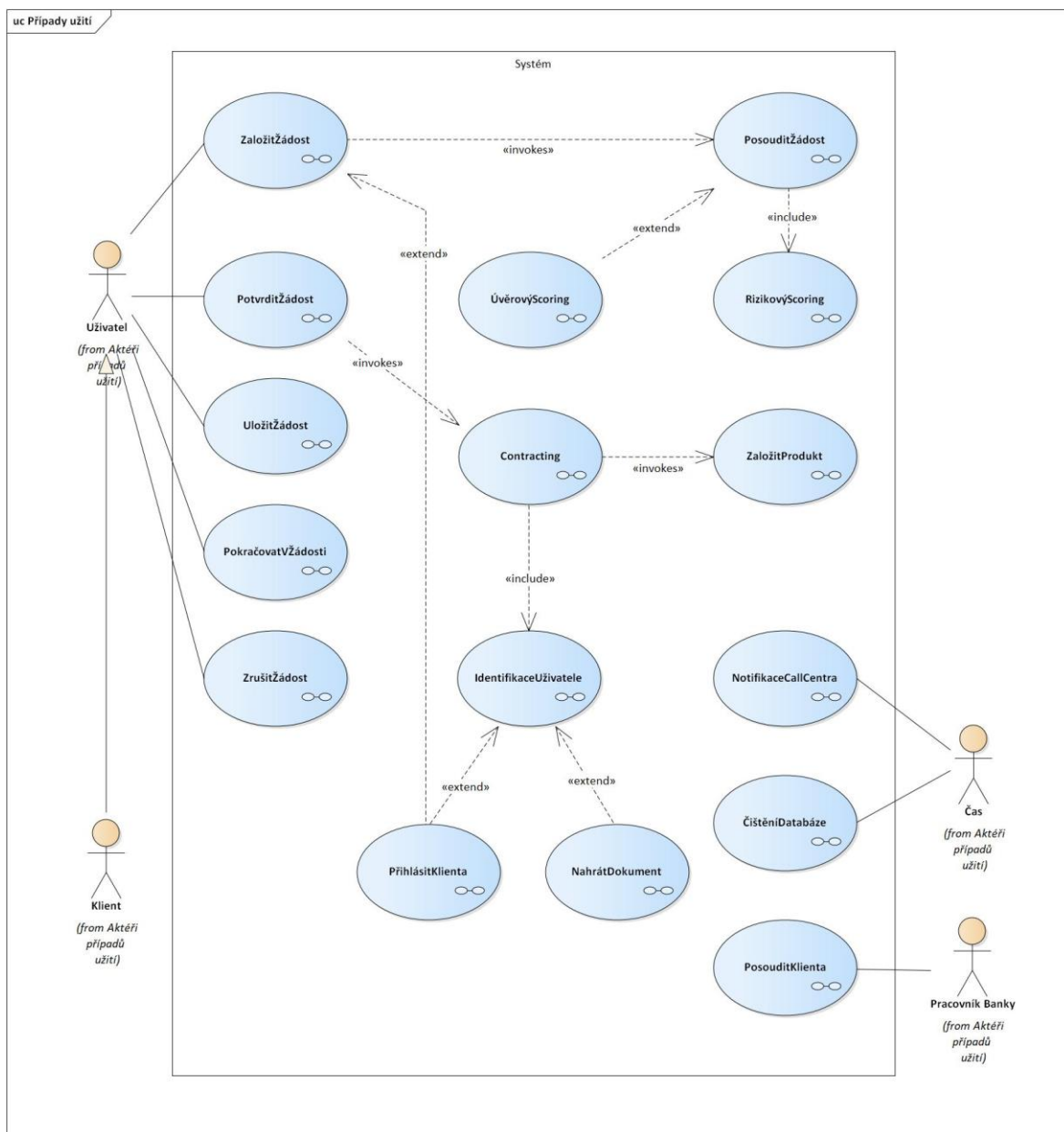
- *ZaložitŽádost*: Uživatel provede parametrizaci produktu a vyplní osobní údaje. V případě, kdy je klient přihlášen a zadané údaje nesouhlasí s daty v centrálním úložišti banky nebo uživatel není přihlášen, pak je proces obohacen o další krok – nahrání kopie identifikačního dokumentu.
- *IdentifikaceUživatele*: Systém porovná data z vytěžené elektronické fotokopie identifikačního dokumentu s daty získanými z detailu transakce provedené klientem banky.
- *PosouzeníŽádosti*: Systém ověří způsobilost klienta vlastnit produkt, jeho rizikový profil, a v případě kreditních produktů také úvěrový profil a rozhodne o poskytnutí produktu na základě obchodních pravidel.
- *PodpisDokumentů*: Uživatel vybere podpisovou metodu a v případě volby podpisu online pomocí SMS mu bude umožněno dokumenty autorizovat.
- *ZaloženíProduktu*: Systém založí produkt v systémech k tomu zřízených.



Obrázek 23: Přehled případů užití; zdroj: [autor]

Diagram případů užití

Hlavním aktérem je uživatel – neautentizovaná fyzická osoba, která vytváří žádost o produkt nebo skupinu produktů. Uživatel zadává do systému osobní údaje nutné pro získání produktu. Systém provede automatické vyhodnocení žádosti a na tomto základě poskytne uživateli závaznou nabídku (pokud je žádost schválena), uživatel potvrdí zájem, podepíše smlouvy a systém automaticky založí produkt.



Obrázek 24: Diagram případů užití; zdroj: [autor]

Každý případ užití je reprezentován scénářem, kde se postupně střídají interakce uživatele a systému. Případy užití specifikují, co má systém vykonávat, nespecifikují ale, jak tomu bude dosaženo. Zatěžování případů užití o specifikou implementaci funkce je považováno za závažnou chybu.

Jako příklad detailu případu užití byl vybrán scénář *ZaložitŽádost*, který je rozšířen případy užití *PřihlásitKlienta* a invokuje případ užití *PosouzeníŽádosti*. Jak již bylo zmíněno v kapitole 4.2.2 návrh autentizace anonymního uživatele jedním ze základních požadavků na systém, tento požadavek je zachycen v případě užití *IdentifikaceUživatele*.

Tabulka 2: Detail případu užití ZaložitŽádost; zdroj: [autor]

Případ užití:	ZaložitŽádost
ID:	UC01
Stručný popis:	Uživatel vyplní žádost o produkt, která se skládá z parametrizovaného produktu a osobních údajů. Systém v případě přihlášeného uživatele zobrazí předschválené varianty kreditních produktů.
Hlavní aktéři:	Uživatel
Vedlejší aktéři:	Žádní
Vstupní podmínky:	Žádné
Hlavní scénář:	<ol style="list-style-type: none"> 1. Případ užití začíná až uživatel vybere produkt, o který má zájem 2. KDYŽ uživatel je přihlášen a žádá o kreditní produkt <ol style="list-style-type: none"> 2.1 Systém zjistí předschválené limity 2.2 Systém identifikuje upsell 2.3 Systém zobrazí formulář parametrizace produktu s předschválenými variantami produktu 3. JINAK uživatel není přihlášen <ol style="list-style-type: none"> 3.1. Systém zobrazí formulář parametrizace produktu bez předschválených variantami produktu 4. Uživatel provede parametrizaci produktu a potvrdí formulář 5. Systém zobrazí obrazovku vybízející klienta k přihlášení 6. Uživatel zvolí přihlášení nebo vybere možnost pokračovat bez přihlášení <i>místo rozšíření: PřihlásitKlineta</i> 7. KDYŽ uživatel je přihlášen <ol style="list-style-type: none"> 7.1. Systém zjistí dostupné clientské údaje 7.2. Systém zobrazí formulář osobní údaje s předvyplněnými daty 8. JINAK uživatel není přihlášen Systém zobrazí formulář osobní údaje a umožní vložit potvrzení příjmu. 9. Uživatel zadá povinné údaje a potvrdí formulář 10. Systém zobrazí obrazovku informující klienta o probíhajícím vyhodnocení žádosti
Výstupní podmínky:	Systém navázal případem užití posouzení žádosti
Alternativní scénáře:	Nefunkční systém poskytující předschválené limity Nefunkční systém ověřující upsell

Tabulka 3: Detail případu užití PřihlásitKlienta; zdroj: [autor]

Případ užití:	PřihlásitKlienta
ID:	UC02
Stručný popis:	Uživatel provede přihlášení do systému pomocí údajů z elektronického bankovníctví
Hlavní aktéři:	Klient
Vedlejší aktéři:	Žádní
Vstupní podmínky:	Žádné
Hlavní scénář:	<ol style="list-style-type: none"> 1. Případ užití začíná, když uživatel zvolí možnost přihlášení 2. Systém zobrazí přihlašovací komponentu 3. Klient vyplní přihlašovací údaje a potvrdí zadání 4. Systém přihlásí klienta
Výstupní podmínky:	Systém identifikoval uživatele pomocí klientského čísla a příslušného hesla
Alternativní scénáře:	NeplatnéUživatelskéÚdaje

Alternativní scénář:	NeplatnéUživatelskéÚdaje
ID:	UC02.1
Stručný popis:	Systém informuje uživatele, že zadal neplatné přihlašovací údaje
Hlavní aktéři:	Klient
Vedlejší aktéři:	Žádní
Vstupní podmínky:	Klient zadal neplatné uživatelské údaje
Hlavní scénář:	<ol style="list-style-type: none"> 1. Alternativní scénář začíná krokem 3 hlavního scénáře 2. Systém zobrazí notifikační obrazovku neplatných údajů a zobrazí nevyplněnou zobrazí přihlašovací komponentu
Výstupní podmínky:	Žádné

Tabulka 4: Detail případu užití NahrátDokument; zdroj: [autor]

Případ užití:	NahrátDokument
ID:	UC03
Stručný popis:	Uživatel nahraje fotokopii dokumentů, systém uloží dokument a vytěží data.
Hlavní aktéři:	Uživatel
Vedlejší aktéři:	Žádní
Vstupní podmínky:	
Hlavní scénář:	<ol style="list-style-type: none"> 1. <u>KDYŽ</u> uživatel je přihlášen <ol style="list-style-type: none"> 1.1 Systém zobrazí formulář sloužící k nahrání kopie identifikačního dokladu <u>JINAK</u> uživatel není přihlášen 2. 2.1. Systém vygeneruje identifikační číslo pro transakci 2.2. Systém zobrazí formulář sloužící k nahrání kopie identifikačního dokladu a informuje klienta o nutnosti provést korunovou platbu na účet banky. 3. Uživatel potvrdí formulář 4. Systém uloží dokument do DMS databáze 5. Systém vytěží data a uloží je do OCR databáze
Výstupní podmínky:	Systém úspěšně uložil dokument, vytěžil data a uložil je do databáze.
Alternativní scénáře:	<p>Chyba uložení souboru ChybaUploaduDokumentu Chyba OCR systému Neplatný dokument</p>

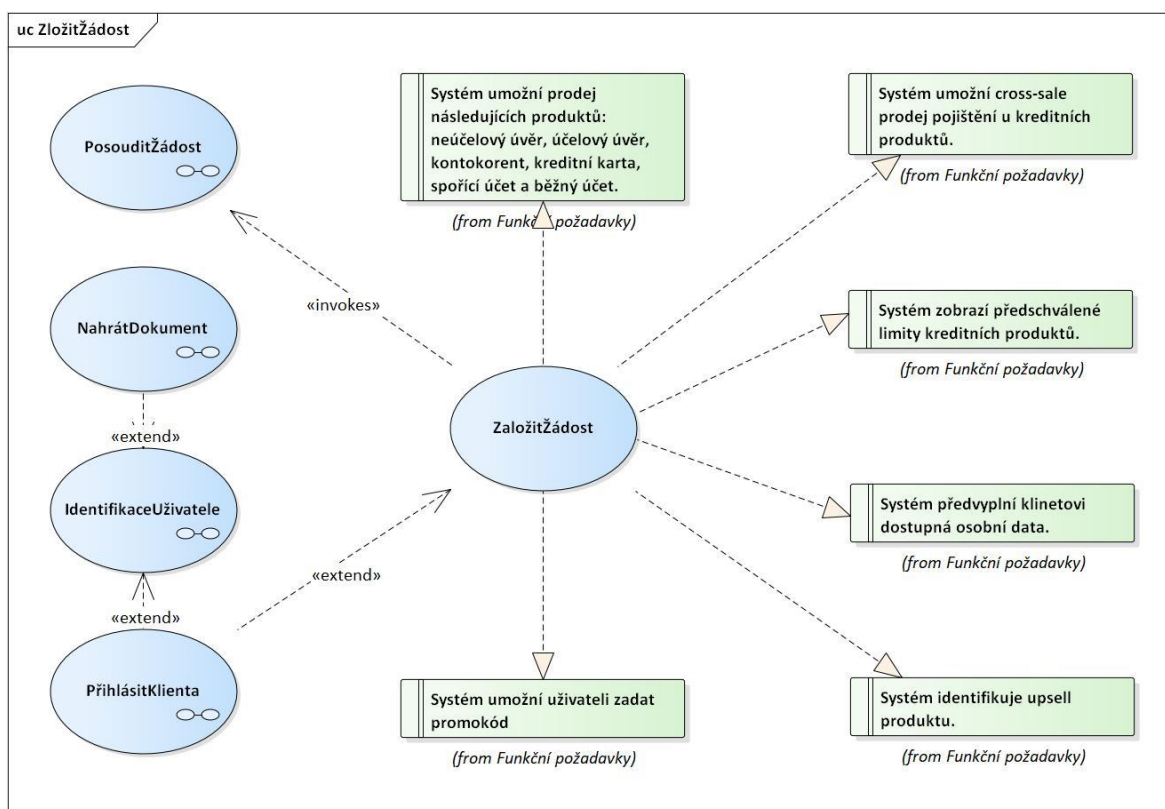
Tabulka 5: Detail případu užití IdentifikaceUživatele; zdroj: [autor]

Případ užití:	IdentifikaceUživatele
ID:	UC04
Stručný popis:	System ověří shodu vytěžených dat z identifikačního dokladu
Hlavní aktéři:	Uživatel
Vedlejší aktéři:	Žádní
Vstupní podmínky:	Uživatel vybral podpisovou metodu online SMS
Hlavní scénář:	<ol style="list-style-type: none"> 1. Případ užití začíná poté co uživatel vyberte jednu z podpisových variant na formuláři závazné nabídky banky. 2. KDYŽ uživatel je přihlášen <ol style="list-style-type: none"> 2.1. System srovná osobní data vyplněná uživatelem s daty v ClientDatabase 2.2. KDYŽ data jsou shodná <ol style="list-style-type: none"> 2.2.1. --- 2.3. JINAK data nejsou shodná <ol style="list-style-type: none"> 2.3.1. <i>místo rozšíření: NahrátDokument</i> 2.3.2. System porovná data z vytěžené dokumentu s daty v úložišti objednávek 2.3.3. System aktualizuje údaje v úložišti objednávek a v ClientDatabase 3. JINAK uživatel není přihlášen <ol style="list-style-type: none"> 3.1. System zobrazí stránku Přihlaste se nebo nahrajte fotokopii dokladu 3.2. Uživatel vyplní data na stránce a odešle formulář 3.3. KDYŽ uživatel zvolil přihlášení <ol style="list-style-type: none"> 3.3.1. <i>místo rozšíření: PřihlásitKlienta</i> 3.3.2. Navazuje bodem 2 3.4. JINAK uživatel nahrál dokument <ol style="list-style-type: none"> 3.4.1. <i>místo rozšíření: NahrátDokument</i> 3.4.2. DOKUD nevyprší nastavený timeout nebo data nejsou navrácena <ol style="list-style-type: none"> 3.4.2.1 System pravidelně dotazuje na detail identifikační platby u vybrané banky 3.4.3 System porovná data z vytěženého dokumentu s identitou poskytnutou bankou 3.4.4 System založí klienta v klientské databázi s příznakem dočasný záznam
Výstupní podmínky:	System ověřil identitu klienta
Alternativní scénáře:	<p>Údaje nesouhlasí</p> <p>Nefunkční system vyhledávající platby</p> <p>Timeout transakce nenalezena</p>

4.3.5 Vztah funkčních požadavků a případů užití

Případy užití a systémové požadavky tvoří dva zdroje požadavků na systém. Uvedením těchto dvou zdrojů do vzájemného vztahu je možné ověřit, zdali soubor systémových požadavků neobsahuje funkcionalitu, která není zachycená v žádném z případů užití nebo naopak existuje takový případ užití, který není podložen žádným systémovým požadavkem.

Na obrázku 18 je znázorněn vztah případu užití *ZaložitŽádost* a systémových požadavků, které jsou jím realizovány.



Obrázek 25: Vztah požadavků a případu užití *ZaložitŽádost*; zdroj: [autor]

4.3.6 Slovníček pojmů

Slovníček pojmů je jedním z důležitých artefaktů, který vzniká ve fázi sběru požadavků a modelování případů užití. Požadavky by měly být zachyceny v obchodním jazyce. Každé odvětví má svou terminologii, která jsou mu vlastní a každá společnost používá některé obecné pojmy pro své specifické účely. Proto je důležité zajistit, aby každý, komu bude

dokumentace systému sloužit jako zdroj informací, byl schopný porozumět jazyku ve kterém byla napsána. K tomuto účelu slouží slovníček pojmů, který je lexikonem klíčových pojmů a definicí.

V tabulce 6 je uveden výňatek ze slovníčku pojmů. Některé z nich jsou například použity v detailu případu užití *ZaložitŽádost*. Kompletní slovníček pojmů je uveden v příloze 8.3.

Tabulka 6: Výňatek ze slovníčku pojmů projektu; zdroj: [autor]

Pojem	Synonyma	Popis
		Produkty, ve kterých banka vystupuje jako dlužník – klient bance poskytuje finance formou vkladů za cenu úroku.
Depozitní produkt	Depozita	V projektu E-shop se jedná o tyto produkty: Spořicí účet, Běžný účet
Kanál	Žádná	Cesta nebo způsob jakým lze sjednat produkt. Současné typy kanálů jsou následující: Pobočka, Call centrum.
Kreditní produkt	Kredity	Produkty, ve kterých banka vystupuje jako věřitel – poskytuje finance jiným osobám za cenu úroku.
Promo kód	Žádná	Číselný kód, který identifikuje marketinkovou kampaň – promo kód se váže k určitému produktu. Uživatel, který se prokáže tímto kódem získává výhodu, například ve formě prominutí zřizovacího poplatku nebo nižšího úroku v případě kreditních produktů.
		Specifický případ žádosti o produkt kreditní karta nebo kontokorent. Klient již požadovaný produkt vlastní a na tomto produktu si přeje navýšit limit.
Upsell	Navýšení limitu	<i>Snížení limitu nebude na kanále e-shop možné (obchodní politika společnosti).</i>
Žádost	Objednávka	Svobodný projev vůle fyzické osoby sjednat si určitý produkt nebo službu poskytovaný bankou.

4.4 Analýza informačního systému

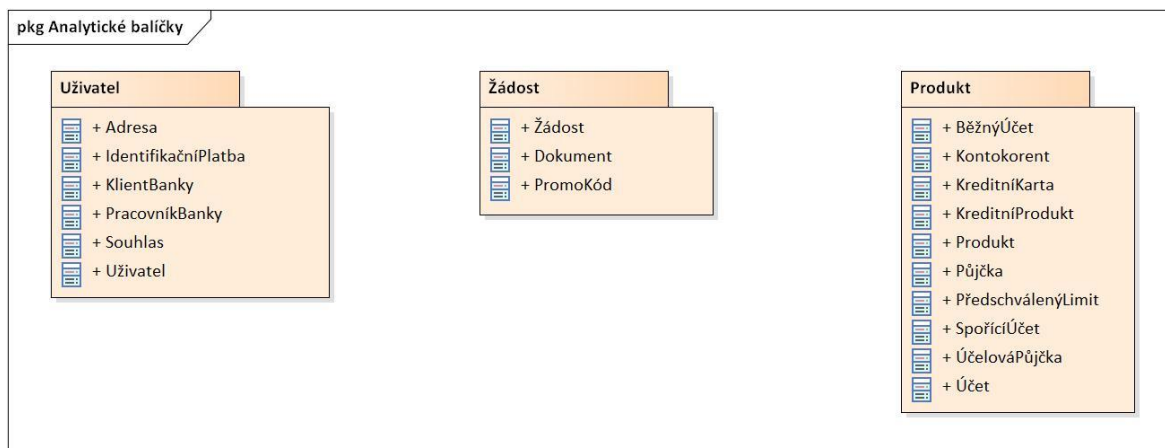
Náplní fáze analýzy informačního systému je tvorba analytického modelu. Analytický model se zaměřuje na chování systému, nicméně nedefinuje, jakým způsobem je tohoto chování dosaženo. Artefakty této fáze vývoje aplikace jsou analytické třídy doplněné o datový slovník, realizace případů užití a modely procesní logiky.

4.4.1 Analytický model tříd

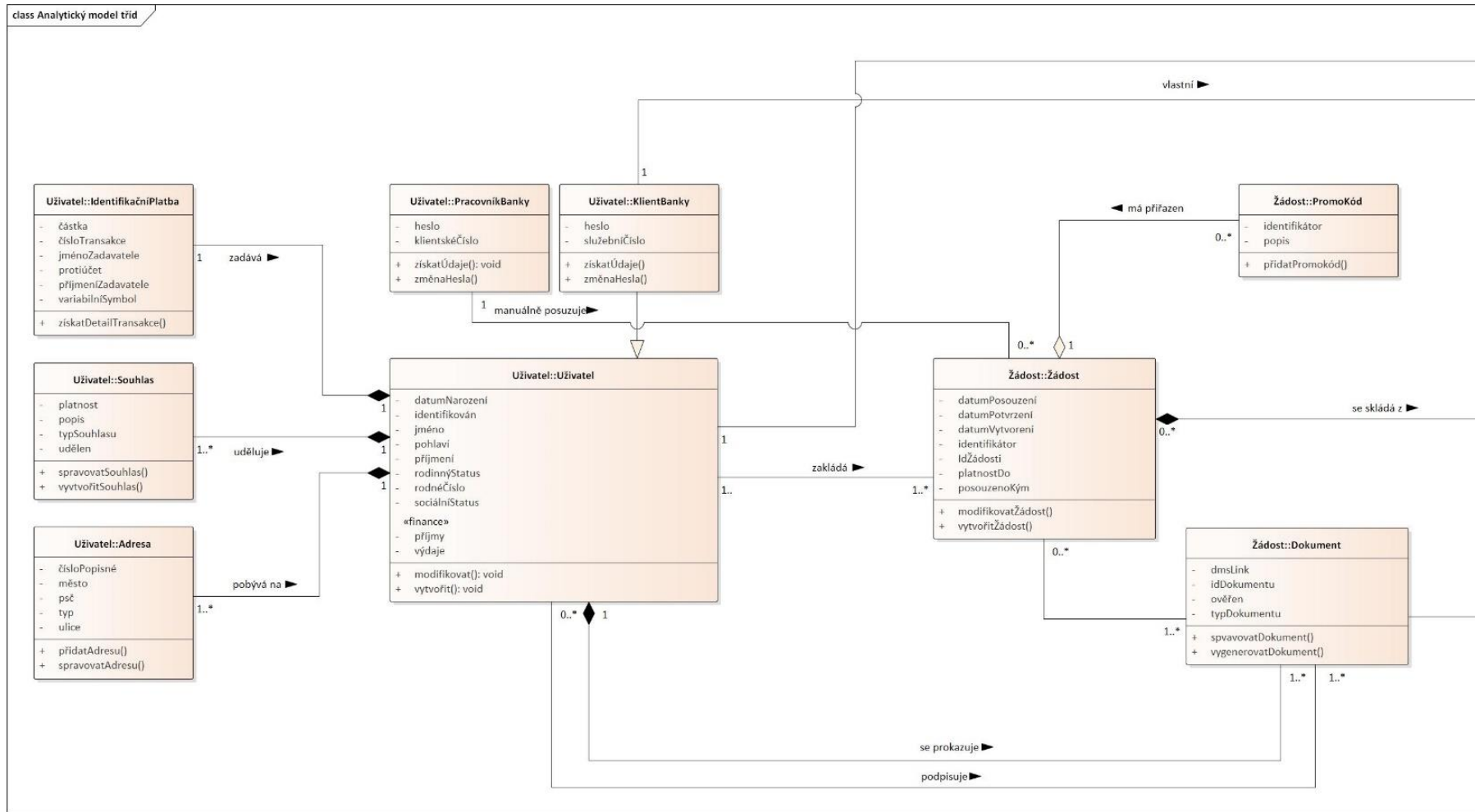
Analytické třídy dle Arlowa [1, s 173] reprezentují abstrakci problémové domény a měly by mapovat pojmy skutečného světa. Za problémovou doménu je považována ta doména, ze které vznikl požadavek na vznik nového softwaru, v tomto případě je problémovou doménou online prodej bankovních produktů.

Prvním krokem objektově orientované analýzy je objasnění této domény – ačkoli v této fázi jde o řešení orientační. Pro model analytických tříd je v praxi často užívána definice konceptuální datový model. Diagram analytických tříd obsahuje základní třídy a operace, pomocí nichž lze přehledně vyjádřit statický pohled na systém při zachování určité míry abstrakce. V další projektové fázi je tento model detailně rozpracován do návrhového modelu tříd.

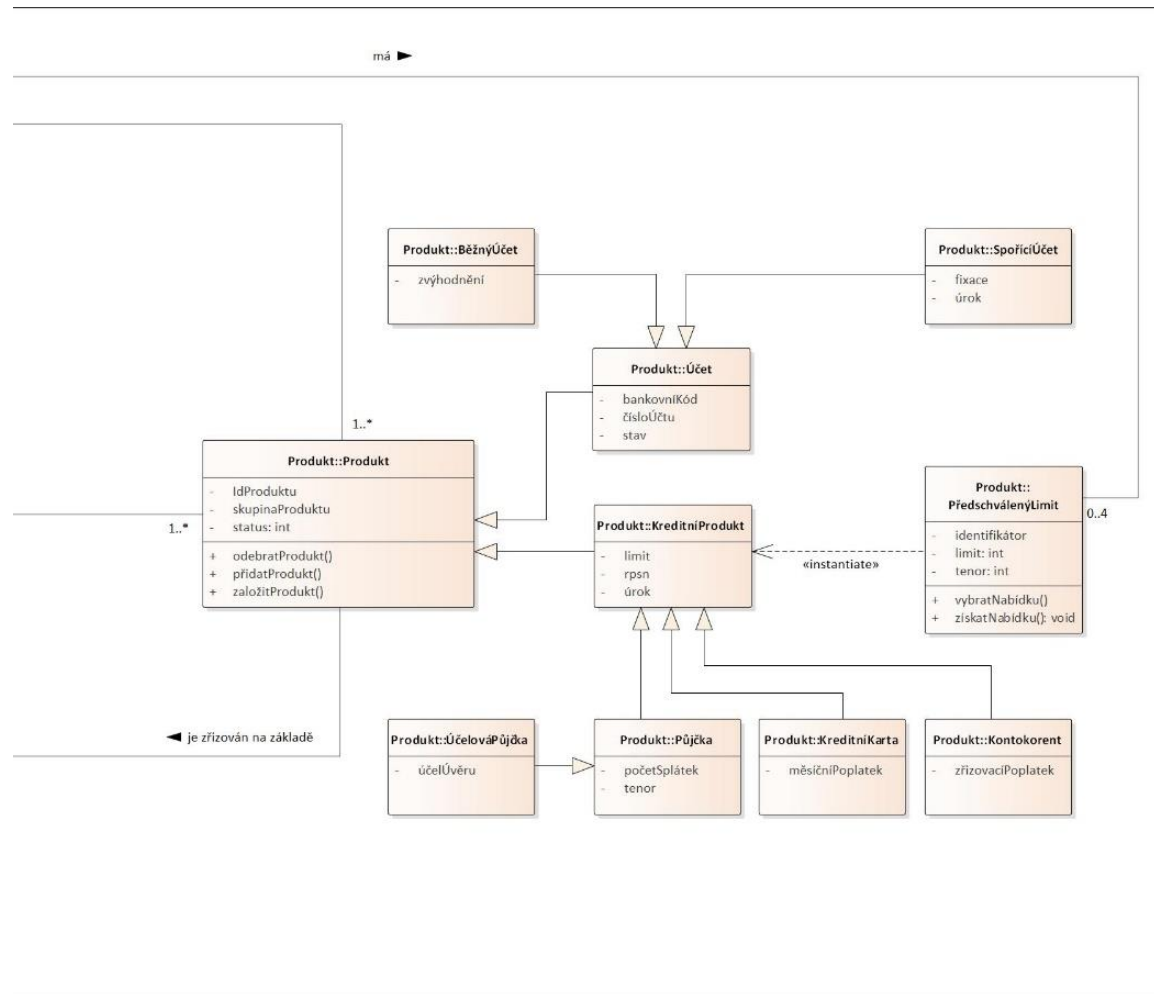
Analytické třídy byly získány metodou analýzy podstatných jmen a sloves systémových požadavků a případů užití a byly rozříděny do tří logických celků: uživatel, žádost a produkt, tak jak je vidět na diagramu 19. Diagram byl pro lepší čitelnost rozdělen na dvě poloviny.



Obrázek 26: Analytické balíčky; zdroj: [autor]



Obrázek 27: Class Diagram analytických tříd 1/2; zdroj: [autor]



Obrázek 28: Class Diagram analytických tříd 2/2; zdroj: [autor]

4.4.2 Datový slovník

Datový slovník zevrubně popisuje identifikované analytické třídy a doplňuje tak model o další informace, které umožňují lepší pochopení modelovaného systému. Níže je uveden výňatek z datového slovníku pro hlavní analytické třídy. Úplný datový slovník je uveden v příloze 8.2.

- Třída *Uživatel* představuje reálného uživatele – fyzickou osobu, která ovládá systém. Třída uchovává hlavní informace o uživateli jako je jméno, přímění, rodné číslo atp. Atribut „identifikován“ podává informaci o tom, zdali je uživatelem autentizovaná osoba nebo anonymní osoba, což je podstatná informace ovlivňující obchodní proces. Třída uživatel zakládá *Žádost*.
- Třída *Klient* je podtřídou třídy *Uživatel*. Představuje klienta banky, který je plně autentizován a není tak nutné provádět identifikaci osoby v průběhu procesu. Klient se přihlašuje do systému pomocí osobního hesla a klientského čísla.
- Třída *IdentifikačníTransakce* představuje elektronickou bezhotovostní transakci, která slouží k autentizaci uživatele. Identifikační transakci zadává třída *Uživatel*. Třída umožňuje získat detail transakce Třída vlastní atributy jméno a příjmení zadavatele, které je využíváno systémem při identifikaci uživatele.
- Třída *PracovníkBanky* představuje fyzickou osobu, která má přístup do neveřejné části systému. Pracovník banky manuálně posuzuje žádosti klientů s nevyhovujícím rizikovým profilem pro automatizovaný proces. Pracovník banky se přihlašuje do systému pomocí služebního čísla a hesla.
- Třída *Žádost* vzniká na popud třídy *Uživatel*. Třída sdružuje produkty, o které má uživatel zájem. Třída má definovaný datum vzniku, stav a platnost. Žádost se skládá alespoň z jednoho produktu. Třída spravuje žádost a využívá třídu *Uživatel*. Je ve vztahu kompozice s třídou *PromoKód*.
- Třída *Produkt* představuje službu, kterou banka poskytuje svým klientům na základě smluvního vztahu. Třída umožňuje přidat, přiřadit produkt k objednávce, modifikuje produkt, vyhodnocuje způsobilost klienta vlastnit produkt a zakládá produkt.

Produkty vznikají na základě dokumentu, který stvrzuje smluvní vztah mezi bankou a klientem.

- Třída *Dokument* představuje elektronický dokument, na jehož základě jsou zakládány produkty. Třída umožňuje vygenerovat dokument a uložit dokument. Dokumenty jsou autorizovány pomocí SMS, která je klientovi zaslána na mobilní telefon.
- Třída *PředschválenýLimit* reprezentuje instanci kreditního produktu, která nemusí projít plným úvěrovým scoringem. Jde o závaznou nabídku specifické varianty produktu ze strany banky. Předschválený limit se váže na klienta banky a je identifikován jedinečným kódem. Třída umožňuje získat nabídku a vybrat předschválený limit.

4.4.3 Realizace případů užití

Analytické třídy zobrazují statický model systému, zatímco realizace případu užití zobrazuje, jak spolu definované analytické třídy komunikují při realizaci chování popsaném v detailu případu užití. Jedná se tedy o dynamický náhled na systém. Ke znázornění této komunikace jsou nejčastěji používány sekvenční diagramy, popř. diagramy spolupráce.

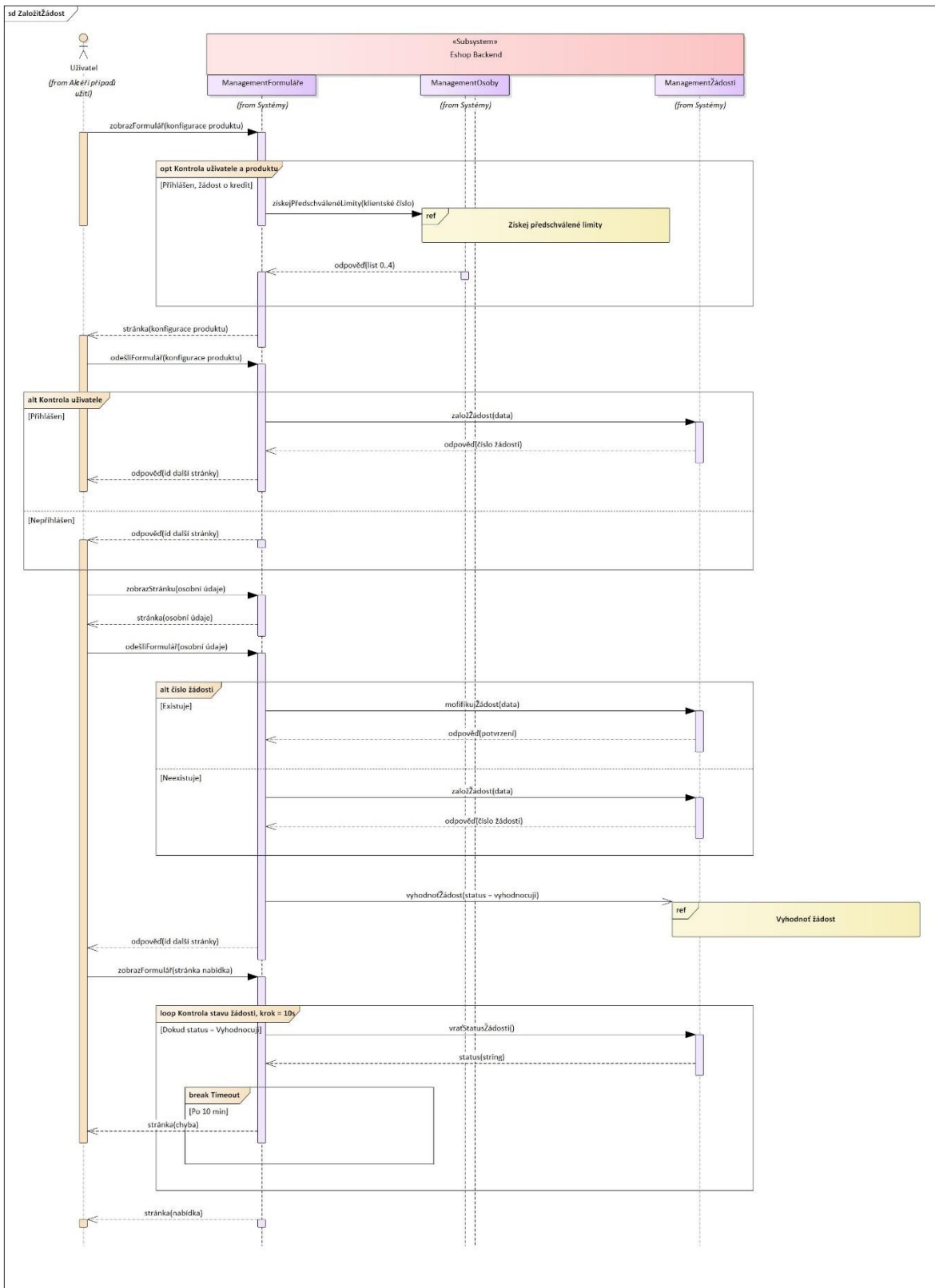
Níže jsou uvedeny sekvenční diagramy pro případy užití *ZaložitObjednávku* a *IdentifikovatUživatele*, které jsou považovány za primární funkcionality modelovaného systému doplněné o realizaci případů užití *PřihlásitKlienta* a *NahrátDokument*. V modelových diagramech byla pro přehlednost vynechána integrační vrstva mezi jednotlivými komponentami. Frontend komunikuje s aplikační logikou pomocí dvou služeb *zobrazFormulář* a *odešliFormulář*. Tyto služby jsou orchestrační a na základě podmínek (např. stupně identifikace uživatele) iniciují další volání. Frontend v odpovědi na služby *odešliFormulář* získává identifikátor stránky, o kterou následně požádá pomocí služby *zobrazFormulář*.

Sekvenční diagram pro případ užití ZaložitŽádost

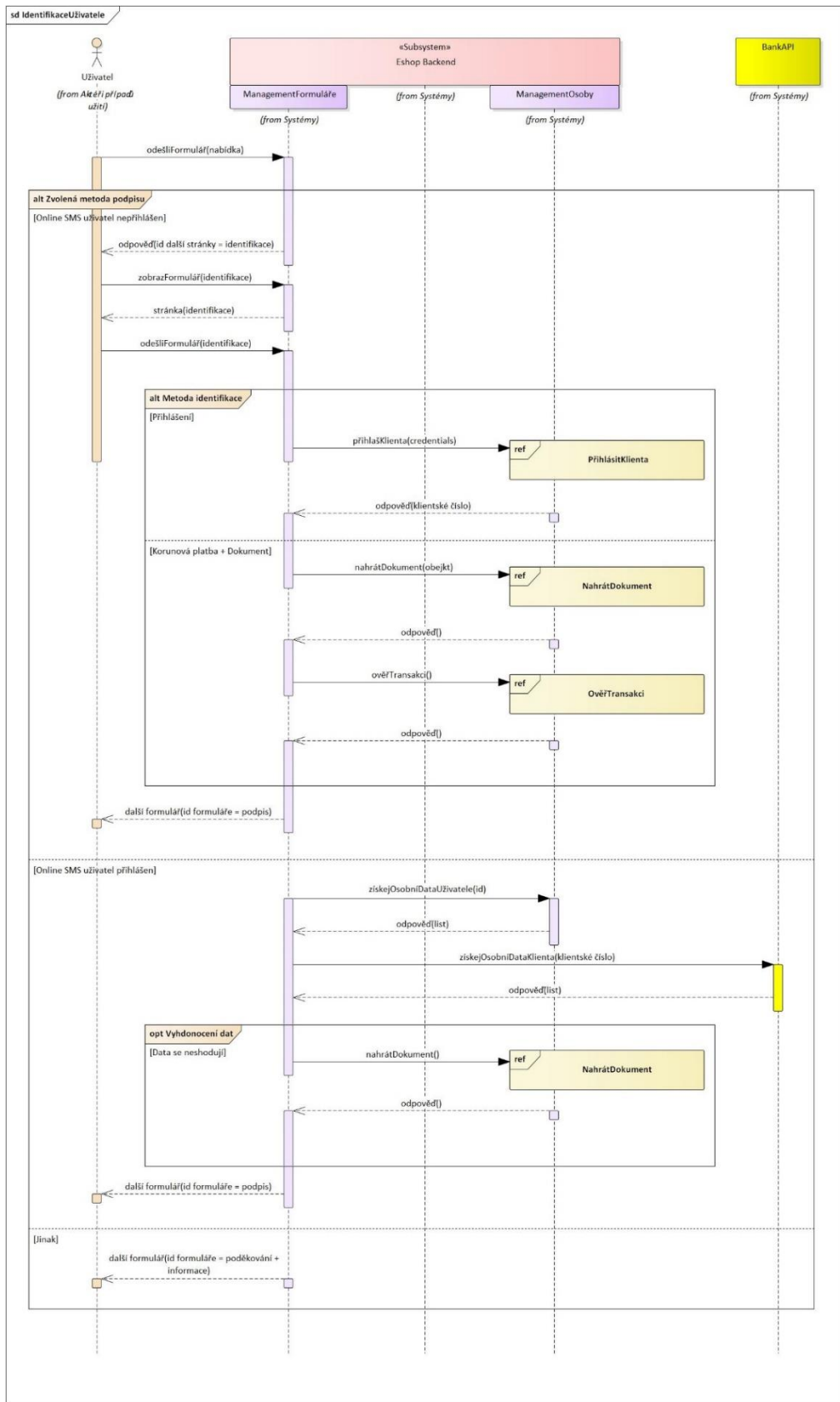
Diagram popisuje případ užití *ZaložitŽádost* (obrázek 29). Primárním aktérem je uživatel, který na domovské stránce e-shopu projevil zájem o některý z nabízených produktů. Před načtením stránky produktová konfigurace získává případné předschválené limity. Pokud je uživatel identifikován (přihlášen) systém na přechodu mezi fází produktové konfigurace a osobními údaji zakládá žádost v databázi. Uživatel vyplní své osobní údaje a odešle formulář. Systém asynchronně volá metodu *vyhodnotŽádost*. V odpovědi na odeslání formuláře dostává frontend identifikátor fáze Nabídka. Při načítání stránky Nabídka systém periodicky zjišťuje, zdali je žádost již vyhodnocena (v mezidobí je uživateli prezentován jednoduchý loading bar), a v případě změny statusu žádosti je uživateli zobrazen výsledek.

Sekvenční diagram případu užití IdentifikaceUživatele

Diagram popisuje případ užití *IdentifikaceUživatele* (obrázek 30). Ve fázi Nabídka uživatel volí mimo jiné preferovanou metodu podpisu smluvní dokumentace. Po odeslání formuláře Nabídka systém kontroluje, zdali je uživateli příslušná podpisová metoda povolena vzhledem k jeho stupni identifikace. Pokud uživatel zvolil metodu online SMS, systém vyzve nepřihlášeného uživatele k autentizaci. Ta může být provedena buď přihlášením, v případě klientů s aktivním elektronickým bankovníctvím nebo pomocí korunové platby. Pokud uživatel není klientem nebo se rozhodne nevyužít možnosti se přihlásit, musí nahrát fotokopii svého identifikačního dokladu a provést korunovou platbu na jeden z nabízených účtů.



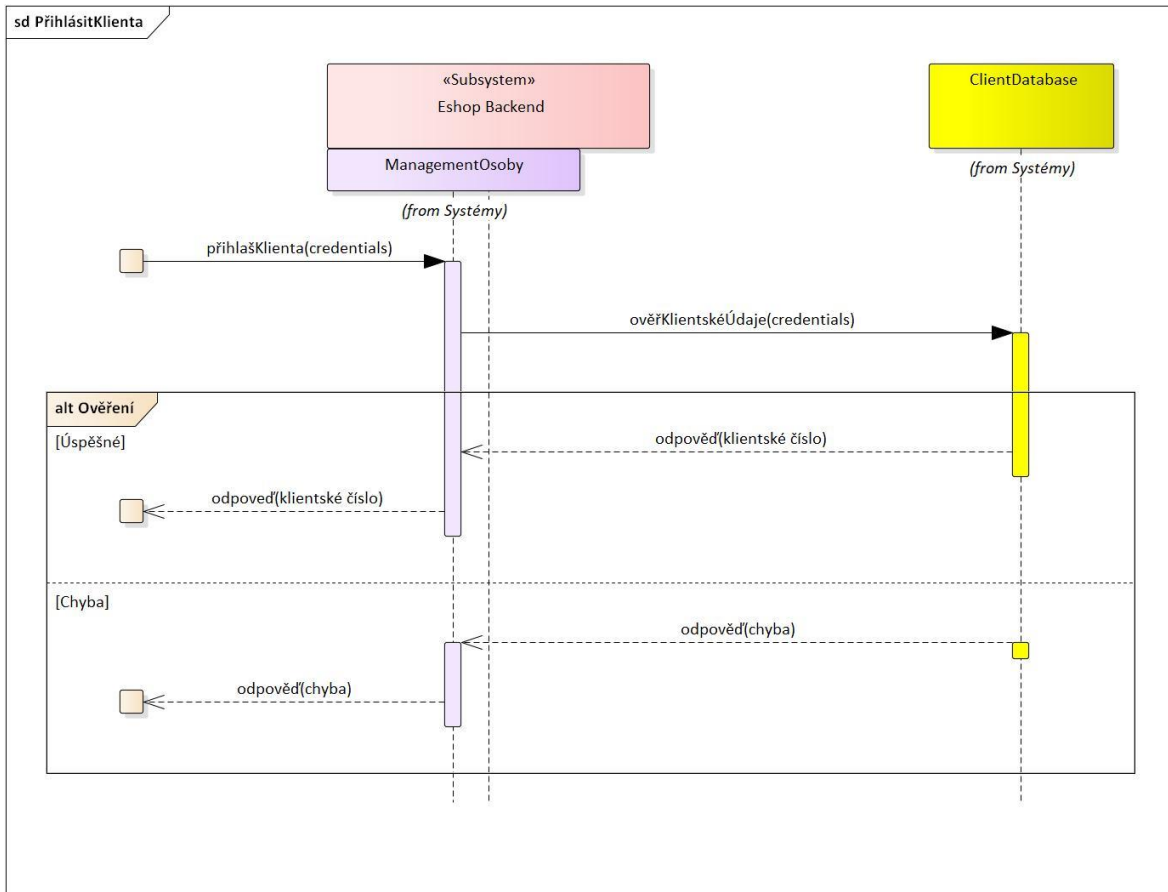
Obrázek 29: Sekvenční diagram ZaložitŽádost; zdroj: [autor]



Obrázek 30: Sekvenční diagram IdentifikaceUživatele; zdroj: [autor]

Sekvenční diagram případu užití PřihlásitKlienta

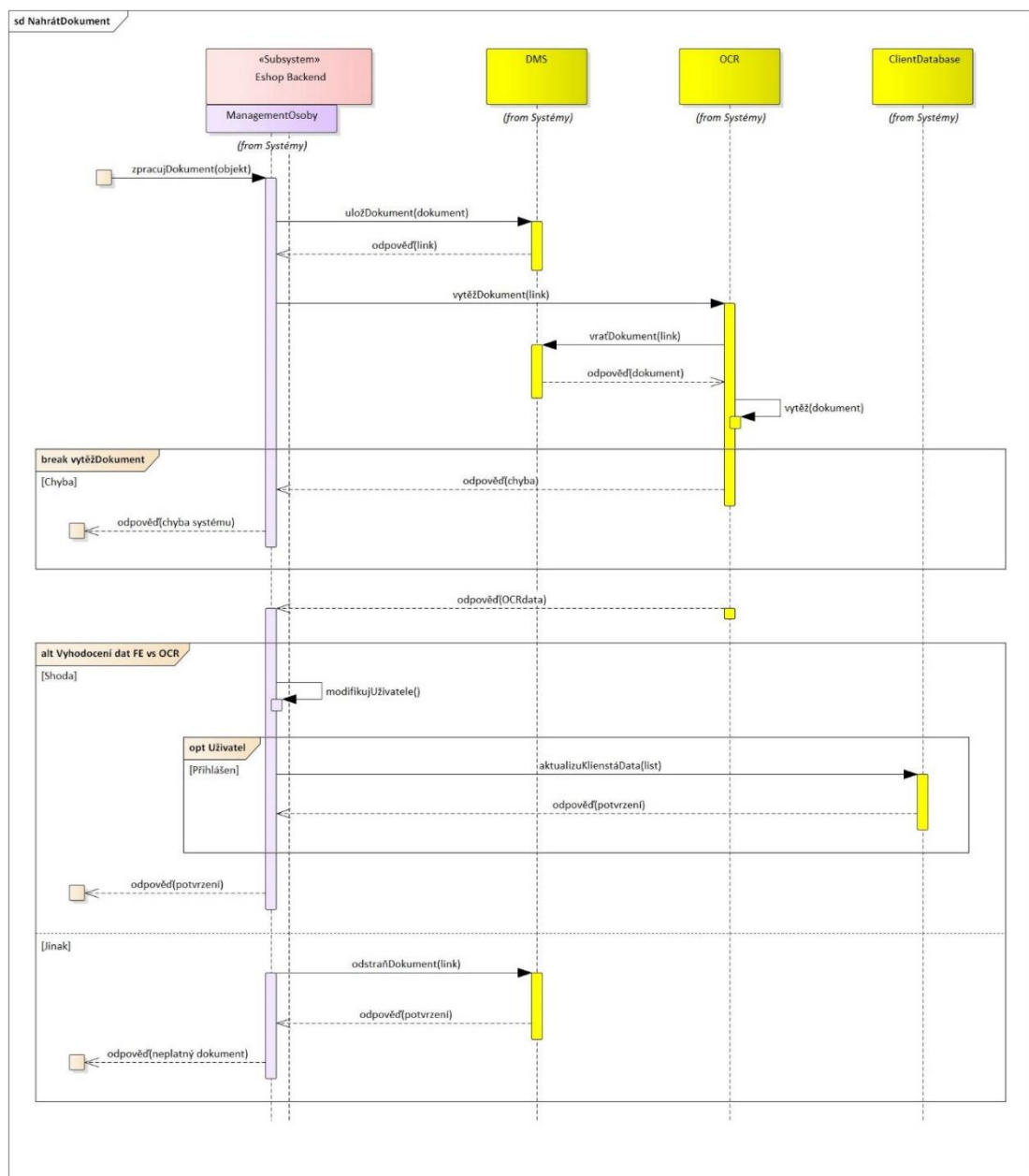
Diagram popisuje komunikaci mezi aplikační vrstvou systému e-shop a systémem *ClientDatabase*, která poskytuje rozhraní *ověřKlientskéÚdaje*. Uživatel se pomocí přístupových údajů do elektronického bankovníctví přihlašuje do systému e-shop



Obrázek 31: Sekvenční diagram PřihlásitKlienta; zdroj: [autor]

Sekvenční diagram realizace případu užití NahrátDokument

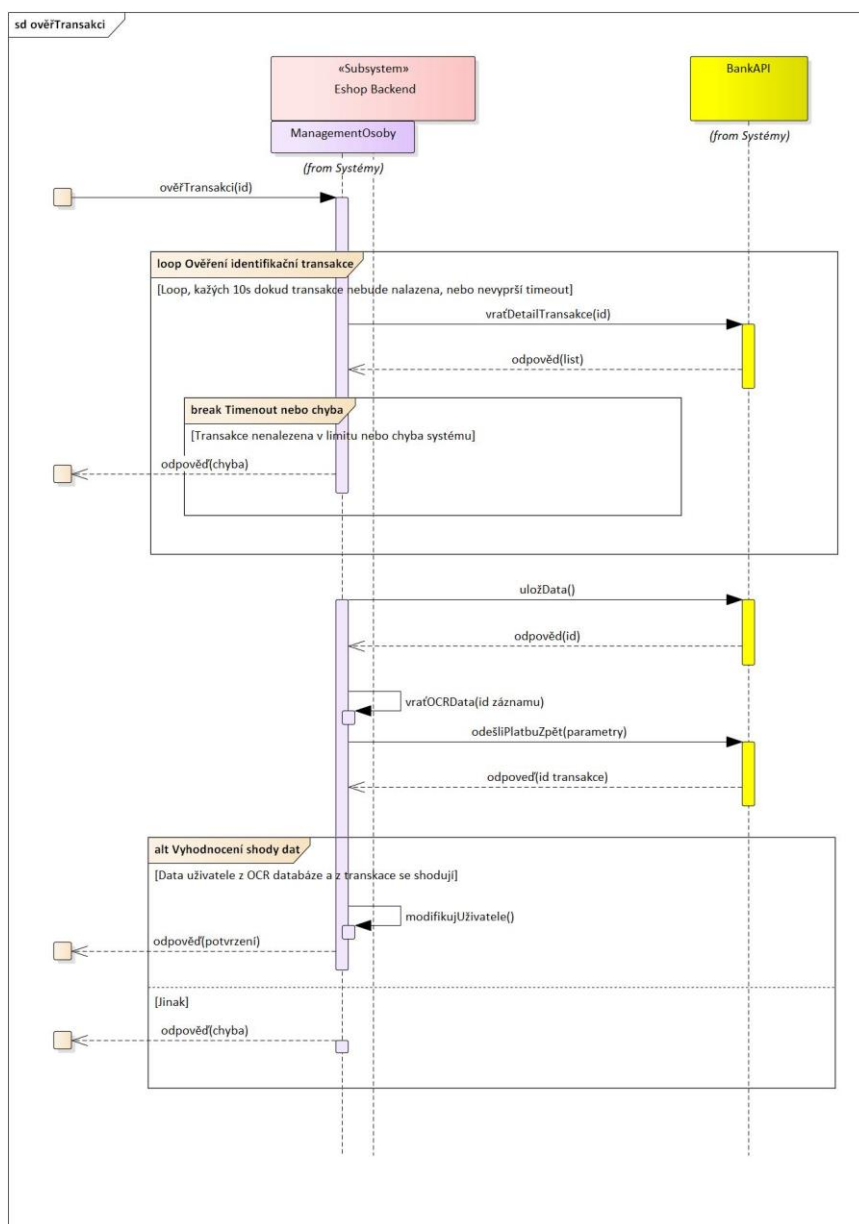
Uživatel nahraje dokument, systém tento dokument uloží do DMS. Dále systém OCR vytěží nahraný elektronický dokument a získaná data uloží do vlastní databáze. Dále je systémem vyhodnocována shoda získaných dat OCR systémem s daty z frontendu, resp. s daty v databázi systému e-shop. Pokud jsou data shodná je dle těchto údajů upravena instance klienta v *ClientDatabase*, pokud existuje.



Obrázek 32: Sekvenční diagram NahrátDokument; zdroj [autor]

Sekvenční diagram pro metodu ověřTransakci

Systém v 10 vteřinovém cyklu (po určenou dobu) vyhledává prostřednictvím *BankAPI* detail příslušné transakce dle variabilního symbolu. Cyklus je prováděn, dokud nejsou data navracena nebo nevypršel časový limit. Po získání detailu identifikační platby jsou tato data uložena do databáze a je proveden příkaz k úhradě na stejnou částku zpět odesílateli. Následuje vyhodnocení shody dat s daty z OCR systému. Pokud jsou data shodná, je uživatel plně identifikován.



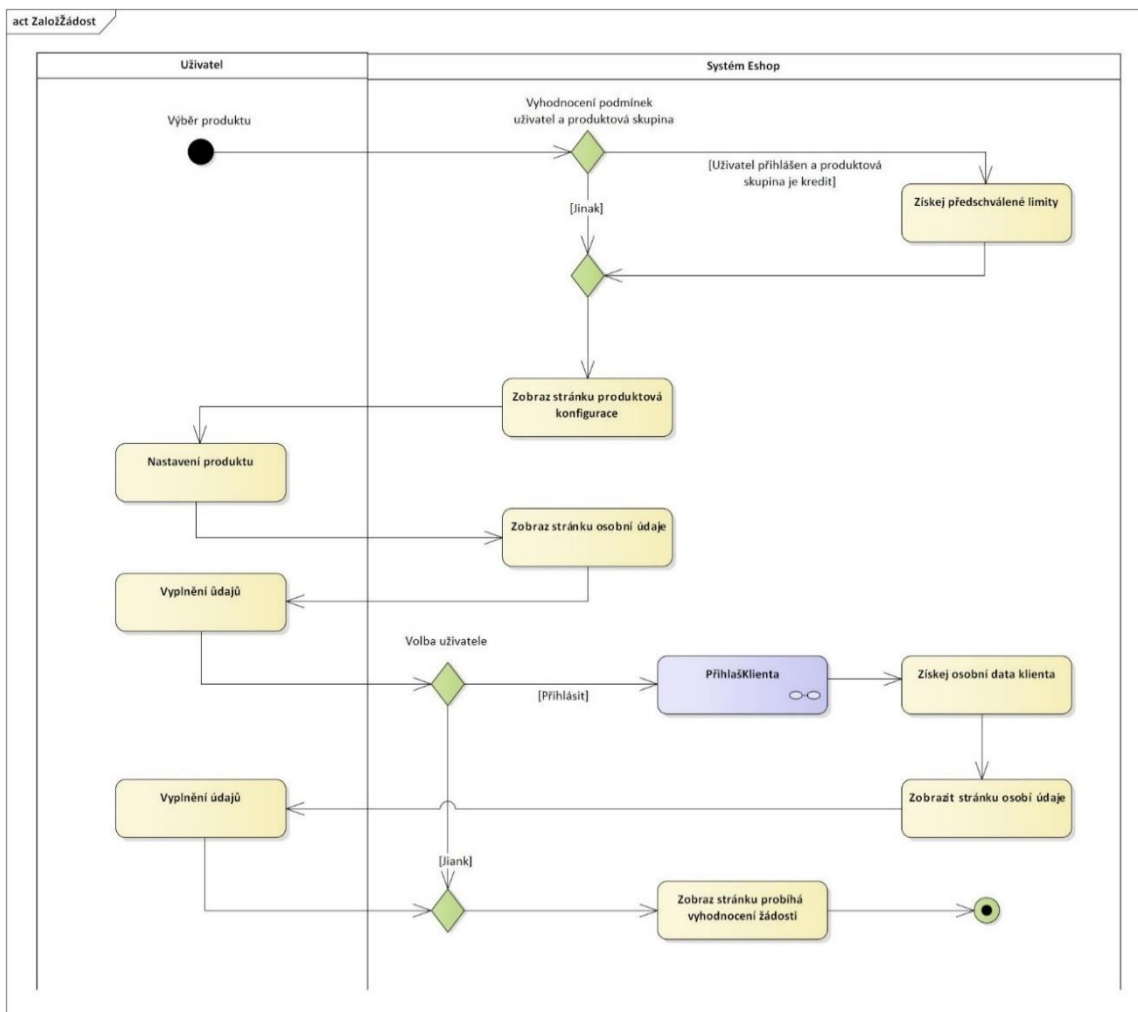
Obrázek 33: Sekvenční diagram ověřTransakci; zdroj: [autor]

4.4.4 Procesní logika

K zachycení obchodních procesů, resp. posloupnosti určitých po sobě jdoucích kroků vedoucích k určitému cíli, jsou nejčastěji používány aktivity diagramy. Diagramy aktivit jsou často vytvářeny jako doplňující prvek případů užití, který graficky znázorňuje jeho detail (tedy scénář), lze ho také použít pro modelování cest mezi případy, pak jde o tzv. stručný diagram interakcí. Na obrázku 32 a 33 je graficky zobrazen případ užití *ZaložitŽádost*, resp. *PosouditŽádost*. Scénář případu užití *ZaložitŽádost* je uveden v tabulce 2 v kapitole 4.3.4.

Diagram aktivit ZaložitŽádost

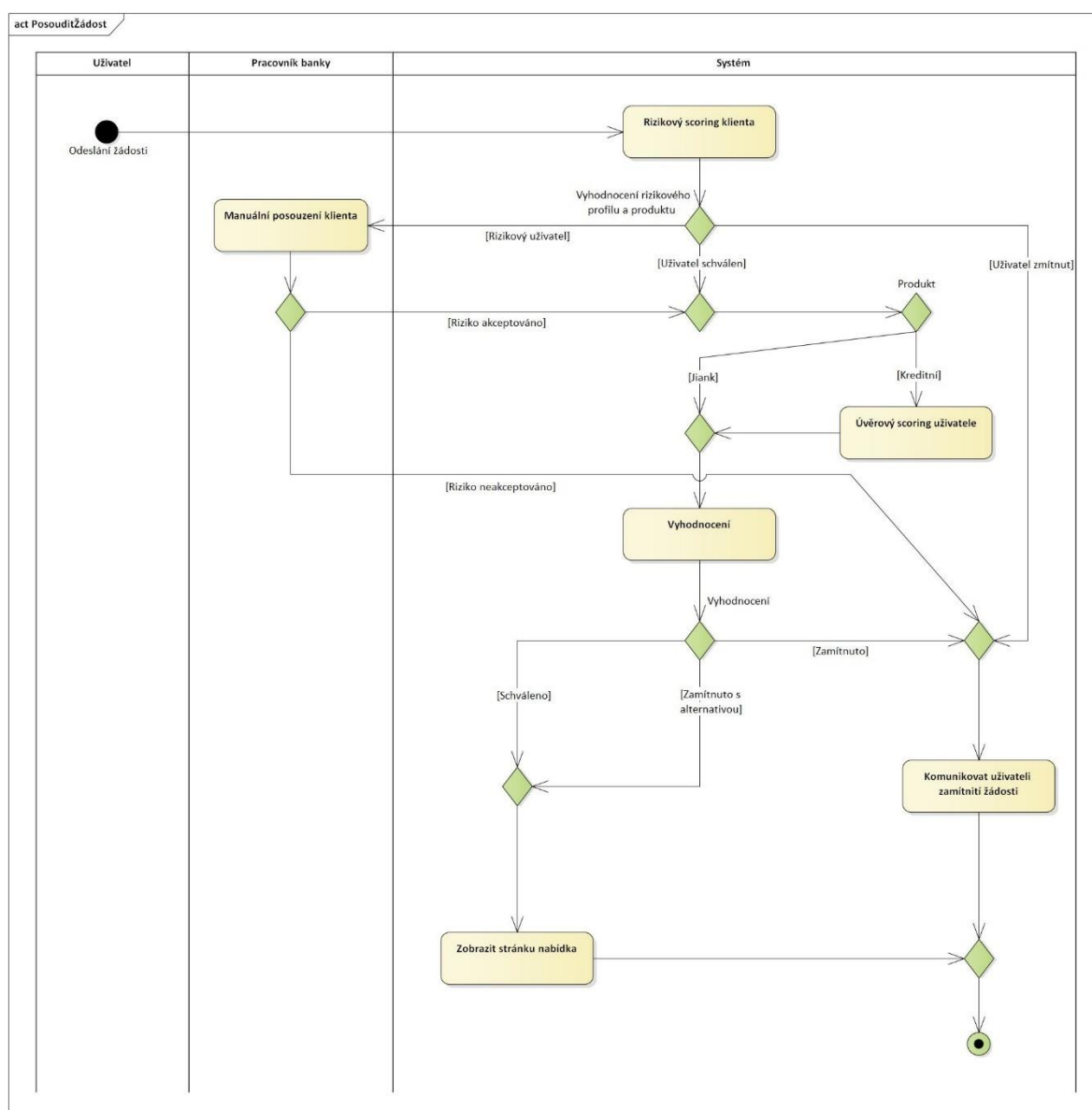
Proces založení žádosti je jednoduchá posloupnost dvou formulářů, která může být rozšířena speciálními případy. Před zobrazením prvního formuláře systém zjišťuje předschválené limity přihlášeného uživatele. Uživatel v první fázi konfigurace produktu zadá základní parametry, po odeslání formuláře následuje druhá fáze – osobní údaje. V této fázi může proběhnout přihlášení klienta, v takovém případě systém zjistí osobní údaje klienta a opět načte formulář osobní údaje s předvyplněnými daty. Odesláním tohoto formuláře přechází žádost do stavu potvrzená a je iniciováno posouzení žádosti.



Obrázek 34: Aktivita diagram případu užití ZaložitŽádost; zdroj: [autor]

Diagram aktivit PosouditŽádost

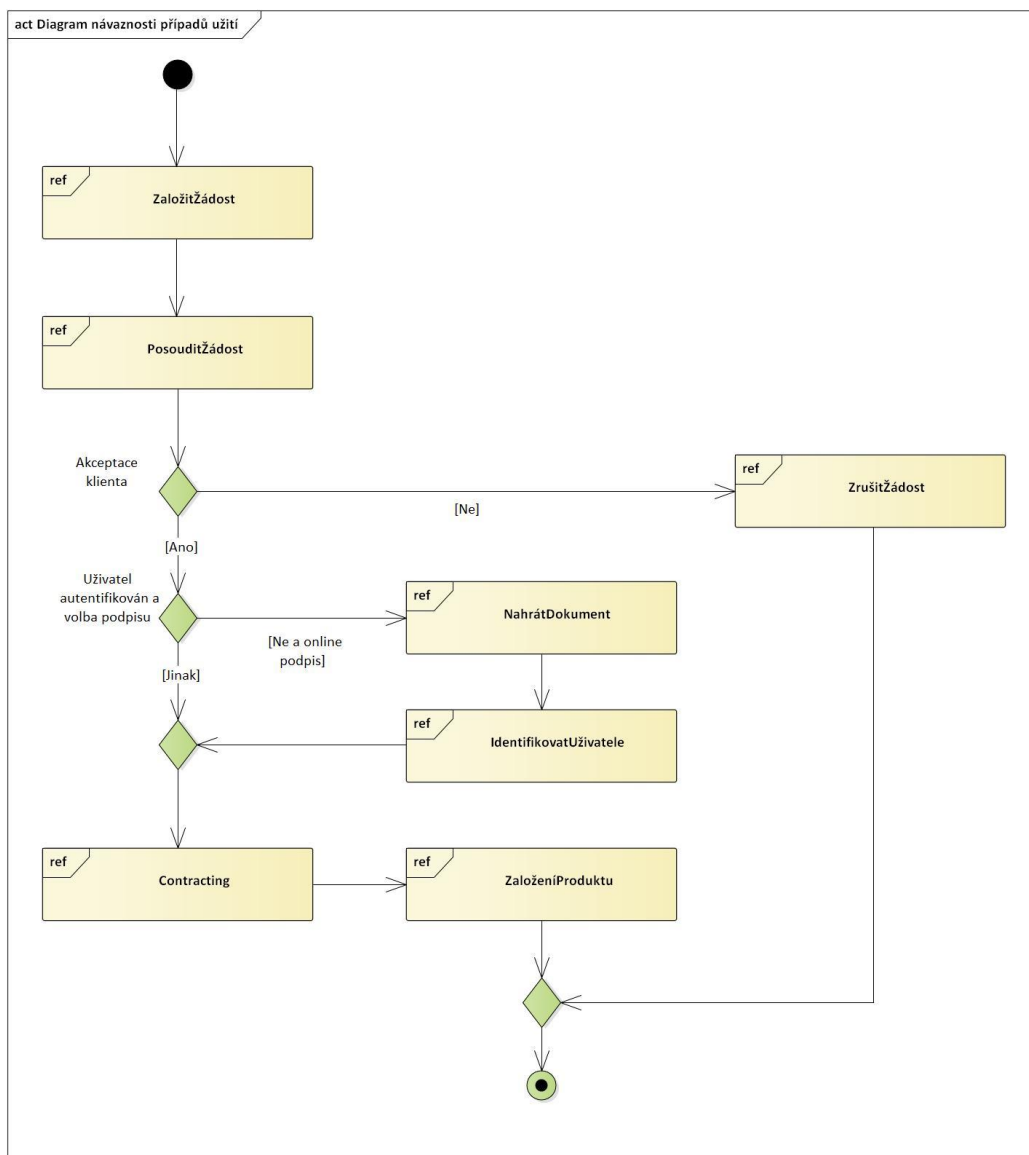
Diagram se zaměřuje na proces posouzení žádosti systému. Odesláním žádosti (poptávky po nabídce) je nastartován automatický proces vyhodnocení žádosti, který zahrnuje rizikový scoring osoby, který je nutný jak pro kreditní, tak depozitní produkty. Pokud je uživatel vyhodnocen jako přípustný, následuje úvěrový scoring. Dále je kontrolována způsobilost žadatele nabýt produkt. Na základě těchto informací je žádost vyhodnocena a výsledek je uživateli prezentován ve fázi Nabídka.



Obrázek 35: Aktivitní diagram případu užití PosouditŽádost; zdroj: [autor]

Diagram návaznosti případů užití

Tento diagram, někdy také nazýván jako Stručný diagram interakcí [1], je používán k modelování cest mezi případy užití, které zobrazují jako diagramy interakce. Pokud jsou případy užití vyjádřeny právě jako výskyty interakcí, jazyk UML umožňuje pro modelování postupu použít syntaxi diagramů aktivit. Na obrázku 36 je uveden stručný diagram interakcí systému e-shop, kde jsou případy užití znázorněny referenčními rámci sekvenčních diagramů jednotlivých výskytů. Přehledně tak zobrazuje posloupnost případů užití a v CASE nástroji poskytuje rychlou referenci k příslušné realizaci případu užití.



Obrázek 36: Diagram návaznosti případů užití; zdroj: [autor]

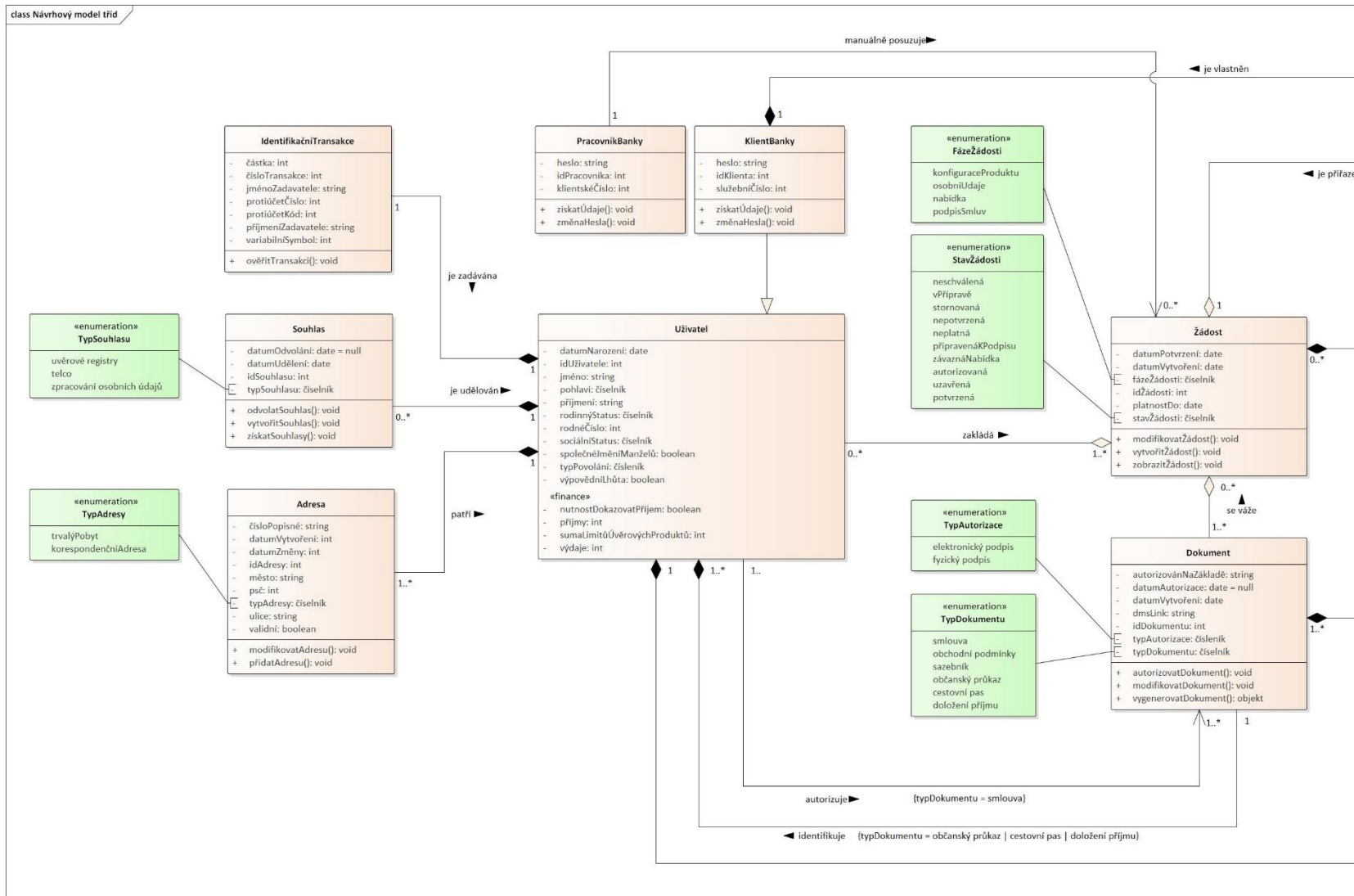
4.5 Návrh informačního systému

V projektové fázi analýzy byl vypracován logický model systému, který odráží chování systému, jaké od něj očekávají uživatelé – v případě e-shopu jsou to jeho vlastníci. Analytický model vyjadřuje, co systém bude provádět, neříká ovšem nic o tom, jak toho bude docíleno. Ve fázi návrhu je nutné specifikovat, jak bude systém požadovaného chování dosahovat, v takové podobě, kterou je vývojář schopný implementovat [1]. V této fázi jsou využity a dále upřesňovány artefakty z analytické fáze.

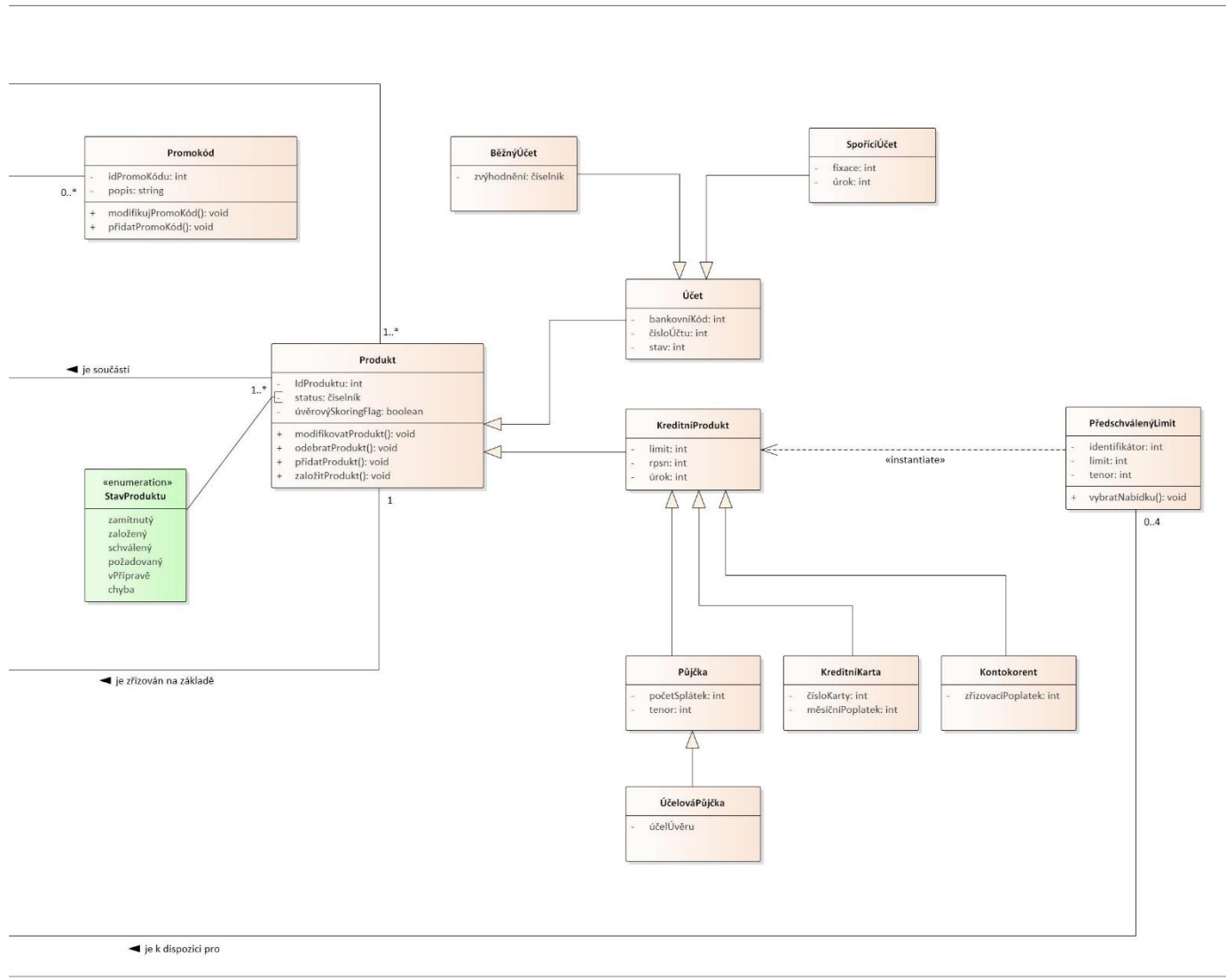
4.5.1 Návrhový model tříd

Návrhové třídy upřesňují analytické třídy do takové detailu, že je lze implementovat, mezi analytickými a návrhovými třídami tedy existuje relace typu `<<trace>>`. Návrhové třídy obsahují kompletní sadu atributů, včetně datového typu a viditelnosti a úplný rozpad operací na jednotlivé metody. Pro diagram analytických tříd je v praxi označován jako logický datový model.

Návrhové třídy systému e-shop byly získány upřesněním analytických tříd. Model byl doplněn o datové typy nezávislé na platformě, byly upřesněny metody a návratové typy a doplněny číselníky pro atributy, u kterých je to přínosem z hlediska porozumění modelu.



Obrázek 37: Class diagram návrhových tříd 1/2; zdroj [autor]

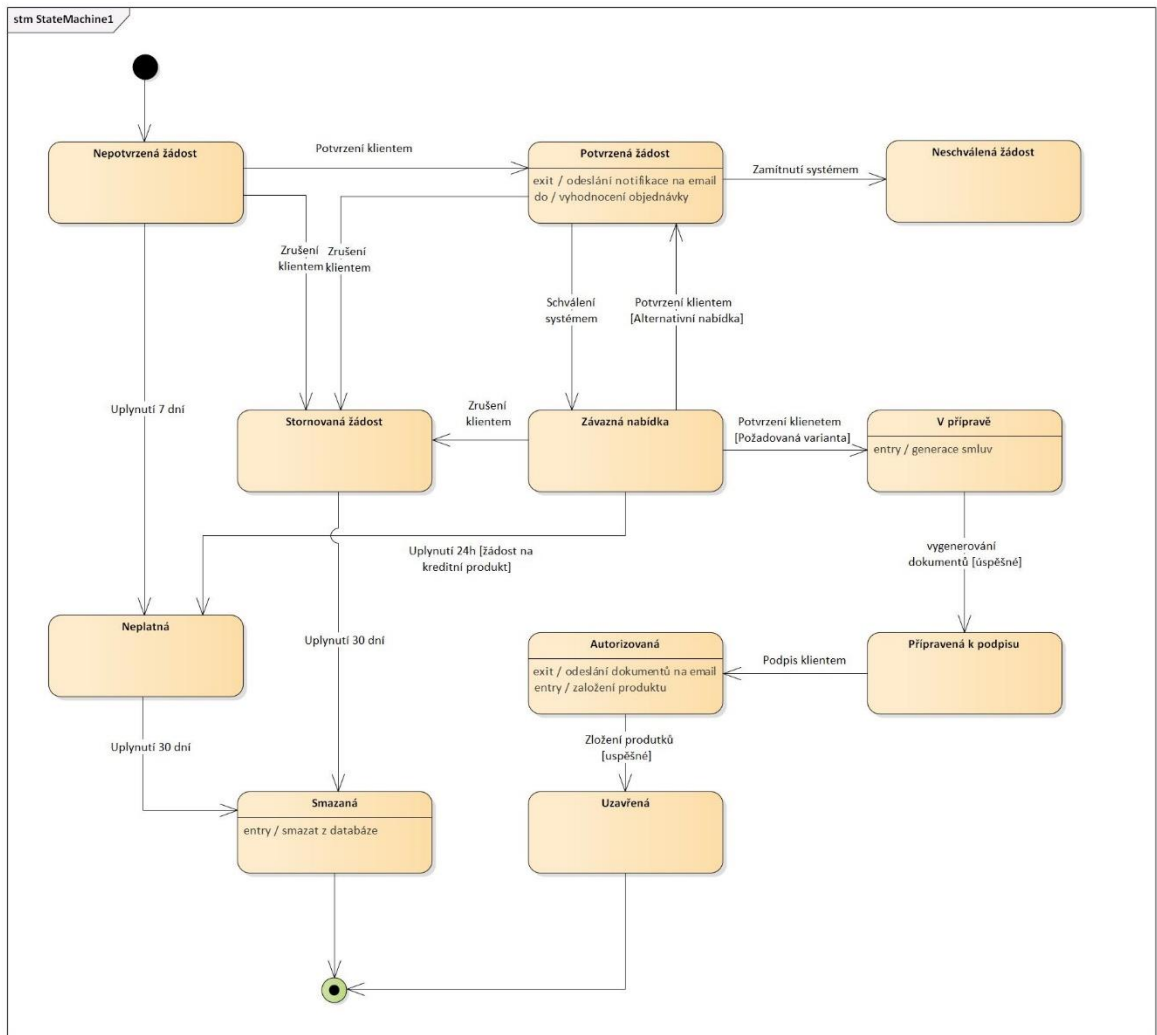


Obrázek 38: Class diagram návrhových tříd 2/2; zdroj: [autor]

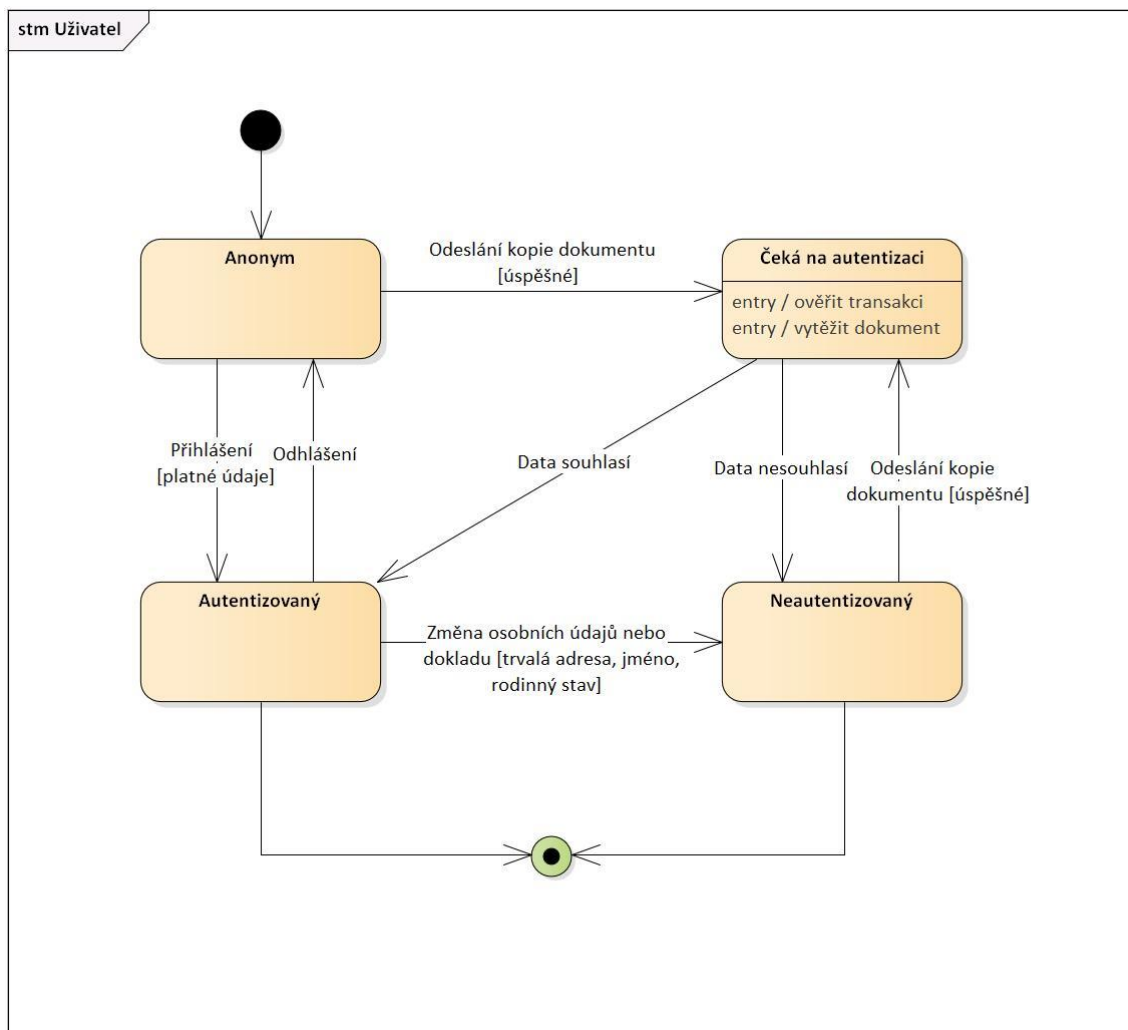
4.5.2 Stavové diagramy

Stavové diagramy jsou technikou využívanou k znázornění chování systému. Podobně jako diagramy aktivit modelují dynamické chování, vzájemně se však liší sémantikou a účelem, pro který jsou nejčastěji využívány. Jak již bylo zmíněno výše, diagramy aktivit se využívají především pro modelování obchodních procesů. Oproti tomu stavové diagramy (nebo též stavové automaty) lze velmi dobře použít pro zachycení životního cyklu objektu – třídy, která má konečný výčet stavů. Každá třída může mít jeden i více stavových diagramů. Před učiněním rozhodnutím tvorby stavového diagramu pro určitý případ je dobré zamyslet se nad otázkou, jestli diagram bude pro návrhový model přínosný nebo naopak bude spíše ztrátou času.

V navrhovaném modelu e-shopu má smysl tvorba stavového diagramu např. pro třídu *Žádost*, kde diagram velmi dobře popisuje přechody mezi stavy v rámci celého prodejního procesu. Nebo velmi jednoduchý, ale výstižný diagram pro třídu *Uživatel*, který jasně ukazuje stav uživatele z pohledu identifikace. Tento diagram byl autorem vybrán, protože ilustruje jeden ze stěžejních cílů projektu – návrh online identifikace klienta.



Obrázek 39: Stavový diagram pro třídu Žádost; zdroj: [autor]



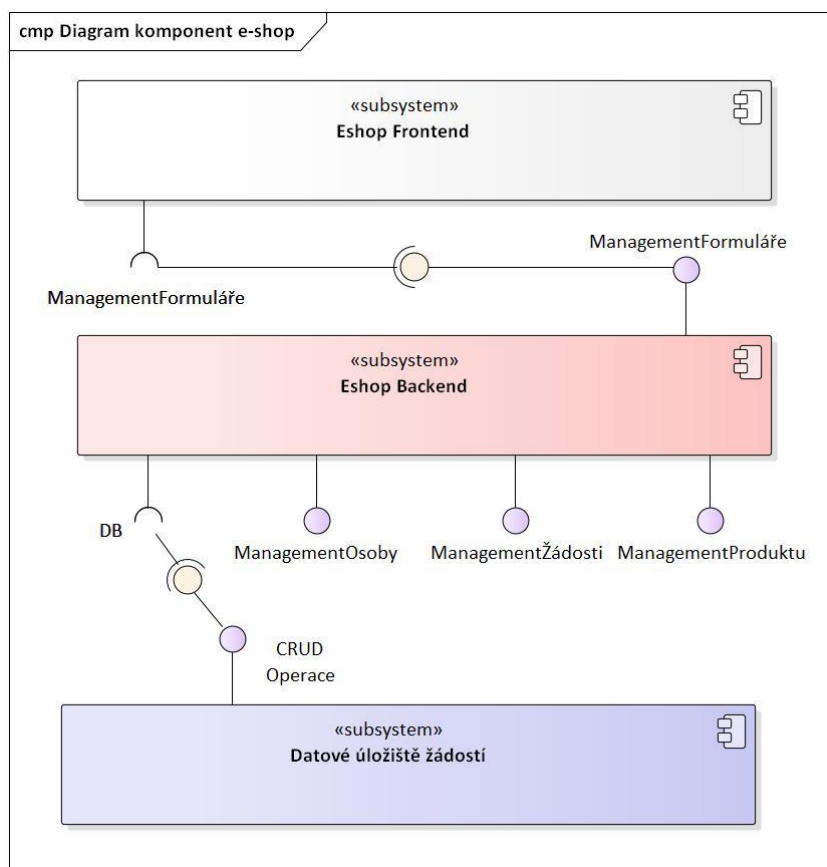
Obrázek 40: Stavový diagram pro třídu Uživatel, zdroj: [autor]

4.5.3 Diagram komponent a diagram rozmístění

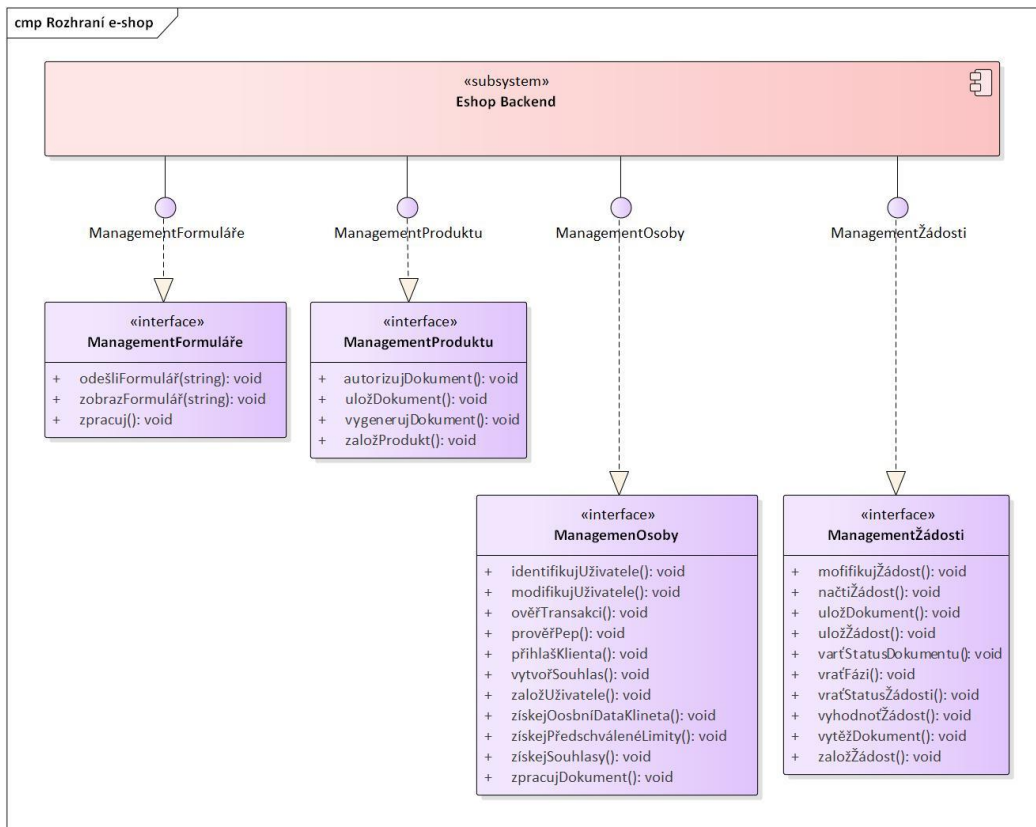
UML definuje komponentu jako modulární část systému, kterou lze znovu využít nebo vyměnit. Interakci mezi komponenty zajišťuje rozhraní, které komponenty vystavují. Komponenty zpravidla reprezentují subsystém – množinu tříd. cílem návrhu komponent je minimalizace vazeb v systému. Modelování komponent je vhodné provádět, až když jsou známy všechny vazby mezi třídami. Diagram rozmístění popisuje architekturu výsledného systému, popisuje prostředí, ve kterém bude systém fungovat, a v kombinaci s diagramem komponent modeluje vazbu mezi softwarem a hardwarem.

Diagram komponent

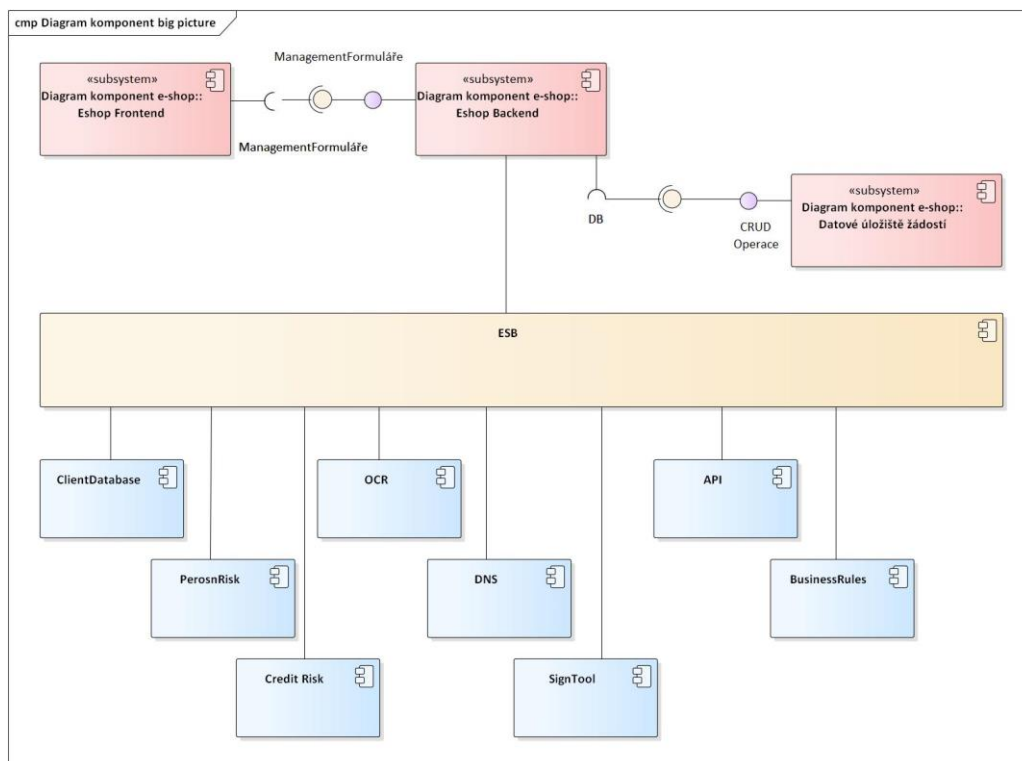
Navrhovaný systém se skládá jako většina webových aplikací z prezentační vrstvy (GUI), aplikační logiky systému a datové vrstvy. Prezentační vrstva označená jako *Eshop Frontend* představuje komunikační prostředí mezi uživatelem a systémem. Subsystem *Eshop Backend* obsahuje aplikační logiku systému. Frontend komunikuje s backendem pomocí rozhraní *ManagementFormuláře*, které realizuje metody *zobrazFormulář* a *odešliFormulář*. GUI neobsahuje žádnou další logiku. Subsystem backend poskytuje tři rozhraní *ManagementŽádosti*, *ManagementOsoby* a *ManagementProduktu*. Tato rozhraní obsluhují kompetence analytických balíčků *Žádost*, *Uživatel* a *Produkt* uvedených v kapitole 4.4.1. *Eshop Backend* vyžaduje rozhraní, pomocí kterého komunikuje s databází žádostí. Tato komunikace probíhá napřímo, nikoli přes ESB, tak aby bylo zajištěno, že žádný třetí systém nebude modifikovat *Žádosti* jinak než prostřednictvím rozhraní vystaveném na *Eshop Backendu*, který k tomu je určený.



Obrázek 41: Diagram komponent systému e-shop; zdroj: [autor]



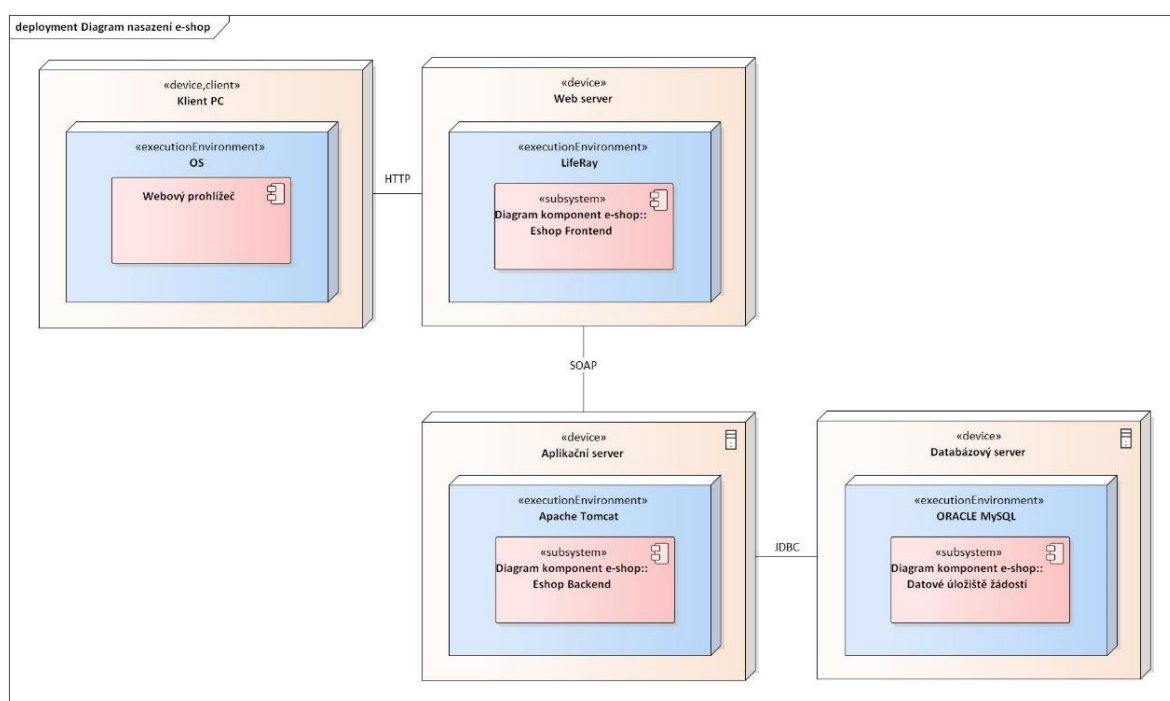
Obrázek 42: Poskytované rozhraní systémem e-shop; zdroj: [autor]



Obrázek 43: Diagram komponent systému e-shop a návazných systémů; zdroj: [autor]

Diagram rozmístění

Na obrázku 44 je zobrazena fyzická architektura systému. Kompletní HW infrastruktura zahrnuje uživatelské PC, dva HW servery a jeden databázový server. Chod aplikace je zajištěn HW servery, kde je nasazen aplikační server Apache Tomcat v případě backend části a portál LifeRay v případě frontend části. Systém využívá ORACLE MySQL databázi umístěnou na samostatném databázovém serveru. Uživatel přistupuje na stránky e-shopu pomocí webového prohlížeče, e-shop. Aplikace navzájem komunikují pomocí webových služeb.

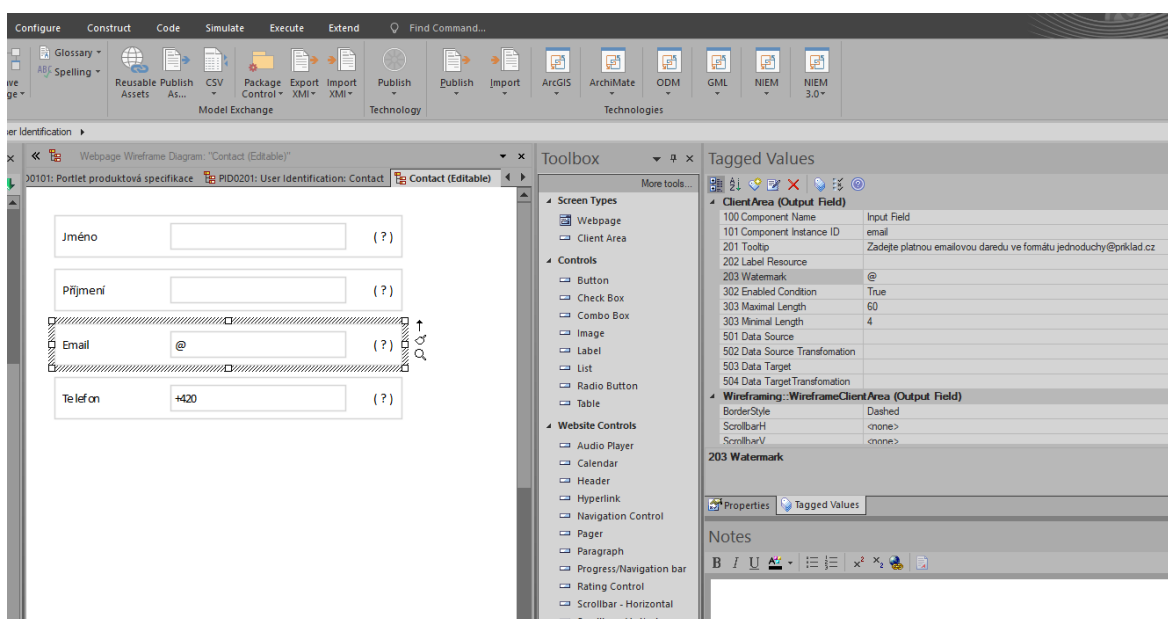


Obrázek 44: Diagram rozmístění; zdroj: [autor]

4.5.4 Wireframes

Wireframe (česky drátěný model) webu je jedním z artefaktů návrhu systému, který definuje rozmístění prvků a funkcionalit na jednotlivých stránkách. Po tvorbě skic jednotlivých stránek jsou dále definovány přechody mezi nimi pomocí navigační matice – tzv. mapy webu.

Wireframes jsou použitelné ve více fázích specifikace systému. V analytické fázi lze pomocí jednoduchého modelu stránky získat přehled o analytických třídách a jejich atributech. Pomocí wireframes si koncoví uživatelé dokáží lépe představit modelovaný systém a následnou práci s ním. V CASE nástrojích lze pomocí wireframes velmi dobře specifikovat zadání pro vývojáře frontendu. Pomocí označených hodnot (tagged values) definovaných zvlášť pro každý prvek na obrazovce lze definovat např. datové zdroje a cíle – xpath elementu služby nebo číselník, transformací dat, formátování, řazení položek v combu, defaultní hodnoty, validace a operace atp. Takto specifikovaný wireframe zároveň slouží jako jedna z částí dokumentace systému, kterou využívají např. testéři při tvorbě testovacích scénářů.



Obrázek 45: Využití označených hodnot při tvorbě wireframes; zdroj: [autor]

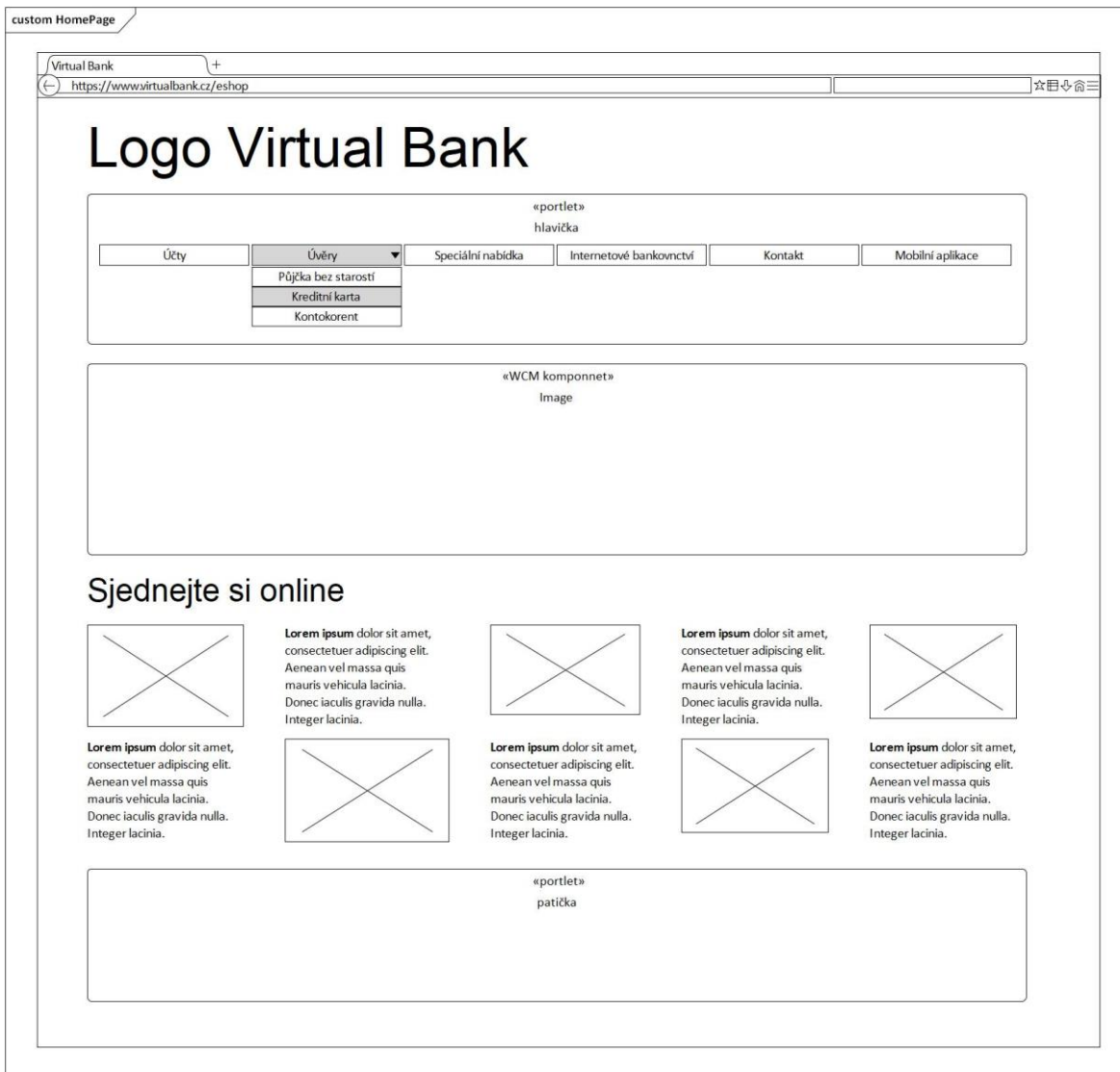
Příklady návrhů obrazovek jsou připojeny níže. Modelované diagramy zobrazují portlety¹⁶ formulářů zajišťujícího prodejní proces. Samotný web se skládá z více komponent např. hlavičky a patičky, navigačního menu, různých WCM komponent a dalších portletů – např.

¹⁶ Portlety jsou webové komponenty které jsou používány jako výměnné části uživatelského rozhraní představující prezentační vrstvu informačního systému komunikující pomocní API.

formuláře zajišťujícího zpětné zavolání nebo chat s podporou. Kompletní sada wireframes je uvedena v příloze 8.6.

Návrh domovské obrazovky frontendu

Domovská obrazovka e-shopu se skládá z několika komponent. První z nich je hlavička, která obsahuje vertikální navigační menu. Následuje kampaňová WCM komponenta (statický obrázek), která je obměňována na toho, na jakém produktu se právě uživatel nachází kurzorem. Pod WCM komponentou je hlavní sekce prezentující vybrané primární produkty. Poklikem na hyperlink nebo ikonu reprezentující produkt je iniciován prodejní proces a uživatel je přesměrován na stránku specifikace produktu, jejíž wireframe je pro neúčelový spotřebitelský úvěr uveden na obrázku č. 47.



Obrázek 46: Wireframe domovské obrazovky; zdroj: [autor]

custom PID0101: Portlet produktová specifikace

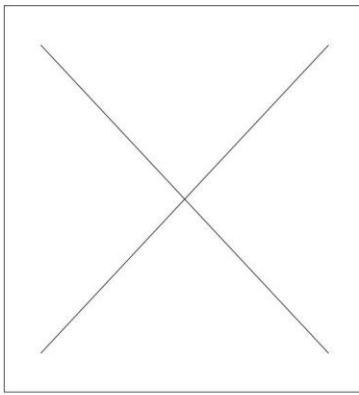
☰ ✕

Půjčka bez starostí

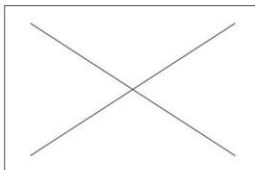
custom Progress Bar nastavte si produkt

Nastavte produkt > Osobní údaje > Nabídka > Identifikace > Podpis smluv

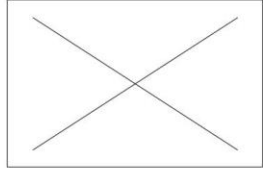
custom Motiv Kampaně



Aliquam ornare wisi eu metus. Aenean vel massa quis mauris vehicula lacinia. Aliquam erat volutpat. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nunc tincidunt ante vitae massa. Quisque porta.



Nulla quis diam. Nullam justo enim, consectetur nec, ullamcorper ac, vestibulum in, elit. Donec quis nibh at felis congue commodo. Praesent id justo in neque elementum ultrices. Nulla turpis magna, cursus sit amet, suscipit a, interdum id, felis.



custom Předschválené limity

Máme pro Vás připraveny následující nabídky

Pokud si vybere jednu z následujících variant, pak Vás od načerpání financí dělí jen 10 minut!

<input type="radio"/> 50 000 CZK <small>za 3 650 CZK na 36 měsíců při úroku 8,1% p.a. a RPSN 9,2%</small>	<input type="radio"/> 200 000 CZK <small>za 3 650 CZK na 36 měsíců při úroku 8,1% p.a. a RPSN 9,2%</small>	<input type="radio"/> 400 000 CZK <small>za 3 650 CZK na 36 měsíců při úroku 8,1% p.a. a RPSN 9,2%</small>	<input type="radio"/> 600 000 CZK <small>za 3 650 CZK na 36 měsíců při úroku 8,1% p.a. a RPSN 9,2%</small>
---	--	--	--

custom Úvěrová kalkulačka

Půjčte si přesně tolik kolik potřebujete...

Měsíční splátka

5 650 CZK

Úroková míra 8,9% p.a.
RPSN 9,9%

Výše půjčky

◀

|||

▶

50 000 CZK 1 000 000 CZK

Jak dlouho chcete splácet?

◀

|||

▶

12 měsíců 96 měsíců

Vzorový příklad Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce consectetur risus a nunc. Nam sed tellus id magna elementum tincidunt. Morbi scelerisque luctus velit. Et harum quidem rerum facilis est et expedita distinctio. Nullam at arcu a est sollicitudin euismod.
Disclaimer Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Sed conwallis magna eu sem. Etiam dicitur tincidunt diam. Morbi leo mi, nonummy eget tristique non, rhoncus non leo.

Obrázek 47: Wireframe formuláře specifikace produktu půjčka; zdroj: [autor]

5 Výsledky a diskuse

5.1 Zhodnocení návrhu aplikace

Vzhledem k faktu, že samotná implementace systému není součástí práce, není možné plně zhodnotit jakost softwaru např. pomocí norem ISO/IEC 9126 nebo *ISO/IEC 14598 Softwarové inženýrství – hodnocení softwarového produktu*. Lze ale hodnotit ty části systému, které jsou v práci reflektovány. Ke zhodnocení software byl vybrán model FURPS, který byl vyvinut společností Hewlett-Packard. Tato metoda specifikuje, jaké oblasti by měly být testovány a definuje tak pět dimenzí: funkčnost, užitečnost, spolehlivost, výkon a rozšiřitelnost. Dimenze spolehlivost a výkon nejsou níže rozebírány vzhledem k tomu, že systém nebyl implementován.

Funkčnost

Ověření funkčnosti systému spočívá v ověření zpracování všech nezbytných a možných funkčních požadavků na systém. K tomuto lze velmi dobře použít funkce CASE nástroje Enterprise Architect, v případě, že v průběhu analýzy a návrhu bylo provedeno slevování funkčních požadavků ve vztahu k případům užití (viz kapitola 4.3.5), pomocí vztahů (funkce „*Relationships*“) lze v balíčku funkčních požadavků najít ty požadavky, které nejsou realizovány žádným případem užití. Z ověření vyplývá, že aplikace byla navržena v souladu s požadavky, každý z požadavků je reflektován případem užití a řešení se tak zdá být plně funkční. Naplnění systémových požadavků akceptují vlastníci jednotlivých požadavků po UAT dle definovaných akceptačních kritérií, které jsou součástí správné definice systémového požadavku.

Použitelnost

Použitelnost systému hodnotí systém ze strany koncového uživatele. Jedná se o hodnocení přehlednosti, přístupnosti a jednoduchosti prezentační zejména prezentační vrstvy aplikace. Obrazovky formuláře byly koncipovány s maximálním důrazem na jednoduchost a přehlednost. Každý formulář se skládá z několika sekcí, které respektují členění nadpisů – což umožňuje orientaci uživatelů používající čtečky obsahu webu z důvodu zrakového postižení. Na každém formuláři je umístěn orientační progres bar informující uživatele,

v jaké fázi procesu se nachází a jaká fáze bude následovat. Pod progres barem ním je umístěn infopanel, do kterého mohou být jednoduše umístěny rady uživatelů. V průběhu testování pravděpodobně vyplynou drobné nedostatky. Veškerá dokumentace, který se týká frontendu je zanesena ve wireframes.

Rozšiřitelnost

Jedním z požadavků bylo vytvoření takového systému, který by mohl v budoucnosti jednoduše obsluhovat více frontendu – například e-shop v rámci smartbankingu nebo nového pobočkového systému. Na tomto základě byl specifikován 5 fázový univerzální prodejní proces požadující pouze nezbytně nutné informace. Veškerá aplikační logika je uložena v subsystému *Eshop Backend*. Frontend je řízen tímto procesem definovaným na backendu, frontend tak vlastně neví, jaká fáze (obrazovka) má následovat. S aplikační logikou komunikuje pomocí jednoho rozhraní. Jedna metoda vrací data pro obrazovku (*zobrazFormulář*) a druhá ukládá data do databáze, provádí orchestrace a v odpovědi na frontend vrací zpět ID stránky, o kterou má následně požádat (*odešliFormulář*). Takovéto řešení je připraveno na napojení jakékoli další aplikace, která bude komunikovat stejným způsobem.

Nevýhodou řešení je komplexnost logiky vyhodnocení následujícího kroku. Případné změny jako je vložení další fáze do procesu, vyžadují zásah, který nemusí být triviální. Řešení tak klade značné nároky na dokumentaci systému.

5.2 Diskuse

Jak již bylo zmíněno v kapitole *Rozšiřitelnost*, změny na úrovni orchestrační služby, který vyhodnocuje současnou a následující fázi objednávky může být komplikované, a to nejen z pohledu implementace. Toto lze ilustrovat např. na teoretickém změnovém požadavku, který vyžaduje, aby bylo uživateli znemožněno založit v systému e-shop více než jednu žádost pro kreditní produkt. Podobné omezení sice již existuje na systému *CreditRisk*, nicméně je vyhodnoceno až při pokusu zapsat žádost do bankovních registrů, kde je žádost odmítnuta, pokud je blokována jinou, již existující. Důvodem je zpoplatnění rozhraní bankovního registru. Pro autentizovaného uživatele se požadavek nejeví jako problém,

system v tomto případě musí před založením žádosti v DB provést dotaz do DB, a pokud existuje nějaká žádost v definovaném stavu ze skupiny kreditních produktů, pak nelze založit další žádost – pouze pokračovat v původní. Zbývá jen vyřešit kam umístit logiku vyhodnocení navzájem se blokujících žádostí, tak aby nebyla uvedena přímo v kódu. Problémem je ale případ, kdy proces prochází ve větvi pro neautentizovaného klienta – v tomto případě dle návrhu probíhá uložení žádosti a klienta ještě před jeho autentizací (jedná se o tzv. lead) kde jsou k dispozici pouze kontaktní data, na kterých nelze provádět efektivně kontrolu. Posunutím ukládání do pozdější fáze pak společnost přichází právě o tyto kontakty na potencionální klienty, kteří konfiguraci žádosti vzdají dříve, než žádost projde uložení kontaktních údajů. Závažnějším problémem může být bezpečnostní riziko – pokud by systém upozornil uživatele, že již má ve společnosti nedokončenou žádost ještě dříve, než se uživatel autentizuje, nelze mít jistotu, že tyto informace nemohou být zneužity. Žádné řešení se nezdá být zcela správné v současné formě procesu.

Část práce, která nastiňuje řešení autentizace klienta je považována za dílčí cíl práce. Navržené řešení splňuje definované zadání – tedy online autentizaci klienta, avšak má určitá úskalí. Prvním z nich je fakt, že není k dispozici posudek soudního znalce, který by potvrzoval legitimitu autentizace uživatele tzv. převzetím identity od třetího subjektu. Toto je riziko, které je buď nutné akceptovat nebo iniciovat vydání posudku. Další z možných slabin řešení je skutečnost, že návrh je závislý na rozhraní dalších obchodních bank. Bez dostatečně rychlé odezvy v rámci desítek sekund by nebylo řešení použitelné. Společnost tak musí mít sjednanou kvalitní smlouvu o podpoře. Bez povšimnutí by neměl také zůstat fakt, že definice dostatečně robustního algoritmu, který bude schopen spolehlivě rozpoznat všechny relevantní typy platných identifikačních dokumentů může být poměrně náročné na zdroje. Zajímavým rozšířením v dalších verzích aplikace by mohlo být z pohledu online autentizace např. zapojení generátoru QR kódů pro předvyplnění transakce ve smartbankingu nebo zapojení platební brány pro online platby kartou.

Z pohledu analýzy je důležité také upozornit na možnou problémovou část aplikace, kterou může představovat skutečnost, že systém nějak neověřuje dokument dokládající příjem klienta (výpis z účtu) a pouze ho uloží do *DMS*. Aplikace využívá současný proces na

systemu *CreditRisk*, který vytěžuje dokument až v momentu, kdy je iniciován scoring. V případě, kdy uživatel omylem nahraje nevyhovující dokument je žádost zamítnuta. Uživatel samozřejmě má možnost nahrát nové dokumenty a opakovat požadavek ale zamítnutí žádosti je zapsáno do bankovních registrů a tato skutečnost snižuje úvěrový rating klienta. V následující verzi aplikace by bylo dobré zvážit možnost implementace vytěžení a kontroly dokumentu ještě před založením úvěrové žádosti v systému *CreditRisk* – otázkou je návratnost nákladů na změnový požadavek, kterou nelze předjímat bez znalosti produkčních dat.

6 Závěr

Cílem této práce byla analýza a návrh informačního systému, který je využitelný pro prodej bankovních produktů, pomocí jazyka UML. V teoretické části byla na základě studia odborné literatury a dalších dostupných zdrojů sumarizována teoretická východiska práce. Byly přiblíženy některé metodiky vývoje informačních systémů, klasické i agilní, přístupy k analýze a syntaxe jazyka UML. Na základě autorovy pracovní zkušenosti z bankovního prostředí byly zevrubně popsány systémy nezbytné pro chod komerčních bank, které se soustředí především na retailovou klientelu. Dále byla přiblížena významná legislativní opatření, která musí být při analýze a návrhu brána v úvahu. Takto strukturovaná literární rešerše slouží jako nezbytný rámec pro uchopení praktické části.

V praktické části byla provedena analýza a návrh informačního systému dle metodiky Unified Process. V úvodní kapitole byly upřesněny výchozí předpoklady a identifikován subjekt, pro který je nový systém navrhován. Byly charakterizovány systémy, se kterými modelovaný systém komunikuje, produkty, které zadavatelský subjekt zamýšlí poskytovat na online kanále a specifikován prodejní proces skupiny kreditních produktů.

Analytická fáze začíná sběrem požadavků na systém, který byl definován na základě autorovy zkušenosti z obdobného projektu v reálné společnosti působící na českém bankovním trhu. Následně byly požadavky analyzovány, byla jim přiřazena priorita a byly roztrženy na funkční a nefunkční systémové požadavky. V návazné činnosti byli identifikováni aktéři systému. Dalším rozbořením systémových požadavků byly získány

primární případy užití a vytvořen model případů užití, který graficky reprezentuje očekávané funkce systému. Dalším artefaktem fáze analýzy jsou analytické třídy, které byly získány pomocí metody analýzy podstatných jmen a sloves a zachyceny v Class diagramu. K tomuto modelu tříd byl vytvořen datový slovník. Dále byly vybrané případy užití rozpracovány do detailu pomocí scénářů případu užití. Pro znázornění dynamického chování systému byly v rámci specifikace realizace případů užití použity sekvenční diagramy. K objasnění procesní logiky byly použity diagramy aktivit.

Ve fázi návrhu jsou upřesněny atributy, operace a vazby analytického modelu tříd. Na tomto základě byl vytvořen návrhový Class diagram. V návaznosti na tuto aktivitu byly vytvořeny stavové diagramy, které zpřehledňují životní cyklus hlavních tříd navrhovaného systému. Dále byly definovány komponenty systému a poskytované a požadované rozhraní komponent. Graficky byl znázorněn celkový pohled na systém a na vztah mezi ním a již existujícími bankovními systémy. Dále práce obsahuje technické návrhy obrazovek, které slouží jako prostředek validace GUI mezi analytiky a koncovými uživateli a zároveň jako jedna z částí dokumentace systému. V závěrečné fázi návrhu byl vytvořen diagram nasazení, zachycuje fyzickou architekturu systému – vztah mezi hardwarem a softwarem. V závěrečné části práce je hodnocen systém z pohledu funkčnosti, použitelnosti a rozšiřitelnosti a byla zde otevřena diskuse týkající se výhod a nevýhod řešení z pohledu připravenosti systému na změny.

Dílním cílem práce bylo přestavení řešení online autentizace uživatele. Autor představil nástin možného řešení pomocí převedení ověřené identity uživatele od třetího subjektu – jiné bankovní společnosti, využitím OCR pro vytěžení dat z identifikačního průkazu uživatele a algoritmu pro vyhodnocení. V teoretické rovině se řešení systému zdá být použitelné, má však několik úskalí. Jedním z nich přímá závislost na systémech ostatních bank, resp. jimi poskytovaných rozhraní – zde je potřeba dojednat kvalitní SLA a spolehlivost OCR systému. Natrénování neurální sítě systému tak, aby spolehlivě rozpoznala všechny typy platných dokladů může být ve výsledku velmi náročné na zdroje. V rámci rozsahu práce však dle autora lze považovat dílní cíl za splněný.

Hlavním cílem práce byla analýza a tvorba návrhu informačního systému dle definovaných požadavků. S ohledem na omezený rozsah této práce nelze považovat za návrh systému za kompletní, nejsou rozpracovány všechny případy užití do takového detailu, který by umožňoval implementaci systému. Nicméně primární případy užití – založení žádosti, posouzení žádosti a autentizace uživatele jsou popsány v dostatečném detailu, proto lze dle autorova názoru považovat i tento cíl za splněný.

Dle autora spočívá hlavní přínos práce v přiblížení fungování banky z IT pohledu, které je v praxi využitelné například pro analytiku, kteří se v tomto prostředí dříve nepohybovali a dává jim tak ucelený pohled na doménu online prodeje bankovních produktů. Dokument nastiňuje hlavní problémy, které je při vývoji systému sloužícímu pro prodej těchto specifických produktů a služeb potřeba řešit a podává návrh jejich řešení – i když často pouze na poměrně vysoké úrovni detailu. V podnikové praxi lze praktickou část diplomové práce považovat za dokument podobný tzv. *solution screeningu* – tedy dokumentu vznikajícím v předprojektové fázi kontroly proveditelnosti, který charakterizuje hlavní problémy a jejich řešení. Tento dokument poté slouží jako vstup pro IT analytiku, kteří jednotlivá řešení rozpracují do úrovně detailu umožňující implementaci.

7 Seznam použitých zdrojů

- [1] ARLOW, Jim a Ila NEUSTADT. UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
- [2] FOWLER, Martin. Destilované UML. Praha: Grada, 2009. Knihovna programátora (Grada). ISBN 978-802-4720-623.
- [5] RICHTA, Karel. Metodiky vývoje software, MDA [online]. Praha, 2011 [cit. 2017-12-02]. Dostupné z: <https://edux.fit.cvut.cz/oppa/BI-SI1/prednasky/BI-SI1-P10m.pdf>. Přednáška. České vysoké učení technické v Praze.
- [6] KANISOVÁ, Hana a Miroslav MÜLLER. UML srozumitelně. 2., aktualiz. vyd. Brno: Computer Press, 2006. ISBN 80-251-1083-4.
- [7] ŠMÍD, Vladimír. MDIS – Multidimensional Development of Information System [online]. Brno, 2005 [cit. 2017-12-02]. Dostupné z: <http://www.fi.muni.cz/~smid/mis-mdis.htm>. Učební materiál. Masarykova Univerzita.
- [8] BRUCKNER, Tomáš. Tvorba informačních systémů: principy, metodiky, architektury. Praha: Grada, 2012. Management v informační společnosti. ISBN 978-802-4741-536.
- [9] SOMMERVILLE, Ian. Softwarové inženýrství. Brno: Computer Press, 2013. ISBN 978-802-5138-267.
- [10] ŠMÍD, Vladimír. Management informačního systému [online]. Brno, 2005 [cit. 2017-12-02]. Dostupné z: <http://www.fi.muni.cz/~smid/managis.html>. Výukový materiál. Masarykova Univerzita
- [11] RICHTA, Karel a Jiří SOCHOR. Softwarové inženýrství I. Dot. 1. vyd. Praha: České vysoké učení technické, 1998. ISBN 80-010-1428-2.
- [12] SLA (Service Level Agreement). In: ManagementMania.com [online]. Wilmington (DE) 2011-2017, 02.12.2015 [cit. 02.12.2017]. Dostupné z: <https://managementmania.com/cs/service-level-agreement>
- [13] LEFFINGWELL, Dean. Agile software requirements: lean requirements practices for teams, programs, and the enterprise. Upper Saddle River, NJ: Addison-Wesley, c2011. Agile software development series. ISBN 03-216-3584-1.

- [14] Rational Unified Process: Best Practices for Software Development Teams. In: Rational Library [online]. CA 95014: Rational Software White Paper, 1998 [cit. 2017-12-02]. Dostupné z: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf
- [15] SCHMULLER, Joseph. Myslíme v jazyku UML. Praha: Grada, 2001. Knihovna programátora (Grada). ISBN 80-247-0029-8.
- [16] RUMBAUGH, James., Ivar. JACOBSON a Grady. BOOCH. The unified modeling language reference manual. 2nd ed. Boston: Addison-Wesley, c2005. ISBN 03-212-4562-8.
- [17] Nejlepší praxe (Best Practice). In: Management Mania [online]. 2015 [cit. 2017-12-03]. Dostupné z: <https://managementmania.com/cs/nejlepsi-praxe-best-practice>
- [18] Camel Case. IT Slovník [online]. 2017 [cit. 2017-12-04]. Dostupné z: <https://it-slovník.cz/pojem/camelcase>
- [19] MACHAČ, Otakar. Reporting jako součást informačního systému podniku. SystemOnLine [online]. 2003, 2003(12), 1 [cit. 2018-02-03]. Dostupné z: <https://www.systemonline.cz/clanky/reporting.htm>
- [20] BASL, Josef a Roman BLAŽIČEK. Podnikové informační systémy: podnik v informační společnosti. 2., výrazně přeprac. a rozš. vyd. Praha: Grada, 2008. Management v informační společnosti. ISBN ISBN978-80-247-2279-5.
- [21] DVOŘÁK, Petr. Bankovníctví pro bankéře a klienty. Praha: Linde, 2005. Vysokoškolská učebnice (Linde). ISBN 80-720-1515-X.
- [22] Centrální registr úvěrů [online]. Praha: ČNB, 2009 [cit. 2018-02-03]. Dostupné z: https://www.cnb.cz/cs/dohled_financni_trh/centralni_registr_uveru/
- [23] Microsite. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-02-04]. Dostupné z: <https://cs.wikipedia.org/wiki/Microsite>
- [24] ADASTRA. Integrace firemních systémů: Proč a jak na to? ADASTRA [online]. 2014, 2014(1), 1 [cit. 2018-02-04]. Dostupné z: <http://www.adastra.cz/clanky/integrace-firemnic-systemu-proc-a-jak-na-to>
- [25] RYCHLÝ, Marek. Servisně orientovaná architektura a její aplikace v systémech sledování a řízení výroby [online]. Brno, 2011 [cit. 2018-02-04]. Dostupné z:

- <http://www.fit.vutbr.cz/~rychly/public/docs/arap11.soa-a-jeji-apl-pri-rizeni-vyroby/arap11.soa-a-jeji-apl-pri-rizeni-vyroby.pdf>. Elektronický článek. VUT v Brně.
- [26] API. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-02-04]. Dostupné z: <https://cs.wikipedia.org/wiki/API>
- [27] TelcoScore [online]. Praha: SID.CZ, 2016 [cit. 2018-02-04]. Dostupné z: <http://www.sid.cz/telco-score>
- [28] Zákon č. 253/2008 Sb [online]., o některých opatřeních proti legalizaci výnosů z trestné činnosti a financování terorismu. In: Sbírka zákonů. 2008. Dostupné z: <https://www.zakonyprolidi.cz/cs/2008-253> § 4, odst. 5. Ve znění pozdějších předpisů. Dostupné z: <http://www.psp.cz/sqw/text/tiskt.sqw?O=7&CT=752&CT1=0>
- [29] Big Brother Awards: Lidská práva a technologie. Slídivové.cz [online]. ? : ?, 2018 [cit. 2018-02-19]. Dostupné z: <http://www.slidilove.cz>
- [30] Úřad pro ochranu osobních údajů: Základní příručka k GDPR [online]. Praha: -, - [cit. 2018-02-22]. Dostupné z: <https://www.uoou.cz/zakladni-prirucka-k-gdpr/ds-4744/archiv=0&p1=3938>
- [31] GDPR prakticky [online]. -: -, 2017 [cit. 2018-02-22]. Dostupné z: <https://www.gdpr.cz/gdpr/povinnosti/>
- [32] CamelCase. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-02-27]. Dostupné z: https://en.wikipedia.org/wiki/Camel_case

8 Přílohy

8.1 Systémové požadavky

custom Funkční požadavky	
Systém umožní klientovi banky přihlášení pomocí přihlašovacích údajů do IB.	tags
Aktér = Klient ID = F01 Priorita = Nezbytný	
Systém zobrazí předsválené limity kreditních produktů.	tags
Aktér = Klient ID = F02 Priorita = Nezbytný	
Systém identifikuje upsell produktu.	tags
Aktér = Klient ID = F03 Priorita = Možný	
Systém umožní vyhledání objednávky.	tags
Aktér = Klient ID = F04 Priorita = Možný	
Systém předvyplní klientovi dostupná osobní data.	tags
Aktér = Klient ID = F05 Priorita = Možný	
Systém umožní online autentizaci neklienta banky.	tags
Aktér = Uživatel ID = F06 Priorita = Nezbytný	
Systém umožní cross-sale prodej pojištění u kreditních produktů.	tags
Aktér = Uživatel ID = F07 Priorita = Nezbytný	
Systém umožní následující způsoby podpisu smluvní dokumentace: podpis pomocí online SMS, podpis prostřednictvím kurýrní služby, podpis na poboce banky.	tags
Aktér = Uživatel ID = F08 Priorita = Nezbytný	
Systém umožní prodej následujících produktů: neúčelový úvěr, účelový úvěr, kontokorent, kreditní karta, spořicí účet a běžný účet.	tags
Aktér = Uživatel ID = F09 Priorita = Nezbytný	
Systém umožní uložení objednávky.	tags
Aktér = Uživatel ID = F10 Priorita = Nezbytný	
Systém umožní pokračovat v rozpracované objednávce.	tags
Aktér = Uživatel ID = 11 Priorita = Nezbytný	

Obrázek 49: Seznam funkčních požadavků 1/2; zdroj: [autor]

Systém umožní uživateli nahrát dokumenty	tags
Aktér = Uživatel ID = 12 Priorita = Nezbytný	
Systém umožní uživateli zrušit objednávku	tags
Aktér = Uživatel ID = 13 Priorita = Nezbytný	
Systém umožní pracovníkovi oddělení spotřebního financování manálně posoudit žádost o půjčku.	tags
Aktér = Pracovník banky ID = F14 Priorita = Nezbytný	
Systém bude pravidelně mazat objednávky s poslední změnou starší než 7 kalendářních dní.	tags
Aktrér = Systém ID = F15 Priorita = Nezbytný	
Systém zajistí rizikový scoring osoby.	tags
Aktér = Systém ID = F16 Priorita = Nezbytný	
Systém zajistí úvěrový scoring osoby.	tags
Aktér = Systém ID = F17 Priorita = Nezbytný	
Systém vyhodnotí způsobilost klienta objednat si konkrétní produkt.	tags
Aktér = Systém ID = F18 Priorita = Nezbytný	
Systém zašle smluvní a předmluvní informace na emailovou adresu před podpisem smlouvy.	tags
Aktér = Systém ID = F19 Priorita = Nezbytný	
Systém bude automaticky zkládat produkty	tags
Aktér = Systém ID = F20 Priorita = Nezbytný	
Systém bude notifikovat call centrum před vypršením platnosti objednávky	tags
Aktér = Systém ID = F21 Priorita = Možný	
Systém umožní uživateli zadat promokód	tags
Aktér = Systém ID = F22 Priorita = Eventuální	

Obrázek 50: Seznam funkčních požadavků 2/2; zdroj: [autor]

custom Nefunkční požadavky

Náklady na pořízení HW a SW na kterém bude systém provozován nepřesáhnou 1 000 000 Kč.
<i>tags</i>
ID = NF01 Priorita = Možný
Úložiště objednávek musí pojmout 100 000 záznamů
<i>tags</i>
ID = NF02 Priorita = Nezbytný
Systém bude po uživateli vyžadovat pouze minimální množství dat.
<i>tags</i>
ID = NF03 Priorita = Nezbytný
Systém bude optimalizován pro jednoduchost použití a intuitivní ovládání.
<i>tags</i>
ID = NF04 Priorita = Možný
Systém bude dostupný 24/7
<i>tags</i>
ID = NF05 Priorita = Nezbytný
Vyhodnocení úvěrového profilu uživatele nesmí přesáhnout 3 minuty
<i>tags</i>
ID = NF06 Priorita = Možný
Systém bude v souladu s ustanovením GDPR
<i>tags</i>
ID = NF07 Priorita = Nezbytný
Systém bude provádět zálohy úložiště objednávek dvakrát denně
<i>tags</i>
ID = NF08 Priorita = Nezbytný

Obrázek 51: Seznam nefunkčních požadavků; zdroj: [autor]

8.2 Kompletní datový slovník

Tabulka 7: Kompletní datový slovník; zdroj: [autor]

Třída	Popis
<i>Adresa</i>	Třída Adresa uchovává informaci o místě pobytu uživatele, dle typu může jít o adresu trvalého pobytu nebo např. o korespondenční adresu.
<i>Dokument</i>	Třída Dokument představuje elektronický dokument, na jehož základě jsou zakládány produkty. Třída umožňuje vygenerovat dokument a uložit dokument. Dokumenty jsou autorizovány pomocí SMS, která je klientovi zaslána na mobilní telefon.
<i>IdentifikačníPlatba</i>	Třída IdentifikačníTransakce představuje elektronickou bezhotovostní transakci, která slouží k autentizaci uživatele. Identifikační transakci zadává třída Uživatel. Třída umožňuje získat detail transakce Třída vlastní atributy jméno a příjmení zadavatele, které je využíváno systémem při identifikaci uživatele.
<i>KlientBanky</i>	Třída KlientBanky je podtřídou třídy Uživatel. Představuje klienta banky, který je plně autentizován, a není tak nutné provádět identifikaci osoby v průběhu procesu. Klient se přihlašuje do systému pomocí osobního hesla a klientského čísla.
<i>KreditníProdukt</i>	Třída KreditníProdukt představuje skupinu úvěrových produktů, pro které jsou společnými znaky výše limitu, který banka uživateli poskytuje a cena za jakých je ochotna toto riziko podstoupit – neboli úrok a RPSN. Třída je generalizací tříd KreditníKarta, Kontokorent a Půjčka.
<i>PracovníkBanky</i>	Třída PracovníkBanky představuje fyzickou osobu, která má přístup do neveřejné části systému. Pracovník banky manuálně posuzuje žádosti klientů s nevyhovujícím rizikovým profilem pro automatizovaný proces. Pracovník banky se přihlašuje do systému pomocí služebního čísla a hesla.
<i>Produkt</i>	Třída Produkt představuje službu, kterou banka poskytuje svým klientům na základě smluvního vztahu. Třída umožňuje přidat přiřadit produkt k objednávce, modifikuje produkt, vyhodnocuje způsobilost klienta vlastnit produkt a zakládá produkt. Produkty vznikají na základě dokumentu, který stvrzuje smluvní vztah mezi bankou a klientem.
<i>PromoKód</i>	Třída PromoKód představuje identifikátor obchodní kampaně, která může mít dopad na produkt (jedná se např. o obchodní slevy, zvýhodnění atp.). třída uchovává informace typu promo kódu a jeho popis.
<i>Souhlas</i>	Třída Souhlas představuje svolení klienta s určitou akcí banky. Třída uchovává informace o typu souhlasu, datu udělení a jeho platnosti.

<i>Účet</i>	Třída Účet představuje skupinu produktů typu účet. Třída uchovává jejich společné atributy – číslo účtu, kód banky a stav. Třída je generalizací tříd SpořicíÚčet a BežnýÚčet.
<i>Uživatel</i>	Třída Uživatel představuje reálného uživatele – fyzickou osobu, která ovládá systém. Třída uchovává hlavní informace o uživateli jako je jméno, přímění, rodné číslo atp. Atribut „identifikován“ podává informaci o tom, zdali je uživatelem autentizovaná osoba nebo anonymní osoba, což je podstatná informace ovlivňující obchodní proces. Třída uživatel zakládá Žádost.
<i>Žádost</i>	Třída Žádost vzniká na popud třídy Uživatel. Třída sdružuje produkty, o které má uživatel zájem. Třída má definovaný datum vzniku, stav a platnost. Žádost se skládá z alespoň z jednoho produktu. Třída spravuje žádost, a využívá třídu Uživatel. Je ve vztahu kompozice s třídou PromoKód.

8.3 Kompletní slovníček pojmů

Tabulka 8: Kompletní slovníček pojmů; zdroj: [autor]

Pojem	Synonyma	Popis
Depozitní produkt	Depozit	Produkty, ve kterých banka vystupuje jako dlužník – klient bance poskytuje finance formou vkladů za cenu úroku.
Kanál	Žádná	Cesta, nebo způsob jakým lze sjednat produkt. Současné typy kanálů jsou následující: Pobočka, Call centrum.
Kreditní produkt	Kredit	Produkty, ve kterých banka vystupuje jako věřitel – poskytuje finance jiným osobám za cenu úroku.
Promo kód	Žádná	12místný číselný kód, který identifikuje marketinkovou kampaň – promo kód se váže k určitému produktu. Uživatel, který se prokáže tímto kódem získává výhodu, například ve formě prominutí zřizovacího poplatku nebo nižšího úroku v případě kreditních produktů.
Upsell	Navýšení limitu	Specifický případ žádosti o produkt kreditní karta nebo kontokorent. Klient již požadovaný produkt vlastní a na tomto produktu si přeje navýšit limit. <i>Snížení limitu nebude na kanále e-shop možné (obchodní politika společnosti).</i>
Žádost	Objednávka	Svobodný projev vůle fyzické osoby sjednat si určitý produkt nebo službu poskytovaný bankou.
Retail	Žádná	Klientela zahrnující fyzické osoby.
Leed	Žádná	Kontakt na potenciačního klienta, skládá se z kontaktních údajů na osobu a parametrizovaného produktu o který projevil zájem. Leed bez souhlasu s využitím osobních údajů pro marketingové účely není validní – osoba nemůže být kontaktována.
Omni-kanál	Multichannel Omnichannel	Pojem požívaný ve spojitosti s procesem, který může probíhat na více kanálech. Např. prodej produktu – může probíhat na samoobslužném kanále nebo na asistovaném kanále. Omichannel proces zajišťuje možnost přesunu procesu z jednoho kanálu na jiný. Např. započetí žádosti na e-shopu a dokončení na pobočce.
Scoring osoby	Risk	Prověření osoby AML systémem, seznamem hledaných osob, bankovními blacklisty atp. Zjišťuje, zdali banka může osobě poskytnout služby.

Úvěrový scoring	Credit	Vyhodnocuje riziko spojené s poskytnutím úvěru dané osobě. Definuje RPSN a úrokovou míru.
Prohlášení	Žádná	Svobodné, čestné prohlášení stvrzující určitá data - např. potvrzení pravdivosti zadaných údajů. Prohlášení nemá reálnou právní váhu.
Souhlas	Žádná	Souhlasem subjektu údajů je dále vědomý projev vůle, jehož obsahem je svolení subjektu údajů se zpracováním osobních údajů. Na rozdíl od prohlášení, souhlas musí být odvolatelný a uživatel vždy musí být při udělení souhlasu poučen jakým způsobem lze tento souhlas odvolat.
Informační povinnost	Žádná	Povinnost informovat subjekty údajů při shromažďování jejich osobních údajů o tom, v jakém rozsahu a pro jaký účel budou osobní údaje zpracovány, kdo a jakým způsobem bude osobní údaje zpracovávat a komu mohou být osobní údaje zpřístupněny.
Pepek	PEP	Politicky exponovaná osoba, nebo osoba se zvláštním vztahem k bance.
Kont'as	Revolver	Kontokorent – jeden z kreditních produktů.
Plast	Karta	Fyzická podoba debetní nebo kreditní karty.
TEU	Techničák	Technický účet, který je zakládán společně s úvěrem – slouží pouze pro splácení úvěru.
Alternativní limit	Žádná	Variety produktů, které jsou uživateli nabídnuty společně s požadovanou variantou. Vypočteny jsou na základě obchodních pravidel.
Předschválený limit	Žádná	Varianta kreditního produktu, u které má klient velmi vysokou pravděpodobnost, že žádost bude schválena. Schvalovací proces v systému CreditRisk je zjednodušen. Limity jsou vypočítány na základě klientovi platební historie nebo může být součástí kampaně.
Tenor	Žádná	Doba splácení úvěru (resp. počet měsíčních splátek).

8.4 Scénáře případů užití

Tabulka 9: Detail případu užití ZaložitŽádost; zdroj: [autor]

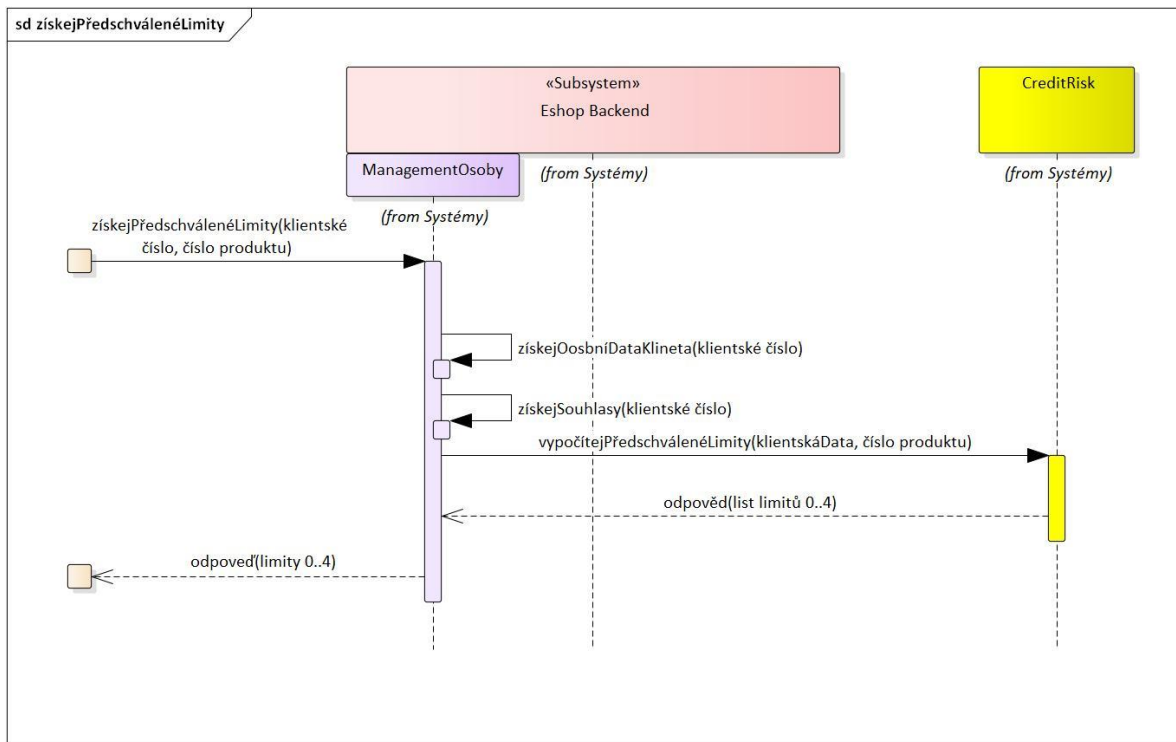
Případ užití:	ZaložitŽádost
ID:	UC01
Stručný popis:	Uživatel vyplní žádost o produkt, která se skládá z parametrizovaného produktu a osobních údajů. Systém v případě přihlášeného uživatele zobrazí předschválené varianty kreditních produktů.
Hlavní aktéři:	Uživatel
Vedlejší aktéři:	Žádní
Vstupní podmínky:	Žádné
Hlavní scénář:	<ol style="list-style-type: none"> 1. Případ užití začíná až uživatel vybere produkt, o který má zájem 2. KDYŽ uživatel je přihlášen a žádá o kreditní produkt <ol style="list-style-type: none"> 2.1 Systém zjistí předschválené limity 2.2 Systém identifikuje upsell 2.3 Systém zobrazí formulář parametrizace produktu s předschválenými variantami produktu 3. JINAK uživatel není přihlášen <ol style="list-style-type: none"> 3.1. Systém zobrazí formulář parametrizace produktu bez předschválených variantami produktu 4. Uživatel provede parametrizaci produktu a potvrdí formulář 5. Systém zobrazí obrazovku vybízející klienta k přihlášení 6. Uživatel zvolí přihlášení nebo vybere možnost pokračovat bez přihlášení <i>místo rozšíření: PřihlásitKlineta</i> 7. KDYŽ uživatel je přihlášen <ol style="list-style-type: none"> 7.1. Systém zjistí dostupné klientské údaje 7.2. Systém zobrazí formulář osobní údaje s předvyplněnými daty 8. JINAK uživatel není přihlášen Systém zobrazí formulář osobní údaje a umožní vložit potvrzení příjmu. 9. Uživatel zadá povinné údaje a potvrdí formulář 10. Systém zobrazí obrazovku informující klienta o probíhajícím vyhodnocení žádosti
Výstupní podmínky:	Systém navázal případem užití posouzení žádosti
Alternativní scénáře:	Nefunkční systém poskytující předschválené limity Nefunkční systém ověřující upsell

Tabulka 10: Tabulka 10: Detail případu užití PosouditŽádost; zdroj: [autor]

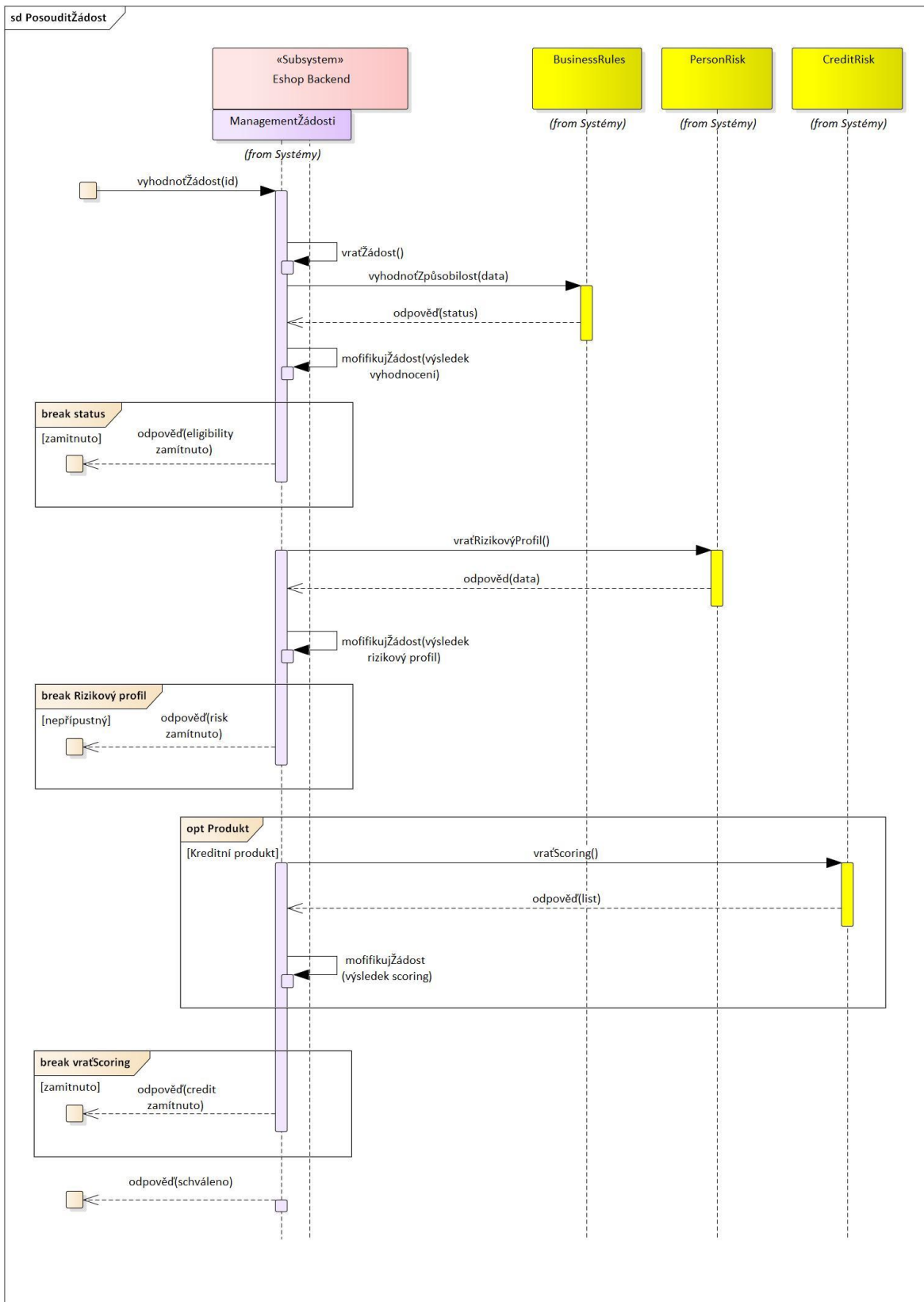
Případ užití:	PosouditŽádost
ID:	UC04
Stručný popis:	Uživatel potvrdí osobní údaje, systém provede vyhodnocení žádosti.
Hlavní aktéři:	Uživatel, CreditRisk, PersonRisk, BusinessRules
Vedlejší aktéři:	Žádní
Vstupní podmínky:	Žádné
Hlavní scénář:	<ol style="list-style-type: none"> 1. Případ užití začíná, když uživatel potvrdí závaznou nabídku banky ve fázi osobní údaje. 2. Systém načte klientovu žádost 3. Systém vyhodnotí způsobilost klienta vlastnit produkt 4. Systém uloží výsledek kontroly oprávnění 5. Systém zjistí rizikový profil klienta 6. Systém uloží výsledek kontroly oprávnění 7. <u>KDYŽ</u> produkt je ze skupiny kreditů <ol style="list-style-type: none"> 7.1. Systém zjistí úvěrový profil klienta 7.2. Systém uloží výsledek vyhodnocení úvěrového profilu 8. <u>JINAK</u> <ol style="list-style-type: none"> 8.1. ---
Výstupní podmínky:	Systém vyhodnotil žádost
Alternativní scénáře:	<p>Uživatel není oprávněn produkt vlastnit Uživatel vyhodnocen jako rizikový Kreditní produkt nebyl schválen Dokumenty potvrzení příjmů jsou neplatné Chyba systému</p>

Tabulka 11: Detail případu užití Contracting; zdroj: [autor]

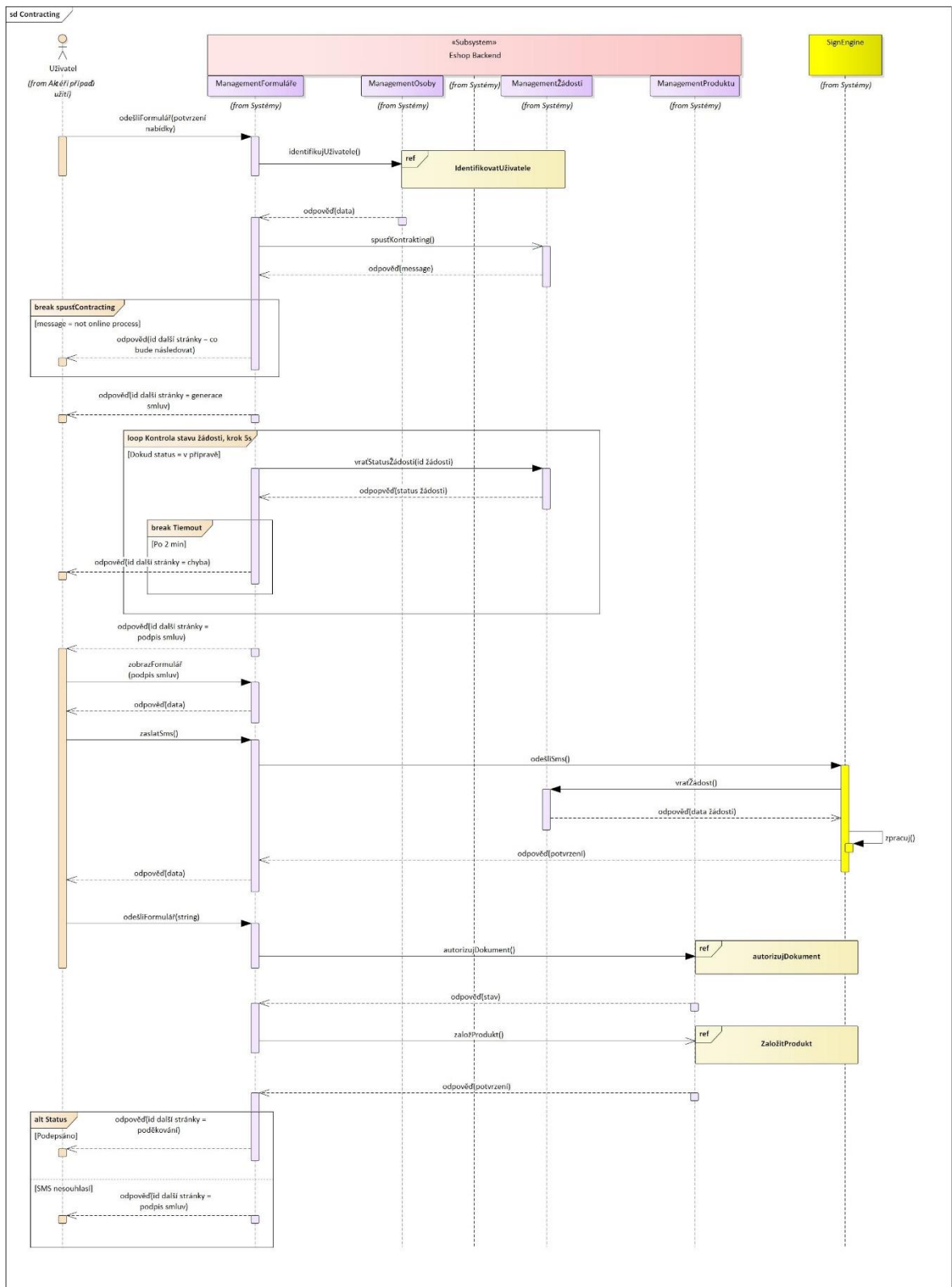
Případ užití:	Contracting
ID:	UC7
Stručný popis:	Uživatel potvrdí závaznou nabídku banky, systém provede vyhodnocení žádosti.
Hlavní aktéři:	Uživatel, CreditRisk, PersonRisk, BusinessRules
Vedlejší aktéři:	Žádní
Vstupní podmínky:	Žádné
Hlavní scénář:	<ol style="list-style-type: none"> 1. Případ užití začíná, když uživatel potvrdí závaznou nabídku banky ve fázi Nabídka 2. zahrnout <i>IdentifikaceUživatele</i> 3. Systém uloží žádost 4. Systém vyhodnotí metodu podpisu 5. <u>KDYŽ</u> metoda podpisu je různá od online SMS <ol style="list-style-type: none"> 5.1. Systém zobrazí stránku s poděkováním a informacemi o off-line kontraktingu 6. <u>JINAK</u> <ol style="list-style-type: none"> 6.1. Systém připraví smluvní dokumentaci k podpisu 6.2. Systém zobrazí uživateli obrazovku s dokumenty k podpisu 6.3. Uživatel potvrdí vůli uzavřít smlouvu a zažádá o SMS 6.4. Systém odešle SMS na klientovo telefonní číslo 6.5. Uživatel vyplní SMS kód a potvrdí formulář 6.6. Systém ověří SMS 6.7. Systém iniciuje založení produktu 6.8. Systém zobrazí obrazovku s poděkováním
Výstupní podmínky:	Systém vyhodnotil žádost
Alternativní scénáře:	<p>SMS kód není validní Platnost SMS kód vypršela Žádost o nový SMS kód Chyba generace dokumentů Chyba systému</p>



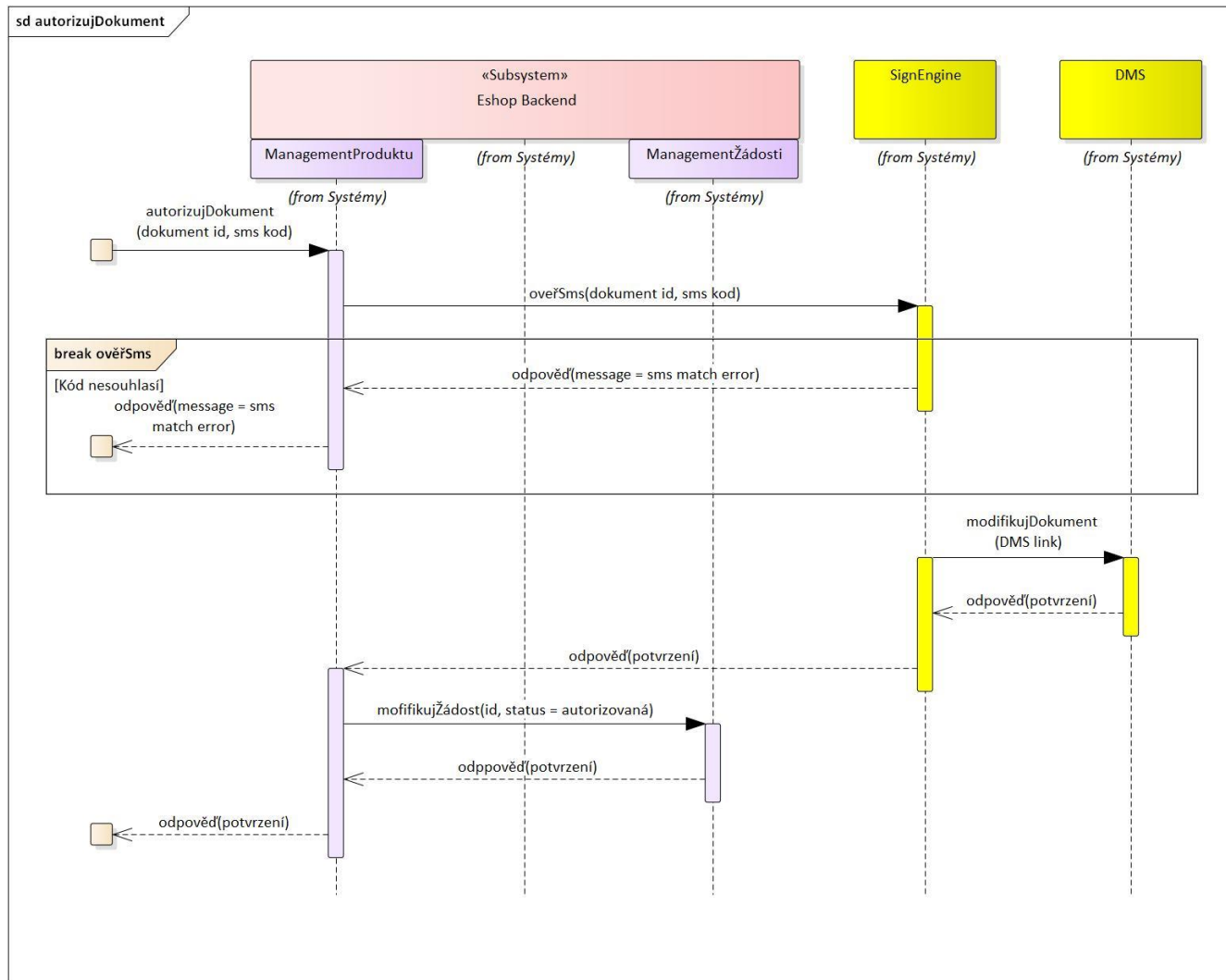
Obrázek 53: Sekvenční diagram metody ziskejPředschválenéLimity; zdroj: [autor]



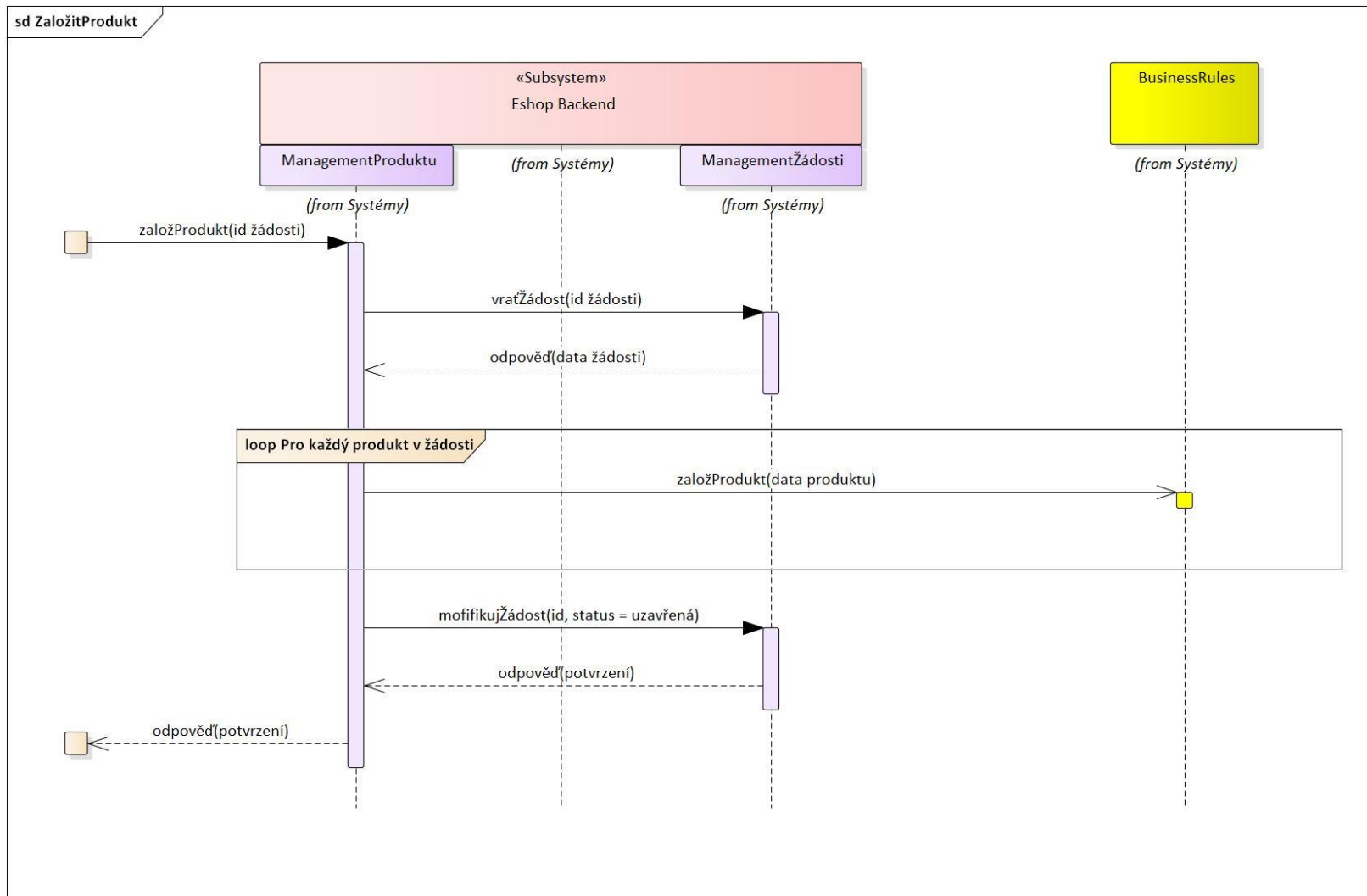
Obrázek 54: Sekvenční diagram PosouditŽádost; zdroj: [autor]



Obrázek 55: Sekvenční diagram Contracting; zdroj: [autor]

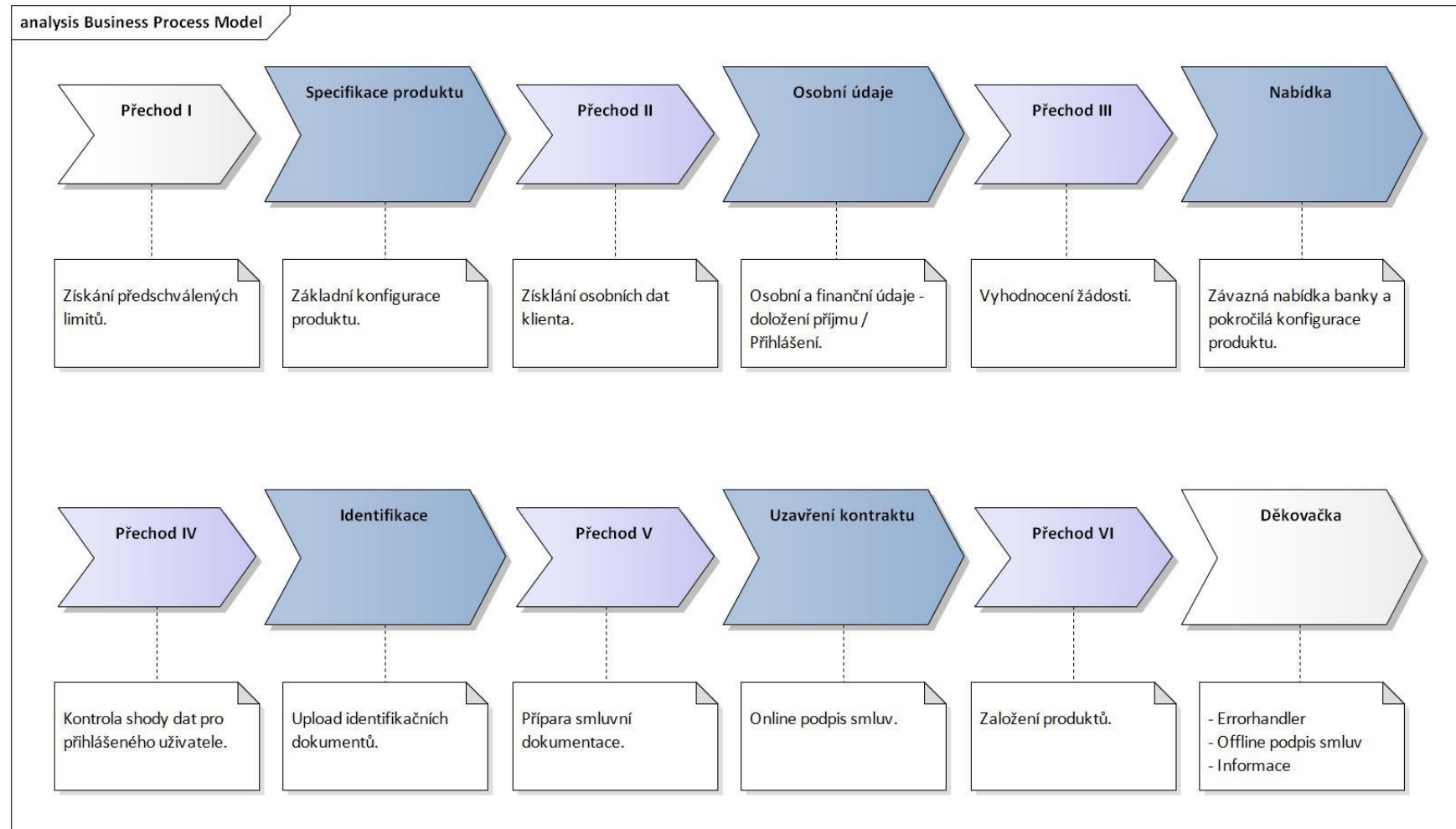


Obrázek 56: Sekvenční diagram pro metodu autorizujDokument; zdroj: [autor]



Obrázek 57: Sekvenční diagram ZaložitProdukt; zdroj: [autor]

8.6 Omni kanálový prodejní proces



Obrázek 58: Omni kanálový prodejní proces; zdroj: [autor]

8.7 Návrhy obrazovek

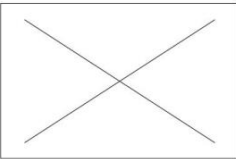
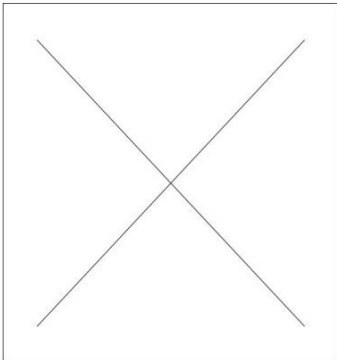
custom PID0101: Portlet produktová specifikace

Půjčka bez starostí

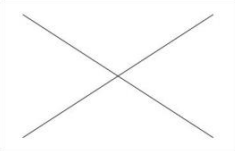
custom Progress Bar nastavte si produkt

Nastavte produkt > Osobní údaje > Nabídka > Identifikace > Podpis smluv

custom Motiv Kampaně



Allquam omare wisi eu metus. Aenean vel massa quis mauris vehicula lacinia. Aliquam erat volutpat. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nunc tincidunt ante vitae massa. Quisque porta.



Nulla quis diam. Nullam justo enim, consectetur nec, ullamcorper ac, vestibulum in, elit. Donec quis nibh at felis congue commodo. Praesent id justo in neque elementum ultrices. Nulla turpis magna, cursus sit amet, suscipit a, interdum id, felis.

custom Předšválené limity

Máme pro Vás připraveny následující nabídky

Pokud si vybere jednu z následujících variant, pak Vás od načerpání financí dělí jen 10 minut!

<input type="radio"/> 50 000 CZK za 3 650 CZK na 36 měsíců při úroku 8,1% p.a. a RPSN 9,2% (?)	<input type="radio"/> 200 000 CZK za 3 650 CZK na 36 měsíců při úroku 8,1% p.a. a RPSN 9,2% (?)	<input type="radio"/> 400 000 CZK za 3 650 CZK na 36 měsíců při úroku 8,1% p.a. a RPSN 9,2% (?)	<input type="radio"/> 600 000 CZK za 3 650 CZK na 36 měsíců při úroku 8,1% p.a. a RPSN 9,2% (?)
--	---	---	---

custom Úvěrová kalkulačka

Půjčte si přesně tolik kolik potřebujete...

Měsíční splátka
5 650 CZK
Úroková míra 8,9% p.a.
RPSN 9,9% (?)

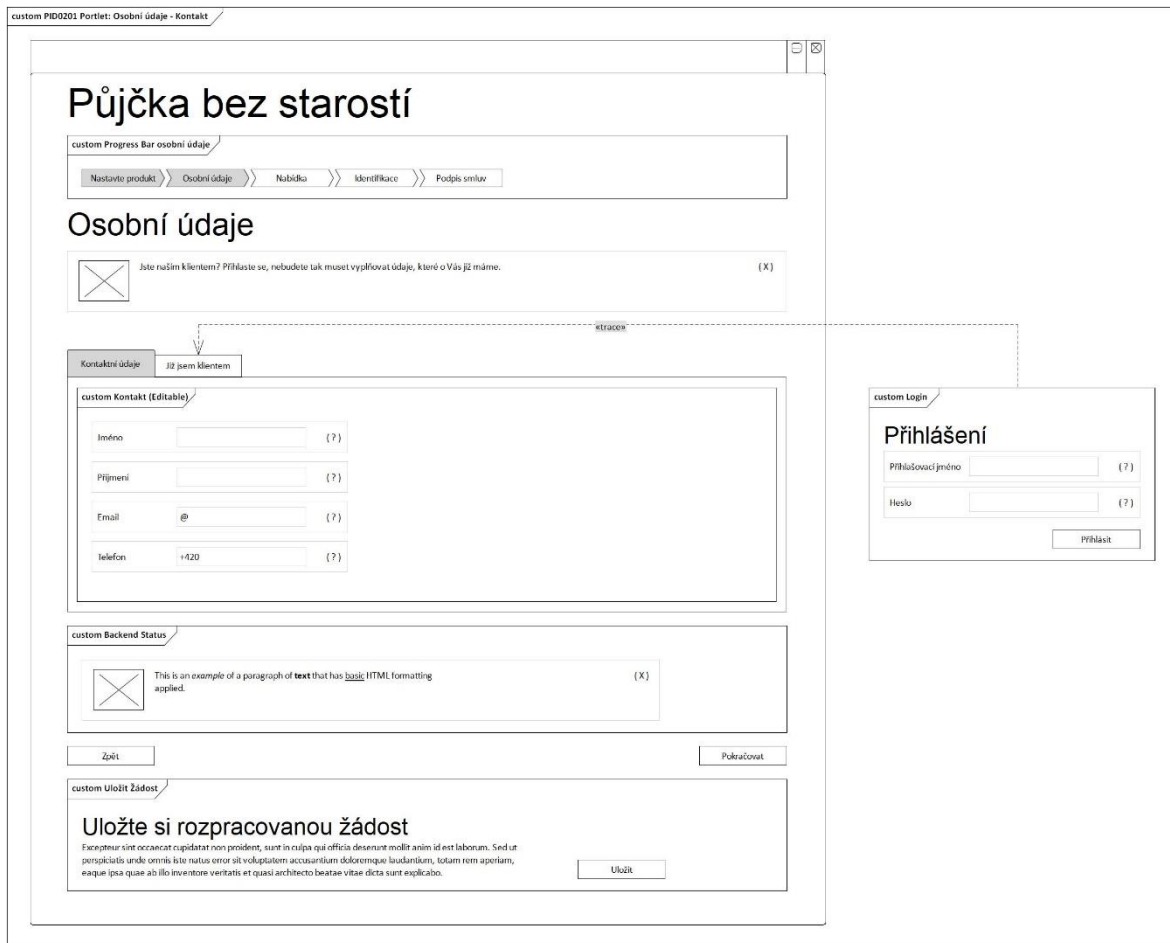
Výše půjčky
50 000 CZK | 1 000 000 CZK
200 000 CZK (?)

Jak dlouho chcete splácet?
12 měsíců | 96 měsíců
72 měsíců (?)

Vzorový příklad Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce consectetur risus a nunc. Nam sed tellus id magna elementum tincidunt. Morbi scelerisque luctus velit. Et harum quidem rerum facilis est et expedita distinctio. Nullam at arcu a est sollicitudin euismod.
Disclaimer Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Sed conwallis magna eu sem. Etiam dictum tincidunt diam. Morbi leo mi, nonummy eget tristique non, rhoncus non leo.

Pokračovat

Obrázek 59: Wireframe formuláře produktové specifikace – půjčka; zdroj: [autor]



Obrázek 60: Wireframe formuláře osobní údaje – kontakt; zdroj: [autor]

Půjčka bez starostí

custom Progress Bar osobní údaje

Nastavte produkt > Osobní údaje > Nabídka > Identifikace > Podpis smluv

Osobní údaje



This is an *example* of a paragraph of **text** that has basic HTML formatting applied.

(X)

Kontaktní údaje

[Zobrazit kontaktní údaje](#)

custom Kontakt (Readonly)

Jméno	Adam	Změnit kontaktní údaje
Příjmení	Dvořák	
Email	adam.dvorak@email.cz	
Telefon	+420 773 111 982	

[Skrýt kontaktní údaje](#)

custom Osobní údaje (Editable)

Osobní údaje

Státní občanství (?)

Rodné číslo (?)

Adresa trvalého pobytu

Ulice (?)

Číslo (?)

PSČ (?)

Město (?)

Doklad totožnosti

Typ dokladu (?)

Číslo dokladu (?)

custom Uložit žádost

Uložte si rozpracovanou žádost

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Obrázek 61: Wireframe formuláře osobní údaje – osobní údaje; zdroj: [autor]


custom PID0203 Portlet: Osobní údaje - Finance

Půjčka bez starostí

custom Progress Bar osobní údaje

Nastavte produkt > Osobní údaje > Nabídka > Identifikace > Podpis smluv

Osobní údaje

 This is an example of a paragraph of text that has basic HTML formatting applied. (X)

Kontaktní údaje

[Zobrazit kontaktní údaje](#)

custom Kontakt (Readonly)

[Změnit kontaktní údaje](#)

Jméno	Adam
Příjmení	Dvořák
Email	adam.dvorak@email.cz
Telefon	+420 773 111 982

[Skrýt kontaktní údaje](#)

Osobní údaje

[Zobrazit osobní údaje](#)

custom Osobní údaje (Readonly)

Státní občanství	Česká Republika
Rodné číslo	911215/0315

Adresa trvalého pobytu

Ulice	Květková
Číslo	349/10
PSČ	190 00
Město	Praha

Doklad totožnosti

Typ dokladu	Občanský průkaz
Číslo dokladu	212010976

[Skrýt osobní údaje](#)

Obrázek 62: Wireframe formuláře osobní údaje – finance 1/2; zdroj: [autor]

custom Finance (Editable)

Výdaje a příjmy

Cisté měsíční příjmy 45 000 CZK (?)

Měsíční výdaje 30 000 CZK (?)

Sociálně demografická data

Sociální status Zaměstnanec (?)

Rodinný status Svobodný (?)

Zaměstnání Manuálně pracující na dob... (?)

Prohlašuji že nejsem ve výpovědní lhůtě

Tags:
 BorderStyle = Dashed
 301 Visible Condition = Pokud [klientské číslo] = null

Doložení příjmů

Doložení příjmu Výpis z účtu (?)

Dokument	Název souboru	Výbrat soubory
	Zde nahrajte výpis z účtu za poslední 3 předchozí kalendářní měsíce v elektronické podobě.	(X)
«drag'n'drop komponenta» ClientArea		

custom Úvěrové registry

Prověření bonity

Potvrzuji, že jsem byl/-a seznámen s [informacemi o úvěrových registrech BRNI, NRKI a SOLUS](#).

Ukládám souhlas s využitím dat [TelcoScore](#).

custom Prohlášení

Stiskem tlačítka "Pokračovat"...

Prohlašujete, že:

- Jsou všechny Vámi zadané údaje pravdivé.
- Nejste zaměstnán ve společnosti, kde vlastněte podíl vyšší než 50% a nejste ve výpovědní lhůtě.
- Nejste politicky exponovanou osobou, osobou se zvláštním vztahem k bankám ani nejste s takovou osobou v příbuzenském vztahu.

Zpět Zrušit žádost Pokračovat

custom Backend Status

This is an *example* of a paragraph of text that has basic HTML formatting applied. (X)

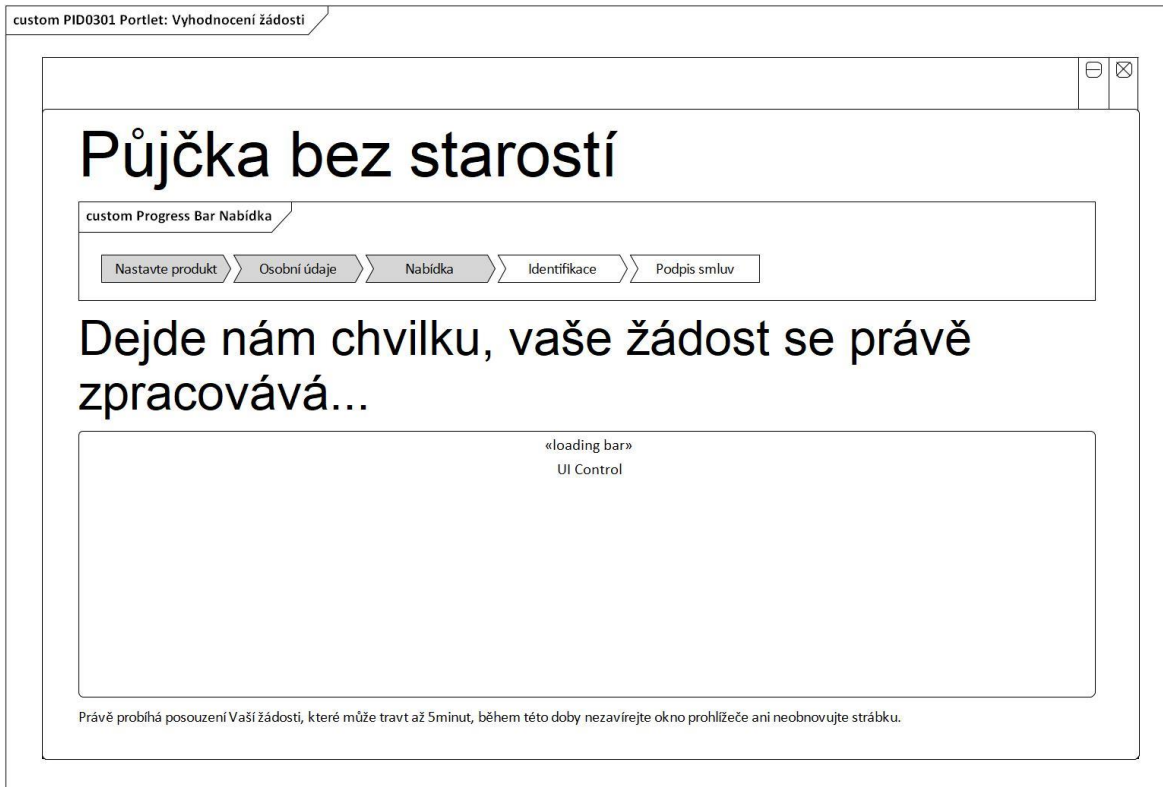
custom Uložit Žádost

Uložte si rozpracovanou žádost

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Uložit

Obrázek 63: Wireframe formuláře osobní údaje – finance 2/2; zdroj: [autor]



Obrázek 64: Wireframe formuláře vyhodnocení nabídky; zdroj: [autor]

custom PID0302: Portlet Nabídka - schváleno

☰ ☒

Půjčka bez starostí

custom Progress Bar Nabídka

Nastavte produkt
Osobní údaje
Nabídka
Identifikace
Podpis smluv

Vaše žádost byla schválena

✕

Nam sed tellus id magna elementum trincidunt. Pellentesque sapien. Integer tempor. Phasellus faucibus molestie nisi. Mauris metus. Aliquam id dolor. In laoreet, magna id viverra trincidunt, sem odio bibendum justo, vel imperdiet sapien wisi sed libero. Nullam sapien sem, ornare ac, nonummy non, lobortis a enim. Duis sapien nunc, commodo et, interdum suscipit, sollicitudin et, dolor. Vestibulum fermentum tortor id mi.

(X)

custom Vyhodnocení žádosti: Credit

Schválili jsme vám následující půjčku, pokud potřebujete více peněz vyberete si alternativních variant.

Vaše varianta

240 000 CZK

za 3 240 CZK měsíčně na 5 let
při úroku 6,9% a RPSN 7,3%

(?)

Alternativní nabídky

300 000 CZK

za 3 650 CZK měsíčně na 5 let

(?)

500 000 CZK

za 6 340 CZK měsíčně na 6 let

(?)

600 000 CZK

za 7 200 CZK měsíčně na 7 let

(?)

Výše půjčky	240 000 CZK
Doba splácení	5 let
Úroková míra	6,9% p.a.
RPSN	7,3% p.a.
Pojištění	Varianta 1 za 120 CZK/měs.

custom Nastavení produktu: Půjčka

Nastavte si půjčku

Datum splátky

25. den v měsíci

(?)

Účet kam pošleme peníze

Předčísíl

-

Číslo účtu

/

Kód banky

(?)

custom Vyběr metody podpisu

Vyberete metodu podpisu smluvní dokumentace

Online SMS

Etiam ligula pede, sagittis quis, interdum ultricies, scelerisque eu.

U kurýra

Praesent id justo in neque elementum ultrices. Aliquam ante. Nunc dapibus tortor vel mi dapibus sollicitudin.

Na pobočce

Nam quis nulla. Morbi scelerisque luctus velit. Aenean id metus id velit ullamcorper pulvinar. Nulla turpis magna, cursus sit amet.

✕

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce tellus. Nullam justo enim, consectetur nec, ullamcorper ac, vestibulum in, elit. Etiam posuere lacus quis dolor. Duis pulvinar. Pellentesque arcu. Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur.

(X)

Pokračovat

custom Uložit žádost

Uložte si rozpracovanou žádost

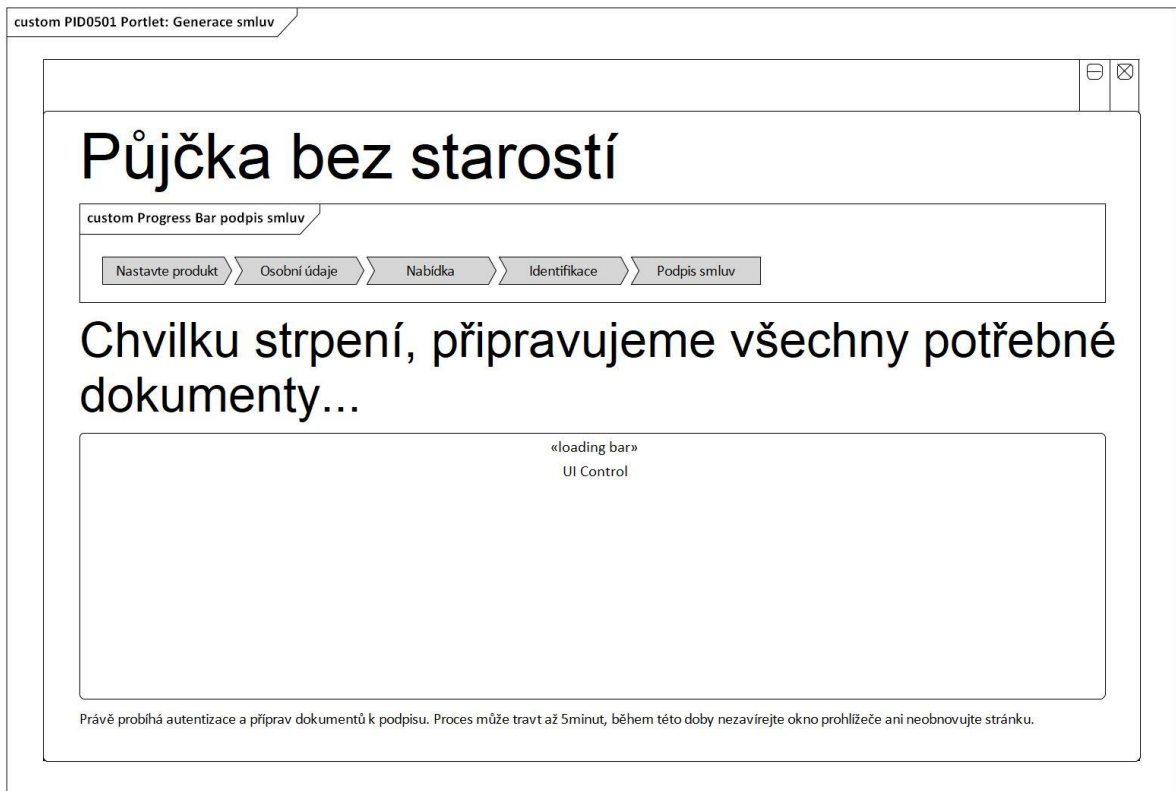
Exceperet sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Uložit

Obrázek 65: Wireframe formuláře nabídka – schváleno; zdroj: [autor]



Obrázek 66: Wireframe formuláře identifikace; zdroj: [autor]



Obrázek 67: Wireframe formuláře generace smluv; zdroj: [autor]

custom PID0502:Portlet: Podpis Smluv

☰ ☒

Půjčka bez starostí

custom Progress Bar podpis smluv

Nastavte produkt
Osobní údaje
Nabídka
Identifikace
Podpis smluv

Dokumeny jsou připraveny k podpisu

✕

Nam sed tellus id magna elementum tincidunt. Pellentesque sapien. Integer tempor. Phasellus faucibus molestie nisi. Mauris metus. Aliquam id dolor. In laoreet, magna id viverra tincidunt, sem odio bibendum justo, vel imperdiet sapien wisi sed libero. Nullam sapien sem, ornare ac, nonummy non, lobortis a enim. Duis sapien nunc, commodo et, interdum suscipit, sollicitudin et, dolor. Vestibulum fermentum tortor id mi.

(X)

custom Podpisová komponenta

✕

Seznamte se s obchodními podmínkami

«Icon»
PDF

Obchodní podmínky a sazebníky

Nulla turpis magna, cursus sit amet, suscipit a, interdum id, felis. Vivamus ac leo pretium faucibus. Etiam bibendum elit eget erat.

S podmínkami jsem se seznámil a souhlasím

✕

Předsmělní dokumentace

«Icon»
PDF

Předsmělní dokumentace Půjčky na cokoli

Nulla turpis magna, cursus sit amet, suscipit a, interdum id, felis. Vivamus ac leo pretium faucibus. Etiam bibendum elit eget erat.

S předsmělními dokumenty jsem se seznámil a souhlasím

✕

Smluvní dokumentace

«Icon»
PDF

Smluvní dokumentace Půjčky na cokoli

Nulla turpis magna, cursus sit amet, suscipit a, interdum id, felis. Vivamus ac leo pretium faucibus. Etiam bibendum elit eget erat.

Se smlouvou jsem se seznámil chci ji podepsat

SMS klíč

xxx-xxx-xxx

Zaslat sms klíč

✕

Na Váš mobilní telefon byl zaslán SMS klíč. Jeho platnost vyprší za 9:59 s.

[Získat nový klíč](#)

Autorizovat

custom Uložit Žádost

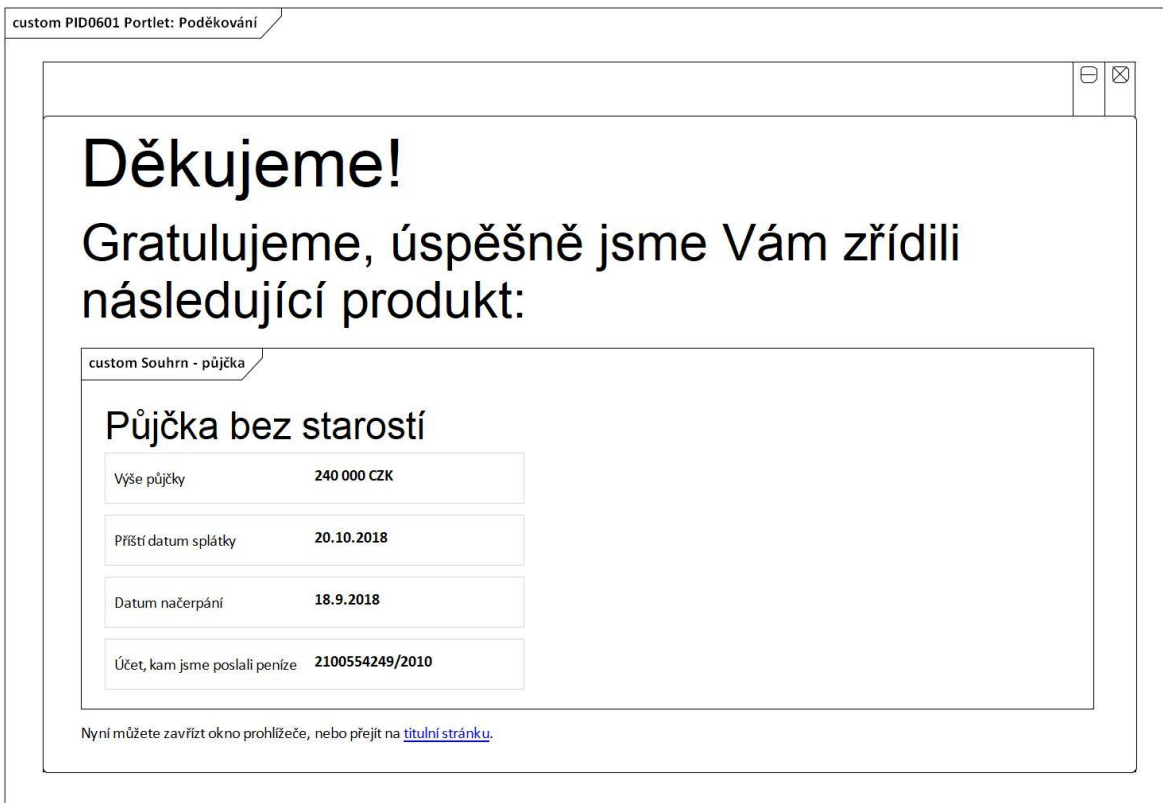
Uložte si rozpracovanou žádost

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Uložit

Obrázek 68: Wireframe formuláře podpis smluv; zdroj: [autor]

140



Obrázek 69: Wireframe formuláře poděkování; zdroj: [autor]

