

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Diplomová práce

Návrh informačního systému v UML

Bc. Jan Milota

© 2017 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Jan Milota

Informatika

Název práce

Návrh informačního systému v UML

Název anglicky

Design of Information System in UML

Cíle práce

Cílem diplomové práce je návrh informačního systému vytvořeného pomocí UML diagramů. Informační systém je navržený z podnětu usnadnění nákupu přes internetové obchody a zároveň odlehčení správy stavu již nakoupeného zboží. Navržený informační systém umožňuje uživatelům vytvoření a řízení svého virtuálního skladu, interaktivní nastavení automatických objednávek z e-shopů, sledování vybraných produktů, informování o možných substitutech těchto produktů a kompletní uchování účetních dokladů.

Metodika

Metodika práce spočívá ve studiu odborných literárních a internetových zdrojů, které se zabývají UML. Dále autor provedl analýzu současného stavu internetových obchodů a jejich služeb. Analýza také obsahuje rozbor recenzí na vybrané internetové obchody a zhodnocení toho, jaké možnosti zákazníci oceňují, či naopak. S využitím poznatků z analýzy byl proveden návrh IS s využitím UML diagramů. Reprezentativní část návrhu byla aplikována jako webová aplikace. Aplikace byla podrobena kvalitativnímu a kvantitativnímu šetření. Na základě výsledků šetření bylo provedeno zhodnocení a navrženy případné úpravy IS.

Doporučený rozsah práce

60 – 80 stran

Klíčová slova

informační systém, návrh, UML, analýza, e-shop, webová aplikace

Doporučené zdroje informací

- ARLOW, J. – NEUSTADT, I. *UML a unifikovaný proces vývoje aplikací : průvodce analýzou a návrhem objektově orientovaného softwaru*. Brno: Computer Press, 2003. ISBN 80-7226-947-.
- ARLOW, J. – NEUSTADT, I. *UML 2 a unifikovaný proces vývoje aplikací : objektově orientovaná analýza a návrh prakticky*. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
- KANISOVÁ, H., MÜLLER, M. *UML srozumitelně*. Brno: Copmuter Press, 2007, ISBN: 80-251-1083-4.
- PAGE-JONES, M. *Základy objektově orientovaného návrhu v UML*. Praha: Grada, 2001. ISBN 80-247-0210-.
- SCHMULLER, J. *Myslíme v jazyku UML : knihovna programátora*. Praha: Grada, 2001. ISBN 80-247-0029-8.
- VRANA, I. – ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE. PROVOZNĚ EKONOMICKÁ FAKULTA, – ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE. KATEDRA INFORMAČNÍHO INŽENÝRSTVÍ. *Projektování informačních systémů s UML*. V Praze: Česká zemědělská univerzita, Provozně ekonomická fakulta, 2008. ISBN 978-80-213-1817-5.

Předběžný termín obhajoby

2016/17 LS – PEF

Vedoucí práce

doc. RNDr. Dana Klimešová, CSc.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 1. 11. 2016

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 1. 11. 2016

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 20. 03. 2017

Čestné prohlášení

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

V Praze dne 30. 3. 2017

Poděkování

Rád bych touto cestou poděkoval doc. RNDr. Daně Klimešové, CSc. za odborné vedení, ochotu a čas, který mi byl při konzultacích věnován. Dále bych chtěl poděkovat všem profesorům, docentům, odborným asistentům, asistentům a lektorům České zemědělské univerzity v Praze za předání cenných znalostí a zkušeností, které mi výrazně pomohly při vypracování této diplomové práce.

Návrh informačního systému v UML

Souhrn

Diplomová práce se zabývá návrhem informačního systému, který je určen ke správě a evidenci zboží ve skladu a faktur. Informační systém je navrhován pomocí sjednoceného modelovacího jazyka (UML). Teoretická část přibližuje problematiku informačních systémů, objektově orientovaný přístup a vývoj v jazyce UML. Praktická část popisuje existující systémy, které se svými funkcemi nejvíce přibližují navrhovanému systému. Dále jsou specifikovány požadavky na nový informační systém, dle kterých je nový informační systém navrhován pomocí UML modelů. Navržený systém je doplněn teoretickým postupem realizace a je vytvořena webová aplikace přibližující vizuální vzhled navrženého systému.

Klíčová slova: informační systém, unifikovaný modelovací jazyk (UML), objektově orientovaný přístup, návrh, analýza, model, e-shop, webová aplikace, virtuální sklad, hlídací pes, wireframe, XML feed

Design of an Information system in UML

Summary

This diploma thesis deals with the information system, which is designed to manage and registration of goods in the warehouse and invoices. The information system is designed by using the unified modeling language (UML). The theoretical part describing information systems, object-oriented approach and developments in UML. The practical part describes existing systems with their functions much closer to the proposed system. Further specifies requirements for a new information system, according to which a new information system designed using UML models. The proposed system is complemented by theoretical and implementation process is designed web application describing the visual appearance of the proposed system.

Keywords: information system, unified modeling Language (UML), object-oriented approach, design, analysis, model, e-commerce, web applications, virtual warehouse, watchdog, wireframe, XML feed

Obsah

1 Úvod.....	12
2 Cíl práce a metodika	13
2.1 Cíl práce	13
2.2 Metodika	13
3 Teoretická východiska	14
3.1 Informační systém.....	14
3.1.1 Klasifikace informačních systémů.....	15
3.1.2 Podnikový informační systém	16
3.1.3 Podsystemy podnikového informačního systému.....	17
3.2 Životní cyklus informačního systému	20
3.2.1 Vodopádový model.....	20
3.2.2 Zbylé modely životního cyklu informačního systému	23
3.3 Metodiky vývoje informačního systému.....	25
3.3.1 Tradiční metodiky vývoje informačního systému	26
3.3.2 Moderní metodiky vývoje informačního systému	27
3.4 Druhy přístupu k analýze a návrhu informačního systému.....	29
3.4.1 Strukturovaný přístup	29
3.4.2 Objektově orientovaný přístup.....	30
3.5 Unified Modeling Language	32
3.5.1 Obecně o UML	33
3.6 Diagramy v UML.....	35
3.6.1 Diagram tříd.....	36
3.6.2 Stavový diagram	39
3.6.3 Digramy pro tvorbu modelu interakcí	40
3.7 Web	43
3.7.1 Internetové prohlížeče a webové služby	44
3.7.2 Pravidla pro tvorbu webu.....	46
3.8 Srovnávače zboží	49
3.8.1 Přidání e-shopu do srovnávače	51
3.8.2 Cenové hledisko.....	53
4 Praktická část	54
4.1 Popis současného stavu	55
4.1.1 Výběr funkcí	55
4.1.2 Analýza současné situace.....	56
4.2 Specifikace požadavků.....	60

4.2.1	Funkční požadavky uživatele.....	61
4.2.2	Nefunkční požadavky	65
4.3	Návrh řešení	67
4.3.1	Model tříd	67
4.3.2	Stavový diagram	70
4.3.3	Diagram případů užití	71
4.3.4	Sekvenční model.....	73
4.3.5	Model aktivit.....	77
4.4	Přehled návrhu UML.....	79
4.5	Logický návrh systému	80
4.6	Kvalitativní šetření	82
4.6.1	Vyhodnocení kvalitativního šetření	84
4.6.2	Změny v systému dle kvalitativního šetření	86
4.7	Tvorba reprezentativní webové aplikace	88
5	Diskuze	89
5.1	Aktuálnost práce.....	89
5.2	Návrh pro realizaci	91
5.2.1	Mobilní aplikace	92
5.2.2	Problém při realizaci	92
6	Závěr.....	94
7	Seznam použitých zdrojů	96
8	Seznam příloh	98
	Přílohy	99

Seznam obrázků

Obrázek 1 - Schéma vztahů modulů PIS (Zdroj: [6])	17
Obrázek 2 - Komponenty SCM (Zdroj: Vlastní tvorba)	19
Obrázek 3 - Vodopádový model (Zdroj: [11])	21
Obrázek 4 - Prototypový model vývoje IS (Zdroj: [11])	23
Obrázek 5 - Iterativní model vývoje IS (Zdroj: [11])	24
Obrázek 6 - Spirálový model vývoje IS (Zdroj: [14])	25
Obrázek 7 - Přehled diagramů UML (Zdroj: Vlastní tvorba)	35
Obrázek 8 - Zobrazení asociace v diagramu tříd (Zdroj: Vlastní tvorba)	37
Obrázek 9 - Zobrazení generalizace v diagramu tříd (Zdroj: Vlastní tvorba)	38
Obrázek 10 - Agregace v diagramu tříd (Zdroj: Vlastní tvorba)	38
Obrázek 11 - Kompozice v diagramu tříd (Zdroj: Vlastní tvorba)	39
Obrázek 12 - Příklad stavového diagramu (Zdroj: Vlastní tvorba)	40
Obrázek 13 - Příklad diagram případů užití (Zdroj: Vlastní tvorba)	41
Obrázek 14 - Příklad sekvenčního diagramu (Zdroj: Vlastní tvorba)	42
Obrázek 15 - Příklad diagramu aktivit (Zdroj: Vlastní tvorba)	43
Obrázek 16 - Statistika prohlížečů za rok 2015-2016 společností StatCounter (Zdroj: [25])	45
Obrázek 17 - Přehled uživatelů srovnávačů zboží v ČR (Zdroj: netmonitor.cz)	50
Obrázek 18 - Přehled srovnávačů s cenovou nabídkou v ČR (Zdroj: netmonitor.cz)	53
Obrázek 19 - Skladová karta ze SW Helios Red (Zdroj: Vlastní zdroj)	58
Obrázek 20 - Nastavení hlídacího psa na Alza.cz (Zdroj: www.alza.cz)	59
Obrázek 21 - Diagram tříd navrhovaného systému (Zdroj: Vlastní tvorba)	68
Obrázek 22 - Stavový diagram – Účet (Zdroj: Vlastní tvorba)	71
Obrázek 23 - Diagram případu užití navrhovaného systému (Zdroj: Vlastní tvorba)	72
Obrázek 24 - Sekvenční diagram – Správa faktur (Zdroj: Vlastní tvorba)	77
Obrázek 25 - Aktivitní diagram – Přihlášení do požadovaného systému (Zdroj: Vlastní tvorba)	78
Obrázek 26 - Wireframe – Aktuální stav skladu (Zdroj: Vlastní tvorba)	80
Obrázek 27 - Wireframe – Historie pohybu zboží (Zdroj: Vlastní tvorba)	81
Obrázek 28 - Wireframe – Aktuální stav skladu po úpravách (Zdroj: Vlastní tvorba)	87
Obrázek 29 - Diagram případů užití – přidání EET (Zdroj: Vlastní tvorba)	90
Obrázek 30 - Částečný diagram tříd – přidání EET (Zdroj: Vlastní tvorba)	90
Obrázek 31 - Schéma potencionální realizace (Zdroj: Vlastní tvorba)	91
Obrázek 32 - Sekvenční diagram – Správa skladu (Zdroj: Vlastní tvorba)	104
Obrázek 33 - Sekvenční diagram – Správa účtů (Zdroj: Vlastní tvorba)	106
Obrázek 34 - Sekvenční diagram – Hlídací pes (Zdroj: Vlastní tvorba)	108

Seznam tabulek

Tabulka 1 - Klasifikace informačních systémů (Zdroj: Vlastní tvorba).....	16
Tabulka 2 - Základní položky BI (Zdroj: Vlastní tvorba)	18
Tabulka 3 - Okruhy ERP (Zdroj: Vlastní tvorba).....	19
Tabulka 4 - Přehled fází vodopádového modelu (Zdroj: Vlastní tvorba).....	22
Tabulka 5 - Agilní metodiky (Zdroj: Vlastní tvorba)	28
Tabulka 6 - Modely strukturovaného přístupu (Zdroj: Vlastní tvorba).....	30
Tabulka 7 - Vlastnosti objektově orientovaného přístupu (Zdroj: Vlastní tvorba).....	31
Tabulka 8 - Modely pro návrh IS (Zdroj: Vlastní tvorba)	36
Tabulka 9 - Pravidla pro přístupný web (Zdroj: Vlastní tvorba)	47
Tabulka 10 - Základní vlastnosti použitelnosti webu (Zdroj: Vlastní tvorba).....	48
Tabulka 11 - Položky v XML feed (Zdroj: Vlastní tvorba).....	52
Tabulka 12 - Funkční požadavky aktéra Uživatel (Zdroj: Vlastní tvorba).....	61
Tabulka 13 - Nefunkční požadavky na systém (Zdroj: Vlastní tvorba)	65
Tabulka 14 - Scénář – Přihlášení do systému (Zdroj: Vlastní tvorba).....	74
Tabulka 15 - Sekvenční diagram – Přihlášení do systému (Zdroj: Vlastní tvorba).....	75
Tabulka 16 - Scénář – Správa faktur (Zdroj: Vlastní tvorba)	76
Tabulka 17 - Scénář – Správa skladu (Zdroj: Vlastní tvorba)	103
Tabulka 18 - Scénář – Správa účtů (Zdroj: Vlastní tvorba).....	105
Tabulka 19 - Sekvenční diagram – Hlídací pes (Zdroj: Vlastní tvorba).....	107

1 Úvod

Informační systémy mají určitou životnost závislou na rychlosti vývoje technologií. Pro podnikatele a malé firmy existuje nespočet těchto systémů, které zajišťují funkci řídicích, skladových nebo účetních nástrojů. Proto je důležitý jejich neustálý rozvoj, ať už formou upgrade stávajících nebo návrhem nových, snažících se vyvarovat předchozích nedostatků a přinášejících nová zlepšení. Tato zlepšení jsou jedním z důležitých prvků pro zaměření na novou cílovou skupinu uživatelů.

Navrhovaný informační systém je určen převážně pro osoby samostatně výdělečně činné (*OSVČ*) a drobné firmy, jejichž hlavní podnikatelskou činností není pouze nákup, skladování a následný prodej zboží. Jeho potencionální užívání je nejefektivnější pro řemeslné živnosti. Důvodem pro výběr takové skupiny uživatelů je absence informačního systému, který by splňoval propojení řízení nákupu, skladování, prodeje, evidence účetnictví s funkcí, která automatizuje optimální výběr produktů a dodavatelů pomocí existujících webových služeb. Kritéria optimálního výběru si uživatel stanovuje sám dle vlastních priorit, jako je například cena, termín dodání nebo množství zboží.

Navrhovaný systém umožňuje drobným živnostníkům zavést efektivní řídicí systém bez nutnosti investic do systémů vytvořených na míru konkrétním zákazníkům. Výhodou pro případné uživatele je dále přenositelnost, přístupnost a snadná integrace z důvodu provozu informačního systému pomocí webového prostředí.

K výběru shora uvedené cílové skupiny, pro kterou je návrh vytvořen, autora vedly osobní profesní zkušenosti. Během vlastní praxe se autor potýkal s problémy způsobenými neexistujícím jednotným systémem, který by propojoval všechny prováděné činnosti nebo povinnosti spojené s drobným podnikáním a jeho evidencí. Nabízené propojení funkcí umožňuje snadnou obsluhu s co nejmenší náročností na čas uživatele. Ten pak může většinu svého času věnovat hlavní podnikatelské činnosti vedoucí k zisku, a nikoli nutné evidenci.

Samotný návrh systému je proveden pomocí unifikovaného modelovacího jazyka (*UML*). Jazyk UML je ověřený jako jeden z nejvhodnějších nástrojů pro návrh systémů. Hlavním důvodem četnosti užití jazyka je možnost vytvoření odlišných pohledů na systém, které mohou nabývat různých úrovní abstrakce. Dalším důvodem užití jazyka je jeho možnost univerzálního popisu systému pomocí grafických prostředků, které bourají případné jazykové bariéry mezi vývojáři.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem diplomové práce je pomocí jazyka UML vytvořit návrh informačního systému ke zjednodušení administrativy, která je spojena s nákupem, prodejem a skladováním zboží. Informační systém je navržený z podnětu usnadnění nákupu v internetových obchodech, přehlednější evidenci transakcí a zároveň odlehčení správy skladových zásob.

Navržený informační systém umožňuje uživatelům vytvoření a řízení svých virtuálních skladů, intuitivní nastavení automatického hlídání produktů v internetových obchodech, informování o možných substitutech těchto produktů a kompletní uchování účetních a skladových transakcí.

2.2 Metodika

Metodika řešené problematiky spočívá ve studiu odborných literárních zdrojů, které se zabývají tématem informačních systémů, životních cyklů systémů, metodik vývoje a samotného jazyka UML. Dále je nutné studium odborných internetových zdrojů pro pochopení problematiky webového prostředí a v současné době velmi používaných srovnávačů zboží. Zmíněná problematika je popsána v teoretické části této diplomové práce.

Praktická část diplomové práce je zaměřena na návrh systému, který vychází z poznatků popsaných v části teoretické. Návrh systému je vytvořen pomocí jazyka UML. Pro tvorbu diagramů a wireframů byl použit CASE nástroj MetaEdit+ 3.0 Single User Version a online editor draw.io ve verzi 6.3.8.

Návrh systému je následně doplněn o logický design v podobě wireframů. V závěru praktické části diplomové práce je provedeno kvalitativní šetření. Výsledky kvalitativního šetření byly zakomponovány do výsledných úprav návrhu systému. V rámci praktické části je také realizována reprezentativní část navrhovaného systému v podobě webové aplikace. K implementaci jsou využity skriptovací jazyky jQuery a PHP, dále značkovací jazyk HTML a framework Bootstrap.

V diskusi diplomové práce je nastíněna možnost, jak navrhovaný systém případně realizovat a jsou uvedena předpokládaná úskalí realizace.

3 Teoretická východiska

Rozvoj informačních systémů nastal v druhé polovině sedmdesátých let minulého století. Rozvoj byl zapříčiněn vývojem technologií a vývojem strojové podpory pro návrh informačních systémů. V posledních letech se pro návrh informačních systémů osvědčil unifikovaný jazyk UML, který se stal velmi populárním mezi návrháři IS. K vývoji jazyka UML také přispěl rozvoj objektově orientovaných přístupů [1], [2].

Pro tvorbu praktické části je zapotřebí výše zmíněnou problematiku popsat, správně pochopit a dle definovaných skutečností vytvořit samotný návrh nového informačního systému.

3.1 Informační systém

Pro správné vymezení pojmu informační systém je zapotřebí uvést vhodnou definici. Jelikož mnoho definic vznikalo postupem času, není mnoho definic, které by se shodovaly s vývojem současných informačních technologií. Definice, která je dle autora stále aktuální a v současné době aplikovatelná, je:

„Informační systém je soubor lidí, technických prostředků a metod (programů), zabezpečujících sběr, přenos, zpracování, uchování dat, za účelem prezentace informací pro potřeby uživatelů činných v systémech řízení“ [3, s. 13].

V definici je uvedeno, že IS je také soubor metod (programů). Jako interpretaci lze uvést, že informační systémy se díky rozvoji informačních technologií navrhují pro různé IT platformy. Tyto návrhy jsou realizovány v podobě počítačových aplikací [3], [4].

V dnešní době jsou informační systémy nepostradatelnou součástí organizací a společností. Toto je zapříčiněno především tím, že informační systémy pracují s informacemi. A informace jsou definovány jako data, která mají významovou složku, díky které snižují nevědomost druhé strany. Pomocí dnešních informačních technologií lze tato data přenášet, uchovávat a také s nimi pracovat [3], [2].

Díky vývoji technologií a kladení důrazu na informace a znalosti, se informační systémy stále zdokonalují a přizpůsobují potřebám doby. Tento rozvoj napomáhá k dalšímu postupu vpřed při vývoji nových informačních systémů [5].

Informační systémy se s rozvojem sítí a internetu stávají velmi závislé na správné spolupráci a propojení s jinými informačními nebo databázovými systémy. Problematika kvalitního propojení, zabezpečení a následné správy informačního systému je stále aktuálním tématem. O tyto činnosti se stará správce informačního systému, systémový integrátor nebo správce bezpečnosti [3], [6].

Informačních systémů je velké množství typů, které se dají dále specifikovat například dle účelu použití. Proto se autor dále rozhodl klasifikaci informačních systémů krátce popsat v následující kapitole.

3.1.1 Klasifikace informačních systémů

Informační systémy lze také dělit dle oblasti, do které daný informační systém spadá. Oblasti je možné upřesnit podle základních čtyř profilů organizací, pro které je informační systém navrhován [4], [7]. Klasifikace informačních systémů je uvedena v tab. č. 1, která je zpracována na základě [6], [8], [7].

Název	Popis
Specifický informační systém	Struktura informačního systému je zcela závislá na typu organizace, dále také na přístupu systémového integrátora, který daný systém upravuje dle požadavků zadavatele. Ve většině případů jsou tyto informační systémy zpřístupněny pouze pro zaměstnance dané organizace, tedy se jedná zejména o podnikové informační systémy , které jsou podrobněji popsány v následující kapitole 3. 1. 2 <i>Podnikový informační systém</i> . V těchto informačních systémech se převážně pracuje s vlastními daty a informacemi. Přístup k datům a informacím je nastaven dle individuálních přístupových práv.
Veřejný informační systém	Bývá kombinací podnikového IS s určitými právy přístupu pro veřejnost. Veřejný informační systém je obecně přístupný pro veřejnost. Tento přístup umožňuje čtení dat nebo případnou změnu dat dle nastavených práv. Správu veřejných informačních systémů provádějí různé instituce, například Ministerstvo spravedlnosti ČR, konkrétně <i>eJustice</i> (Veřejný rejstřík a Sběrka listin). Instituce se starají o financování a správu informačního systému.

Státní informační systémy	Jedná se o informační systémy, které nejsou přístupné veřejnosti, ale pouze interním zaměstnancům daného státního sektoru.
Osobní informační systém	Jedná se o podobný informační systém jako je podnikový informační systém. Hlavním rozdílem je možnost využívání správy dat pouze jedním uživatelem, který se stává i správcem celého informačního systému.

Tabulka 1 - Klasifikace informačních systémů (Zdroj: Vlastní tvorba)

Pro účely diplomové práce je nutné vysvětlení problematiky podnikového informačního systému. Navrhovaný informační systém je koncipován pro menší podniky a osoby samostatně výdělečně činné, proto spadá do kategorie specifický informační systém.

3.1.2 Podnikový informační systém

Podnikový informační systém (*neboli PIS*) je druh informačního systému, který je nezbytnou součástí všech prosperujících podniků. Jeho využití přispívá ke zlepšování řízení podnikových procesů neboli toků informací a dat [6], [3].

Podnikový informační systém se používá ve všech vrstvách podniku. Proto PIS obsahuje ve většině případů více různých modulů, které jsou přizpůsobeny pro určitou potřebu. Příkladem takového modulu může být modul pro správu účetnictví, skladování nebo logistiku podniku [6], [3].

Využívání PIS napomáhá k udržení informací, rychlejšímu zpracování a celkové centralizaci informačních toků. Vyžaduje nutnost proškolení zaměstnanců. Výhodou PIS je přidělování práv zaměstnancům dle sektoru jejich zaměření [6].

PIS se skládá z několika podsystémů. Dříve se jednalo o chápání PIS jako ERP (*Enterprise Resources Planning*). ERP představovalo ucelenou a správnou představu o tom, jak má vypadat podnikový informační systém [4], [3].

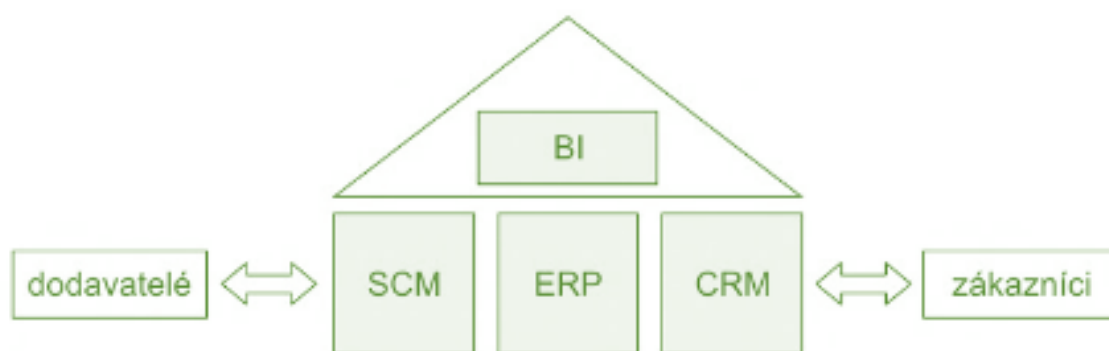
V dnešní době se PIS popisuje spíše tzv. „*ERP II*“. ERP II je rozšíření klasického ERP o několik modulů, které se staly standardem v PIS. ERP II obsahuje normální systém ERP rozšířený o CRM (*Customer Relationship Management*), SCM (*Supply Chain Management*) a BI (*Business Intelligence*) [6], [3], [4].

Postupem rozvoje technologií mohou podniky využívat různých druhů PIS. Druhy se liší podle správy, podle fyzického umístění serverů, na kterých PIS funguje a dle využívání pronajatého systému nebo koupě a vlastnictví celého systému [6].

Správu PIS má na starost interní tým z oddělení IT. Nebo může být správa ponechána na starost systémovému integrátorovi, který implementoval informační systém do podniku [6], [4].

3.1.3 Podsystemy podnikového informačního systému

Pro ilustraci a představu vzájemných vztahů podsystémů PIS, je níže uveden obrázek č. 1, který znázorňuje propojení a vztah jednotlivých podsystémů.



Obrázek 1 - Schéma vztahů modulů PIS (Zdroj: [6])

Business Intelligence (BI)

Business Intelligence představuje několik metod a konceptů, které jsou určeny pro zkvalitnění rozhodovacích procesů firmy. BI popisuje „procesy, znalosti, aplikace, platformy, nástroje, technologie, které podporují porozumění datům, jejich vztahům a trendům“ [3], [4].

BI se snaží odstranit komunikační bariéry uvnitř podniku. Vytváří lepší předpoklady pro zkvalitnění řízení firmy. V praxi se jedná o informační systém, který má přístup k informacím a snaží se je správně předat pro podporu řízení podniku [6].

V následující tabulce č. 2 jsou popsány základní položky, které tvoří BI informačního systému. Tabulka vychází ze zdrojů [6], [4].

Položka	Popis
Aktuální informace	Musí být vytvořen přístup k aktuálním datům ještě před vytvořením určitého rozhodnutí podniku. Jedná se o informace o stavu dodavatelů, odběratelů, prodeje, skladů.
Nezávislost	Informace v podniku musejí být správně formulovány a uloženy. Přístup zaměstnanců v podniku musí být k totožným informacím. Tím se zabrání nežádoucímu komunikačnímu „šumu“.
Pružnost	Pružnost je využívána v případech zjišťování nových informací, které nejsou zatím blíže specifikovány.

Tabulka 2 - Základní položky BI (Zdroj: Vlastní tvorba)

Supply Chain Management (SCM)

Jedná se o řízení dodavatelských řetězců. SCM obsahuje soubor nástrojů a procesů, které slouží k optimalizaci řízení a maximální efektivitě provozu všech součástí celého dodavatelsko-odběratelského řetězce. To vše je konkrétním příkladem správného propojení dodavatelů s odběrateli na bázi informačních a komunikačních technologií [4], [3].

SCM se skládá z pěti základních komponent. Každá komponenta se stará o jednu danou část cesty materiálu. SCM řetězec končí předáním vyhotoveného produktu odběrateli. Základní komponenty jsou popsány v následujícím obr. č. 2., který vychází ze zdrojů [6], [3].



Obrázek 2 - Komponenty SCM (Zdroj: Vlastní tvorba)

ERP

Systém ERP obecně pokrývá čtyři hlavní okruhy, které podniky nejčastěji používají. Tyto oblasti jsou uvedeny v následující tabulce č. 3, která vychází ze zdrojů [6] , [4].

Okruh	Popis
Finance	ERP obsahuje různé moduly například pro správu pohledávek, řízení hotovosti, nákladové účetnictví, dlouhodobý majetek, analýzy ziskovosti.
Personalistika	Do oblasti personalistiky patří například evidence odpracované doby, mzdy, plánování směn, výdaje na cestování zaměstnanců.
Výroba a logistika	Mezi tento okruh patří například sklad a řízení zásob, kalkulace nákladů, plánování výroby, údržba, doprava.
Marketing a prodej	ERP umožňuje správu například ceny reklam, řízení prodeje, plánování prodeje, servis.

Tabulka 3 - Okruhy ERP (Zdroj: Vlastní tvorba)

CRM

Je obecně nazýváno jako řízení vztahů se zákazníky. Jedná se o management, který zaměřen jen na zákaznický přístup. Vztahy se zákazníky jsou velmi důležité a je potřeba je udržovat.

Jako jednu z funkcí CRM lze zařadit statistický sběr dat o zákaznících. Tento sběr dat je dále zpracováván a vyhodnocován. Dobře zpracované statistiky o zákaznících umožňují lépe pochopit potřeby a cíle zákazníků. Takovéto procesy pomáhají k navázání oboustranné komunikace mezi firmou a zákazníky. Jako CRM lze považovat veškerý HW, SW, oddělení i pracovníky, kteří se touto problematikou zabývají [6], [4], [2].

Pro správné pochopení postupu tvorby informačního systému je nutné popsat modely životního cyklu informačního systému.

3.2 Životní cyklus informačního systému

Životní cyklus informačního systému popisuje jednotlivé fáze vývoje informačního systému. Existuje velké množství modelů životního cyklu informačních systémů. Jedním z prvních modelů vývoje informačního systému je model vodopádový [9], [7].

V tomto modelu se jako v prvním specifikovaly jednotlivé fáze vývoje. Tyto ucelené fáze se nadále přebíraly do dalších modelů, kde byly fáze různě upravovány z hlediska časového použití nebo přístupu vývojového týmu [10].

Proto se autor zaměřuje na vodopádový model. Zbývající životní cykly jsou popsány jen krátce. Jedná se o prototypový model, iterativní model a spirálový model.

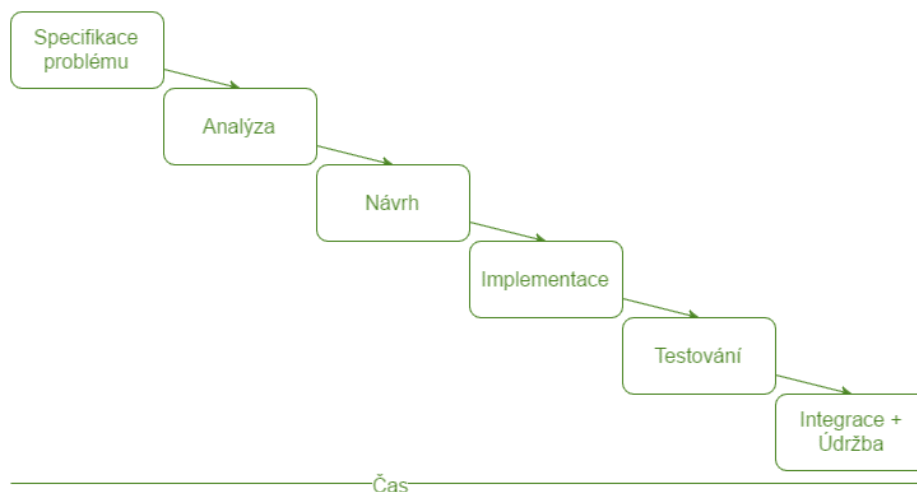
3.2.1 Vodopádový model

Jedná se o sekvenční vývojový proces. Model byl nazván vodopádovým právě pro jeho grafického zobrazení, které připomíná vodopád. Podstatou tohoto modelu je již zmíněné rozdělení na fáze, které na sebe přímo navazují. Následná fáze začíná až bezprostředně po dokončení fáze předchozí. Vodopádový model obsahuje analýzu požadavků, návrh, implementaci, testování, integraci a údržbu [8], [9], [11].

Vodopádový model byl v době svého vzniku velice průlomový. Umožnil celý proces vývoje rozdělit na dílčí části. Díky dělení umožnil systematicky opakovaný postup vývoje.

Při formálním popisu tohoto modelu byl model spíše kritizován pro jeho nepoužitelnost pro tvorbu náročnějších projektů.

Hlavním důvodem nevhodnosti vodopádového modelu pro složité informační systémy je nemožnost zpětné vazby se zadavatelem, případně nemožnost zobrazení částečně funkčního systému. Výsledný systém navrhovaný vodopádovým modelem je možné předvést až po dokončení poslední fáze vývoje. Na základě toho je znemožněna změna struktury navrhovaného systému a systém se musí celý navrhnout znovu s pozměněným zadáním [11], [8] .



Obrázek 3 - Vodopádový model (Zdroj: [11])

Vodopádový model se stal předlohou pro mnoho jeho vlastních modifikací. V současné době bývá někdy obzvláště obtížné rozeznat, zdali se jedná ještě o modifikaci vodopádového modelu nebo již o jiný model pro tvorbu informačního systému [11], [12]. Vodopádový model je zobrazen výše na obr. č. 3.

S ohledem na používání jednotlivých fází vodopádového modelu i v dalších modelech je vhodné jednotlivé fáze vodopádového modelu stručně popsat.

Popis fází vodopádového modelu je zpracován formou tabulky, která je uvedena níže. Tab. č. 4 je vypracována na základě zdrojů [9] , [7], [13]

Fáze	Popis
Specifikace problému	Jedná se o sběr požadavků, kterým se musí v navrhovaném systému vyhovět. Požadavky jsou funkční, jsou to požadavky přímo zadané od zadavatele informačního systému, a nefunkční. Nefunkční požadavky jsou požadavky kladené čistě na samotný systém.
Analýza	Tato fáze představuje transformaci požadavků. Dále obsahuje definování cíle, metodiky, problematiky a času dokončení. Při fázi analýzy jsou vytvářeny jasně definované dokumenty řešené problematiky pro jednotlivé sekce vývojového týmu.
Návrh	Ve fázi návrhu se modelují případné diagramy rozložení samotného systému. Diagramy jsou vytvářeny na dekomponovanou analýzu navrhovaného systému. Při návrhu je systém navrhován z více úhlů pohledu. Návrhová fáze přesně identifikuje například určení vstupů, výstupů, operací nebo vztahů. V této fázi vzniká kompletní a jednoznačný textový nebo grafický popis navrhovaného systému, který je předán k následné fázi implementace.
Implementace	Jednotlivé části systému jsou jednotlivě vytvářeny za pomoci vybraných programovacích jazyků a následně spojovány do větších celků. Výsledkem fáze implementace je ucelený spustitelný systém.
Testování	Během této fáze se testuje spustitelná verze informačního systému pro svojí správnou funkčnost. Testuje se, zdali je systém odladěn rychlostně, zdali je systém stabilní a zdali splňuje podmínky zadané zadavatelem. Výstupem této fáze je odhalení a přesná definice chyb nalezených při testování. Chyby jsou následně odstraněny dle popisů v dokumentaci nazvané „ <i>test plan</i> “. Testovací plán se skládá z položek zvaných „ <i>test case</i> “.
Provoz a údržba	Při této fázi je finální informační systém předán zákazníkovi formou integrace, kterou provádí systémový integrátor. Po předání jsou dohodnuta případná školení budoucích uživatelů informačního systému. Dle smlouvy o vytvoření informačního systému je zákazníkovi prováděna údržba systému v podobě aktualizací nebo oprav při výskytu neodhalených chyb fází testování. Tato doba je pevně stanovena ve smlouvě mezi zákazníkem a společností, která vyvinula informační systém.

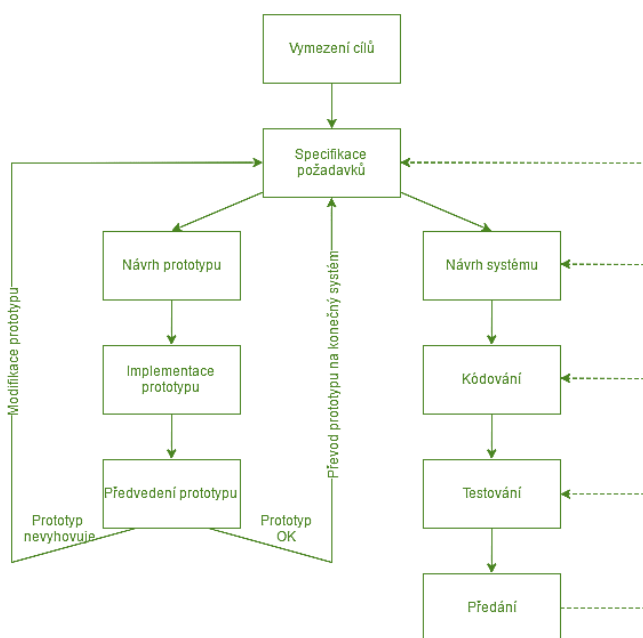
Tabulka 4 - Přehled fází vodopádového modelu (Zdroj: Vlastní tvorba)

3.2.2 Zbylé modely životního cyklu informačního systému

Prototypový model

Prototypový model je obecně používán u menších projektů. Přispívá k tomu převážně tvorbou neúplných verzí celého systému, tedy prototypů. Tento postup umožňuje rychlé změny v malých verzích a na jejich základě je možné tvořit větší systém spojením právě těchto malých funkčních částí [11], [13].

Prototypový model umožňuje rychle reagovat na případné chyby a umožňuje rychlé přepracování malé části systému [11]. Prototypový model je znázorněn níže na obr. č. 4.



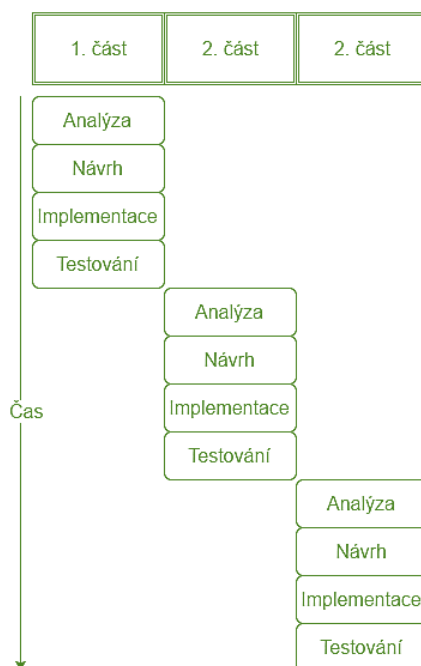
Obrázek 4 - Prototypový model vývoje IS (Zdroj: [11])

Iterativní model

Iterativní model je vhodný jako kombinace iteračních a sekvenčních metodik. Model spočívá v rozdělení celého projektu na menší části. Každá tato část je tvořena dle vodopádového modelu. Po dokončení první části zakončené integrací a údržbou je vodopádová metoda použita znovu od začátku na další část celého projektu [11], [13].

Tento model zachovává nízké procento chyb, právě díky samotnému testování každé části. Díky tomu se chyby nepředávají a po finálním testování projektu není

tak obtížné chyby opravit [11]. Iterativní model životního cyklu informačního systému je znázorněn níže na obr. č. 5.

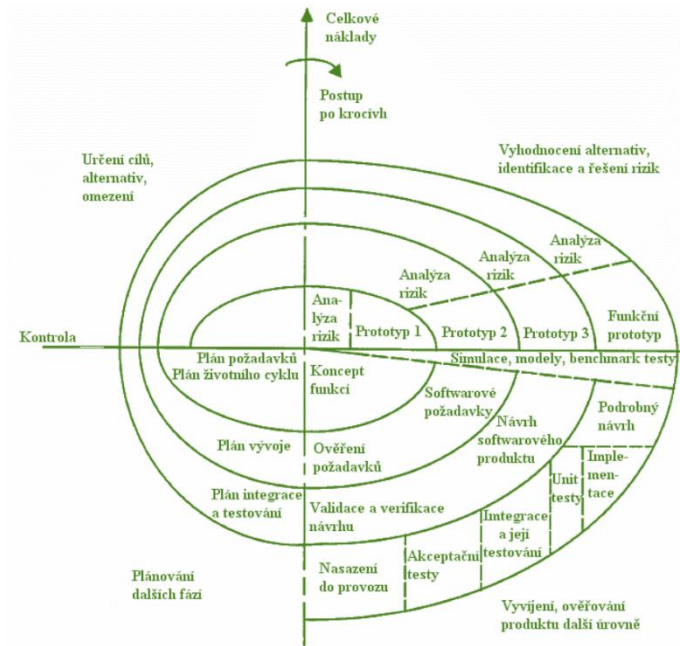


Obrázek 5 - Iterativní model vývoje IS (Zdroj: [11])

Spirálový model

Spirálový model vychází z již zmíněných modelů, a to z kombinace vodopádového a iterativního modelu. Spirálový model dále využívá analýzu rizik. Díky hlavnímu principu této metody, opakování kroků vývoje až do konečné podoby projektu, získal model své jméno spirálový [11], [14]. Spirálový model je zobrazen na obr. č. 6 níže.

Celý projekt je rozdělen na menší části stejně jako u iterativního modelu. Části prochází stále dokola spirálou, která je rozdělena do čtyř kvadrantů. Kvadranty jsou analýza, hodnocení, vývoj a plánování. Díky zvýšenému opakování kroků je tento model vhodnější u většího počtu projektů než předchozí modely [9], [13].



Obrázek 6 - Spirálový model vývoje IS (Zdroj: [14])

Z modelů životních cyklů informačního systému vycházejí přístupy k tvorbě informačního systému. Proto se autor dále zaměřuje na popis přístupů k tvorbě neboli k metodikám vývoje informačního systému.

3.3 Metodiky vývoje informačního systému

Jelikož proces tvorby informačního systému tvoří velké množství dílčích úkolů, které je důležité splnit, je zapotřebí správně rozdělit pracovní role.

Týmy, které se podílejí na tvorbě informačního systému, se skládají z mnoha druhů pracovních rolí, které se dělí dle jejich zaměření. Mezi nejvýznamnější profese patří projektový manažer, analytik, programátor, grafik a tester [13], [14].

Projektový manažer má na starost plynulý chod tvorby informačního systému. Manažer projektu je zodpovědný za projekt ve všech jeho fázích. Manažer řídí projektový tým, zpracovává projektový plán a v závěru vyhotovení projektu hodnotí dosažení cílů a výsledný projekt. Jelikož sám řídí celý tým, je důležité správné zvolení způsobu vedení. Nejčastěji bývá tým veden demokratickým způsobem. V kritických případech je zapotřebí důslednějšího způsobu vedení, a to direktivním způsobem [10], [14].

Pro kvalitní a hodnotný informační systém je také důležitý správný postup vývoje. Těmito postupy se zabývají metodiky vývoje informačního systému. Metodiky vývoje informačního systému popisují všechny fáze řešení. V rámci metodiky vývoje se jedná o souhrn postupů, které slouží k úspěšnému vyhotovení funkčního informačního systému [14], [13].

Vývoj informačního systému byl již v šedesátých letech popsán jako posloupnost dílčích kroků, kterých je potřeba dosáhnout, aby byl výsledný informační systém zhotoven.

Tato posloupnost dílčích procesů se nazývá metodiky vývoje informačního systému. Metodiky vývoje informačního systému popisují, která pracovní profese se podílí na určité části projektu. Dále určuje, co následuje po předchozí fázi vývoje. Tento postup určuje způsob dosažení výsledného informačního systému [12], [13].

Metodiky vývoje informačního systému lze klasifikovat z časového hlediska na dva celky [13]:

- tradiční metodiky vývoje;
- moderní metodiky vývoje.

3.3.1 Tradiční metodiky vývoje informačního systému

Tradiční metodiky vývoje vycházejí z modelů životních cyklů, které jsou popsány detailně v kapitole 3.2 *Životní cyklus informačního systému*, proto autor tyto typy metodik vývoje informačního systému podrobněji nepopisuje.

Tradiční metodiky jsou založeny na definici postupů a projektovém řízení informačního systému. Požadavky jsou stanoveny na začátku vývojového procesu a v průběhu vývojového procesu nejsou upravovány. Ve většině případu dochází ke špatně definovaným požadavkům již na začátku vývoje. To vede k tomu, že výsledný informační systém není dle představ zadavatele, jelikož vyhotovený informační systém vidí zadavatel až po jeho úplném dokončení [12], [14], [10].

Tradiční metodiky z těchto důvodů nevyhovují současným požadavkům. Proto se postupem času začaly vyvíjet metodiky informačního systému, které se zaměřují na interakci se zákazníkem. Tyto metodiky se nazývají moderní metodiky vývoje informačního systému [13].

3.3.2 Moderní metodiky vývoje informačního systému

V současnosti mnoho vývojových firem využívá moderní metodiky vývoje informačního systému. Jedná se například o:

- metodiky rigorózní,
- metodiky agilní.

Rigorózní metodiky

Jedná se o rozsáhlé řešení tvorby informačních systémů. Rigorózní metodiky jsou rozsáhlé, jelikož detailně popisují problematiku, činnosti a procesy použité při vývoji informačních systémů. Tyto metodiky vyžadují různé dokumentace. Finální implementace informačního systému se vyznačuje tím, že bývá často zdlouhavá [13], [14], [7].

Rigorózní metodiky jsou vhodné pro neměnné požadavky zadavatele. Pokud je problém jasně definován a není v průběhu projektu potřeba výraznějších změn v zadání, je rigorózní metodika použita správně. Mezi rigorózní metodiky vývoje informačního systému patří zejména metodika OPEN, UP nebo RUP [14], [7].

OPEN (*Object-oriented Process, Environment and Noation*) je metodika, která je založena na *objektově orientovaném vývoji*. Je to metodika, která je vhodná pro jakýkoli projekt. Vyznačuje se přizpůsobivým řešením dle schopností vývojového týmu, požadavků zákazníka a velikosti projektu [10], [12].

UP / RUP (*Unified Process / Rational Unified Process*) jsou metodiky, které jsou také založeny na *objektově orientovaném přístupu*. Tyto metodiky využívají jazyk UML pro tvorbu informačního systému. Obě tyto metodiky se vyznačují nízkou podporou správy informačního systému. V případě RUP se jedná o komerční verzi. Metodika UP je uživatelsky přívětivější a obsahuje rozsáhlou dokumentaci [14], [10], [7].

Agilní metodiky

Agilní metodiky jsou od předešlých metodik rozdílné v tom, že kladou důraz na spolupráci se zákazníkem, který je částečně brán jako člen vývojového týmu. Úzká spolupráce snižuje potřebu práce na složitých dokumentacích. Dokumentace je nahrazena osobní komunikací nebo případnými komunikačními toky mezi oběma stranami. Agilní metodiky jsou časté u tvorby projektů, na kterých pracují méně početné týmy vývojářů [13], [7].

Tyto metodiky jsou velmi vstřícné pro případné změny. Agilní metodiky jsou aplikovatelné nejen v případech změny zadání ze strany zákazníka, ale i v případech změn technologických. Touto tvorbou lze rychle dosáhnout funkčního řešení, které se flexibilně přizpůsobuje změnám [14], [7]. Typy agilních metodik vývoje informačního systému jsou uvedeny v následující tabulce č. 5, která je zpracována na základě [13], [7], [10].

Název	Popis
SCRUM	Spočívá v každodenním setkávání týmu vývojářů, kde každý pracovník mluví o své práci na projektu. Například jaké problémy během práce zjistil a jaké navrhuje jejich řešení. Díky každodenní poradě mají ostatní pracovníci přehled i o práci ostatních. Problém jednoho pracovníka může být vyřešen radou jiného pracovníka. Toto může vést k urychlení práce jednotlivých úseků. Sezení probíhají v omezené době (2–4 týdny). Tato doba bývá označována jako " <i>sprint</i> ".
Extrémní programování	Důležitým aspektem je pohled na zákazníka, který se částečně stává členem programátorského týmu. Požadavky nejsou přesně definovány. Zákazník má možnost diskutovat s dalšími členy týmu, a i pomocí testování se zapojit do průběhu vývoje. Zákazník svou interakcí s vývojáři vytváří tzv. „kartu historie“, díky které plně předává své požadavky na projekt. V extrémním programování lze tvořit několik verzí systému, jelikož vývojový tým má z "karty historie" nové podklady, které implementují do nové verze systému. Po vydání nové verze musí programátoři verzi otestovat. Pokud je testování úspěšné, je informační systém přijat a verze se stává základem pro další iteraci systému. Další prvkem, který výrazně odlišuje extrémní programování od ostatních typů je spolupráce dvou programátorů na jedné pracovní stanici.
ASD	Snaží se, co možná nejvíce uspokojovat změny v průběhu vývoje. Postupy ASD nejsou přesně definovány. Obecně lze říci, že tvoří cyklus, který spočívá v opakování těchto tří částí: spekulace, spolupráce a učení.

Tabulka 5 - Agilní metodiky (Zdroj: Vlastní tvorba)

Jelikož autor v případě popisu metodik vývoje informačního systému zmiňoval objektivě orientovaný přístup, rozhodl se tento pojem podrobněji vysvětlit v následující kapitole.

3.4 Druhy přístupu k analýze a návrhu informačního systému

Postupem času se v analýze a návrhu informačního systému objevily dva základní přístupy. Jedná se o strukturovaný přístup a objektově orientovaný přístup (*OOP*). Oba přístupy se vyznačují jinými charakteristikami [15], [16].

Společnou myšlenkou těchto dvou přístupů je dekompozice celého vývoje informačního systému na dílčí části. Tyto části je důležité identifikovat a popsat. Nezbytnou součástí dekompozice je zachování vazeb mezi dílčími částmi, aby nedošlo k narušení integrity celku [7] .

3.4.1 Strukturovaný přístup

Strukturovaný přístup byl navržen v sedmdesátých letech dvacátého století. Základy tohoto přístupu definoval Ton deMaco v roce 1979 ve své práci „*Strukturovaná analýza a specifikace systémů*“. Stejně problematice se věnovali i Ken Orr, Larry Constatine, Vaughn Frick, Stevan Ward, Peter Chen nebo například Edward Yourdon. Právě Edward Yourdon popsal strukturní analýzu v devadesátých letech dvacátého století [15], [16], [7] .

Strukturovaný přístup využívá základního prvku strukturalizace. Tento prvek je pospán 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 [7], [15] .

Základními částmi strukturovaného přístupu jsou funkční modelování, procesy a možnost dekomponování procesů. Z pohledu abstrakce lze tento přístup rozčlenit na tři modely, které jsou popsány v následující tab. č.6, která je zpravována na základě [9], [4], [17].

Název	Popis
Konceptuální model	Rozpoznává datové objekty a jejich vztahy. Definuje „návrh“ jako nezávislý na technologickém prostředí a dále určuje co je obsahem systému.
Logický model	Představuje relační schéma včetně všech integritních omezení. Logický model není zatížen konkrétní implementací, avšak přesně určuje strukturu konkrétních dat. Tento model bývá často vyjádřen v podobě DFD a ERD diagramů. Model DFD je představitelem funkčního modelování díky dekomponování procesů v jednotlivých diagramech. Následné rozklady a upřesnění procesů v DFD se stupňují dle úrovně dekompozice.
Fyzický datový model	Stará se o implementaci v daném prostředí.

Tabulka 6 - Modely strukturovaného přístupu (Zdroj: Vlastní tvorba)

3.4.2 Objektově orientovaný přístup

„Objektově orientovaný přístup, rozvíjený od poloviny 80. let minulého století, je založen na objektech. Objekt je určitý prvek systému, který vykazuje vlastní chování (reakci na vstupní impulsy – metody) a má určité vlastnosti (atributy). Objekty, které mají stejné chování a obdobné či stejné vlastnosti, tvoří třídu objektů“ [4, s. 31].

Objektově orientovaný přístup se snaží vše transformovat do podoby objektů. Objektově orientovaný přístup je postaven na hledání množiny objektů, jejich vazeb, atributů a metod. Objekty mohou díky tomuto spojení představovat jednotlivé části celku [9].

Stejně jako ve strukturovaném přístupu je i zde potřeba provést dekompozici systému na menší části. Avšak objektově orientovaný přístup využívá jiný způsob abstrakce, a to převážně při rozdělování důležitějších částí od méně důležitých [16].

Vlastnosti objektově orientovaného přístupu

Pro objektově orientovaný přístup jsou charakteristické některé vlastnosti. Tyto vlastnosti jsou popsány v následující tab. č. 7, která je zpracována na základě [1], [16], [14].

Vlastnost	Popis
Dědičnost	Dědičnost je vlastnost, která se využívá pro zpřehlednění celé struktury, která pracuje na hierarchickém základě. Hierarchie vytváří vztah <i>nadtřída – podtřída</i> . Objekty mohou mezi sebou dědit metody a atributy. Převážně se jedná o dědění všech vlastností <i>nadtřidy</i> , které získá i nově vytvořená <i>podtřída</i> . Aby měla podtřída smysl, musí mít minimálně jednu specifickou vlastnost, kterou nadtřída nemá.
Zapouzdření	Je důležité pro zobrazování obsahu objektů. Zapouzdření odlišuje vnější chování celku od vnitřního v objektu. Zapouzdření skrývá určité informace na základě přístupu a umožňuje oddělit objekt od okolí.
Abstrakce	Odděluje důležité informace od nepodstatných. Abstrakce především rozlišuje, co objekt dělá a jaký objekt je, nebo jak je objekt implementován.
Polymorfismus	Umožňuje objektům reagovat různě na stejnou zprávu. Objekty pod jednou třídou mohou mít díky dědičnosti přenesené stejné metody, avšak každá metoda bude dělat u každého objektu něco jiného. Pokud jiný objekt pošle zprávu těmto objektům a tato zpráva se bude jevit pro odesílatele jako stejná, každý objekt na ní může zareagovat jinak. Díky polymorfismu může být operace implementována více než jednou metodou.

Tabulka 7 - Vlastnosti objektově orientovaného přístupu (Zdroj: Vlastní tvorba)

Prvky objektově orientovaného přístupu

Základním prvkem je **objekt**. Jedná se o stavební jednotku samotného objektově orientovaného přístupu, která znázorňuje určitou část reálného světa. Objekt obsahuje vlastnosti. Vlastnosti jsou atributy a metody. Objekt představuje buď reálný předmět nebo abstraktní pojem. **Instance objektu** je konkrétní výskyt objektu [16], [9].

Dalším důležitým prvkem objektově orientovaného přístupu je **třída objektů**. Třída je základní konstrukční prvek OOP. Třída objektů představuje skupinu objektů, které mají stejné vlastnosti [18], [19].

Třídy mohou náležet jiné třídě. Tyto třídy jsou nazývány **podtřídy** a **nadtřídy**. Třídy stejně jako objekty mají své jméno nebo název [16].

Každý objekt, který je zároveň instancí třídy, má vlastní hodnoty atributů, avšak sdílí strukturu atributů a operace společně s ostatními instancemi této třídy [1], [17].

Mezi třídami objektů se vyskytují různé vazby. Tyto vazby jsou dále popsány v kapitole, která se zabývá unifikovaným jazykem UML.

3.5 Unified Modeling Language

Před vznikem jazyka UML bylo velice složité definovat požadavky mezi zadavatelem a tvůrcem informačního systému. Proto softwaroví inženýři hledali prostředek, který by upřesnil komunikaci mezi zadavatelem a tvůrcem informačního systému [14].

Jako vhodné řešení se ukázaly postupy, které měly přesně definovanou syntaxi a sémantiku. Řešení spočívalo v grafických diagramech, které znatelně snižují nejasnost a nepřesnost v interpretaci. Diagram umožňuje kvalitnější sdělování požadavků než textový popis. Grafický jazyk vznikl za pomoci softwarového inženýrství [2], [12].

Softwarové inženýrství

Softwarové inženýrství dle autora jednoznačně vystihuje definice:

„Softwarové inženýrství je inženýrská disciplína zabývající se praktickými problémy vývoje rozsáhlých softwarových systémů.“ (Prof. Ing. Ivo Vondrák, CSc. – VŠB – Technická univerzita Ostrava)

Softwarové inženýrství říká, jak vytvořit projekt za pomoci UML. Tento nejrozsáhlejší proces se nazývá Vodopádový model, který je popsán v kapitole 3.2.1 *Vodopádový model* [20].

Pro návrh a tvorbu informačních systémů pomocí grafického zobrazení se používají CASE (*Computer Aided Software/Systems Engineering*) nástroje. CASE nástroje představují software pro vývoj počítačových programů a systémů za účelem získání vyšší kvality, bezpečnosti, bezchybnosti apod. Za pomoci CASE nástrojů lze dosáhnout modelování navrhovaného systému pomocí grafických diagramů, které je následně možné převést do počítačového kódu. Zpětný proces je také umožněn, což dopomáhá k rychlejšímu pochopení zdrojového kódu [9].

3.5.1 Obecně o UML

UML je grafický jazyk pro objektově orientovanou analýzu a návrh informačního systému. Jedná se o jazyk, který pomocí grafických diagramů popisuje architekturu informačního systému [15], [11].

Díky definici UML jazyka lze informační systém vyvíjet, upravovat nebo pouze pochopit stávající architekturu informačního systému. Grafické diagramy mají své standardizované grafické prvky. Díky jednoznačnosti lze vytvořit diagramy, které jsou univerzální a srozumitelné pro každého, kdo zná prvky daného grafického jazyka [12], [17].

Tento grafický popis je při vývoji nového informačního systému daleko vhodnější než použití slovního popisu problému, který může být při překladu pochopen odlišně [12], [19], [18].

Prostředky jazyka UML nabízejí možnost popisu statické a dynamické stránky informačního systému. Popis statické části zobrazuje a jednoznačně definuje strukturu

informačního systému. Popis dynamické části zobrazuje chování informačního systému a jeho částí v čase. Při následném propojení obou částí se vytvoří ucelená představa o architektuře celého informačního systému [21], [7].

Jazyk UML se snaží o sjednocení některých částí, ke kterým během návrhu informačního systému dochází. Jedná se o oblasti [18]:

- **Vývojový cyklus** – jazyk UML pomocí grafického zpracování nabízí vizuální syntaxi při modelování informačního systému. Tato vizualizace umožňuje zachytit veškeré vývojové procesní stavy, tedy od požadavků až po finální implementační dokončení [19].
- **Aplikační domény** – jelikož jazyk UML využívá OOP, je možné jej použít pro všechny druhy informačních systémů. UML umožňuje modelovat systém zasazený do reálného času nebo podpůrné systémy pro rozhodování [17].
- **Implementační jazyky** – jazyk UML byl vytvořen tak, aby byl aplikovatelný na jakýkoliv programový jazyk. Díky tomu není jazyk UML závislý na použití konkrétního programovacího jazyka. Jelikož je jazyk UML založen na OOP, je návrh vytvořený pomocí něho lépe realizovaný objektově orientovanými programovacími jazyky jako jsou například Smalltalk, Java nebo C# [15], [11].

Historie UML

Organizace Object Management Group (*neboli OMG*) se jako první začala zajímat o standardizaci grafického jazyka pro objektově orientovaný přístup. OMG ze začátku nemělo v okolí velkou podporu, avšak díky pokusům několika znalců OOP o vytvoření kvalitní metodiky, měla organizace základy pro vytvoření UML. Jazyk UML byl velmi ovlivněn metodikou OMT (Object modeling technique od Rumbaugh a Premerlani) a OOSD (Object oriented doftware development od G. Booch) [21], [12].

První verzí UML byla verze 1.0 vydaná v roce 1997 firmou OMG. V následných letech se tvorba novějších verzí UML odvíjela od využití již existujících metodik. Úpravy po stránce grafické a přidávání dalších metodik probíhalo od verze 1.1 až do vydání verze UML 2.0. Verze 1.1 byla prohlášena oficiálním standardem firmy OMG. Výraznější změny v jazyce UML se dostavily právě až s příchodem verze 2.0. V roce 2007 byla vydána verze 2.1.2 a o čtyři roky později verze 2.4.1 [17], [15], [12].

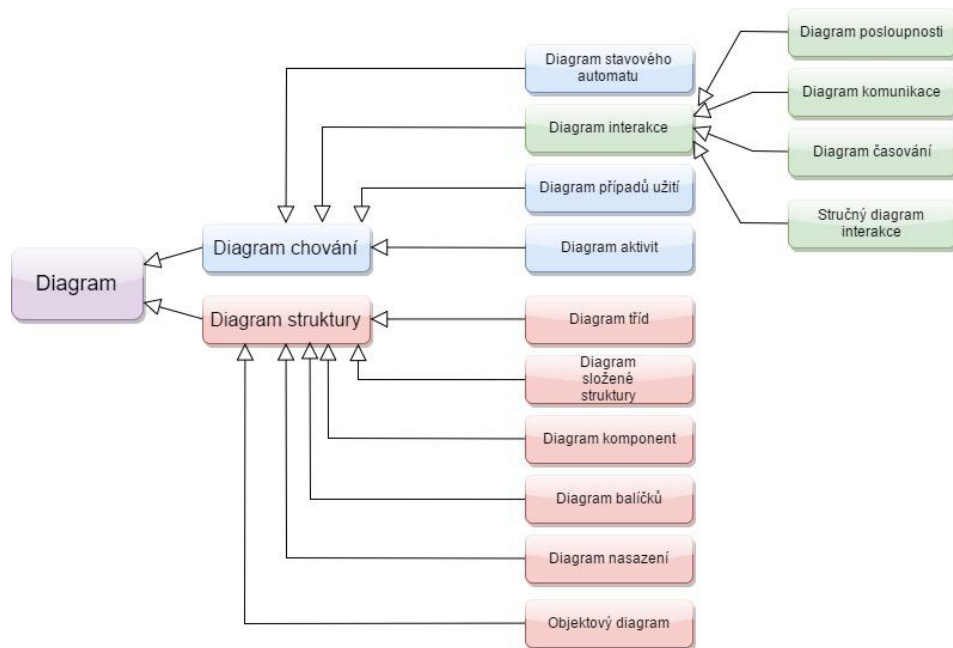
3.6 Diagramy v UML

Jazyk UML má přesně definovány diagramy, díky kterým lze popsat informační systém z různých pohledů. Proto se jazyk UML nazývá grafickým jazykem.

Diagramy jsou dvourozměrné a zobrazují určité aspekty celého systému. Některé diagramy jsou utvořeny k popisu celého systému, nicméně některé všechny systémy takto popsal nelze převážně z důvodu velikosti popisovaných systémů. Pro tyto případy jazyk UML umožňuje hierarchické vnořování a uspořádání většího počtu diagramů stejného typu. V tomto hierarchickém celku je možnost zobrazení si celého systému postupně po jednotlivých diagramech.

Diagramy zobrazují určitou úroveň abstrakce popisovaného systému. Díky tomu lze dosáhnout porozumění popisu i u širší skupiny uživatelů, kteří se na jazyk UML nespécializují. Umožňují však dostatečně přesný popis informačního systému pro specialisty v oboru [12], [21].

Na obrázku č. 7 níže jsou zobrazeny diagramy, které UML nabízí. Diagramy se dělí na dvě větší skupiny, a to diagramy chování a diagramy struktury. Každá ze skupin obsahuje různé typy diagramů. Skupina diagramy chování zahrnuje skupinu diagramů interakcí [21], [9].



Obrázek 7 - Přehled diagramů UML (Zdroj: Vlastní tvorba)

Postup modelování informačního systému spočívá v rozdělení informačního systému na tři základní modely. Každý z modelů se zaměřuje na určitý pohled informačního systému a využívá určité diagramy.

Jednotlivé modely jsou stručně popsány v následující tabulce č. 8, která je zpracována dle [2], [19], [15].

Název	Popis
Model tříd	Popisuje statický pohled na informační systém. V tomto modelu nelze vyjádřit interakci mezi třídami objektů, ke kterým dochází v čase. Pro popis statické struktury objektů a jejich vztahů se využívá diagram tříd.
Model stavů	Popisuje dynamické chování systému při jeho činnosti. Specifikuje změny objektů a jejich vztahů v čase. Pro popis dynamického chování systému se využívá stavový diagram, který se standardně vytváří jeden pro každou třídu objektů.
Model interakcí	Popisuje interakce, ke kterým dochází v systému. Jedná se o uživatelské interakce nebo o interakce mezi komponenty systému. Ukazuje, jaké problémy mohou při komunikaci nastat. Pro popis modelu interakcí se využívají diagramy případů užití, sekvenčních modelů a diagramů aktivit.

Tabulka 8 - Modely pro návrh IS (Zdroj: Vlastní tvorba)

Diagramy, které jsou uvedeny na obr. č. 7, nejsou nutné v rozsahu této diplomové práce využít všechny. Pro popis navrhovaného informačního systému a pro znázornění ucelené představy o informačním systému postačí diagramy, které jsou zmíněny v tab. č. 8. Tedy diagramy, které se používají k popisu jednotlivých modelů.

3.6.1 Diagram tříd

Diagram tříd se používá pro modelování statického pohledu na navrhovaný systém. Tvorba diagramu tříd je nejobsáhlejší ze všech diagramů UML, jelikož se snaží popsat statickou strukturu celého navrhovaného systému [9], [19]. Pokud se jedná o rozsáhlejší systém, může být tvorba diagramu tříd komplikovanější.

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

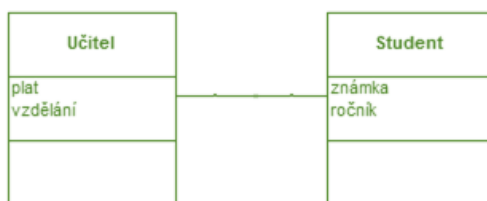
1. **Doménový model** – modeluje základní logickou strukturu navrhovaného systému. Tato struktura obsahuje pouze třídy objektů, atributy a vztahy mezi třídami objektů [2], [15].
2. **Návrhový model** – doplňuje doménový model o datové typy atributů, metody a další doplňující třídy objektů potřebné v modelu. Dále model obsahuje upřesnění vazeb mezi třídami objektů [7], [22].
3. **Implementační model** – zaměřuje se na podrobný popis struktury modelu. Tento model bývá určen koncovým vývojářům systému [17].

Stavební prvky diagramu tříd

Základním stavebním prvkem diagramu tříd je třída objektů. Tato třída objektů má svůj název a vlastnosti (atributy a metody). Třída může být i abstraktního typu, který se sám o sobě v systému nevyskytuje jako reálná třída objektů, avšak má společné atributy či chování s dalšími třídami objektů. Jméno třídy vždy začíná velkým písmenem a žádné dvě třídy se nesmí jmenovat stejně. Atributy popisují určitý znak třídy objektů [15], [21].

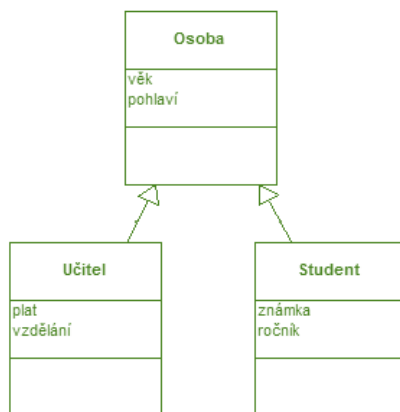
Podstatnou část diagramu tříd tvoří vazby mezi třídami objektů. Jedná se o čtyři typy vazeb, které jsou dále podrobněji popsány.

Asociace – jedná se o nejzákladnější typ vazby mezi dvěma třídami objektů. Představuje vazbu, která upřesňuje vztahy mezi instancemi tříd objektů [15]. Notace v jazyku UML je uvedena na následujícím obr. č. 8.



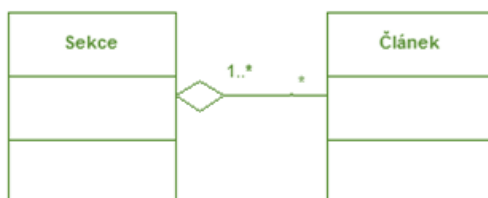
Obrázek 8 - Zobrazení asociace v diagramu tříd (Zdroj: Vlastní tvorba)

Generalizace - je typ vazby, při které podtřída dědí všechny atributy a metody své nadtřídy. Při generalizaci musí být dodrženo pravidlo, že podtřída obsahuje všechny vlastnosti nadtřídy, ale i minimálně jednu novou vlastnost [15], [2]. Notace v jazyce UML je znázorněna na obr. č. 9.



Obrázek 9 - Zobrazení generalizace v diagramu tříd (Zdroj: Vlastní tvorba)

Agregace - reprezentuje vztah mezi dvěma třídami objektů, který se nazývá vztah *součást – celek*. Tento vztah je typu: „*skládá se z*“ nebo „*je součástí*“. Pokud zanikne celek, pak součást může existovat nezávisle na celku [9], [18]. Na obr. č. 10 je znázorněna notace agregace v UML.



Obrázek 10 - Agregace v diagramu tříd (Zdroj: Vlastní tvorba)

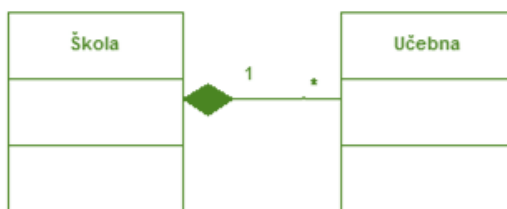
Jako příklad agregace lze uvést popis, který je znázorněn na obr. č. 10, tedy že celek může držet tzv. kolekci. Jako celek může být brána *sekce* a jako součást *článek*. Článek může existovat sám o sobě a může být součástí i jiných kolekcí.

Agregace se vyskytuje nejčastěji v kombinaci s určením *multiplicity*. Multiplicita neboli násobnost popisuje počet výskytu jedné třídy vzhledem k druhé třídě. Jedná se o omezení v počtu instancí tříd ve vzájemné vztahu. Násobnost lze takto popsat u agregace, asociace a kompozice [15], [2].

Multiplicita bývá označena následovně:

- **1 (číslo)** – číslo označuje přesný počet výskytů. V tomto případě se jedná právě o jeden výskyt [21].
- *** (hvězdička)** – hvězdička označuje libovolný počet. Tím je myšlen i nulový výskyt.
- **1..* (interval)** – dvě tečky mezi symboly označují interval, v jakém se může četnost pohybovat. Počet může být určen i přímo výčtem prvků. (1, 2, 4, 5..*) [17].

Kompozice – jedná se o speciální případ agregace (tzv. *silná agregace*). Všechny součásti patří jednomu celku a pokud zanikne celek, zaniknou všechny části. Části nemohou bez celku existovat. Multiplicita je značena shodně jako u agregace [19], [18]. Notace kompozice v jazyce UML je znázorněna níže na obr. č. 11.



Obrázek 11 - Kompozice v diagramu tříd (Zdroj: Vlastní tvorba)

3.6.2 Stavový diagram

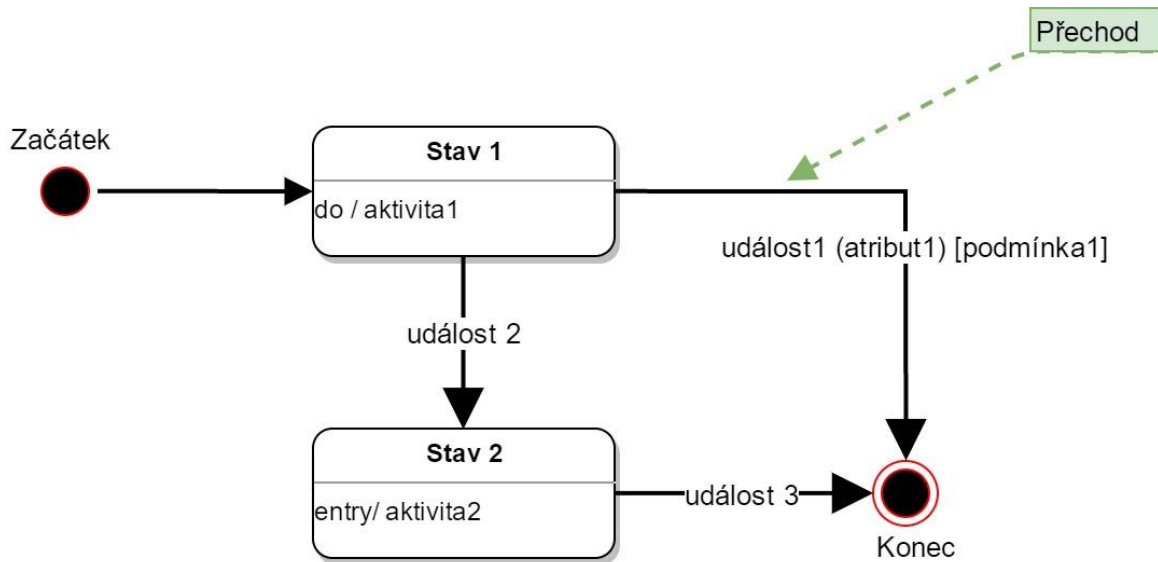
Stavový diagram slouží k popisu dynamického chování. Umožňuje sledovat změny částí systému v čase. Stavové diagramy se tvoří k jednotlivým třídám objektů. Stavový diagram představuje vztahy událostí a stavů, které tvoří hlavní prvky stavového diagramu. Dalšími prvky stavového diagramu jsou přechody, aktivity, podmínky a atributy [21], [17], [19].

Stavový diagram je popisován od výchozího stavu až po koncový stav objektu. Přechodem je označena situace, kdy v určitém stavu dojde k události, která určí následující stav objektu. Přechod v diagramu je znázorněn šipkou mezi stavy a jeho popis je určen danou událostí, která přechod vyvolala [17], [9].

Aktivita popisuje odezvu na příchozí událost. Aktivity mohou být uvedeny v těle stavu, nebo za událostí v popisu přechodu. Při popisu aktivity uvnitř stavu pomáhá notace *do/A*. Popsaná aktivita *A* začne na začátku stavu, ve kterém je uvedena. Podmínka udává

určitý fakt, který musí být splněn pro umožnění přechodu. Atributy jsou hodnoty, které jsou přeposílány pomocí přechodu z jednoho stavu do druhého [15], [21], [9].

Na obrázku č. 12 je uveden příklad obecného stavového diagramu, ve kterém je uvedeno zobrazení jednotlivých prvků stavového diagramu.



Obrázek 12 - Příklad stavového diagramu (Zdroj: Vlastní tvorba)

3.6.3 Digramy pro tvorbu modelu interakcí

Mezi diagramy, které se používají pro tvorbu modelu interakcí a popisu chování systému patří [2]:

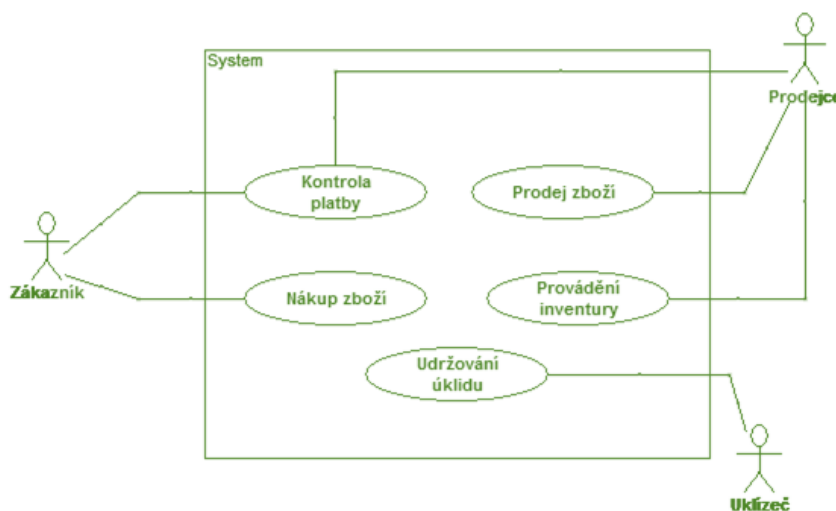
- diagram případů užití;
- sekvenční modely skládající se ze scénářů, které jsou následně popisovány sekvenčními diagramy;
- diagram aktivit.

Diagram případů užití

Z modelu interakcí diagram případů užití popisuje nejobecnější pohled na navrhovaný systém. Tento pohled zobrazuje systém jako celek a všechny externí účastníky systému, kteří komunikují se systémem [23], [15].

Diagram případů užití rozděluje navrhovaný systém na vnější a vnitřní část. Vnější část představují *aktéři*, vnitřní část diagramu tvoří systém, který je rozdělen na jednotlivé *případy užití* (neboli *Use Case*). Aktéři vstupují do systému a využívají určité případy užití v systému. Případy užití jsou dílčí části, které jsou součástí systému, a popisují základní funkcionality navrhovaného systému [17], [22].

Na obr. č. 13 je uveden příklad diagramu případu užití. Jsou zde uvedeny tři aktéři (*Zákazník*, *Prodejce*, *Uklízeč*). Tito aktéři přistupují k určitým případům užití (např. *Uklízeč* vykonává *Udržování úklidu* nebo *Zákazník* uskutečňuje *Nákup zboží*).



Obrázek 13 - Příklad diagramu případů užití (Zdroj: Vlastní tvorba)

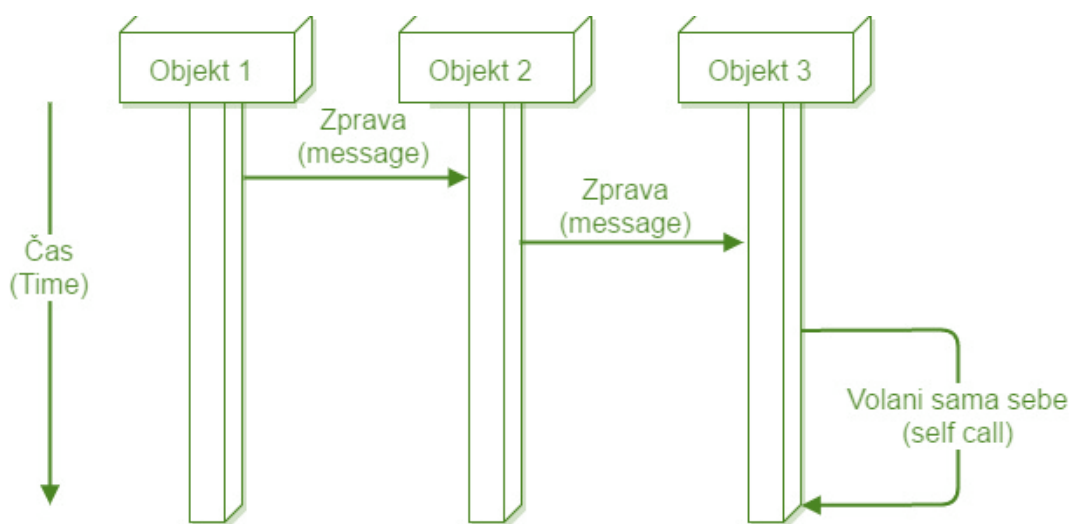
Sekvenční modely

Sekvenční modely tvoří **scénáře** a **sekvenční diagramy**. Scénáře popisují textovými výroky komunikaci a aktivity mezi třídami objektů, objekty nebo aktéry. Ve většině případů se pro jednotlivé případy užití navrhuje několik scénářů [17], [7].

Sekvenční diagram popisuje interakci mezi jednotlivými klasifikátory. Klasifikátor může být v této interakci třída, aktér, objekt nebo komponenty systému. Sekvenční diagramy mají výrazně jinou strukturu než ostatní diagramy. V horní liště se nachází postupný výčet všech zainteresovaných klasifikátorů. Z každého účastníka interakce vede svislá linie. Této linii se říká *lifeline*. Objekty a komponenty mezi sebou komunikují a znázorňují posloupnost událostí, které se vyskytují během určitého případu užití systému [17], [2], [22].

Každá zpráva mezi objekty je znázorněna šipkou mezi čárami života. Za zprávu se považuje požadavek objektu o volání operace druhého objektu. Objekty mohou využívat i tzv. self-call. Self-call je volání sama sebe [17], [18].

Na obrázku č. 14 níže je uveden obecný příklad sekvenčního diagramu. Je zde znázorněna možnost zobrazení interakcí mezi objekty.



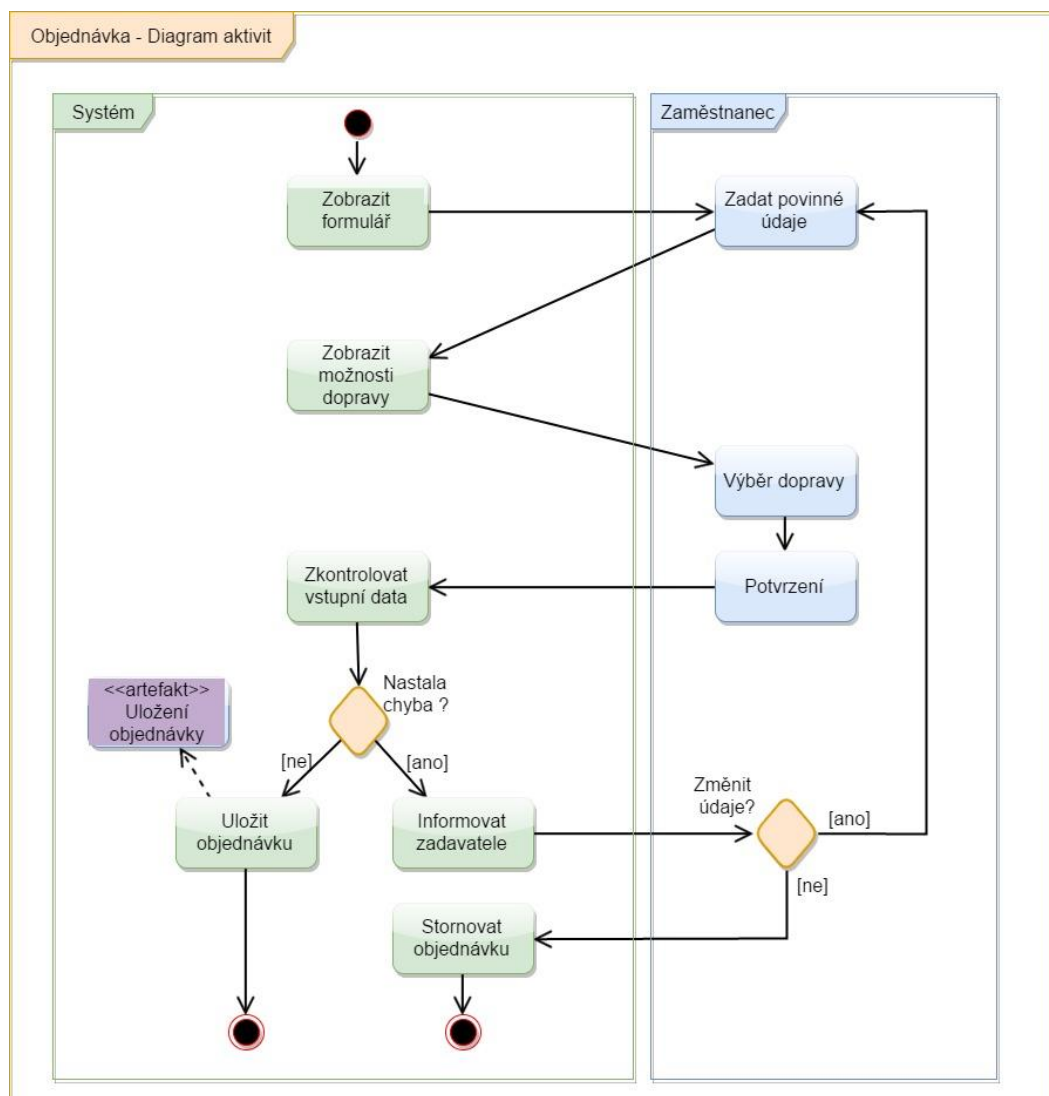
Obrázek 14 - Příklad sekvenčního diagramu (Zdroj: Vlastní tvorba)

Diagram aktivit

Diagram aktivit umožňuje modelovat buď případ užití jako sled událostí anebo také procesy zachycené pomocí workflow jednotlivých tříd objektů. Každý proces je znázorněn jako složení kolekce uzlů, které jsou spojeny hranami. Workflow a proces je tedy aktivita, která je modelována pomocí diagramu aktivit. Sekvenci jednotlivých kroků v diagramu aktivit určuje řídicí tok. Interakcí objektů se rozumí, že objekt může provádět aktivitu nebo zašle zprávu jinému objektu [15], [19].

Na začátku tvorby diagramu aktivit je vhodné začít tzv. plaveckými dráhami (oddíly), které spočívají v zobrazení komunikace mezi participanty (aktéry). Do levého oddílu je zapisován aktér, který spouští například konkrétní případ užití. Dále do levého horního rohu se vloží počáteční uzel a následně lze vytvářet logickou posloupnost akcí, během které lze umístit rozhodovací uzly. Diagram může obsahovat vnořené aktivity, které je vhodné dále dekomponovat [12], [9].

Na obrázku č. 15 je uveden příklad diagramu aktivit, který obsahuje znázornění „plavečkových drah“, začátek v podobě počátečního uzlu, rozhodovací uzle a konec v podobě konečného uzlu.



Obrázek 15 - Příklad diagramu aktivit (Zdroj: Vlastní tvorba)

3.7 Web

Web lze chápat jako soubor více internetových stránek, které začínají stejným názvem domény. Každý web má svou vlastní strukturu (úvod, sekce, zápatí). Tato struktura je ve většině případech uložena v mapě webu [24].

Web je pojem, který se používá v označení *World Wide Web* (WWW). Jedná se o celosvětovou síť hypertextových dokumentů, které jsou vzájemně provázány pomocí

protokolu HTTP. Jednotkou webu je internetová stránka, která obsahuje texty, obrázky apod. Pro prohlížení internetových stránek webu slouží internetové prohlížeče [25], [26].

Pro potřeby diplomové práce je nutné popsat pojmy – internetový prohlížeč, webové aplikace, internetové stránky a e-shop. Návrh informačního systému v jazyce UML je navrhován pro teoretickou realizaci pomocí webové aplikace, která je zobrazovaná pomocí internetového prohlížeče.

3.7.1 Internetové prohlížeče a webové služby

Internetové prohlížeče

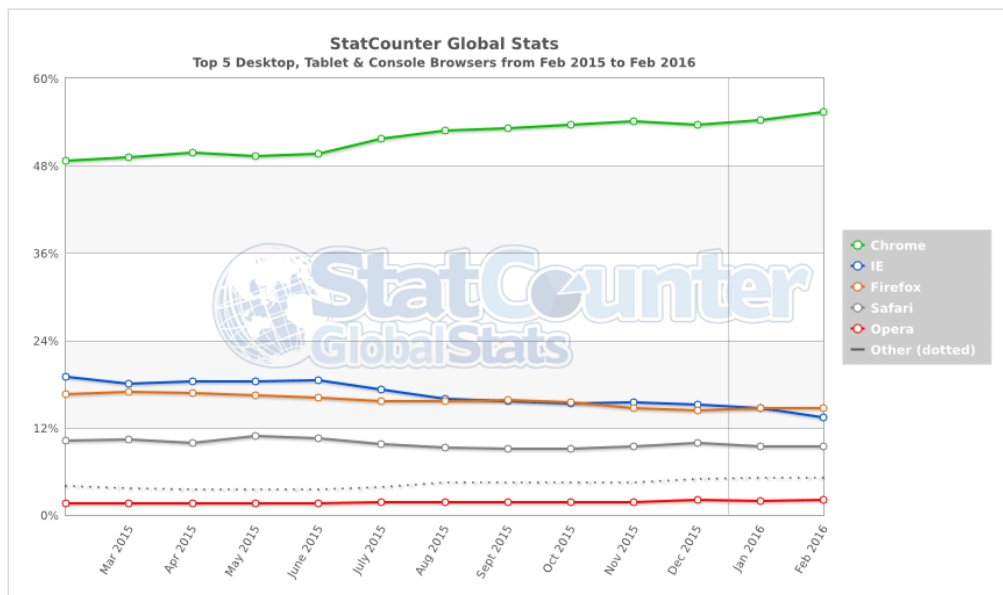
V současné době je nepřeberné množství různých internetových prohlížečů, které umožňují uživateli zobrazovat a využívat internetové stránky a internetové aplikace. Mezi několik nejpoužívanějších patří *Chrome*, *Internet Explorer*, *Firefox*, *Safari* a *Opera*.

Internetový prohlížeč představuje prostředníka mezi pracovní stanicí uživatele a webovými stránkami a aplikacemi, které jsou uloženy na serverech. Prohlížeč vytváří uživatelsky přívětivé prostředí, které pomocí určitých modulů umožňuje uživateli správné zobrazování textu, obrázků, videí, skriptů nebo aplikací spustitelných v samotném prohlížeči. Prohlížeč v určitém směru vytváří částečně unikátní operační systém, ve kterém se uživatel pohybuje a jeho pomocí může zobrazovat nebo upravovat internetový obsah.

Dle statistického šetření, které je zobrazeno na obr. č. 16 a které provádí společnost *StatCounter*, je prohlížeč *Chrome* od společnosti *Google* využíván více jak polovinou všech uživatelů. Dle úrovně podpory určitého prohlížeče se odvíjí i možnost doplňků, které lze ve webové aplikaci použít pro správné zobrazení [25]. Ve statistickém šetření jsou zahrnuty internetové prohlížeče, které jsou určeny pro desktopy, tablety a mobilní zařízení [24], [26], [27]. Z důvodu vysokého využívání internetového prohlížeče *Chrome* jsou pro představu popsány výhody a nevýhody prohlížeče.

Prohlížeč *Chrome* je oblíbený zejména díky jednoduchému grafickému designu. Jednoduchost spočívá v černém textu na bílém pozadí. *Chrome* odstranil zbytečné ovládací prvky a nechal vyniknout samotnou webovou stránku. Další důvodem je rychlost internetového prohlížeče. *Chrome* provádí časté aktualizace, které uživatel často ani nepozná [27], [26].

Nevýhodou internetového prohlížeče Chrome je zbytečné zabírání operační paměti, které u pomalejších zařízení způsobuje výrazné omezení výkonu pracovní stanice. Toto omezení ovlivňuje i výkonnější stroje při zapnutí většího počtu záložek v prohlížeči. Tato nevýhoda byla v nedávné době částečně redukována v jedné z aktualizací prohlížeče Chrome [27].



Obrázek 16 - Statistika prohlížečů za rok 2015-2016 společností StatCounter (Zdroj: [25])

Webová aplikace

Webová aplikace je aplikace, která je poskytována uživatelům pomocí sítě Internet z webového serveru. Webová aplikace tedy nebývá uložena na konkrétním zařízení uživatele, který jí využívá.

Oblíbenost webových aplikací spočívá v dostupnosti. K přístupu do webové aplikace uživatel potřebuje pouze internetový prohlížeč. Zařízení uživatele lze brát jako tzv. „tenkého klienta“, který nezná logiku webové aplikace [28], [26].

Programování webových aplikací probíhá nejčastěji pomocí tzv. „frameworků“. Framework umožňuje psaní webové aplikace na vyšších úrovních a tím se snižuje obtížnost samotného programování [26], [29].

Správa a aktualizování webové aplikace je jednodušší než u aplikací, které jsou nainstalovány přímo v zařízení, jelikož webové aplikace lze upravovat odkudkoliv. Webové aplikace jsou využívány pro informační systémy, emailové schránky, internetové obchody [29], [28].

E-shop

Internetový obchod (*neboli e-shop*) je určitý druh webové aplikace, která je primárně použita za účelem předání informací zákazníkovi a prodeje určitého druhu zboží. Vztahy, které e-shopy tvoří, jsou B2B (obchodník k obchodníkovi) a B2C (obchodník k zákazníkovi). Veškerá komunikace mezi jednou a druhou stranou probíhá pomocí internetové sítě [24], [30].

Internetové obchody začaly v devadesátých letech ve Spojených státech amerických. První společnosti, které začaly provozovat internetové obchody, byly Amazon a eBay. Obě tyto společnosti vznikly v roce 1994 [26], [30]. V ČR první e-shopy byly Vltava a ComputerPress. Rozvoj internetových obchodů nastal po roce 2000, jelikož infrastruktura internetové sítě byla již na dostatečné úrovni. V dnešní době lze přes internet koupit téměř vše (elektroniku, jídlo, zájezdy atp.) [26], [29].

Hlídací pes

Hlídací pes je webová služba, která slouží ke sledování jakéhokoliv obsahu na internetu. Tato služba vznikla z podnětu neustále měnícího se obsahu na internetu [31], [30].

Nejčastěji je hlídací pes využíván pro sledování ceny určitého produktu. Hlídací pes umožňuje zákazníkům e-shopu sledovat změny ceny či dostupnosti vybraného zboží. Pokud cena klesne pod zadanou hranici zákazníkem nebo se změní dostupnost produktu, odešle se e-mail nebo SMS zpráva zákazníkovi o proběhlé změně [26], [31], [30].

Některé e-shopy nabízejí svou vlastní službu hlídacího psa. Přidaná hodnota této služby není jen pro zákazníka, ale i pro provozovatele internetového obchodu. Hlídací pes umožňuje vlastníkově e-shopu přehled aktivních hlídacích psů zákazníků. Tento přehled obsahuje informace o tom, kolik zákazníků sleduje daný produkt nebo jaká cena daného produktu zákazníka zajímá. Na základě těchto informací provozovatel e-shopu může provést marketingové akce, ve kterých může rozhodnout, který produkt bude zlevněn [26], [28].

3.7.2 Pravidla pro tvorbu webu

Spolu s webem a webovými stránkami jsou často skloňovány pojmy *přístupnost* a *použitelnost*. V této kapitole autor krátce popisuje základní charakteristiky těchto dvou

pojmu. Dalšími pojmy, které úzce souvisí s pravidly pro tvorbu webu, responzivita a SEO. Tyto pojmy jsou také dále krátce popsány.

Přístupný web je takový web, který klade uživatelům minimální překážky. Dále se také zaměřuje na hendikepované uživatele. Ve své podstatě se jedná o šest základních skupin pravidel, které když tvůrce webu dodrží, zaručí přístupný web. Těchto šest skupin pravidel je dále popsáno v tab. č. 9, která je zpracována dle [24], [28], [26], [28].

Pravidlo	Popis
Obsah webových stránek je dostupný a čitelný	Každý netextový prvek, objekt nebo skript nesoucí významové sdělení musí mít svou textovou alternativu, která se objeví podržením klávesy „alt“. Barvy popředí a pozadí jsou dostatečně kontrastní. Na pozadí není vzorek, který snižuje čitelnost.
Práci s webovou stránkou řídí uživatel	Obsah webu se mění až po zásahu uživatele. Na stránce se neobnovuje žádný objekt častěji než jednou za sekundu. Nová okna webové stránky se otevírají jen v odůvodněných případech.
Informace jsou srozumitelné a přehledné	Webová stránka je dostatečně a výstižně popsána. Uživatel během chvilky ví, na jaké webové stránce se pohybuje. Velké bloky v textové podobě jsou děleny na menší bloky, a to kvůli zvýšení přehlednosti a čitelnosti.
Ovládání webu je jasné a pochopitelné	Webové stránky mají správný název a navigaci. V každé vnitřní stránce se nachází aktivní prvek pro možnost návratu na úvodní stránku.
Odkazy jsou zřetelné a návodné	Hypertextové odkazy jsou zvýrazněny, a to nikoliv pouze barevně. Každý odkaz má svůj vlastní popis, na jakou stránku odkazuje.
Kód je technicky způsobilý a strukturovaný	Jedná se o správné naprogramování internetových stránek, tedy bez chyb v kódu.

Tabulka 9 - Pravidla pro přístupný web (Zdroj: Vlastní tvorba)

Přístupnost webu lze otestovat pomocí online validátorů. Díky nim lze rychle zjistit případné nedostatky.

Použitelnost webu je skupina pravidel, která se zaměřují na zlepšení interakce uživatele s webovou stránkou. Snaží se o zvýšení srozumitelnosti a přehlednosti webu. Použitelností se rozumí snadné orientování na webové stránce [32], [26].

Použitelnost webu lze testovat různými metodami. Jednou z metod je například A/B testování, které spočívá ve vytvoření dvou rozdílných variant webových stránek s podobným obsahem. Varianta A je zobrazována jedné skupině uživatelů, a druhá varianta druhé skupině uživatelů. Obě skupiny mají za úkol splnění nějakého cíle. Na základě často časových rozdílů se může rozhodnout, která z variant je lepší (použitelnější) [32], [24].

V testech zaměřených na použitelnost se zkoumá pět základních vlastností, kterých je potřeba v případě použitelného webu dosáhnout. Tyto vlastnosti jsou uvedeny v tab. č. 10, která je zpracována na základě [24], [28], [26].

Pojem	Popis
Naučitelnost	Jak složité je pro uživatele zvládnout základní úkoly při prvním setkání s testovaným prostředím.
Efektivita	Jak rychle se uživatel seznámí s prostředím a může plnit zadané úkoly.
Zapamatovatelnost	Jak složité bude provádět efektivní práci v testovaném prostředí v případě, že uživatel s prostředím již delší dobu nepracoval.
Chybovost	Kolika chyb se uživatel dopustí, jaká je jejich závažnost a jak složitá je oprava provedených chyb.
Spokojenost	Jak je pro uživatele příjemné pracovat s webem jako s celkem.

Tabulka 10 - Základní vlastnosti použitelnosti webu (Zdroj: Vlastní tvorba)

Responzivita (*neboli responzivní web*) je pojem, který představuje stylování HTML dokumentu do takové podoby, aby výsledná webová stránka se zobrazovala správně na různých zařízeních (počítače, mobilní telefony, tablety). Jedná se o optimalizaci webu pro odlišné druhy zařízení [24], [26], [28].

Responzivitu lze zajistit především vlastností *Media Queries*, která je obsažena ve specifikaci CSS3. Díky této vlastnosti lze rozpoznat na jakém zařízení je webová stránka zobrazována a tím ji přizpůsobit [24], [28].

Search Engine Optimization (*neboli zkráceně SEO*) je soubor metod, které slouží ke zlepšení viditelnosti webu v internetových vyhledávačích. Díky těmto metodám se web zobrazuje ve výsledcích vyhledávačů lépe (vizuálně výše ve výsledcích vyhledávání) a častěji [31], [24], [28].

Výše zmíněná pravidla a vlastnosti pro tvorbu webu se vztahují také na e-shopy. Díky tomu majitel e-shopu může zvyšovat návštěvnost a tím i zisky. Pokud je e-shop i přístupný, lze skupinu potenciálních zákazníků rozšířit.

3.8 Srovnávače zboží

Srovnávače zboží (*neboli vyhledávače zboží*) jsou webové stránky, které se specializují na fulltextové vyhledávání zboží na internetu. Vyhledání lze specifikovat jako rozsáhlé hledání a porovnávání zboží z různých internetových obchodů. Srovnávání zboží je založeno na vytvoření databáze produktů, ve které jsou zaznamenány veškeré produkty a detailní informace o nich. Vyhledávané informace jsou vypisovány právě z této databáze produktů [30], [24], [33].

Srovnávače zboží obsahují další uživateli vyhledávané informace. Jedná se například o uživatelské recenze daného zboží nebo hodnocení internetových obchodů [24], [30].

Princip srovnávačů zboží spočívá v zadání obecného nebo přesného názvu zboží uživatelem. Poté srovnávač zboží zobrazí seznam několika internetových obchodů, které hledaný produkt nabízejí. Podle toho, jaký srovnávač zboží uživatel použije, se zobrazí různé specifikace zboží (např. název, cena, doprava, dostupnost atp.) [30], [24], [28].

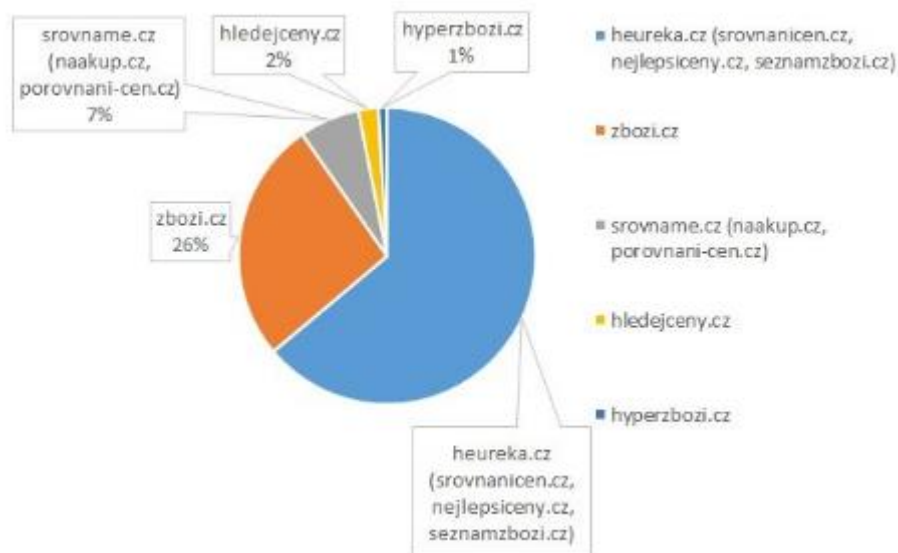
Princip srovnávání cen se začal používat koncem devadesátých let. V České republice v roce 2005 byl prvním srovnávačem Jyxo.cz. Od roku 2009 v ČR dominují dva největší srovnávače zboží - *Zboží.cz* a *Heureka.cz*. *Zboží.cz* se zaměřuje více na vyhledávání zboží, než *Heureka.cz*, která se také výrazně zabývá doplňkovými službami. *Heureka.cz* bývá také označována jako nákupní rádce [30], [28].

Když se po zadání textového požadavku pro vyhledávání produktu uživateli zobrazí řazený výpis dle daného srovnávače, tak provozovatel srovnávače má možnost ovlivnit výsledky vyhledávání. Konkrétně Zboží.cz nabízí různé stupně spolupráce s vlastníky e-shopů. Tito vlastníci mají možnost zaplatit si služby konkrétního srovnávače a tím částečně ovlivnit pozici e-shopu ve výsledku vyhledávání [24], [30].

Průměrná denní návštěvnost Heureka.cz je 250 tisíc návštěvníků. Návštěvnost u Zboží.cz tvoří 130–150 tisíc uživatelů denně. České srovnávače zboží obsahují databáze čítající deseti tisíce e-shopů. Konkrétně Zboží.cz uvádí více jak 16000 e-shopů [30].

Při agregování srovnávačů se jako nejsilnější skupinou jeví spojení heureka.cz, srovnanicen.cz, nejlepsiceny.cz a seznamzbozi.cz. Tyto srovnávače využívají totožný obsah díky propojení přihlašování a databází XML feedu.

Na níže uvedeném obr. č. 17 je znázorněn graf, který uvádí procentuální podíl uživatelů jednotlivých srovnávačů zboží, které využívají shodnou databázi XML feedu. Skupina vyhledávačů agregovaná heurékou spolu se zboží.cz tvoří 90 % návštěvnosti všech srovnávačů. Z tohoto důvodu je v diplomové práci směřovat napojení na právě tyto dvě databáze.



Obrázek 17 - Přehled uživatelů srovnávačů zboží v ČR (Zdroj: netmonitor.cz)

3.8.1 Přidání e-shopu do srovnávače

Přidání e-shopu do srovnávače zboží se liší podle toho, do kterého ze srovnávačů se e-shop chce zařadit. V následující kapitole je popsán obecný postup u vyhledávače zboží Heureka.cz [24], [28].

První krok tvoří bezplatná registrace, při které si nový zákazník vyhledávače vytvoří svůj účet. Následně si do svého účtu přidá e-shopy, které chce zákazník do vyhledávače přidat.

Dalším krokem je vyplnění několika důležitých informací [24], [30]:

- *Informace o obchodu* – název obchodu, webová adresa, adresa pro XML soubor, a kontaktní e-mail, adresa sídla firmy.
- *Kontaktní údaje* – údaje fyzické osoby, se kterou bude komunikovat zástupce Heureka.cz.
- *Fakturační údaje* – slouží k vystavení daňového dokladu.
- *Cena dopravy* – způsob a cena dopravy, kterou e-shop nabízí [24].

Poté dochází k odeslání správně vytvořeného XML feedu, který je krátce vysvětlen níže. Po odeslání se provede spárování s nově odeslaným XML feedem a databází vyhledávače. Shodné položky se spárují v jeden nabízený produkt, který je při vyhledávání porovnáván se všemi spárovanými produkty. Primárně toto párování probíhá na základě názvu a označení produktu [30], [34].

XML feed

XML feed je datový soubor, který obsahuje informace o jednotlivých položkách zboží v daném e-shopu. Soubor XML feed lze získat vygenerováním přímo z databáze e-shopu. Každý e-shop, který je zařazený do seznamu pro vyhledávání, musí poskytnout databázi produktů ve formátu XML, tedy jako XML feed [30], [26].

E-shopy mají povinnost aktualizovat svůj XML feed. To lze provádět periodicky v řádu dnů nebo hodin. Aktualizovat XML feed je potřeba při výrazné změně produktů, například dostupnost, změna ceny nebo ukončení prodeje [34], [33].

V následující tabulce č. 11 jsou uvedeny základní položky XML feedu, které jsou potřeba k porovnávání zboží, konkrétně k vyhledávači Heureka.cz. Tabulka je vypracována na základě [33], [26], [30].

Název tagu	Popis
<PRODUCTNAME>	Obsahuje přesný název produktu. Do názvu produktu patří také výrobce, značka nebo specifické číslo či kód. Pomocí <i>productname</i> dochází ke správnému vyhledání zboží.
<PRODUCT>	Obsahuje dodatečné informace k produktu, které převážně slouží k lepšímu vyhledání produktu. <i>Product</i> musí obsahovat přesné znění z tagu <i>productname</i> . Pokud tomu tak není, zboží může mít problémy se spárováním.
<ITEM_ID>	Slouží k jednoznačnému určení produktu. Jedná se o identifikační číslo produktu, které je jedinečné právě v jenom e-shopu.
<CATEGORYTEXT>	Popisuje hierarchické zařazení produktu do produktové nabídky. Zároveň se jedná o produktovou nabídku v celé Heureka.cz. Hierarchický strom by měl být shodný se stromem Heureka.cz. Tato shodnost napomáhá k rychlejšímu řazení do katalogu.
<DELIVERY>	Sděluje souhrn nabízených možností dopravy z e-shopu ke koncovému zákazníkovi. V případě nevyplnění je nutné sdělit informace o přepravě přímo vyhledávači zboží (Heureka.cz).
<DELIVERY_DATE>	Udává časové hledisko celé objednávky. Informuje o stavech zboží a jejich dostupnosti.
<URL>	Je určen k uvedení adresy odkazující na e-shop. Heureka.cz po zobrazení vyhledávaného zboží umožňuje přesun zákazníka přímo na kontaktní e-shop pomocí této adresy.
<IMGRUL>	Obsahuje odkaz na úvodní obrázek daného produktu. Obrázek, na který je odkazováno, by neměl obsahovat vodoznak.

Tabulka 11 - Položky v XML feed (Zdroj: Vlastní tvorba)

3.8.2 Cenové hledisko

Jelikož srovnávače zboží přináší internetovým obchodům zvýšení návštěvnosti a zvýšení výnosů, je pro ně výhodné investovat do zařazení e-shopů ve vyhledávacích zboží.

Nastavení plateb e-shopů vůči vyhledávačům probíhá na základě několika platebních modelů. Jedním z modelů je PPC (*neboli pay per click*). Jedná se o placení malé částky za kliknutí na odkaz do e-shopu v daném vyhledávači zboží.

Dalším modelem je například paušální platba (*neboli ROAS – Return of Advertising Speding*). ROAS udává peněžní hodnotu, která uvádí kolik korun e-shop získá zainvestováním jedné koruny do srovnávače zboží.

Zapojení e-shopu do srovnávače může být i zdarma, avšak e-shop nemá žádné výhody oproti konkurenci, která srovnávači odvádí platbu. Na stránkách vyhledávačů je vždy možné zobrazit aktuální ceny požadované vyhledávačem [30]. Na obrázku č. 18 jsou znázorněny služby a cenová nabídka několika nejvýznamnějších srovnávačů, které jsou dostupné v ČR.

Název	Služby zdarma	Placené služby	Platba
zbozi.cz	aktualizace XML feedu 3× týdně	vyšší pozice, každodenní aktualizace, přístup do statistik	1 Kč za proklik
hyperzbozi.cz	žádné	veškeré	od 1 Kč za proklik
heureka.cz	žádné	veškeré	od 1 Kč za proklik
monitor.cz	žádné	veškeré	do 1 Kč za proklik
srovname.cz	žádné	veškeré	1 Kč za proklik
koupis.cz	žádné	veškeré 1000,5000,200-00 položek v xml feedu	999 Kč nebo 1499 Kč nebo 1999 Kč za rok
www.cenovyradce.cz	žádné	veškeré	od 0,40 Kč za proklik nebo 590 Kč za rok

Obrázek 18 - Přehled srovnávačů s cenovou nabídkou v ČR (Zdroj: netmonitor.cz)

4 Praktická část

Praktická část se zabývá řešením vlastního návrhu informačního systému, který je proveden pomocí UML modelů. Informační systém je navržen pro specifickou skupinu uživatelů. Tuto skupinu činí převážně osoby samostatně výdělečně činné (OSVČ) a menší podniky do přibližně deseti zaměstnanců. Nově navržený systém nabízí uživatelům úsporu peněz při nákupu materiálu nebo zboží, úsporu času při podnikové administrativě a zpřehlednění vstupních a výstupních transakcí.

Nejprve je tedy potřeba provést analýzu současného stavu. Jedná se o analýzu hlavních komponent, které nově navržený systém propojuje a využívá potřebné části z již existujících webových a podnikových služeb.

Následně jsou shromážděny požadavky a problémy, které musí být řešeny. Po shromáždění požadavků je přistoupeno k vytvoření jednotlivých modelů tvořících návrh vlastního systému. Pro rozšíření pohledu na navrhovaný systém jsou vytvořeny wireframe, které popisují systém z uživatelského pohledu. Celý návrh je podroben kvalitativnímu šetření, které bylo provedeno na vybrané skupině respondentů. Výsledky kvalitativního šetření jsou popsány a znázorněny na upraveném wireframu. Praktická část je zakončena vytvořením webové aplikace, která primárně slouží k vizualizaci a demonstraci nově navrženého systému a umožňuje nastínit proces přihlášení do systému a průchod několika vybranými podsystemy.

4.1 Popis současného stavu

Tato kapitola se zabývá určením důležitých funkcí pro navrhovaný systém a následným popisem současného stavu těchto funkcí.

4.1.1 Výběr funkcí

Nově navržený systém má za úkol ulehčení nákupu a administrativy pro OSVČ. Toto ulehčení spočívá ve vytvoření *virtuálního skladu*, který bude evidovat zásoby uživatele. Zásoby se budou automaticky odečítat při vystavení faktury a zároveň je možné nastavit minimální hranici určitého zboží.

Pokud počet určitého druh zboží klesne pod nastavenou hranici, umožňuje systém několik možností následného řešení:

1. **upozornění** uživatele na snížení množství zboží ve skladu;
2. **automatické připravení objednávky** přednastaveného množství zboží, které zajistí, že uživatel při potřebě vykonání určité práce nebude mít nedostatek zboží nebo materiálů. Při poklesu množství zboží je uživatel vyzván k následnému nákupu zboží z doporučeného e-shopu. Pokud by po čase internetové obchody umožnily nákup zboží pomocí automatického objednání, poté by automatické objednání mohlo být provedeno i bez nutnosti zásahu uživatele do konkrétní objednávky.
3. **prohledání XML feedu** na vyhledávacích a nalezení cenově optimálního dodavatele potřebného zboží, nebo jeho alternativu. Tato funkce informuje uživatele o jeho možnostech a samotný nákup přenechá na uživateli.

Systém dále umožňuje evidenci přijatých a vydaných faktur. Dle nich je schopný zvyšovat, nebo snižovat stav skladu a případně přidávat nové položky. Systém umožňuje vytváření faktur pro zákazníky a případné načítání již existujících faktur do systému v elektronické podobě. Díky tomuto propojení lze dosáhnout jednodušší správy účetnictví. Tato systémová databáze faktur slouží jako kvalitní podklady pro účetní uzávěrku.

Systém umožňuje nastavení hlídacího psa na určité zboží. Pokud vybrané zboží bude k dispozici za přednastavenou cenu, bude uživatel o tomto stavu informován.

Informování uživatele je nastavitelné. Nabízené možnosti jsou pomocí mobilního zařízení, emailu nebo upozorněním pomocí aplikace systému v internetovém prohlížeči nebo v mobilním zařízení. Pro účet uživatele je umožněno nastavení přístupových práv. Přihlášení do systému probíhá následovně:

1. zadání názvu uživatelského účtu;
2. zadání korektního hesla a odeslání přihlašovacích údajů;
3. výběr existujícího systému / vytvoření nového systému / požádání o přizvání k již existujícímu systému;
4. dle nastavených práv je umožněn vstup do vybraného systému.

4.1.2 Analýza současné situace

V této kapitole autor diplomové práce popisuje současný stav nejdůležitějších komponent potřebných k realizaci nově navrhovaného systému. Jsou zde zhodnoceny výhody i nedostatky daných komponent, které jsou v současné době nabízeny uživatelům spolu s dalšími službami. Mezi přední společnosti a produkty, které autor popisuje, patří modul softwaru Helios Red, internetový vyhledávač Heureka.cz a internetový e-shop Alza.cz.

Virtuální sklad

V současné době existuje mnoho variant virtuálních skladů, které jsou převážně reprezentovány v podnikových informačních systémech. Tyto systémy jsou převážně specializované na skladové funkce a převážně vytvářejí vnitropodnikovou databázi. Velké podnikové informační systémy jsou drahé a neumožňují autorem popisovanou a vyžadovanou interakci s e-shopy a internetovými službami. Příkladem skladového systému může být specializovaný modul systému Helios Red, který nese název Skladová evidence zásob.

Modul například umožňuje vytvářet skladové karty, které jsou vhodné pro identifikaci místa uložení konkrétního zboží ve skladu. Modul dále nabízí obratovou soupisku zásob, konfiguraci úloh, příjem nebo zisky. Skladová karta v tomto modulu umožňuje zadání velkého množství informací, které lze o určitém zboží evidovat. Zboží musí být vždy zadáno do systému a musí být vytvořena nová skladová karta. Systém

neumí na základě načtení digitální verze faktury sám vyplnit skladovou kartu především z důvodu nedostatečných informací ve faktuře. V případě vytváření nového záznamu pro naše účely stačí mnohem jednodušší druh skladové evidence, která uchovává převážně informace zjistitelné z faktury.

Mezi informace uvedené na faktuře patří informace o:

1. **dodavatel:** název firmy, ulice a číslo domu, PSČ, město, IČ, DIČ, telefon, fax, banka, číslo účtu, IBAN, variabilní symbol, kontaktní symbol, dodací list číslo, číslo objednávky
2. **odběrateli:** název firmy, ulice, PSČ, město.
3. **provedené transakci:** způsob odběru, způsob platby, výše zálohy, datum splatnosti, datum vystavení, datum zdanitelného plnění.
4. **zboží:** název, označení, množství, cena bez DPH, sazba DPH, DPH celkem, celkem bez DPH a celkem k úhradě s DPH.

Výše zmíněné informace musejí být převeditelné do záznamu o provedených transakcích a z informací o samotném zboží musí být vytvořeny skladové položky. Ve skladové evidenci navrhovaného systému však musí být uvedeno více informací (atributů) u jednotlivých položek, které se starají například o propojení s e-shopy nebo o specifikaci konkrétní položky. Mezi tyto informace patří: datum, url adresa položky (patřící e-shopu), url adresa e-shopu, označení produktu v dokumentu XML feed, které pomůže k následnému vyhledávání položky.

Na následném obrázku je vidět náhled skladové karty ze softwaru Helios Red, která výrazně převyšuje požadavky na modul nově navrhovaného systému.

Celý skladový systém softwaru Helios Red v mnoha směrech převyšuje požadavky na nově navrhovaný systém. Jedná se především o velikost podniku, na který je sklad vytvářen. Nově navrhovaný systém je tvořen převážně pro jednotlivce nebo pro několik jedinců. Z toho důvodu jsou velké možnosti nabízené skladovou evidencí systému Helios Red spíše nežádoucí z hlediska zbytečného navýšení složitosti celého systému.

Obrázek 19 - Skladová karta ze SW Helios Red (Zdroj: Vlastní zdroj)

Srovnávání zboží

Pro porovnání určitého druhu zboží slouží v současné době srovnávače zboží. Srovnávače fungují pomocí vytvoření databáze XML feedu. Databáze je tvořena ze záznamů z každého přihlášeného e-shopu, který XML feed poskytl srovnávači zboží. Největší podíl srovnávačů zboží tvoří na českých e-shopech **Heureka.cz** a **Zboží.cz**.

Oba srovnávače umožňují komparaci zboží dle ceny, hodnocení pomocí recenzí e-shopu a případné dostupnosti zboží. Umožňují seřazení položek dle ceny bez započítání dopravy i s ohledem na cenu dopravy od prodejce. Tyto možnosti jsou důležité a vhodné pro navrhovaný systém.

Pro každé porovnání zboží je však nutné opětovně zapnout internetový prohlížeč a vyhledat zboží znovu. Pro navrhovaný systém je třeba automatického porovnání

přednastaveného zboží, jehož množství kleslo pod přednastavenou minimální hranici zásob. Srovnávače zboží pro navrhovaný systém by byli velkým přínosem v případě, že by byl poskytnut přístup k databáze XML feedu, nebo by bylo možné zboží vyhledávat periodicky pomocí skriptu.

Alza.cz

Jedná se o největší český internetový e-shop zaměřující se na prodej převážně elektroniky. Alza.cz v současné době nabízí i mnoho dalších produktů, které byly postupem času přidávány, a které nesouvisí s elektronikou. Příkladem mohou být hračky, zahradní nářadí nebo software. Alza.cz je zde uvedena hlavně z důvodu kvalitního využívání tzv. hlídacího psa.



Obrázek 20 - Nastavení hlídacího psa na Alza.cz (Zdroj: www.alza.cz)

Hlídací pes na Alza.cz je automatický robot, které umožňuje označení určitého zboží a následné hlídání změny stavu označeného zboží. Změnou stavu je myšlena změna dostupnosti a změna ceny. Tento hlídací pes umožňuje nastavit cenovou hranici. Pokud cena klesne pod nastavenou hranici je uživatel informován emailem.

Možnost hlídání dostupnosti skladem je však nedostatečná pro nově navrhovaný systém. Hlídání dostupnosti neumožňuje nastavit minimální počet kusů na skladě. Další nevýhodou tohoto hlídání je, že hlídání probíhá pouze pro e-shop Alza.cz, nikoliv i pro ostatní e-shopy s obdobným zbožím. Tento fakt je pochopitelný i z hlediska marketingu společnosti Alza.cz.

4.2 Specifikace požadavků

Požadavky kladené na systém z pohledu uživatelů jsou velice důležitou složkou pro tvorbu systému. Tyto požadavky nejsou součástí jazyka UML. Požadavky upřesňují konkrétní představy budoucích uživatelů o tom, jak bude systém fungovat. Systém umožňuje přístup více uživatelům (účtům) k jednomu systému. Jeden konkrétní účet může mít přístup také k většímu počtu skladových systémů. Každý účet má přednastavenou úroveň práv k danému systému.

Uživatel, který založil svým účtem systém se stává automaticky správcem systému a tím jeho účet získává plná práva na využití všech nabízených funkcí systému. Tento správce, pokud potřebuje umožnit i ostatním uživatelům přihlášení k jeho systému, vytvoří nebo přidá další uživatelský účet vytvořením jména, hesla a nastavením práv. Existujícímu účtu nastaví pouze práva a uloží je pod čísla od 1 do 5. Po přihlášení k účtu je třeba zvolit konkrétní systém. K přihlášení k účtu slouží přihlašovací jméno uživatele, heslo a potvrzení přihlášení stiskem potvrzovacího ovládacího prvku. Následuje přihlášení k existujícímu systému, vytvoření systému nového nebo požádání o udělení práv k již existujícímu systému, který spravuje jiný uživatel.

Z tohoto důvodu lze popsat požadavky pouze jednoho uživatele vlastního účtu s plným oprávněním.

4.2.1 Funkční požadavky uživatele

Identifikované funkční požadavky uživatele jsou zapsané v následné tabulce. Jedná se o popis jednotlivých požadavků, které jsou následně dle jedinečných identifikátorů popsány. Všechny zmíněné požadavky jsou popisovány pro uživatele, který vlastní veškerá oprávnění k systému.

Identifikátor	Popis požadavku
U01	Uživatel se může přihlásit a odhlásit k uživatelskému účtu.
U02	Uživatel si vybírá (vytváří / žádá o přidání) systém do kterého se přihlásí.
U03	Uživatel spravuje uživatelské role a nastavuje oprávnění.
U04	Uživatel vytváří a spravuje modul Hlídací pes.
U05	Uživatel spravuje a přidává faktury.
U06	Uživatel spravuje a přidává evidenci transakcí.
U07	Uživatel spravuje položky ve skladu.
U08	Uživatel spravuje cenu zboží k prodeji ve skladu.

Tabulka 12 - Funkční požadavky aktéra Uživatel (Zdroj: Vlastní tvorba)

U01, U02: Přihlášení odhlášení a výběr systému

Uživatel se přihlašuje do systému zadáním svého přihlašovacího jména, správného hesla a potvrzením přihlášení stiskem potvrzovacího ovládacího prvku.

Položky jméno a heslo jsou k sobě vázány. Po přihlášení k účtu jsou uživateli zobrazeny tři základní možnosti, jak se přihlásit k systému. První možnost je vybrání požadovaného systému z nabídky již přiřazených systémů k aktuálně přihlášenému účtu. Druhou možností je založení úplně nového systému, ve kterém se uživatel stane automaticky administrátorem. Poslední možnost je vyhledání již existujících systémů a zaslání žádosti o přidání aktuálně přihlášeného účtu k vybranému systému. Pokud administrátor zvoleného systému potvrdí žádost, bude popisovaný uživatel informován o udělení práv a následně nalezne nový systém ve svém seznamu systémů.

Při úspěšném přihlášení do vybraného systému je uživateli umožněno odhlášení ze systému nebo přímo odhlášení z přihlášeného účtu bez ohledu na práva, které má uživatel přiřazená v aktuálně zobrazeném systému.

U3: Správa uživatelských rolí a nastavení oprávnění

Pokud je uživatel přihlášen bez omezení oprávnění, tedy je přihlášen administrátorským účtem, může vykonávat následující operace s uživateli:

1. Uživatel může umožnit přístup existujícím účtům ke svému systému. Při přidání účtu vybere i úroveň oprávnění mezi úrovněmi 1 až 5.
2. Vytvoření úrovní oprávnění je čistě v kompetenci uživatele. U každé skupiny práv (1 až 5) mohou být nastavena jiné oprávnění. Mezi oprávnění patří například: možnost zobrazit aktuální sklad, možnost přidávat položky do skladu, možnost odebírat položky ze skladu, možnost zobrazení časového pohybu položek ve skladu, možnost změny nastavení u položek ve skladu, možnost změny ceny prodeje u položek ve skladu a další podrobnější oprávnění.
3. Uživatel může k jednotlivým uživatelským účtům přidávat potřebné atributy, které slouží k upřesnění uživatelských účtů nebo například k možnostem nastavení upozornění pro nové účty (pomocí tel. čísla nebo emailové adresy).

U4: Správa modulu Hlídací pes

V modulu Hlídací pes má uživatel k dispozici textová pole a ovládací prvky sloužící k ovládání modulu. Textové pole slouží k zadání názvu zboží, interního označení položky, maximální přípustné ceny položky, požadovaného množství položky a url adresy. Url adresa odkazuje na konkrétní produkt, který je v ideálním případě správně přiřazen k XML feedu jednoho ze dvou vybraných srovnávačů. Zadaný název se převede do skladu poté, co proběhne koupě hlídaného produktu. Interní označení slouží pro přesnou identifikaci jednoho kusu zboží nebo jedné konkrétní objednávky. Toto označení slouží jako jednoznačný identifikátor, proto nesmí být více různých položek označeno stejným identifikátorem. Pokud se vybrané zboží cenou za kus zastaví přesně nebo pod maximální přípustnou cenou, systém uživatele informuje o dosažení této hranice. Zadání požadovaného počet kusů slouží obdobně jako maximální přípustná cena. Pokud se zboží dostane na požadované množství, je uživatel upozorněn přednastaveným způsobem.

V případě, že uživatel požaduje určité zboží na url adrese, avšak nevyžaduje hlídání pouze tohoto e-shopu, je k dispozici výběru možnosti aktivního srovnávače zboží. V případě aktivního srovnávače systém nalezne nejnižší cenu zboží a upozorní uživatele v situaci, kdy je cena shodná nebo nižší než přednastavená hranice. Aktivní srovnávač plní

svou funkci pouze v případě, že je položka vložená pomocí url adresy korektně přiřazena do již zmíněných databází XML feedu.

Časová perioda prohledávání e-shopů je závislá na vytížení serverů srovnávačů zboží a na případné realizaci. Uživatel má možnost nastavit, jaká cena produktu má být hlídána. Může být hlídána pouze koncová cena pro zákazníka nebo cena se započítanou cenou za dopravu, případně může být brán zřetel na množstevní slevu nebo dopravu zdarma. Tyto specifické požadavky si nastavuje uživatel sám.

U05: Správa a přidání faktury

Uživatel může vytvořit přehled příchozích a odchozích faktur. Díky tomuto přehledu může uživatel snadněji vytvořit účetní uzávěrku za určité období. Správa faktur umožňuje přidání nové faktury do databáze faktur v souboru xml nebo html. Díky použití formátu značkovacího jazyka se faktura načte do databáze a modul umožňuje následné přidání položek z faktury do skladu, nebo naopak odebrání položek ze skladu při vystavení odchozí faktury. Vkládaná faktura nemusí být pouze z internetového obchodu, ale i z kamenné prodejny, nicméně musí být převedena do zmíněného formátu.

Faktury je možné pomocí tohoto modulu vytvářet. Tvorba umožňuje přidání několika šablon, do kterých se mohou přidávat položky z aktuálního skladu. Uživatel může pro vyplňování odběratelů a dodavatelů využít možnosti automatického vyplnění v případě, že mezi zainteresovanými stranami již proběhla transakce, která je uložena v systému.

Uživatel při načítání existující faktury do systému má možnost překontrolování správnosti zápisu informací před jejich samotným uložením do databáze nebo před přidáním položek do skladu.

U06: Správa evidencí a transakcí

Tato správa je pro uživatele rozšířením možností práce s fakturami i mimo ně. Pomocí správy evidencí je uživateli umožněno změnit nastavení budoucího nákupu na jiné e-shopy, nebo například připravovat nové zakázky a tím zjišťovat dostatečný stav skladu a zásob.

U07: Správa položek ve skladu

Správa skladu umožňuje uživateli nastavovat důležité parametry skladových zásob. Jedná se především o vytváření nových položek zboží, úpravu již vytvořených nebo načtených položek z uložených faktur.

Správa skladu umožňuje zadání nebo změnu všech atributů, které jsou zobrazovány u jednotlivých položek. Typickým příkladem je zadání nebo úprava unikátního označení zboží jako jednoznačného identifikátoru.

V modulu skladu je uživateli umožněno nastavení minimální zásoby daného zboží ve skladu. V případě nastavení této hranice je potřeba mít povolený alespoň jeden způsob upozornění, který se nechá dále přesněji specifikovat v nastavení upozornění. U každé položky je uživateli umožněno vypnout nebo zapnout vyhledávání nejvýhodnějšího zboží pomocí srovnávače zboží. Samotné srovnávání probíhá až po poklesu určitého zboží pod rezervní množství. Samotné objednání zboží provádí uživatel sám. Dále je uživateli umožněno nastavení priorit vyhledávání při stisku nastavení priorit porovnávače. V tomto nastavení je možné vybrat jako prioritní atribut vyhledávání cenu, cenu s dopravou nebo lze nastavit brání v úvahu procentuální slevy při objednávce většího počtu zboží. Dále je umožněno nastavení minimálního množství zboží potřebného k objednání.

Správa skladu v neposlední řadě umožňuje uživateli zjištění celkové hodnoty zboží na skladě.

U08: Správa cen zboží k prodeji

Jedná se o modul obdobný nabízenými možnostmi jako správa skladu. V tomto modulu je uživateli umožněno hlavně nastavovat cenovou marži na jednotlivé zboží. Ceny zboží v odchozích fakturách jsou definovány právě pomocí atributů z tohoto modulu.

4.2.2 Nefunkční požadavky

Nefunkční požadavky jsou požadavky kladené na systém. Tyto požadavky jsou definovány dle shodných pravidel jako funkční požadavky. Požadavky jsou shrnuty v tabulce nefunkčních požadavků, které jsou dle jednoznačných identifikátorů dále upřesněny.

Identifikátor	Popis požadavku
S01	Systém musí dosahovat dostupnosti minimálně 99,9 %.
S02	Systém musí být dostupný z aktuálních webových prohlížečů.
S03	Systém musí mít ochranu proti útoku DoS a DDoS.
S04	Systém musí obsloužit desetitisíce účtů a systémů.
S05	Systém musí pojmout desetitisíce relací (tabulek).
S06	Systém nesmí přesáhnout chybovost 0,0001 % požadavků.
S07	Systém musí být stabilní s nízkou latencí.
S08	Systém má záložní server s vysokokapacitním uložištěm.
S09	Systém musí mít responzivní design pro různá zařízení
S10	Systém by měl mít přístup k databázím XML feed.

Tabulka 13 - Nefunkční požadavky na systém (Zdroj: Vlastní tvorba)

S01: Dostupnost minimálně 99,9 %

Stejně jako většina serverů v dnešní době musí dosahovat určité procentuální dostupnosti s ohledem na jejich nutné využívání, i tento systém musí být online 24 hodiny denně. Dostupnost nad 99,9 % je standardní úroveň dostupnosti nabízená v současné době poskytovateli serverů. Jedná se o vyhrazení určitého času během roku na možné aktualizování softwaru, hardwaru nebo datové infrastruktury. Při takovéto úrovni dostupnosti je maximální doba nefunkčnosti systému přibližně 8,766 hodin za celý rok.

Obvykle tato doba bývá rozdělena na menší úseky právě pro již zmíněný servis. Při přesažení této doby mají zákazníci v případě placených serverů nárok na určitou finanční náhradu popsanou ve smluvních podmínkách.

S02: Dostupnost z aktuálních webových prohlížečů

Systém musí být správně zobrazen na všech používaných internetových prohlížečích. Tím je myšleno správné zobrazení i funkčnost na všech aktuálních verzích desktopových

i mobilních prohlížečů. Systém musí nabízet stejnou funkcionalitu pro uživatele všech rozšířených operačních systémů.

S03: Ochrana proti DDos

Systém musí být zabezpečen současnými technologiemi proti napadnutí útoky Dos a DDos. Systém musí mít vyřešené ochrany proti přesycení požadavky na server. Přesycením je například hrozba velkého počtu požadavků, díky kterým se zaplní například datové toky, čímž se znemožní plnění požadavků. Ohrožená místa jsou: síťová rozhraní, aplikace IS nebo samotné HW komponenty serveru.

S04, S05, S08: Velké databázové uložení

Systém musí mít k dispozici velké množství úložného prostoru pro obsluhu desítek tisíců potenciálních uživatelů, systému a relací, které systémy potřebují pro úschovu dat. Tyto servery a uložení musí být zálohovány v případě napadení systému fyzickou nebo softwarovou formou.

S06, S07: Nízká chybovost požadavků, latence a vysoká stabilita

Systém musí být naprogramován stabilně s přesně stanovenými možnostmi a řádným testováním před finálním zveřejněním. Systém musí být definován po všech možných vstupech od uživatelů, kteří systém budou využívat. Mezi tyto jasně definované kroky uživatele nesmí chybět ani kontrola nahrávaných souborů (faktur atp.) a kontrola zaplnění a vytížení serverů a uložení. Při zjištění rychlejšího zaplňování uložení či zvyšování latence je nutné provést patřičné opatření například rozšířením velikosti nebo změnou infrastruktury datových toků. Stabilita systému závisí na intenzivním sledování chování systému v průběhu fungování a případných optimalizací v beta verzi systému.

S09: Responzivita systému

Systém musí být psán tak, aby každý jeho modul byl responzivní a tím se dal bez omezení využívat na mobilních zařízeních, nebo na zařízeních dosahujících nízkého počtu rozlišovacích bodů. Samozřejmostí je, že u zařízení s nízkou zobrazovací plochou a nízkým rozlišením bude nutné tabulky posouvat a nebude umožněno například zobrazení všech atributů najednou.

S10: Přístup k databázím obsahující XML feed

System musí být schopen vyhledávat zboží v různých e-shopech. Toho lze dosáhnout přístupem k databázím XML feedu, který lze dosáhnout smlouvou s vybranými srovnávači, nebo naprogramováním automatických robotů, které by pro porovnávání využívaly servery srovnávačů přímo.

Způsob, jakým by tento požadavek byl splněn, je spíše na samotných programátorech, kteří by navrhovaný systém realizovali.

4.3 Návrh řešení

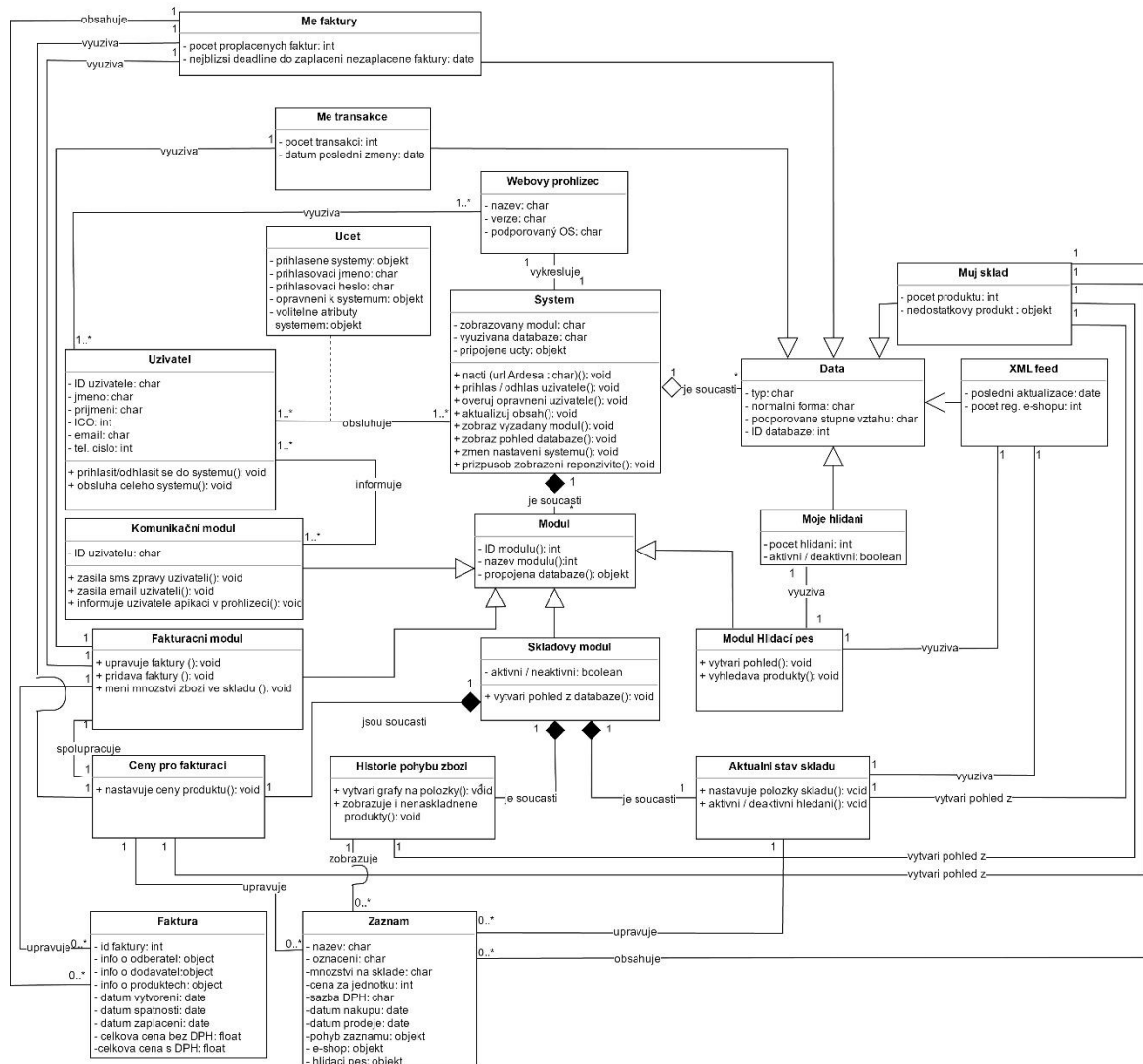
V této kapitole autor popisuje kompletní návrh systému za použití modelů jazyka UML. Nejprve autor uvádí popis statické struktury navrhovaného systému pomocí modelu tříd, který obsahuje diagram tříd doplněný o datový slovník. Dále je pomocí stavového diagramu znázorněno chování třídy objektů *Účet*. Na tyto dva modely navazuje model interakcí, a to konkrétně diagramem případů užití, sekvenčními modely a diagramem aktivit.

4.3.1 Model tříd

Jedním z prvních kroků při popisu struktury nově navrženého systému je vytvoření diagramu tříd. Diagram tříd je důležitý pro jasnou identifikaci tříd objektů a vazeb mezi nimi. K diagramu tříd je nutné vytvořit datový slovník. Datový slovník obsahuje popis všech zobrazených tříd a vazeb mezi nimi.

Tvorba diagramu tříd probíhá definováním si tříd objektů, které jsou nutné pro návrh systému. Tyto třídy a jejich vazby jsou zaneseny do modelu tříd. K modelu tříd se následně vytvoří datový slovník. Za pomoci datového slovníku a diagramu tříd je posléze dokončován model tříd.

Vytvořený diagram tříd obsahuje pouze nezbytně nutné třídy pro dostatečnou přehlednost o systému. Navrhovaný systém je složitější a vytvoření diagramu tříd, který by detailně popisoval každou jeho část, by byl nad požadovaný rámec abstrakce.



Obrázek 21 - Diagram tříd navrhovaného systému (Zdroj: Vlastní tvorba)

Datový slovník

Uživatel znázorňuje osobu, která má vytvořen vlastní účet v navrhovaném systému. Uživatel je třída, která se přihlašuje do systému, odhlašuje se ze systému a celý systém obsluhuje. Atributy třídy *Uživatel* jsou především osobní údaje. Mezi tyto atributy patří ID uživatele, jméno, příjmení, IČO (pokud se jedná o OSVČ), email a telefonní číslo (pro potřeby systémového upozornění uživatele). Uživatel využívá *Webový prohlížeč* k ovládání systému.

System je celek, který spravuje a sdružuje všechny ostatní moduly a databáze. Jedná se o třídu, jejíž atributy popisují přiřazené účty, právě zobrazovaný modul a současně využívanou databázi pro správný databázový pohled. Třída *System* znázorňuje samotné jádro systému, ke kterému se ostatní třídy (moduly a databáze) připojují. Třída *System* však obsahuje mnoho metod, díky nimž může ovládat a spravovat ostatní třídy. Mezi tyto metody patří například přihlášení a odhlášení uživatele, ověření přihlašovacích údajů, aktualizace obsahu po vyžádání, zobrazování pohledů databází, změna nastavení atp.

System je obsluhován uživatelem pomocí *Webového prohlížeče*, který vykresluje vizuální podobu systému. *System* je ve vztahu agregace s třídou *Databáze*. System má plný přístup do databází, při nefunkčnosti nebo odstranění systému však nedojde k odstranění *Databáze*. *System* je dále ve vztahu s třídou *Modul*. Tento vztah je kompozice. Pokud *System* zanikne, zanikne i třída *Modul* a všechny její podtřídy.

Účet je třída, která umožňuje uživateli přihlášení do systému nebo systémů. Pokud by uživatel neměl aktivní účet, nemohl by obsluhovat systém, protože by se neměl kam přihlásit. *Účet* vzniká registrací uživatele. Třída *Účet* obsahuje atributy o počtu systémů, kam se lze účtem přihlásit, přihlašovací údaje k účtu a nastavené oprávnění k jednotlivým systémům.

Webový prohlížeč je třída využívaná třídou *Uživatel* a vykreslující třídu *System*. Tato třída vlastní atributy o názvu prohlížeče, o jeho současné verzi a o podporovaném operačním systému. Třída *Webový prohlížeč* obstarává hlavně zobrazení uživatelského prostředí pro třídu *Uživatel*. Samotný systém bez prezentačního prostředí by uživateli neměl co nabídnout a zároveň uživatel by neměl žádné možnosti k ovládnutí systému.

Modul (abstraktní nadtřída) je nadtřída k ostatním modulům, které využívá *System*. Třída modul je součástí třídy *System*, bez které nemůže existovat především díky použití vazby kompozice. Třída obsahuje atributy ID modulu, název modulu a atribut k právě propojené databázi, jejíž pohled je zobrazován systémem. Moduly jsou v současné navrhované verzi systému právě čtyři, avšak je zde ponechána možnost na rozšíření počtu těchto modulů.

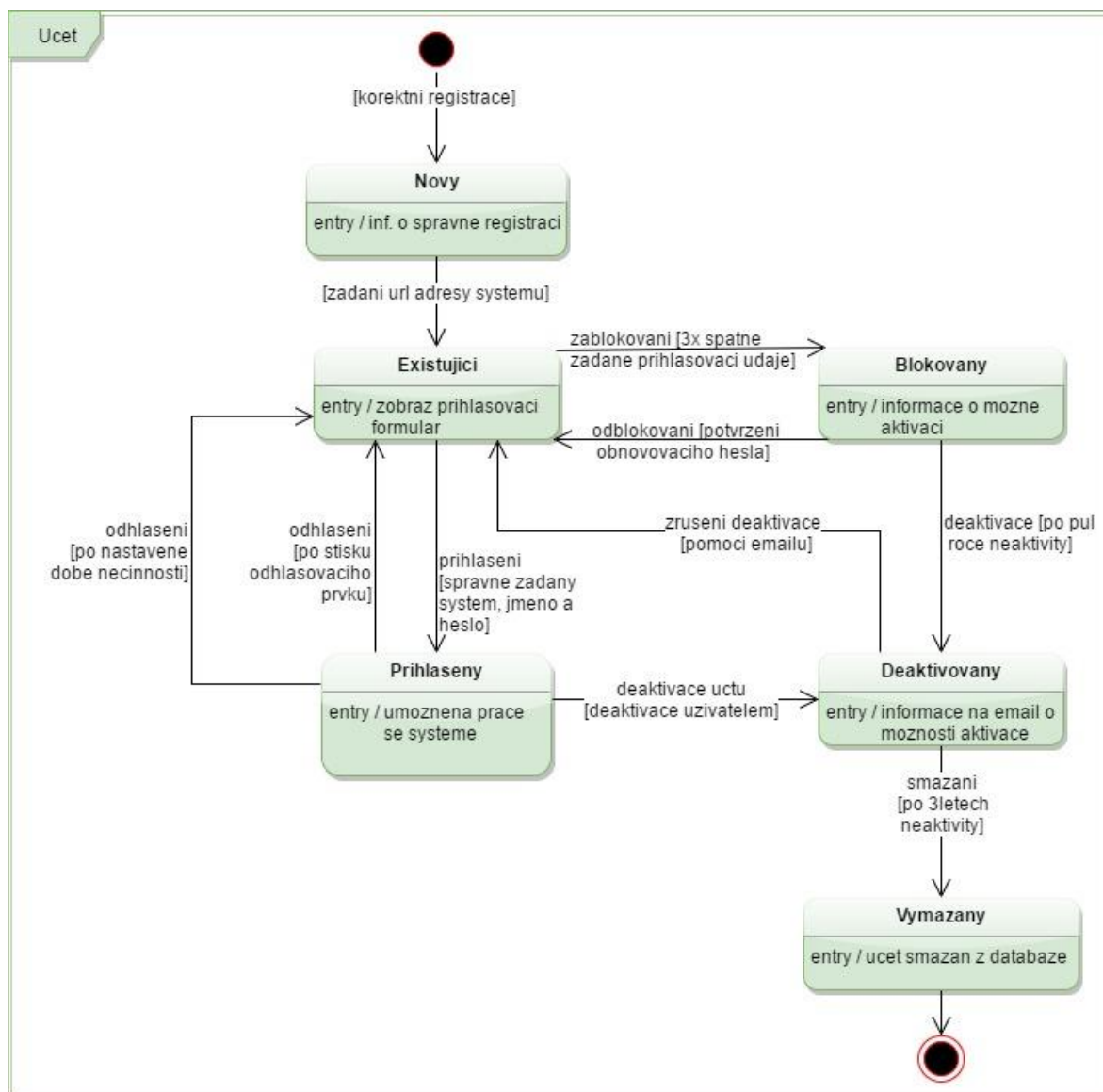
Komunikační modul (podtřída nadtřídy *Modul*) je třída komunikující s *uživatel*em na podněty systému. Podněty systému si uživatel nastaví sám v nastavení. Mezi podněty mohou patřit: nízký stav zboží na skladě, nalezení položky zadané do modulu hlídací pes nebo vyřízení faktury. Komunikační modul využívá atribut ID uživatele k načtení údajů o telefonním čísle uživatele a jeho emailové adrese. Třída může uživatele informovat i prostřednictvím aplikace v lištičce.

Kompletní datový slovník popisující třídy objektů v diagramu tříd je uveden v *Příloha A*.

4.3.2 Stavový diagram

Dynamický model se vyznačuje popisem změn objektů v čase pomocí stavových diagramů. Tyto změny znázorněné stavovými diagramy mohou popisovat chování celého systému pomocí popisu chování jednotlivých tříd objektů.

Účet vznikne po korektní registraci uživatele. Po registraci je uživatel informován o správné registraci a je mu umožněno, po zadání url adresy navrhovaného systému, přihlásit se. Účet je v tuto chvíli ve stavu *Existující*. Při správném zadání všech přihlašovacích údajů se stav změní na stav *Přihlášen*. Ze stavu *Přihlášen* lze odhlásit účet po nastavené době neaktivity, odhlásit lze i stiskem ovládacího prvku pro odhlášení, nebo lze účet deaktivovat. Deaktivace je možná pouze v případě, že uživatel deaktivuje sám sebe. Při chybném zadání přihlašovacích údajů třikrát po sobě je nutné provést odblokování pomocí obnovovacího hesla. Pokud se deaktivace neprovede do doby šesti měsíců, tak se stane účet *Deaktivovaný*. Pro přechod ze stavu *Deaktivovaného* je nutné zrušit deaktivaci pomocí emailu nebo při tříleté nečinnosti se účet automaticky vymaže ze systémové databáze.



Obrázek 22 - Stavový diagram – Účet (Zdroj: Vlastní tvorba)

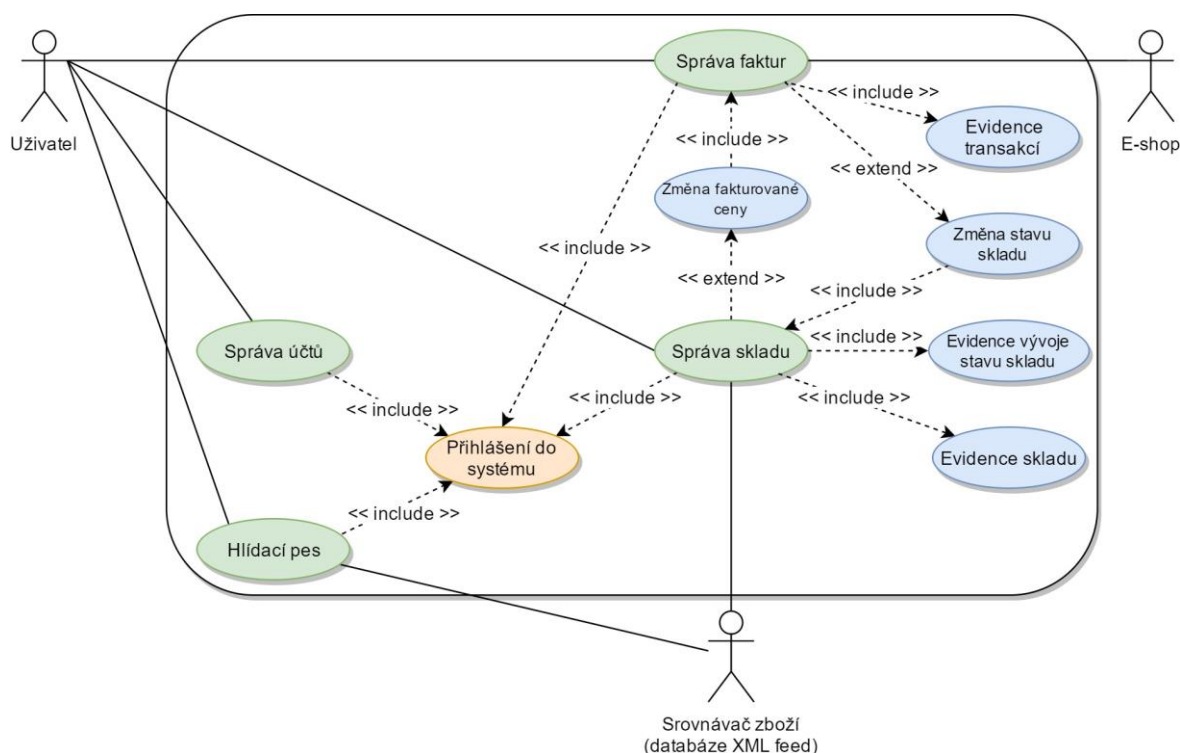
4.3.3 Diagram případů užití

Diagram případů užití popisuje navržený informační systém z pohledu interakce aktérů s funkcionalitami systému. Jedná se o nejobecnější pohled na systém. Samotný popis je zaměřen na jednotlivé případy užití systému, které aktéři využívají. Model případů užití vychází z funkčních požadavků uživatele.

Jako aktéři jsou v diagramu případů užití uvedeni: *Uživatel*, *Srovnávač zboží a E-shop*. *Uživatel* je aktér, který využívá jednotlivé oblasti systému jako nastavení *Hlídacího psa*, *správu účtu*, *správu skladu a správu faktur*.

Srovnávač zboží je aktér, který by ve výsledné realizaci mohl být nahrazen databází XML feedu nebo vytvořením automatického robota, který se bude chovat jako uživatel srovnávače. Tento robot by vyhledával nastavené produkty s určitou periodou a z výsledků porovnání by přeposílal informace ze zdrojového html kódu do systému. Systém by poté obdržené informace zobrazil uživateli.

E-shop je aktérem, který v případě spolupráce se systémem odesílá do systému faktury o nákupech bez nutnosti ručního nahrání faktur uživatelem (v případě dostatečného rozšíření systému).



Obrázek 23 - Diagram případu užití navrhovaného systému (Zdroj: Vlastní tvorba)

Diagram případů užití je barevně rozdělen pro lepší pochopení systému. Zelené případy užití znázorňují hlavní funkcionality systému. Oranžový případ užití je nutný pro využívání služeb systému. Modré případy užití mají doplňkovou funkci k hlavním případům užití.

Vyobrazený aktér *uživatel* představuje zobrazení uživatele jako administrátory vlastního systému, který nemá omezené oprávnění. V případě ostatních uživatelů, kteří mají jiná oprávnění, by se změnil výše zobrazený diagram případů užití. Snížení práv uživatele by bylo zobrazeno odebráním vazby k určitému případu užití, ke kterému by uživatel neměl přístup.

Pokud uživatel vyžaduje jakoukoli interakci se systémem je nejprve nutné se přihlásit. Nutnost přihlášení do systému je zobrazeno vazbou typu <<include>> pro případ užití *Přihlášení do systému* s ostatními případy užití systému, které využívá uživatel.

Případy užití *Evidence vývoje stavu skladu* a *Evidence skladu* jsou vždy dostupné při obsluze případu užití *Správa skladu*. Složitější vazby jsou mezi případy užití *Správa faktur* a *Správa skladu*. Oba případy užití umožňují změnu dat v druhém případě užití. *Správa skladu* umožňuje uživateli měnit ceny položek ve výstupních fakturách díky změně ceny položky v modulu „Ceny pro fakturaci“. *Správa faktur* umožňuje uživateli měnit množství zboží na skladě pomocí příchozích a odchozích faktur.

4.3.4 Sekvenční model

Sekvenční model je použit pro podrobnější úroveň popisu mezi navrhovaným systémem a okolím. Sekvenční model popisuje chování a posloupnosti.

Sekvenční model se skládá ze scénářů a sekvenčních diagramů. Scénář popisuje posloupnost jednotlivých událostí, které nastávají při použití určité části systému. Sekvenční diagram znázorňuje komunikaci mezi klasifikátory popsanou v teoretické části. Sekvenční diagram neumožňuje alternativy rozhodování.

Sekvenčním modelem autor popíše jednotlivé případy užití z diagramu případů užití. Pro správné pochopení navrhovaného systému je důležité popsat hlavní případy užití systému. Tyto případy užití jsou v diagramu případů užití znázorněny zelenou barvou. Dále je nezbytné správně popsat případ užití *Přihlášení do systému*. Bez tohoto případu užití by nemohl být systém využíván uživatelem.

Výčet sekvenčních modelů, které jsou použity pro popis systému:

- Přihlášení do systému (hlavní aktor: uživatel)
- Správa faktur (hlavní aktor: uživatel)

- Správa skladu (hlavní aktor: uživatel, model uveden v příloze)
- Správa účtů (hlavní aktor: uživatel, model uveden v příloze)
- Hlídací pes (hlavní aktor: uživatel, model uveden v příloze)

Následující scénáře představují komunikaci mezi uživatelem systému a samotným systémem. Všechny případy užití zde popisované představují jednotlivé moduly systému, které jsou ovládány právě samotným systémem. Pokud komunikuje uživatel s některým z modulů systému, je ve scénáři popsána tato komunikace jako komunikace uživatele a systému.

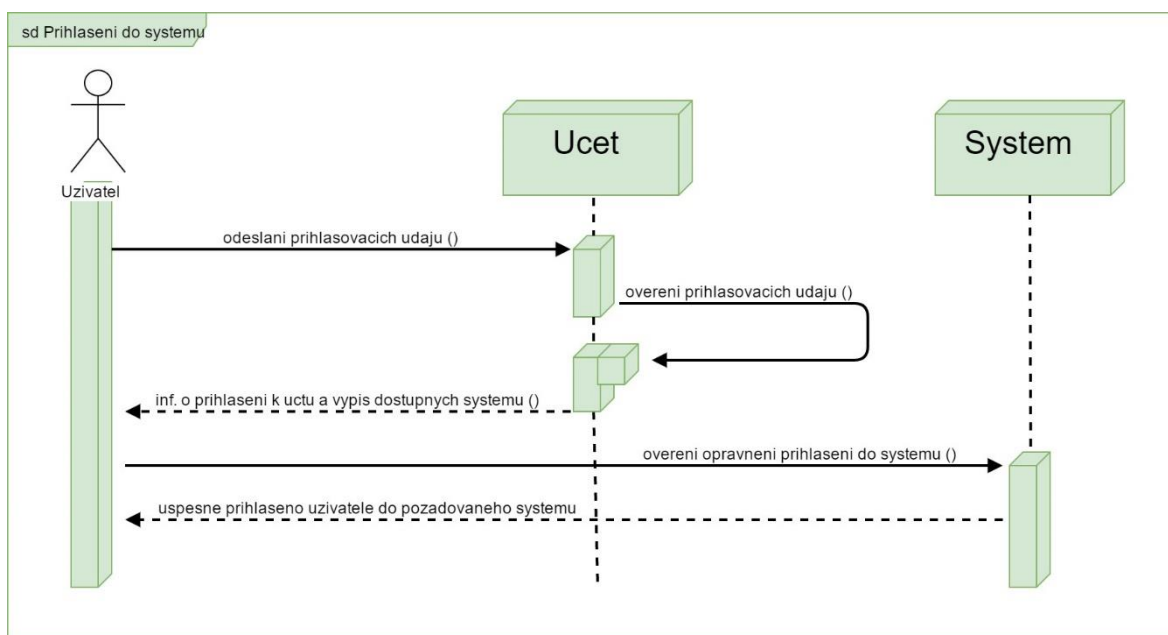
Sekvenční model: Přihlášení do systému

Přihlášení do systému popisuje scénář, který je platný pro všechny uživatele. Z diagramu případů užití je jasné, že pro jakoukoliv činnost v systému je nutné se přihlásit.

Případ užití: Přihlášení do systému	
ID	P01
Stručný popis	Uživatel se přihlašuje do systému.
Hlavní aktér	Uživatel
Vstupní podmínky	Uživatel není přihlášen do systému. Uživatel má vytvořený účet. Uživatelský účet je připojen k alespoň jednomu systému. Uživatel má zobrazený přihlašovací formulář.
Hlavní scénář	<ol style="list-style-type: none"> 1. Uživatel zadá přihlašovací údaje a stiskne potvrzovací prvek <i>přihlásit</i>. 2. Systém ověří přihlašovací údaje. 3. Systém zobrazí uživateli dostupné systémy. 4. Uživatel zvolí požadovaný systém a stiskne potvrzovací prvek <i>přihlásit</i>. 5. Systém přihlásí uživatele k požadovanému systému.
Výstupní podmínky	Uživatel je přihlášen alespoň k jednomu systému.
Alternativní scénář	Neplatné přihlašovací údaje k účtu. Uživatel se rozhodne o založení nového systému.

Tabulka 14 - Scénář – Přihlášení do systému (Zdroj: Vlastní tvorba)

Přihlášení k účtu probíhá zadáním přihlašovacích údajů a potvrzením odeslání. Zadaná data jsou odeslána do systému. Následně proběhne ověření přihlašovacích údajů. Pokud jsou přihlašovací údaje správné je uživatel úspěšně přihlášen k účtu. Po přihlášení k účtu má uživatel možnost vybrat jeden z již připojených systémů, založit systém nový nebo vyhledat již existující systém, ke kterému jeho účet připojen není.



Tabulka 15 - Sekvenční diagram – Přihlášení do systému (Zdroj: Vlastní tvorba)

Sekvenční model: Správa faktur

Sekvenční model správa faktur popisuje scénář vložení nové faktury do systému. Vložení probíhá z interního uložení pracovní stanice.

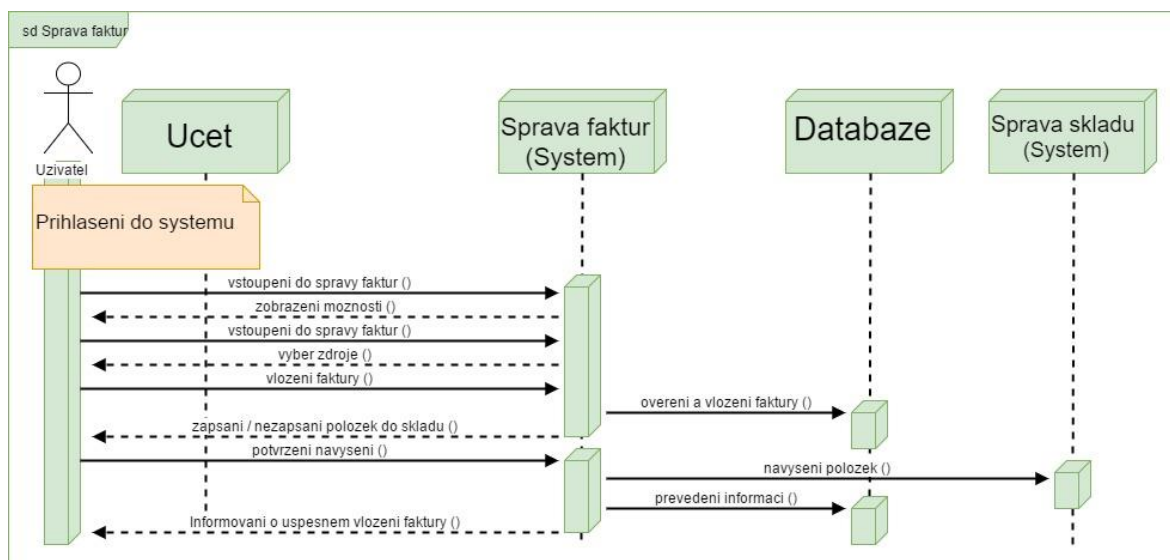
Případ užití: Správa faktur	
ID	P02
Stručný popis	Uživatel přidává fakturu.
Hlavní aktér	Uživatel
Vstupní podmínky	Uživatelův účet má oprávnění pracovat s modulem správy faktur.

Hlavní scénář	<ol style="list-style-type: none"> 1. Uživatel vstoupí do modulu <i>Správa faktur</i> ovládacím prvkem. 2. Systém zobrazí seznam faktur a nabízené možnosti. 3. Uživatel vybere možnost vložení nové přijaté faktury. 4. Systém zobrazí výběr zdroje dokumentu. 5. Uživatel vybere fakturu z uložení PC a stiskne vložit. 6. Systém ověří formát souboru a vloží fakturu do databáze. 7. Systém nabídne uživateli navýšení skladových položek, které jsou obsaženy ve faktuře. 8. Uživatel potvrdí navýšení položek ve skladu. 9. Systém navýší položky ve skladu. 10. Systém převede další informace o dodavateli z faktury do systému. 11. Systém informuje uživatele o úspěšném vložení faktury.
Výstupní podmínky	<p>Faktura je úspěšně vložena do systému.</p> <p>Položky z faktury jsou přidány do modulu sklad.</p> <p>Informace o dodavateli jsou převedeny do systému.</p>
Alternativní scénář	<p>Nahrání faktury se nezdařilo z důvodu nepodporovaného formátu souboru.</p>

Tabulka 16 - Scénář – *Správa faktur* (Zdroj: Vlastní tvorba)

Nahrání faktury do systému je možné provést vytvořením faktury v systému. Vytváření probíhá pomocí vyplňování šablony. Šablony obsahují základní hlavičky jako je odběratel nebo dodavatel. Šablona umožňuje snadné přidávání položek ze skladu do faktury.

Systém podporuje více formátů pro načítané faktury. Systém podporuje primárně formáty XML a HTML. Ostatní formáty je možné převést pomocí čtečky do formátu XML. Nově vytvořený formát XML je možné upravit uživatelem pro správné přiřazení položek a dodavatelových či odběratelských údajů.



Obrázek 24 - Sekvenční diagram – Správa faktur (Zdroj: Vlastní tvorba)

Před samotnou prací se správou faktur je nutné se přihlásit. Zobrazené přihlášení oranžovou barvou je shodné s předchozím diagramem *Přihlášení do systému*.

Při správě faktur se uživatel pohybuje v prostředí *Správa faktur*. Toto prostředí vytváří náhledy z dat v databázi faktur a při změně nebo přidání faktury je databáze aktualizována. Mezi *Správa faktur* a *Správa skladu* probíhá komunikace. Oboustranná komunikace umožňuje pomocí faktur měnit množství zboží na skladě, a naopak pomocí změny ceny zboží upravovat ceny ve vystavených fakturách.

Zbýlé tři sekvenční modely jsou uvedeny v *Příloha B* z důvodu dodržení rozsahu diplomové práce.

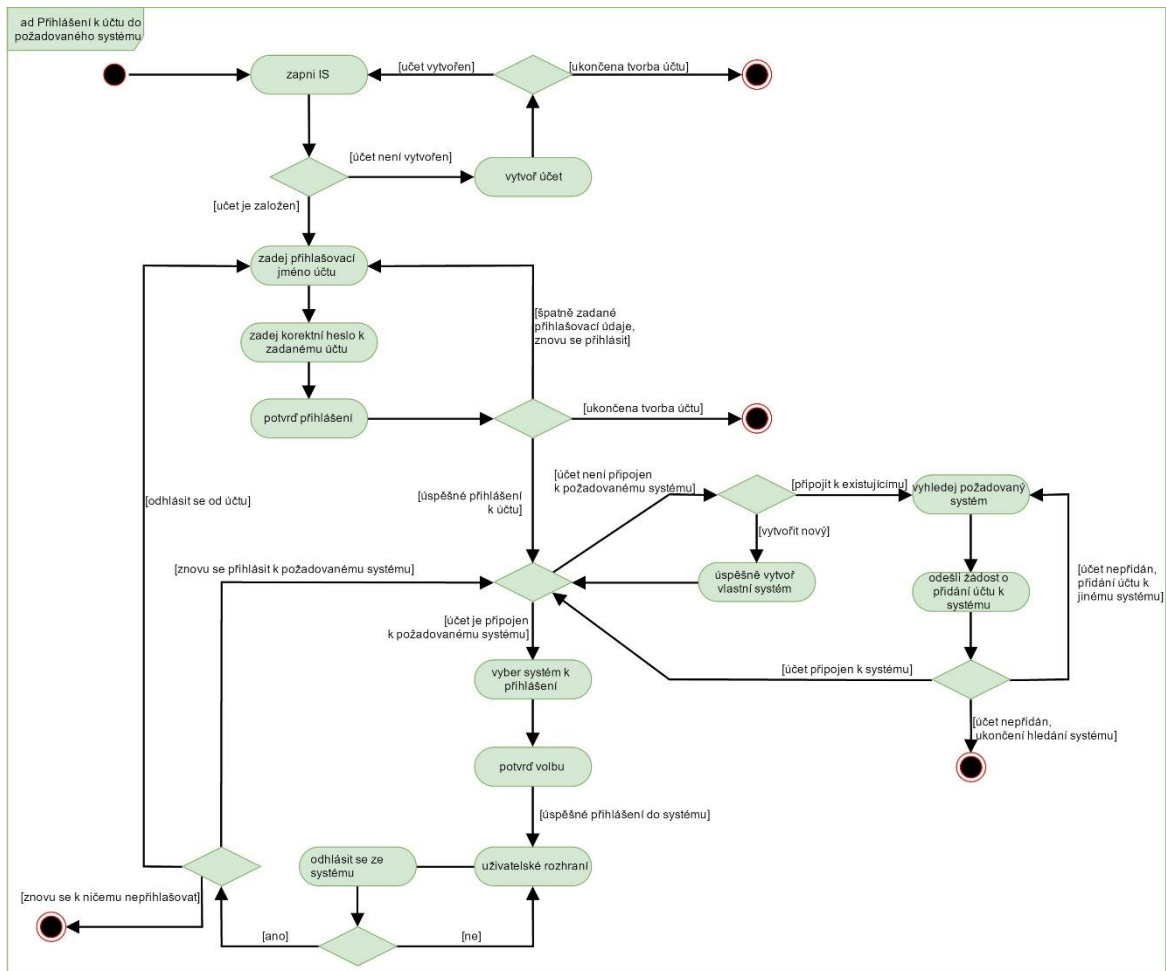
4.3.5 Model aktivit

Model aktivit představuje nejdetailněji popis interakce. Jedná se o poslední uvedený model pro popis navrhovaného systému. Model aktivit zobrazuje a popisuje kroky v rámci složitého algoritmu či procesu.

Následující diagram aktivit zobrazuje proces přihlášení do systému. Přihlášení do systému již bylo popsáno pomocí stavového a sekvenčního diagramu. Model aktivit popisuje postup přihlášení nejdetailněji z již zmíněných diagramů. Diagram aktivit například popisuje tok řízení obdobně jako předchozí diagramy, avšak umožňuje tvorbu

konkurenčních toků řízení. Z hlediska abstrakce diagramu je vytvořen grafický popis přihlášení s použitím nejnižší míry abstrakce.

Níže zobrazený diagram aktivit popisuje postup přihlášení ze stavu, kdy uživatel nemusí mít vytvořený účet. Diagram zobrazuje východiska pro mnoho možných situací procesu přihlášení. Aktivity *úspěšně vytvoř vlastní systém* a *vytvoř účet* je možné dále dekomponovat a vytvořit podrobný popis tvorby nového účtu nebo založení nového systému. Tyto tvorby nejsou v diagramu dekomponovány pro potřebu zachování přehlednosti diagramu. Dalším důvodem je udržení řídicího toku úspěšného přihlášení do požadovaného systému.



Obrázek 25 - Aktivita diagram – Přihlášení do požadovaného systému (Zdroj: Vlastní tvorba)

4.4 Přehled návrhu UML

Souhrn návrhů

Popisem jednotlivých UML modelů bylo dosaženo dostatečné znalosti o logice navrhovaného systému. Diagram případů užití a diagram tříd popisují systém jako celek. Ostatní diagramy jsou orientovány na popis jednotlivých případů užití, tříd, scénářů nebo aktivit, které popisují jednotlivé řídicí toky.

Dělení systému

Návrh systému provedený za pomoci UML modelů postupně zobrazuje navrhovaný systém pomocí podsystémů. Podsystémy se lépe realizují pro vývojové týmy, avšak nesmějí být opomenuty vazby mezi jednotlivými podsystémy. Rozklad systému je nutné zvolit správně dle společných vlastností. Podsystémy bývají obvykle určeny službami, které poskytují. V navrhovaném systému autor také postupoval dělením systému na podsystémy dle jejich nabízených služeb.

Optimalizace

Systém je dle UML modelů navrhován tak, aby měl správnou logiku. Po návrhu správné logiky je nezbytně nutné celý návrh optimalizovat. Optimalizace návrhu obsahuje úpravu přístupů, vazeb nebo funkcí tak, aby byl počet kroků potřebný k určité operaci (řídicímu toku) co nejnižší.

Obsluha uložišť dat

Navrhovaný systém využívá hned několik databází. Bez správného propojení databází na logickou vrstvu by byl výsledný systém nevyhovující až téměř nefunkční. Navrhovaný systém je navrhnut tak, aby nebyl závislý na konkrétním typu použitých databází. Díky tomu bude případně možný přesun na jiný typ databází bez nutnosti přepsání business logiky (*BI*) systému.

4.5 Logický návrh systému

UML modely je spolehlivě popsána logická struktura a funkčnost systému. Pro přiblížení vizuální podoby je vhodné použití „wireframu“ (jedná se o logický náhled, používaný například při tvorbě webových stránek). Wireframe může lépe pomoci pochopit navrhovaný systém pro osoby, které neovládají jazyk UML.

Pro navrhovaný informační systém je dostačující zobrazení dvou wireframů. Systém při využívání jednotlivých modulů zachovává hlavní ovládací prvky. Změnou prochází převážně spodní část wireframu, a to změnou vykreslovaného pohledu. Další výrazná změna wireframu nastává například při stisku aktivního prvku *Nastavení*, kdy se zobrazí vyskakovací okno s nabídkou možností pro nastavení. Zbytek stránky na okrajích si zachová původní vzhled, který bude mírně rozostřen.

pol.	Název zboží	Označení	Množství na skladě	Cena za jednotku	sazba DPH	Cena s DPH	Celkem cena s DPH	Minimální zásoba	Upozornění o nedostatku	Porovnávač zboží
1.	UTP kabel cat.6	859222000158	800m	7,28,- Kč	21 %	8,82,- Kč	7056,- Kč	150	N tel. email apl.	akt.
2.	F-konektor 6.8	KONZIKZKS01	25	5,80,- Kč	21 %	7,- Kč	175,- Kč	10	N tel. email apl.	akt.
3.	F-konektor 5	KONZIKZKS02	10	5,80,- Kč	21 %	7,- Kč	70,- Kč	---	N tel. email apl.	akt.
4.	anténí koax kabel 6,8mm	xkvx100	250m	10,7,- Kč	21 %	12,9,- Kč	3225,- Kč	150	N tel. email apl.	neakt.
5.	koaxial RG6 Amiko	889129	390m	3,3,- Kč	21 %	3,99,- Kč	1556,1,- Kč	150	N tel. email apl.	neakt.
6.										
7.										
8.										

Celková hodnota zboží ve skladu: 12082,1,- Kč

Nastavení upozornění Nastavení priorit porovnávače

Obrázek 26 - Wireframe – Aktuální stav skladu (Zdroj: Vlastní tvorba)

První zobrazený wireframe představuje prostředí modulu Sklad, a to konkrétně zobrazení aktuálního stavu skladu. Druhý wireframe popisuje historii pohybu zboží. Tyto dva wireframe jsou vybrány z důvodu různého zobrazení produktů při použití identické databáze.

Zobrazený wireframe zobrazuje ovládací prvky v horní části obrazovky. Důležitými informativními prvky jsou především logo, název systému, kam je momentálně uživatel přihlášen, název účtu, pod kterým je přihlášen, aktivní prvek pro odhlášení ze systému a možnost nainstalování lištičky do prohlížeče.

Wireframe zobrazuje výběr modulů pomocí stromově hierarchické struktury. Na nejvyšším stupni se nachází *Sklad*, *Faktury*, *Hlídací pes* a *Správa účtů*. *Sklad* a *Faktury* se dále dělí na celky nižšího stupně. Konkrétně modul *Sklad* obsahuje *Aktuální stav skladu*, *Sklad – ceny pro fakturaci* a *Historie pohybu zboží*.

Zobrazený modul *Aktuální stav skladu* popisuje položky, které má uživatel zadané jako vlastněné nebo k dispozici. Tento modul vytváří pohled, ve kterém zobrazuje pouze záznamy, které nemají u atributu množství hodnotu 0. Modul umožňuje úpravu záznamů, přidání dalšího záznamu nebo správu minimálních zásob, upozornění při nedostatku zboží a nastavení porovnávače zboží.

Dále je zobrazen wireframe *Historie pohybu zboží*, který využívá data ze stejné databáze, jako předešlý wireframe.

Logo		Název systému	Název účtu	odhlásit se	Lištička do prohlížeče
Sklad		Faktury	Hlídací pes	Správa účtů	Nastavení
Aktuální stav skladu		Sklad - ceny pro fakturaci	Historie pohybu zboží		
pol.	Název zboží	Označení	1/16 - 2/16 - 3/16 - 4/16 - 5/16 - 6/16 - 7/16 - 8/16 - 9/16 - 10/16 - 11/16 - 12/16 - 1/17 - 2/17 - 3/17 - 4/17 - 5/17	množství	častý dodavatel
1.	UTP kavel cat.6	859222000158		1000 750 500 250 0	CZC.cz
2.	F-konektor 5mm	KONZIZKS02		40 30 20 10 0	CZC.cz
3.	IP Cam ZONE	zone77445		8 6 4 2 0	Elron s.r.o
4.	IP Cam ZONE B	zone77545		8 6 4 2 0	Elron s.r.o

Obrázek 27 - Wireframe – Historie pohybu zboží (Zdroj: Vlastní tvorba)

Z náhledu je zřetelné zachování horní části wireframu. Jedná se především o ovládací prvky a informace o systému a účtu. V navigačním menu je znázorněn jiný modul jako aktivní.

Tabulka ve spodní části wireframu se výrazně změnila. Ačkoliv je náhled tvořen ze stejné databáze, není zde uvedena většina atributů záznamů. Předchozí wireframe přidával několik metod pro výpočet celkové ceny položky s DPH nebo celkovou hodnotu skladu. Wireframe *Historie pohybu zboží* přidává metodu s grafickým výstupem. Metoda využívá sběrová data z naskladnění a vyskladnění jednotlivých položek. Pro potřeby zobrazení časového průběhu množství jsou v tomto pohledu zobrazovány i položky, které mají v atributu *množství* hodnotu 0.

Uživateli tento modul může z dlouhodobého hlediska výrazně pomoci při sledování sezonního prodeje zboží, nebo naopak sezonních slev, podle kterých naskladňoval větší množství zboží.

4.6 Kvalitativní šetření

Následující kvalitativní šetření bylo provedeno na základě informací získaných z následujícího literárního zdroje [35].

Pro zvýšení kvality navrhovaného systému se autor rozhodl vytvořit kvalitativní šetření. Jako výzkumnou metodu kvalitativního šetření autor zvolil jako nejvhodnější metodu dotazování. Dotazování bylo uskutečněno elektronickým dotazníkem a následně elektronickou formou byl dotazník předán vybraným respondentům.

Typ výběrového souboru respondentů byl vytvořen „*Výběrem podle uvážení*“; jedná se o záměrný výběr respondentů dle úsudku autora. Respondenti byli vybráni pro jejich pracovní činnost, která odpovídá skupině koncových uživatelů, pro které je systém navrhován.

Kvalitativní šetření bylo provedeno vytvořením formuláře v programu Microsoft Word 2016, vyplněním odpovědí respondenty a následnou analýzou odpovědí a vytvořením závěru šetření. Formulář byl navržen tak, aby respondent mohl vyplňovat pouze pole určené k odpovědi na zadané otázky. Veškeré úpravy nejsou respondentům umožněny. Dotazníkové otázky jsou zadány jako „otevřené“.

Díky otevřeným otázkám nejsou respondenti v odpovědích tolik omezení jako v uzavřených otázkách. Pomocí většího prostoru pro vyjádření respondentů je možné dosáhnout méně zkreslených odpovědí.

Otázky jsou rozděleny do dvou skupin. V první skupině jsou otázky zaměřené na poznání respondentů, jejich pracovní činnosti a ověření správného výběru respondenta. Otázky jsou otevřeného typu *zcela nestrukturované*. Jsou tedy položeny otázky, na které mohou respondenti odpovídat téměř v neomezeném rozsahu. Avšak konkrétně v této skupině otázek jsou očekávané odpovědi pouze v řádu několika slov.

Druhá skupina otázek je uvedena až po seznámení respondentů s navrhovaným systémem. Druhá skupina otázek je také otevřeného typu *zcela nestrukturované* otázky. Avšak v této skupině jsou očekávané odpovědi výrazně delší, jelikož je dotazováno na respondentův otevřený názor.

Prvním důležitým krokem tvorby dotazníku bylo jasné zformulování problémů. Jedná se o definování informací, které nevíme a informací, které jsou potřeba zjistit z kvalitativního šetření. Před vyhodnocením formuláře nevíme:

1. Zdali byl vybrán správný soubor respondentů.
2. Jak respondenti využívají nákup pomocí internetu a zdali využívají skladovou evidenci.
3. Zda respondenti mají zkušenosti s používáním informačních systémů.

Dotazníkem chceme zjistit:

1. Informace o pracovním zaměření respondentů (popřípadě o jakou podnikatelskou činnost se jedná).
2. Jak jsou respondenti znalí informačních systémů.
3. Bylo vysvětlení navrhovaného systému srozumitelné?
4. Co je třeba případně pozměnit dle respondentů.
5. Zdali má navrhovaný systém smysl realizovat.
6. Zdali by respondenti po realizaci systém používali.

Formulář byl rozdělen do čtyř částí. První část formuláře je průvodní, která seznamuje respondenta o vyplňovaném formuláři a co je obsahem šetření.

Druhá část tvoří otázky na respondenta ohledně jeho současné pracovní činnosti, používání informačních systémů a využívání nákupu nebo prodeje zboží přes internet.

Třetí část popisuje navrhovaný systém. Popis systému je výrazně zkrácený a obsahuje pouze základní přehled o funkcích navrhovaného systému. Na závěr popisu je přidán logický návrh systému pro respondentovu lepší představu.

Závěrečná část obsahuje otázky směřované na zjištění respondentova názoru na navrhovaný systém. Umožňuje napsat připomínky a zjistit, zdali by realizovaný systém respondentovi vyhovoval k pracovní činnosti.

Poslední otázka je kontrolní. Jedná se o zjištění věrohodnosti seznámení respondenta s navrhovaným systémem. Pokud se respondent správně seznámil s navrhovaným systémem bude mu poslední otázka připadat chybně zadaná.

Celý formulář je uveden v *Příloha C*. V případě potřeby autor poskytl respondentům ústní popis, který sloužil k upřesnění některých informací o navrhovaném systému. Tyto informace, které byly nad rámec formuláře, sloužily jen k rozšíření představy o systému.

4.6.1 Vyhodnocení kvalitativního šetření

Pro dodržení tvrzení v dotazníku o anonymitě a nezveřejnění odpovědí respondentů, jsou zde uvedeny pouze shrnutí odpovědí. Dále je zde uvedeno zjištění správnosti vybraného souboru respondentů a vyhodnocení správného pochopení navrhovaného systému.

Správné pochopení navrhovaného systému

Pouze jeden respondent neodhalil poslední otázku jako chybně zadanou. Proto jeho odpovědi při tvorbě souhrnného vyhodnocení nedosahovaly stejné váhy jako odpovědi ostatních respondentů.

Ostatní respondenti na kontrolní otázku odpověděli ve smyslu zpochybnění zadané otázky nebo odpověděli, jako kdyby otázka byla položena správně. Tím je myšleno, že na otázku jejich názoru na instalaci systému odpověděli, že systém nevyžaduje instalaci z důvodu využívání služeb systému ve webovém prohlížeči.

Výběr respondentů

Celých 80 % respondentů odpovídá očekávanému souboru respondentů. Jedná se o OSVČ nebo malé podniky, které využívají ke své pracovní činnosti informační systémy, nákup a prodej zboží pomocí internetu a mají potřebu po určitou dobu skladovat produkty před jejich následným prodejem.

Pouze 20 % respondentů neodpovídalo požadovanému vzorku respondentů, a to především v kritériu velikosti podniku. Tito respondenti uvedli velikost podniku vyšší než 20 zaměstnanců. U této menšiny se zároveň shodně objevila potřeba přidání skladových karet. Při výběru respondentů nebyl záměrně osloven žádný respondent, jehož hlavní pracovní činností je pouze prodej, skladování a následný prodej zboží. Pro tyto respondenty je již vytvořena celá škála informačních skladových systémů, které svou složitostí a možnostmi výrazně převyšují rozsah skladového modulu v navrhovaném systému.

Souhrn odpovědí respondentů

Jak již bylo zmíněno, při větším rozsahu podniku nad dvacet zaměstnanců bylo respondenty vyžadováno přidání skladových karet. I pokud není hlavní činností podniku nákup, skladování a prodej zboží, je při postupném rozrůstání podniku zvýšená nutnost zavedení skladových karet.

Vyjádření respondentů k otázce týkající se popisu byla v souhrnu dvojího typu. Jedna část respondentů byla spokojena s popisem a jeho stručností. Druhá část respondentů částečně kritizovala příliš obecný popis navrhovaného systému, ze kterého nejsou všechny funkce dostatečně popsány. Druhá skupina tvořila 30 % respondentů, kterým byly nejasnosti popisu vysvětleny individuálně.

Všichni respondenti uvedli teoretickou použitelnost modulů navrhovaného systému pro jejich pracovní činnost.

Funkcí, které respondenti v navrhovaném systému postrádají, bylo více. První postrádaná funkce je absence možnosti vytvoření nabídky zboží pro zákazníky respondentů. Tato funkce by byla proveditelná v navrhovaném systému a je vhodné jí do systému přidat. Druhá postrádaná funkce tvoří možnosti úpravy. Konkrétně se jedná o úpravu atributů, řazení atributů a celková úprava vytvářeného pohledu. Tyto možnosti není složité přidat do návrhu systému. Tato změna představuje další krok, který by měl být

proveden po realizaci základní struktury systému. Uvedené změny více přizpůsobují navrhovaný systém potřebám jednotlivých uživatelů, což je v posledních letech rozvíjející se trend.

V systému si respondenti všimli absence možnosti odhlášení se od systému bez nutnosti odhlášení se od účtu. Oprava této chyby byla zahrnuta do výsledků šetření.

Další vhodná úprava systému spočívá v přidání vyhledávacího pole do všech modulů, které tvoří pohledy z databáze. Při delším používání navrhovaného systému se může databáze rozšířit až na stovky položek, mezi kterými je orientace pouze pomocí posuvníku znatelně nedostatečná.

S tím souvisí i poslední postrádaná funkce dle respondentů. Jedná se o absenci možnosti přidávat produkty do seznamu položek, který si může uživatel sám vytvářet, upravovat a zobrazovat. Tuto funkci by bylo vhodné přidat pod modul *Sklad*.

4.6.2 Změny v systému dle kvalitativního šetření

Na základě souhrnu odpovědí jsou výše zmíněné funkce přidány do návrhu systému. V zobrazených modelech UML se však tyto změny projeví pouze v úpravě datového slovníku, nebo například přidáním nového sekvenčního modelu na tvorbu nabídky. Proto se autor rozhodl úpravu systému znázornit pomocí Wireframu, kde se dají úpravy systému nejlépe znázornit. Je to logické především z důvodu, že respondenti měli zmíněný wireframe k dispozici při odpovídání na nedostatky navrhovaného systému.

Změny, které je vhodné provést:

1. Umožnit uživateli systému provést úpravu atributů, které jsou v daném pohledu zobrazovány. Úpravou je myšlena možnost seřazení položek v pohledu dle atributů. Další úpravu představuje možnost přidání a odebrání atributů nebo přesouvání atributů horizontálně dle uživatelských potřeb.
2. Do modulu *Sklad* je přidán nový pohled, který je určen přímo pro tvorbu skupin jednotlivých produktů. Jedná se pouze o grafický doplněk a přidání jednoho atributu do relační databáze. Tento atribut nese název skupiny, do které byla položka přidána uživatelem.

- Přidání vyhledávání položek dle všech atributů, které položky obsahují, přispěje k rychlosti vyhledávání položek (zboží, transakcí, faktur atd.). Toto vyhledávací pole je přidáno i do všech ostatních podsystémů, ve kterých urychluje nalezení požadované položky.
- Rozdělení možnosti odhlášení od systému nebo od uživatelského účtu je důležitým prvkem, který autor při návrhu opomněl.
- Přidání možnosti tvorby nabídek je velmi důležitým krokem. Tento podsystém bude fungovat obdobně jako tvorba faktury, umožňuje však uživateli systému přidávat i položky, které mají ve skladu atribut *množství* roven 0. Díky tomu je možné vytvořit nabídku i na zboží, které nemusí být v současný okamžik k dispozici.

4

odhlásit se od systému odhlásit se od účtu Lištička do prohlížeče

5

Logo Název systému Název účtu

Sklad **Faktury - Nabídky** Hlídací pes Správa účtů Nastavení

Aktuální stav skladu Sklad - ceny pro fakturaci Historie pohybu zboží **2** Správa skupin **3** Vyhledej

1	Název zboží	Označení	Množství na skladě	Cena za jednotku	sazba DPH	Cena s DPH	cena s DPH	minimální zásoba	Upozornění nedostatku	Porovnávač zboží
1.	UTP kabel cat.6	859222000158	800m	7,28,- Kč	21 %	8,82,- Kč	7056,- Kč	150	N	tel. email apl. akt.
2.	F-konektor 6,8	KONZIKS01	25	5,80,- Kč	21 %	7,- Kč	175,- Kč	10	N	tel. email apl. akt.
3.	F-konektor 5	KONZIKS02	10	5,80,- Kč	21 %	7,- Kč	70,- Kč	---	N	tel. email apl. akt.
4.	anténní koax kabel 6,8mm	xkvx100	250m	10,7,- Kč	21 %	12,9,- Kč	3225,- Kč	150	N	tel. email apl. neakt.
5.	koaxial RG6 Amiko	889129	390m	3,3,- Kč	21 %	3,99,- Kč	1556,1,- Kč	150	N	tel. email apl. neakt.
6.										
7.										
8.										

Celková hodnota zboží ve skladu 12082,1,- Kč

Nastavení upozornění Nastavení priorit porovnávače

Obrázek 28 - Wireframe – Aktuální stav skladu po úpravách (Zdroj: Vlastní tvorba)

4.7 Tvorba reprezentativní webové aplikace

Navrhovaný systém je pro správné pochopení vhodné implementovat jako webovou aplikaci. Vytvořená aplikace popisuje především uživatelské prostředí. Popis je proveden pomocí shodného grafického zobrazení navrhovaného systému. Oproti navrhovanému systému tvoří vytvořená aplikace funkci demonstrativní a přispívá k představě o vizuálním provedení navrženého systému. Tím je myšlena absence napojení na databáze, nemožnost úprav prostředí a dále je uživatelský pohyb po aplikaci omezen. Webová aplikace umožňuje průchod jednotlivými moduly systému. V modulech není možné upravovat náhledy, vkládat dokumenty ani měnit systémové či uživatelské nastavení.

Webová aplikace je vytvořena a uložena na cvičném univerzitním serveru kitlab.pef.czu.cz. Webová aplikace byla vytvořena pomocí prvků HTML, CSS3, jQuery, frameworku Bootstrap a jazyka PHP.

Pomocí využití zmíněných prvků byla vytvořena přihlašovací stránka aplikace. Odesláním správného uživatelského jména a hesla se vytvoří session, které dále umožňuje pohyb po dalších jednotlivých stránkách. Dále obdobným způsobem byla vytvořena stránka pro výběr systému, kam se uživatel přihlásí.

Následující stránky potřebují k otevření oprávnění přenášené pomocí session. Jedná se o dalších šest stránek, které se přepínají dle stisknutí ovládacích prvků v navigačním menu. Pro představu je umožněn přístup do všech náhledů modulu *Sklad*, jednoho náhledu pro modul *Faktury – Nabídka* a jednoho náhledu do modulu *Hlídací pes*.

Aktuální stav skladu umožňuje přidání, úpravu a odstranění položek. Tyto položky jsou uloženy v připojené databázi. Ostatní záložky zobrazují pouze informativní obsah o navrhovaných funkcích systému.

Funkční webová aplikace je přístupná, na již zmíněném serveru kitlab.pef.czu.cz. Přesná adresa přihlašovací stránky webové aplikace je: kitlab.pef.czu.cz/1617zs/ete32e/06/is.

Univerzální přihlašovací údaje pro umožnění zobrazení celé webové aplikace jsou:

- Přihlašovací jméno: *admin*
- Heslo: *admin*

5 Diskuze

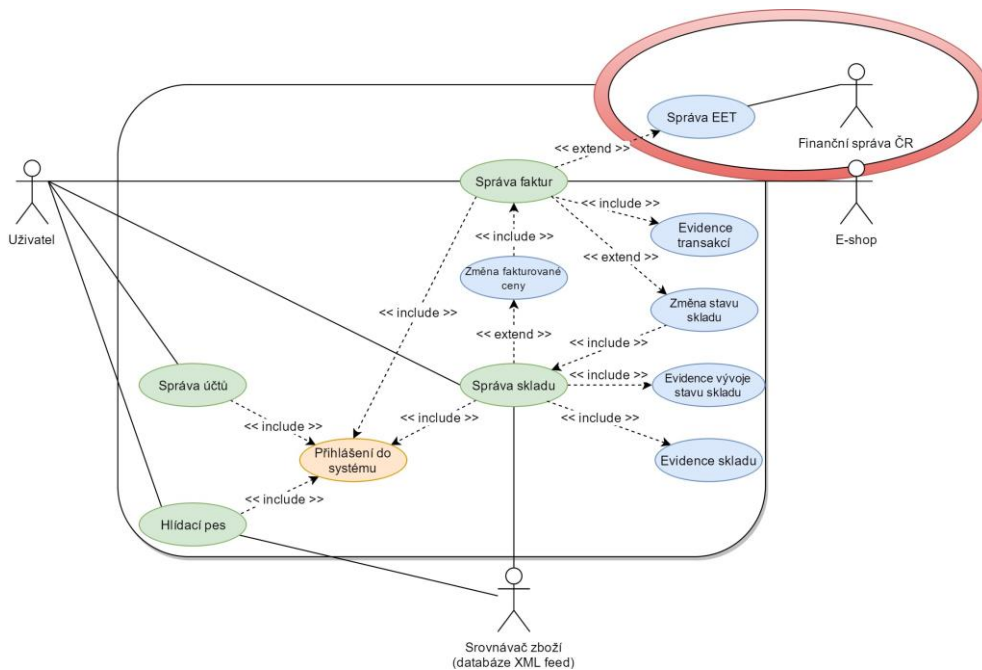
5.1 Aktuálnost práce

Při výběru této diplomové práce autor neočekával brzké a povinné zavedení EET (Elektronická evidence tržeb). Z toho důvodu nebyl na zavedení EET brán zřetel v době vyhotovení zadání diplomové práce. Avšak pro dodržení aktuálnosti je vhodné nastínit přidání EET do navrhovaného systému. Jelikož navrhovaný systém pracuje s evidováním položek, které jsou potencionálně nakupovány nebo prodávány, je vhodné EET do systému zahrnout.

Elektronická evidence tržeb zaznamenává prodej zboží, služeb nebo gastronomie za platbu v hotovosti nebo platební kartou. Proto důležitým krokem, který je nutné přidat, je přidání modulu pro správu EET k fakturačnímu modulu právě pro možnost platby v hotovosti nebo platební kartou.

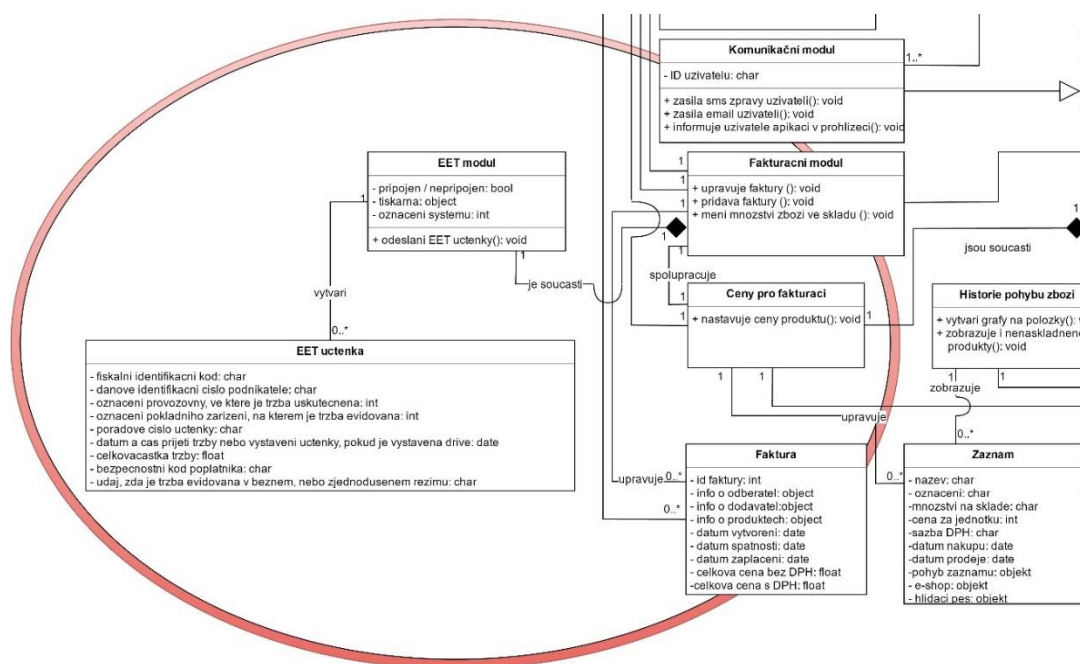
Při platbě v hotovosti se zapne modul pro EET, který bude napojen na finanční správu ČR. Při vytvoření odchozí faktury se systém dotáže na způsob platby. Pokud bude zvolena možnost platby v hotovosti nebo platební kartou, bude vytvořena EET účtenka. EET účtenka převezme atributy faktury a přidá své povinné atributy. Vytvořená EET účtenka je v elektronické podobě uložena ve správě EET a odeslána finanční správě ČR. Tištěná verze EET účtenky je vytištěna pomocí tiskárny podporující formát EET a předána odběrateli.

Přidání správy EET do navrhovaného systému je znázorněno na diagramu případu užití. Do diagramu je přidán jak případ užití, tak aktor *Finanční správa ČR*, který přijímá elektronickou verzi EET účtenky.



Obrázek 29 - Diagram případů užití – přidání EET (Zdroj: Vlastní tvorba)

Následující část diagramu tříd popisuje přidání EET do navrhovaného systému. Jedná se především o přidání třídy *EET modul* a třídy *EET uctenka*. Úctenka obsahuje atributy zadané státem. Modul EET vytváří účtenky na základě dat převzatých ze třídy *Fakturační modul*.



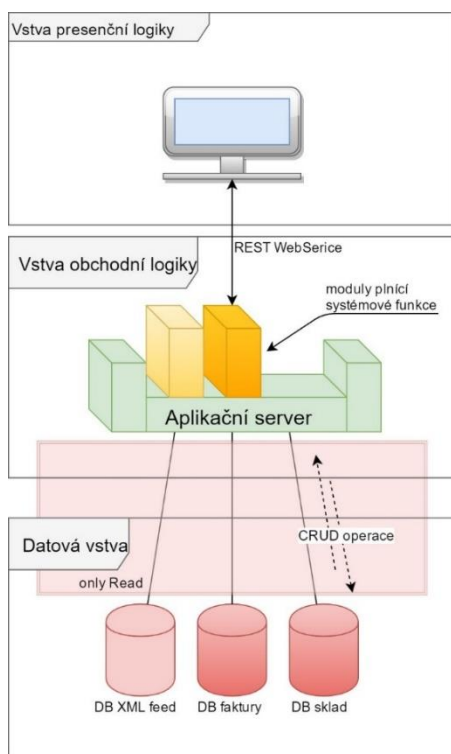
Obrázek 30 - Částečný diagram tříd – přidání EET (Zdroj: Vlastní tvorba)

5.2 Návrh pro realizaci

Samotná realizaci navrhovaného systému není tématem této diplomové práci. Autor však předpokládá potencionální realizaci následovně:

1. Navrhovaný systém by bylo vhodné postavit na 3 a vícevrstvé architektuře.
2. Komunikace mezi jednotlivými vrstvami by byla zajištěna pomocí standardizovaných protokolů.
3. Datová vrstva obsahuje pouze databáze ovládané pomocí operací CRUD (vytvořit, číst, upravit, smazat). Výjimku tvoří databáze XML feedu.
4. Vrstva aplikační logiky je obsažena na aplikačním serveru. Pomocí serveru je komunikováno s databázemi.
5. Samotný systém a jeho funkce bude obsažen pomocí modulů ve vrstvě obchodní logiky. Tato vrstva vytváří pomocí *Rest Web Services* interaktivní uživatelské rozhraní.
6. Vrstva prezenční logiky zobrazuje uživateli navrhovaný systém a umožňuje mu ovládání systému.

Pro lepší pochopení předpokládané architektury je zobrazen návrh na následujícím schématu.



Obrázek 31 - Schéma potencionální realizace (Zdroj: Vlastní tvorba)

5.2.1 Mobilní aplikace

V systému je popisována možnost informování uživatele o určitých změnách pomocí mobilní aplikace. Mobilní aplikace by měla pouze funkci upozornění. Nejednalo by se o odlehčenou verzi celého systému. V případě, že chce být uživatel informován, tak mu pomocí mobilní aplikace přijde pouze vyskakovací upozornění o konkrétní změně, kterou nechal sledovat. Po otevření upozornění bude uživatel převeden na webovou stránku systému a po následném přihlášení bude uživatelova činnost v systému prováděna pouze pomocí mobilního webového prohlížeče.

5.2.2 Problém při realizaci

Pro realizaci navrhovaného systému by bylo důležité vyřešení několika bariér, které samotnou realizaci neumožňují. Největší bariéru při realizaci navrhovaného systému tvoří propojení na XML feed databáze.

Databází XML feedu je větší množství. V teoretické části bylo zobrazeno procentuální rozvržení e-shopů pro jednotlivé databáze XML feedu. Dostatečně velká množina e-shopů by vznikla propojením databáze Heureka.cz a Zboží.cz.

Dnes jakýkoliv e-shop se může přidat do zmíněných databází. Téměř všichni majitelé zmíněných databází udávají postupy, jak správně samotný XML feed vyplnit a odeslat majiteli databáze. Problém nastává v dostupnosti této databáze ostatním uživatelům. Žádný z majitelů databází XML feedu (převážně srovnávače zboží) neumožní přístup ke své databázi jen tak.

Pro řešení daného problému jsou dvě možnosti:

1. První možnost spočívá ve vytvoření výhodné smlouvy pro daný srovnávač. Srovnávač by poskytl pouze možnost *Read* databázi XML feedu pro navrhovaný systém. Poskytnutím přístupu a možnosti čtení záznamů by nebyla data ohrožena případnou změnou ze strany navrhovaného systému a jeho uživatelů. Čtení dat by bylo pro navrhovaný systém dostatečné. Jedná se částečně o podobný požadavek, který srovnávače nabízejí jako službu všem, kdo využívají jejich srovnání zboží. Je však pravděpodobné, že by srovnávače nechtěly umožnit přístup do databáze

z důvodů okopírování informací a tím i jedné z nejdůležitějších a nejcennější částí jejich společnosti.

2. Druhou možností by bylo vytvoření již zmíněného robota, který by využíval služby předem nastavených srovnávačů stejně jako lidský uživatel. V současné době není obtížné vytvořit robota (skript), který by se choval jako normální uživatel. Robot by zadával požadavky, které by si nastavil uživatel v navrhovaném systému a dále by z příchozího html dokumentu vypisoval důležité informace (cena, množství...) a ty dále porovnával. Při použití této možnosti by navrhovaný systém fungoval, jak má. Problém je v počtu vyhledávání na srovnávačích. Tímto přístupem by veškeré požadavky nebyli mířeny přímo na databázi, ale na samotný server vyhledávače. Postupem času by se z nárazového periodicky opakovaného vyhledávání pomocí robotů mohl stát téměř útok DDoS, což je nežádoucím výsledkem této varianty.

6 Závěr

Cílem diplomové práce bylo pomocí jazyka UML navrhnout informační systém, který pomáhá zjednodušit a částečně automatizovat administrativu spojenou s nákupem, prodejem a skladováním zboží. Navržený systém splňuje zadání, které bylo definováno jako cíl autorovy práce. Navrhovaný systém umožňuje propojení podsystémů, které jsou nutné pro řádné vedení administrativy spojené s podnikatelskou činností uživatele, v jeden uživatelsky přívětivý celek. Systém při jeho zavedení může výrazně snížit množství času potřebného k přehlednému řízení podnikatelské činnosti a s ní spojených vytvářených záznamů (objednávky, faktury). Snížení množství času je dosaženo i pomocí automatizace rutinních operací tam, kde je to vhodné. V systému je však důležité umožnit určité operace spravovat manuálně nebo automatizaci kontrolovat uživatelem.

Navržený systém spadá do kategorie podnikových informačních systémů a jeho návrh probíhal pomocí modifikovaného vodopádového modelu životního cyklu systému. Model byl přizpůsoben potřebám diplomové práce. Jedná se o přeskočení fáze „Implementace“ přímo k fázi „Testování“, kterou byl ověřován návrh, nikoli realizovaný systém. Výsledky získané pomocí testování byly zakomponovány do následné implementace, která proběhla pouze nad určitou částí navrhovaného systému, a to formou vytvoření webové aplikace. Webová aplikace je oproti doporučení autora k navrhovanému systému vytvořena pouze na dvouvrstvé architektuře, umožňuje uživateli jen několik funkcí a je napojena pouze na jednu databázi.

Návrh systému popsany v diplomové práci pomocí UML modelů by mohl být z hlediska velikosti systému obsáhlejší, nicméně rozsah diplomové práce neumožňuje dostatečný prostor k vytvoření takového množství modelů. Modely UML je popsána základní struktura systému a vybrané dynamické části systému a jeho podsystémů. Pro rozsáhlejší popis systému byla diplomová práce doplněna i pomocí wireframů, které například lépe znázorňují logické rozložení ovládacích prvků systému. Jeden ze základních wireframů sloužil spolu s popisem systému jako podklad k vytvoření dotazníku potřebného pro provedení kvalitativního šetření. Na základě vyhodnocení kvalitativního šetření provedl autor navrhované úpravy, které byly lépe zobrazeny změnou ve wireframu a následně byly změny aplikovány do webové aplikace.

V diskusi diplomové práce autor nastínil možnost napojení systému na elektronickou evidenci tržeb (EET). Diskuze dále obsahuje autorovo doporučení pro návrh realizace systému a problém, se kterým by se mohl vývojářský tým setkat při tvorbě navrhovaného systému.

Navrhovaný systém vychází z propojení již existujících a ověřených produktů informačních systémů, avšak určitá část propojení navrhovaného systému není v současné době možná. Jedná se především o využívání databází XML feed od společností Heureka.cz a Zboží.cz. Toto propojení je z technického hlediska realizovatelné, ale není možné jeho uskutečnění z důvodu ochrany dat výše zmíněných společností.

Navrhovaný systém je univerzální, jelikož není zaměřen na konkrétní podnikatelskou činnost. Jeho využití je možné v široké oblasti podniků a fyzických osob podnikajících s odlišnými požadavky. V případě realizace navrženého informačního systému by mohlo být dosaženo tzv. „vyplnění díry na trhu“ v oblasti informačních systémů, které jsou pro malé podnikatele dostupné především po finanční stránce.

Současná situace v České republice je spíše nepříznivá pro soukromé podnikatele a přispívá k jejich úbytku. Navržený systém by mohl výrazně zmírnit obtížnost činností spjatých s povinnými rutinami plynoucími z podnikatelské činnosti a tím přispět k rozvoji řemeslné živnosti.

7 Seznam použitých zdrojů

- [1] VRANA, Ivan. *Projektování informačních systémů s UML*. Vyd. 1. V Praze: Česká zemědělská univerzita, Provozně ekonomická fakulta, 2008. ISBN 9788021318175.
- [2] VRANA, Ivan a Karel RICHTA. *Zásady a postupy zavádění podnikových informačních systémů: praktická příručka pro podnikové manažery*. 1. vyd. Praha: Grada, 2005. Management v informační společnosti. ISBN 8024711036.
- [3] MOLNÁR, Zdeněk. *Podnikové informační systémy*. Vyd. 2., přeprac. V Praze: České vysoké učení technické, 2009. ISBN 9788001043806.
- [4] VYMĚTAL, Dominik. *Informační systémy v podnicích: teorie a praxe projektování*. 1. vyd. Praha: Grada, 2009. Průvodce. ISBN 9788024730462.
- [5] WORDSWORTH, J. *Software engineering with B*. Reading, Mass.: Addison-Wesley, 1996. ISBN 0201403560.
- [6] BASL, Josef a Roman BLAŽÍČEK. *Podnikové informační systémy: podnik v informační společnosti*. 3., aktualiz. a dopl. vyd. Praha: Grada, 2012. Management v informační společnosti. ISBN 9788024743073.
- [7] BRUCKNER, Tomáš. *Tvorba informačních systémů: principy, metodiky, architektury*. 1. vyd. Praha: Grada, 2012. Management v informační společnosti. ISBN 9788024741536.
- [8] AALST, Wil a Christian STAHL. *Modeling business processes: a petri net-oriented approach*. Cambridge, Mass.: MIT Press, 2011. Cooperative information systems. ISBN 0262015382.
- [9] SOMMERVILLE, Ian. *Softwarové inženýrství*. 1. vyd. Brno: Computer Press, 2013. ISBN 9788025138267.
- [10] RICHTA, Karel a Jiří SOCHOR. *Softwarové inženýrství I*. Vyd. 1. Praha: České vysoké učení technické, 1996. ISBN 8001014282.
- [11] MCCONNELL, Steve. *Dokonalý kód: umění programování a techniky tvorby software*. Vyd. 1. Brno: Computer Press, 2005. ISBN 802510849X.
- [12] 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 9788025115039.
- [13] BUCHALCEVOVÁ, Alena. *Metodiky vývoje a údržby informačních systémů: kategorizace, agilní metodiky, vzory pro návrh metodiky*. Praha: Grada Publishing, 2005. ISBN 8024710757.
- [14] SELBY, Richard. *Software engineering: Barry W. Boehm's lifetime contributions to software development, management, and research*. Hoboken, N.J.: IEEE Computer Society, 2007. ISBN 047014873X.
- [15] SCHMULLER, Joseph. *Myslíme v jazyku UML*. 1. vyd. Praha: Grada, 2001. Knihovna programátora (Grada). ISBN 8024700298.
- [16] MERUNKA, Vojtěch. *Objektové modelování*. 1. vyd. Praha: Alfa Nakladatelství, 2008. Informatika studium (Alfa Nakladatelství). ISBN 9788087197042.
- [17] KANISOVÁ, Hana a Miroslav MÜLLER. *UML srozumitelně*. 2., aktualiz. vyd. Brno: Computer Press, 2006. ISBN 8025110834.
- [18] WEILKIENS, Tim. a Bernd. OESTEREICH. *UML 2 certification guide: fundamental and intermediate exams*. Boston: Elsevier/Morgan Kaufmann, 2007. ISBN 9780123735850.
- [19] PENDER, Tom. *UML bible*. Indianapolis, IN: Wiley, 2003. ISBN 0764526049.

- [20] BROOKS, Frederick. *The mythical man-month: essays on software engineering*. Anniversary ed. Reading, Mass.: Addison-Wesley Pub. Co., 1995. ISBN 0201835959.
- [21] FOWLER, Martin. *Destilované UML*. 1. vyd. Praha: Grada, 2009. Knihovna programátora (Grada). ISBN 9788024720623.
- [22] ARLOW, Jim a Ila NEUSTADT. *UML a unifikovaný proces vývoje aplikací: průvodce analýzou a návrhem objektově orientovaného softwaru*. Vyd. 1. Brno: Computer Press, 2003. ISBN 807226947X.
- [23] HUNTER, David. *Beginning XML*. 4th ed. Indianapolis, IN: Wrox/Wiley Pub., 2007. Wrox beginning guides.
- [24] MIKULÁŠKOVÁ, Petra a Mirek SEDLÁK. *Jak vytvořit úspěšný a výdělečný internetový obchod*. 1. vyd. Brno: Computer Press, 2015. ISBN 9788025143834.
- [25] JAHODA, Bohumil. Statistika nejpoužívanějších prohlížečů. In: *Je čas.cz* [online]. 2016 [cit. 2017-01-31]. Dostupné z: <http://jecas.cz/statistiky-prohlizecu>
- [26] JANOUCHEK, Viktor. *Internetový marketing*. 2. vyd. V Brně: Computer Press, 2014. ISBN 9788025143117.
- [27] KASÍK, Pavel. Proč Google Chrome předběhl IE a stal se nejpoužívanějším prohlížečem. In: *Idnes: Technet* [online]. 2012 [cit. 2017-01-31]. Dostupné z: http://technet.idnes.cz/google-chrome-predbeh-ie-internet-explorer-fzt-/sw_internet.aspx?c=A120521_150519_sw_internet_pka
- [28] ŠPINAR, David. Pravidla tvorby přístupného webu. In: *Přístupnost na webu* [online]. 2007 [cit. 2017-01-31]. Dostupné z: <http://pristupnost.nawebu.cz/texty/pravidla-standardy.php?full>
- [29] SPOLSKY, Joel. How Microsoft Lost the API War. In: *Joelonsoftware.com* [online]. New York, 2004 [cit. 2017-01-31]. Dostupné z: <https://www.joelonsoftware.com/2004/06/13/how-microsoft-lost-the-api-war/>
- [30] Vlastníte e-shop? Využijte nabídky srovnávačů zboží, můžete prodat více. In: *Podnikatel.cz* [online]. 2010 [cit. 2017-01-31]. Dostupné z: <http://www.podnikatel.cz/clanky/vyuzijte-nabidky-srovnavacu-zbozi/>
- [31] PROCHÁZKA, David. *SEO: cesta k propagaci vlastního webu*. 1. vyd. Praha: Grada, 2012. Průvodce (Grada). ISBN 9788024742229.
- [32] LIDWELL, William, Kritina HOLDEN a Jill BUTLER. *Univerzální principy designu: 125 způsobů jak zvýšit použitelnost a přitažlivost a ovlivnit vnímání designu*. Vyd. 1. Brno: Computer Press, 2011. ISBN 9788025135402.
- [33] *Ishopy* [online]. Deutsch-Wagram, 2016 [cit. 2017-01-31]. Dostupné z: <http://www.ishopy.eu/>
- [34] COLE, Timothy a Myung-Ja HAN. *XML for catalogers and metadata librarians*. Santa Barbara: Libraries Unlimited, 2013. Third millennium cataloging. ISBN 9781598845198.
- [35] KOTLER, Philip. *Moderní marketing: 4. evropské vydání*. 1. vyd. Praha: Grada, 2007. ISBN 9788024715452.

8 Seznam příloh

Příloha A - Kompletní datový slovník Diagramu tříd	99
Příloha B - Sekvenční modely	103
Příloha C - Kvalitativní šetření	109

Přílohy

Příloha A - Kompletní datový slovník Diagramu tříd

Uživatel je člověk, který má vytvořen vlastní účet v navrhovaném systému. Uživatel je třída, která se přihlašuje do systému, odhlašuje se ze systému a celý systém obsluhuje. Atributy třídy *Uživatel* jsou především osobní údaje. Mezi tyto atributy patří ID uživatele, jméno, příjmení, IČO (pokud se jedná o OSVČ), email a telefonní číslo (pro potřeby systémového upozornění uživatele). Uživatel využívá *Webový prohlížeč* k ovládání systému.

Systém je celek, který spravuje a sdružuje všechny ostatní moduly a databáze. Jedná se o třídu, jejíž atributy popisují přiřazené účty, právě zobrazovaný modul a současně využívanou databázi pro správný databázový pohled. Třída *Systém* znázorňuje samotné jádro systému, ke kterému se ostatní třídy (moduly a databáze) připojují. Třída *Systém* však obsahuje mnoho metod, díky nimž může ovládat a spravovat ostatní třídy. Mezi tyto metody patří například přihlášení a odhlášení uživatele, ověření přihlašovacích údajů, aktualizace obsahu po vyžádání, zobrazování pohledů databází, změna nastavení atp.

Systém je obsluhován uživatelem pomocí *Webového prohlížeče*, který vykresluje vizuální podobu systému. *Systém* je ve vztahu agregace s třídou *Databáze*. *Systém* má plný přístup do databází, při nefunkčnosti nebo odstranění systému však nedojde k odstranění *Databáze*. *Systém* je dále ve vztahu s třídou *Modul*. Tento vztah je kompozice. Pokud *Systém* zanikne, zanikne i třída *Modul* a všechny její podtřídy.

Účet je třída, která umožňuje uživateli přihlášení do systému nebo systémů. Pokud by uživatel neměl aktivní účet, nemohl by obsluhovat systém, protože by se neměl kam přihlásit. *Účet* vzniká registrací uživatele. Třída *Účet* obsahuje atributy o počtu systémů, kam se lze účtem přihlásit, přihlašovací údaje k účtu a nastavené oprávnění k jednotlivým systémům.

Webový prohlížeč je třída využívaná třídou *Uživatel* a vykreslující třídu *Systém*. Tato třída vlastní atributy o názvu prohlížeče, o jeho současné verzi a o podporovaném operačním

systemu. Třída *Webový prohlížeč* obstarává hlavně zobrazení uživatelského prostředí pro třídu *Uživatel*. Samotný systém bez prezentačního prostředí by uživateli neměl co nabídnout a zároveň uživatel by neměl žádné možnosti k ovládní systému.

Modul (abstraktní nadtřída) je nadtřída k ostatním modulům, které využívá *Systém*. Třída modul je součástí třídy *Systém*, bez které nemůže existovat právě díky vazbě kompozice. Třída obsahuje atributy ID modulu, název modulu a atribut k právě propojené databázi, jejíž pohled je zobrazován systémem. Moduly jsou v současné navrhované verzi systému právě čtyři, avšak je zde ponechána možnost na rozšíření počtu těchto modulů.

Komunikační modul (podtřída nadtříd *Modul*) je třída komunikující s *uživatелеm* na podněty systému. Podněty systému si uživatel nastaví sám v nastavení. Mezi podněty mohou patřit: nízký stav zboží na skladě, nalezení položky zadané do modulu hlídací pes nebo vyřízení faktury. Komunikační modul využívá atribut ID uživatele k načtení údajů o telefonním čísle uživatele a jeho emailové adrese. Třída může uživatele informovat i prostřednictvím aplikace v lištičce.

Fakturační modul (podtřída nadtříd *Modul*) je třída využívající přístup k datům *Mé faktury* a *Mé transakce*. Jedná se o třídu, která umožňuje vytvářet pohledy dle potřeb uživatele. Dále třída umožňuje spolupráci s třídou *Ceny pro fakturaci*. Tato spolupráce vychází z dat, které *Fakturační modul* přebírá a na jejichž základě vytváří nebo upravuje faktury. Dále tato třída umožňuje nahrání externích souborů do systému. Díky využití dat z databází může uživatel snadno předávat a vytvářet hlavičky faktur dle již přijatých či vystavených faktur.

Skladový modul (podtřída nadtříd *Modul*) je abstraktní třída, jejíž součástí jsou třídy *Ceny pro fakturaci*, *Historie pohybu zboží* a *Aktuální stav skladu*. Tato třída se zmíněnými třídami je ve vztahu kompozice, proto tyto třídy nemohou existovat bez třídy *Skladový modul*. Skladový modul nese informaci o právě aktivním modulu, který se zobrazuje uživateli. Skladový modul vytváří pohledy z dat, která využívají jeho podtřídy.

Modul Hlídací pes (podtřída nadtříd *Modul*) je třída využívající dvě datové základny, a to *Moje hlídání* a *XML feed*. Z *Moje Hlídání* je vytvořený pohled totožný s obsahem samotné databáze, proto tato třída nemusí vytvářet složitý pohled. Třída využívá třídu *XML*

feed pro prohledávání webu díky čemuž nalezne nejnižší ceny zvolených produktů. Hlídaní může být aktivní nebo neaktivní.

Ceny pro fakturaci (součást třídy *Skladový modul*) je třída, které je uživatelem využívána především pro nastavení cen produktů. Nově nastavené ceny jsou přebírány do *Fakturačního modulu*. Modul je důležitý pro správu příjmů uživatele nastavováním výstupních cen.

Historie pohybu zboží (součást třídy *Skladový modul*) je třída, která zobrazuje všechny položky z dat *Můj sklad*. Vypsání položky nemá všechny své atributy pro snadnější zobrazení. Třída vytváří grafické zobrazení pohybu množství jednotlivých produktů na skladě. Zobrazení je v podobě grafu u každé položky. Třída zobrazuje i položky, které již na skladě nejsou, ale byly někdy uživatelem naskladněny a vyskladněny.

Aktuální stav skladu (součást třídy *Skladový modul*) je třída, která zobrazuje současný stav skladu. Třída zobrazuje pouze naskladněné položky s většinou atributů. Nezobrazené atributy jsou převážně informace o jednotlivých objednávkách položek a jejich vyskladnění. Třída využívá k svým náhledům data ze tříd *Můj sklad* a *XML feed*.

Data (abstraktní třída) je nadtřída ke všem datům, která využívána *Systém*. Třída je součástí třídy *Systém*. Na rozdíl od třídy *Modul* může třída *Data* existovat i v případě aktuální nefunkčnosti *Systému*. Třída *Data* obsahuje atributy: typ dat, normální forma dat, podporované vztahy v datech a ID databáze. V současné navrhované verzi systému je využíváno pět databází.

Můj sklad (podtřída nadtřídy *Data*) je třída obsahující informace o všech položkách, které byly někdy evidované ve skladu. *Můj sklad* obsahuje atribut počet produktu a nedostatkový produkt. Modul sklad poskytuje data k vytvoření pohledů jednotlivými moduly.

XML feed (podtřída nadtřídy *Data*) je třída obsahující soubory XML feed, které jsou k dispozici od srovnávačů zboží. V případě, že by nebyla možná dostupnost k této

databázi, by byla třída nahrazena automatickým robotem, který by využíval služeb srovnávačů zboží jako uživatel. Správná funkčnost tohoto robota by se odvíjela od zabezpečení serveru srovnávačů a jejich případných výjimek pro zmíněného automatického robota.

Mé faktury (podtřída nadtřídy *Data*) je třída obsahující soubory ve formátu XML. Soubory obsahují jednotlivé faktury, které jsou pomocí třídy *Fakturační modul* zobrazeny do uživatelsky přívětivého designu. Třída obsahuje atributy o počtu proplacených faktur a o datu nejpozdějšího zaplacení nejaktuálnější faktury v seznamu. Tento atribut dopomáhá uživateli nezapomenout zaplatit přijatou fakturu.

Mé transakce (podtřída nadtřídy *Data*) je třída uchovávající data o všech příchozích i odchozích platbách a přesunech položek. Tato třída je využívána pouze třídou *Fakturační modul*. Třída *Mé transakce* obsahuje atributy o počtu transakcí a o datumu poslední změny v datech.

Moje hlídání (podtřída nadtřídy *Data*) je třída uchovávající data o položkách, které si uživatel zadá. Tyto položky jsou hlídány třídou *Modul Hlídací pes*, která provádí periodické prohledávání *XML feedu* pro nalezení položek za požadovanou cenu a v požadovaném množství. V této třídě jsou uvedeny právě zmíněné parametry u jednotlivých položek. Třída vlastní atributy počet hlídání a atribut určující aktivní nebo neaktivní využití této třídy.

Záznam je třída uchovávající data o jednotlivých záznamech. Záznamem je myšlen jeden druh zboží. Třída neobsahuje žádné metody. Třída obsahuje velké množství atributů. Záznam a jeho atributy jsou využívány ve třech skladových modulech.

Faktura představuje třídu, která znázorňuje faktury obsažené v databázi faktur. Faktura stejně jako *Záznam* neobsahuje žádné metody, obsahuje však mnoho atributů, které tvoří samotnou fakturu. Faktura je využívána třídou *Mé faktury*. Třída *Mé faktury* upravuje atributy faktury nebo vytváří celou entitu třídy *Faktura*.

Příloha B - Sekvenční modely

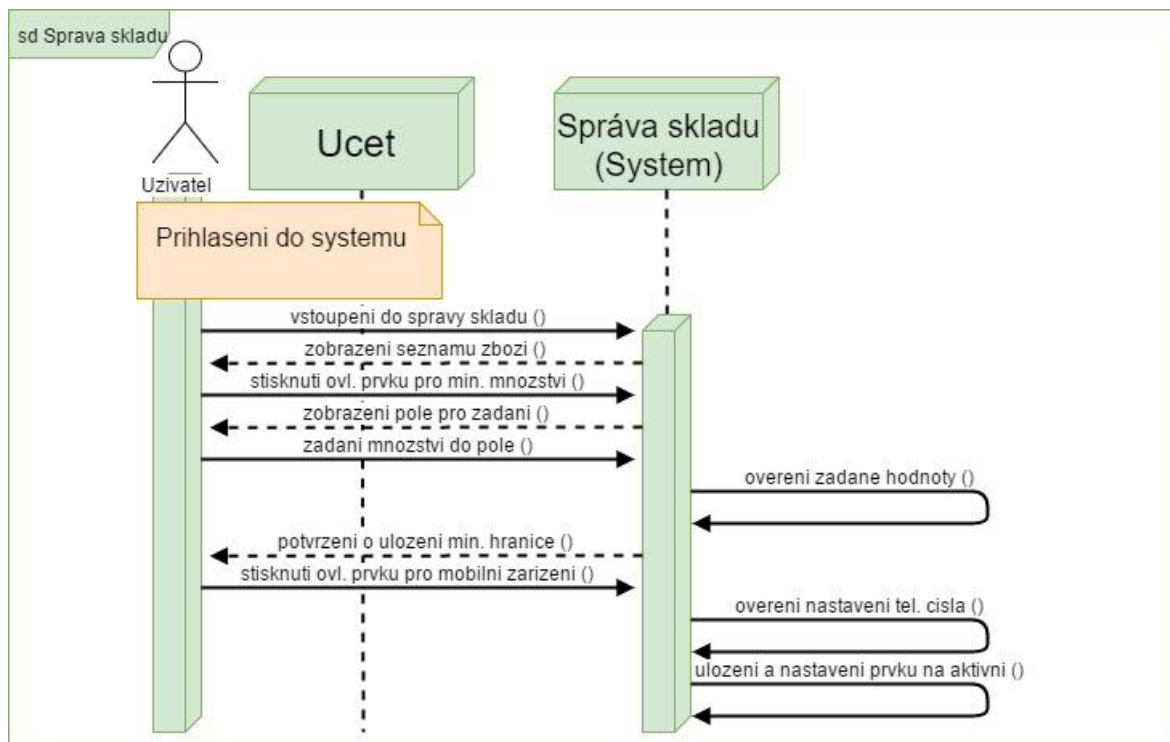
Sekvenční model: Správa skladu

Sekvenční model Správa skladu popisuje scénář nastavení minimální hodnoty množství zboží F-konektor. Dále je popsáno nastavení upozornění o nedostatku tohoto zboží.

Případ užití: Správa skladu	
ID	P03
Stručný popis	Uživatel nastavuje minimální hranici množství zboží a určuje, jak má být informován o poklesu zboží pod nastavenou hranici.
Hlavní aktér	Uživatel
Vstupní podmínky	Uživatelův účet má oprávnění pracovat s modulem správy skladu.
Hlavní scénář	<ol style="list-style-type: none">1. Uživatel vstoupí do modulu <i>Správa skladu</i> ovládacím prvkem.2. Systém zobrazí seznam zboží a nabízené možnosti pro nastavení zboží.3. Uživatel stiskne ovládací prvek pro výběr minimální hodnoty zboží F-konektor.4. Systém zobrazí pole pro zadání požadovaného množství zboží.5. Uživatel zadá požadované množství zboží a potvrdí ovládacím prvkem potvrdit.6. Systém ověří, zdali se jedná o celočíselnou hodnotu a uloží uživatelskou volbu.7. Systém uživateli potvrdí uložení hodnoty.8. Uživatel stiskne ovládací prvek mobilního zařízení pro informování uživatele při poklesu množství zboží.9. Systém ověří, že má uživatel nastavené telefonní číslo.10. Systém uloží uživatelskou volbu a změní barvu ovládacího prvku <i>tel.</i> na zelenou.
Výstupní podmínky	Minimální množství zboží na skladě bylo úspěšně uloženo. Při poklesu množství zboží pod nastavenou hranici bude uživatel informován pomocí mobilního zařízení.
Alternativní scénář	Nastavení minimální hodnoty zboží nebylo provedeno z důvodu nepodporovaného znaku zapsaného do pole. Uživatel nemá uloženo telefonní číslo v systému, proto je vyzván systémem k jeho zadání pro aktivaci upozornění pomocí mobilního zařízení. Uživatel si vybere jinou formu upozornění na snížení množství zboží pod nastavenou hranici.

Tabulka 17 - Scénář – Správa skladu (Zdroj: Vlastní tvorba)

Uživatel použitím *Správy skladu* vytváří pohled na databázi položek. Náhled umožňuje upravovat atributy jednotlivých položek a přidávat atributy systémem. Pro lepší představu o *Správě skladu* je tento případ užití zobrazen logickým návrhem.



Obrázek 32 - Sekvenční diagram – *Správa skladu* (Zdroj: Vlastní tvorba)

Vyobrazený sekvenční diagram mimo jiné zobrazuje časté ověřování prováděné systémem. Toto ověřování je nutné z hlediska dodržení datových tipů atributů. Časté ověřování podporuje integritní omezení databáze, kterou systém využívá pro tvorbu náhledů.

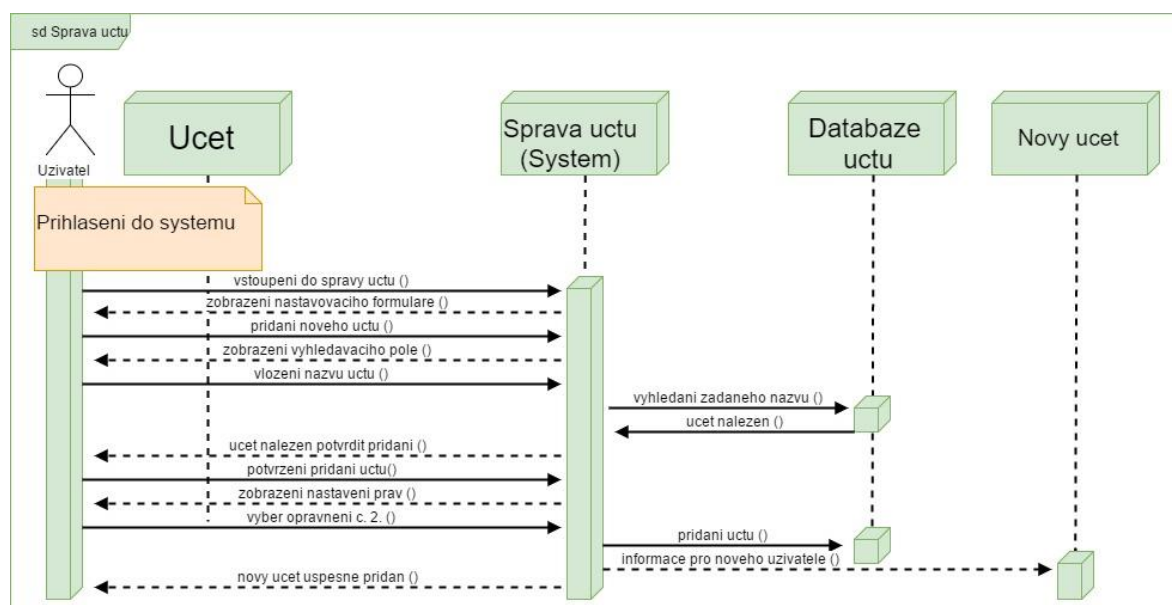
Sekvenční model: *Správa účtů*

Sekvenční model *správa účtů* popisuje scénář přidání dalšího účtu k systému. Při přidání účtu je nezbytné provést přiřazení oprávnění. Bez přiřazení oprávnění neproběhne přidání účtu a systém tuto operaci neuloží.

Případ užití: Správa účtů	
ID	P04
Stručný popis	Uživatel přiřazuje nový účet, k již existujícímu systému.
Hlavní aktér	Uživatel (přihlášený z účtu s plným oprávněním)
Vstupní podmínky	Uživatel má vytvořený administrátorský účet. Uživatelský účet je připojen k alespoň jednomu systému.
Hlavní scénář	<ol style="list-style-type: none"> 1. Uživatel vstoupí do správy účtů. 2. Systém zobrazí nastavovací formulář pro správu účtů. 3. Uživatel stiskne volbu pro přidání nového existujícího účtu. 4. Systém zobrazí vyhledávací pole pro zadání názvu přihlašovacího účtu. (Název účtu odpovídá přihlašovacímu jménu) 5. Uživatel zadá existující účet, který není doposud přidán k současně používanému systému a potvrdí vyhledávání. 6. Systém vyhledá účet v databázi účtů. 7. Systém po nalezení shody názvu zobrazí uživateli vyhledávaný účet a umožní jeho přidání. 8. Uživatel potvrdí přidání účtu stiskem ovládacího prvku <i>přidat účet</i>. 9. Systém zobrazí uživateli nastavení práv pro nově přidávaný účet. 10. Uživatel vybere přednastavené oprávnění č. 2. a potvrdí výběr. 11. Systém přiřadí účet do databáze účtů přiřazených k aktuálnímu systému. 12. Systém odešle informaci o přidání účtu do systému jeho uživateli. 13. Systém informuje uživatele o úspěšném přidání nového účtu.
Výstupní podmínky	Uživatel úspěšně přidal nový účet k systému a nastavil práva na úroveň číslo 2.
Alternativní scénář	<p>Vyhledávaný účet nebyl nalezen v databázi účtů.</p> <p>Uživatel nenastavil úroveň oprávnění.</p> <p>Uživatelský účet nebyl přiřazen z důvodu maximální kapacity účtů přiřazených k systému.</p>

Tabulka 18 - Scénář – Správa účtů (Zdroj: Vlastní tvorba)

System pro přidání nového účtu vyžaduje nastavení oprávnění. System má omezený počet účtů, které mohou být připojeny k jednomu systému. Při vyhledávání účtu systém zobrazí i možné alternativy, které se názvem podobají vyhledávanému názvu. Zobrazení podobných výsledků hledání snižuje dobu nalezení správného účtu při překlepech uživatele nebo nepřesné informaci o názvu nového účtu.



Obrázek 33 - Sekvenční diagram – Správa účtů (Zdroj: Vlastní tvorba)

Sekvenční diagram správy účtů zobrazuje vyhledávání v databázi účtů. Je zde zobrazeno, že vyhledávání probíhá bez komunikace se samotným vyhledávaným účtem. Nový účet je kontaktován až po jeho přidání do databáze systému. Uživateli nového účtu se po přihlášení zobrazí, že jeho účet byl přidán do jiného systému. Pokud uživatel účtu nechce být přidán do tohoto systému, může zvolit v nastavení ignorovat přiřazení. Pokud uživatel nechce být přidáván k žádnému systému, může v nastavení vypnout možnost přidávání jeho účtu k existujícím systémům.

Sekvenční model: Hlídací pes

Sekvenční model hlídací pes popisuje scénář přidání nové položky do seznamu vyhledávaných produktů. Při přidávání nového zboží je nutné uvést adresu na jeden internetový obchod, který daný produkt nabízí. Tento produkt musí být správně zařazen do produktových katalogů srovnávačů zboží. Pokud produkt není dobře zařazen nebo je

zadaná url adresa na e-shop, který neposkytuje aktuální XML feed srovnávačům zboží, nemůže hlídací pes pracovat správně.

Případ užití: Hlídací pes	
ID	P05
Stručný popis	Uživatel přidává produkt do modulu <i>hlídací pes</i> .
Hlavní aktér	Uživatel
Vstupní podmínky	Uživatelův účet má oprávnění pracovat s modulem Hlídací pes.
Hlavní scénář	<ol style="list-style-type: none"> 1. Uživatel vstoupí do <i>Hlídací pes</i>. 2. Systém zobrazí uživateli rozhraní správy <i>Hlídací pes</i>. 3. Uživatel stiskne do pole pro zadání nové položky. 4. Systém umožní editování položky. 5. Uživatel zadá název položky, počet kusů, nastaví maximální přípustnou cenu, periodu srovnávání a stiskne potvrdit. 6. Systém zobrazí pole pro zadání url adresy zboží nebo kódové označení produktu dle databáze XML feed. 7. Uživatel zkopíruje url adresu produktu z internetového obchodu a potvrdí nákup i z jiných internetových obchodů. 8. Systém uloží adresu a možnosti prohledávání. 9. Systém provede prohledání XML feedu. 10. Systém načte nejnižší cenu a e-shop prodávající zadaný produkt. 11. Systém uživateli zobrazí současnou nejnižší cenu produktu uvedenou v databázi XML feed.
Výstupní podmínky	Uživatel úspěšně přidal nový produkt do modulu <i>Hlídací pes</i> . Systém dle nastavené periody prohledává XML feed a zobrazuje nejvhodnější výsledky.
Alternativní scénář	Zboží nebylo nalezeno v XML feedu. Uživatel využije pro zadání hledání zkopírovaný kód produktu místo url adresy. Systém nalezne nižší cenu produktu, než kterou zadal uživatel.

Tabulka 19 - Sekvenční diagram – Hlídací pes (Zdroj: Vlastní tvorba)

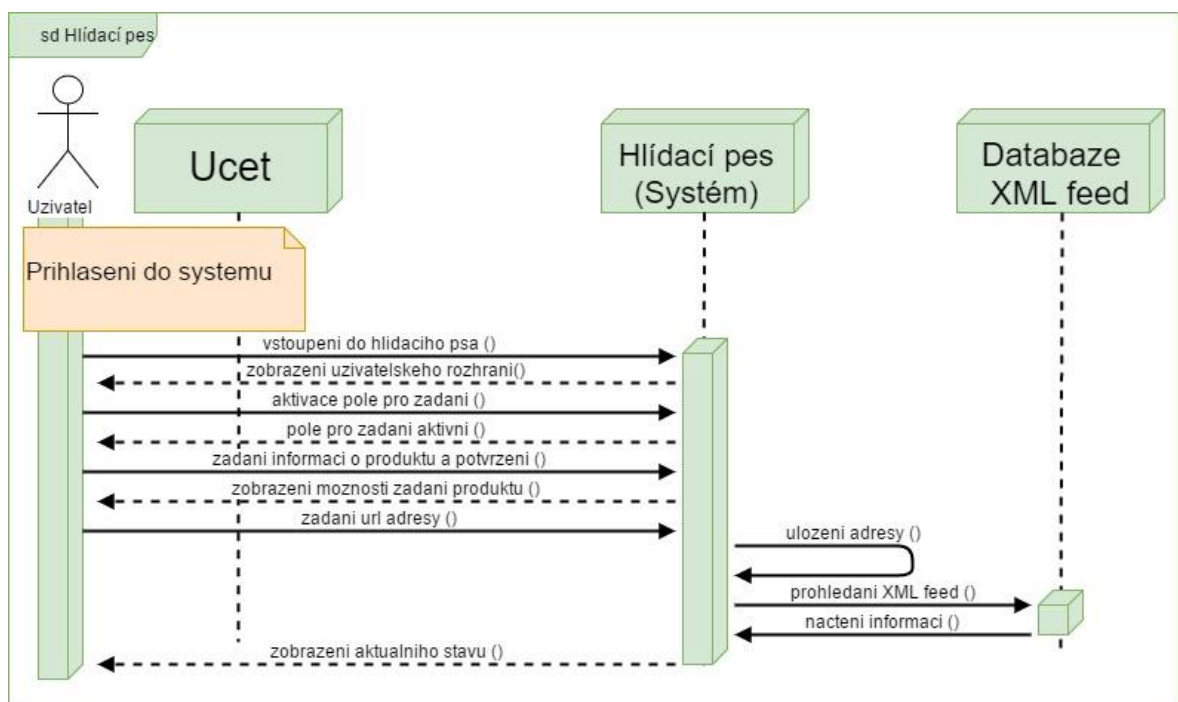
Při práci v modulu *Hlídací pes* je možné nastavit alternativní kontakt na uživatele. Uživatel může nastavit telefonní číslo nebo emailovou adresu. Uživatel může nastavit oznámení už

při poklesu ceny nebo jen při dostatečném množství nabízeného produktu. Pro jednoznačné zadávání produktu je vhodnější použít zadání produktu přes identifikační kód produktu. Identifikační kód každého produktu je uveden v databázi XML feed.

Systém nijak nezasahuje do databáze XML feed. Systém využívá pouze funkce *Read* a porovnává zadaná data od uživatele s daty v databázi. Komunikace 9,10 a 11 se opakují dle uživatelského nastavení. Opakování je aktivní i v případě odhlášení uživatele ze systému. Komunikace s databází XML feed je ukončena až po uživatelské změně nastavení nebo koupi daného produktu.

Vyobrazenými sekvenčními diagramy je přiblížena komunikace aktora uživatel se systémem. Komunikace je zobrazena ve všech hlavních případech užití, které navrhovaný systém umožňuje.

Samotná komunikace v případě posledního sekvenčního diagramu je závislá na implementaci celého systému a na řešení komunikace se srovnávací zboží. Tento problém byl popisován již v kapitole zabývající se diagramem případů užití.



Obrázek 34 - Sekvenční diagram – Hlídací pes (Zdroj: Vlastní tvorba)

Příloha C - Kvalitativní šetření

Dobrý den,

toto kvalitativní šetření je určeno pro osoby samostatně výdělečně činné (*dále jen OSVČ*) a malé podniky, které k výkonu své činnosti využívají nákup a prodej zboží. Kvalitativní šetření je podkladem pro zpracování diplomové práce. Cílem tohoto kvalitativního šetření je seznámit Vás s navrhovaným systémem a na základě Vašeho úsudku zjistit Váš názor na navrhovaný systém.

Toto kvalitativní šetření je vytvořeno pro vyplnění dokumentu v elektronické podobě. Proto ke svým odpovědím využijte prosím prostor pod zadanou otázkou. Všechny otázky jsou pro lepší přehlednost zvýrazněny zelenou barvou. Veškeré odpovědi jsou anonymní a nebudou nikde zveřejňovány.

1. část

Úvodní otázky

Nejprve Vás poprosím o vyplnění otázek, které se týkají výkonu Vaší pracovní činnosti.

1) **Jaká je Vaše pracovní náplň (několika slovy shrňte Vaši pracovní činnost)?**

Klikněte nebo klepněte sem a zadejte text.

2) **Kolik pracovníků tvoří Váš podnik?**

Klikněte nebo klepněte sem a zadejte text.

a) **Pokud jste OSVČ, pracujete sám nebo s někým spolupracujete?**

Klikněte nebo klepněte sem a zadejte text.

3) **Využíváte k Vaší pracovní činnosti nákup zboží a jeho prodej?**

Klikněte nebo klepněte sem a zadejte text.

a) Pokud ano, popište jak u Vás proces nákupu a prodeje probíhá?

Klikněte nebo klepněte sem a zadejte text.

4) Využíváte internetové obchody k nákupu zboží, které je určeno k Vaší pracovní činnosti?

Klikněte nebo klepněte sem a zadejte text.

a. Pokud ano, napište, co je pro Vás směrodatné při výběru internetového obchodu?

Klikněte nebo klepněte sem a zadejte text.

b. Uveďte, co je pro Vás směrodatné při výběru konkrétního produktu?

Klikněte nebo klepněte sem a zadejte text.

5) Používáte ke své pracovní činnosti nějaký informační, skladový nebo účetní systém?

Klikněte nebo klepněte sem a zadejte text.

a. Pokud ano, napište název systému a krátce popište k čemu ho využíváte?

Klikněte nebo klepněte sem a zadejte text.

6) Jakým způsobem byla Vaše pracovní činnost ovlivněna zavedením elektronické evidence tržeb (EET)?

Klikněte nebo klepněte sem a zadejte text.

2. část

Seznámení s navrhovaným systémem

Navrhovaný systém představuje informační systém, který je zobrazován uživateli pomocí webového prohlížeče. Navrhovaný systém pomáhá uživateli se správou skladu, tvorbou a úpravou faktur, online objednááním zboží a k potencionální celkové úspoře času a peněz.

Systém se skládá z několika podsystémů:

- 1) **Skladový modul:** zobrazuje uživateli zboží na skladě, umožňuje hlídání minimálního množství zboží, umožňuje vytvářet ceny zboží k prodeji a v neposlední řadě zobrazuje historii pohybu zboží na skladě pomocí grafického zobrazení.
- 2) **Modul pro správu faktur:** umožňuje uživateli vytvářet faktury, upravovat šablony na tvorbu faktur, spravovat příchozí a odchozí faktury, převedení zboží z příchozí faktury rovnou do skladových zásob nebo umožňuje prodej zboží za hotové přímo na místě.
- 3) **Modul hlídací pes:** nabízí uživateli možnost sledování určitého produktu na e-shopech. Modul hlídá cenu a dostupnost nastavených produktů. Při poklesu ceny informuje uživatele o aktuální ceně. Díky využití dat z databáze XML feed je schopen porovnávat zboží stejně jako Heureka.cz a Zboží.cz.

A jak to celé funguje?

- 1) Vytvoříte si svůj uživatelský účet.
- 2) Po úspěšném přihlášení k uživatelskému účtu:
 - a) Můžete založit vlastní systém, kterému budete správcem.
 - b) Nebo můžete požádat o přidání vašeho účtu k již existujícímu systému.

(Pojmem *Systém* je myšlen jeden podnik, nebo případně OSVČ.
K jednomu systému může být přiřazeno více uživatelských účtů.)
- 3) Ke svému systému můžete přizvat i jiné uživatelské účty, kterým nastavíte požadované oprávnění (nastavení oprávnění znamená, že další účet například nemá oprávnění vstoupit do *Modulu pro správu faktur*).
- 4) Nyní můžete začít využívat všech možností a funkcí, které systém nabízí spolu s podsystémy.

Pro lepší představu, jak by vypadal navrhovaný systém, je zde přiložen návrh logického rozvržení ovládacích prvků.

Logo		Název systému		Název účtu		odhlásit se		Lištička do prohlížeče					
Sklad		Faktury		Hlídací pes		Správa účtů		Nastavení					
Aktuální stav skladu		Sklad - ceny pro fakturaci			Historie pohybu zboží								
pol.	Název zboží	Označení	Množství na skladě	Cena za jednotku	sazba DPH	Cena s DPH	Celkem cena s DPH	Minimální zásoba	Upozornění o nedostatku			Porovnávač zboží	
1.	UTP kabel cat6	859222000158	800m	7,28,- Kč	21 %	8,82,- Kč	7056,- Kč	150	N	tel.	email	apl.	akt.
2.	F-konektor 6,8	KONZIZKS01	25	5,80,- Kč	21 %	7,- Kč	175,- Kč	10	N	tel.	email	apl.	akt.
3.	F-konektor 5	KONZIZKS02	10	5,80,- Kč	21 %	7,- Kč	70,- Kč	---	N	tel.	email	apl.	akt.
	anténí koax kabel 6,8mm	xkvx100	250m	10,7,- Kč	21 %	12,9,- Kč	3225,- Kč	150	N	tel.	email	apl.	neakt.
5.	koaxial RG6 Amiko	889129	390m	3,3,- Kč	21 %	3,99,- Kč	1556,1,- Kč	150	N	tel.	email	apl.	neakt.
6.													
7.													
8.													
Celková hodnota zboží ve skladu						12082,1,- Kč		Nastavení upozornění			Nastavení priorit porovnávače		

Navrhovaný systém spojuje několik již existujících služeb:

- Základní služba je **skladová evidence**, která je koncipována pro menší podniky nebo OSVČ. Proto není požadována složitá evidence využívající například skladové karty apod.
- Další služba je tvorba, úprava a manipulace s **fakturami**. Tato služba dokáže nahradit určité software zabývající se účetnictvím nebo může převést již existující účetnictví z excelových tabulek do webového prostředí navrhovaného systému.
- Poslední výraznou službou je propojení systému na databázi **srovnávačů zboží**. Tímto propojením je možné dosáhnout hlídání častých nákupů za nízké ceny. Hlídání je výrazně rozšířené o možnosti nastavení priorit hledání, zadávání benefitů pro zákazníka atd.

3. část

Závěrečné otázky

Nyní Vás poprosím o vyplnění otázek, které se týkají vašeho pochopení a názoru na navrhovaný informační systém.

7) Jaký je Váš názor na srozumitelnost vysvětlení navrhovaného systému?

Klikněte nebo klepněte sem a zadejte text.

8) Jaký je Váš názor na použitelnost funkcí systému k vaší pracovní činnosti?

Klikněte nebo klepněte sem a zadejte text.

9) Napište a odůvodněte, jaké funkce informačního systému v navrhovaném systému na první pohled postrádáte.

Klikněte nebo klepněte sem a zadejte text.

10) Jaký je Váš názor na napojení modulu pro správu faktur s elektronickou evidencí tržeb (EET)?

Klikněte nebo klepněte sem a zadejte text.

11) Jaký je váš názor na možnost tvorby a editace faktur v navrhovaném systému?

Klikněte nebo klepněte sem a zadejte text.

12) Jaký je Váš názor na absenci skladových karet v navrhovaném systému?

Klikněte nebo klepněte sem a zadejte text.

13) Myslíte, že by navrhovaný systém našel uplatnění ve Vaší pracovní činnosti?

Klikněte nebo klepněte sem a zadejte text.

14) Jaký je Váš názor na instalaci navrhovaného systému v případě jeho realizace?

Klikněte nebo klepněte sem a zadejte text.