Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství

Diplomová práce

Návrh informačního systému pro sklad

Bc. Jan Mládek

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Jan Mládek

Systémové inženýrství a informatika
Informatika

Název práce

Návrh informačního systému pro sklad

Název anglicky

Design of information system for warehouse

Cíle práce

Cílem práce je návrh informačního systému pomocí jazyka UML. Návrh vychází z praktických zkušeností v nejmenované firmě a je zaměřen na zjednodušení manipulace se zbožím. Informační systém je navržen na základě funčních a nefunkčních požadavků pro manipulaci se zbožím tak, aby umožňoval uživatelům intuitivní a jednoduchou práci.

Metodika

Metodika diplomové práce je založena na studiu a analýze odborných pramenů pro tvorbu informačních systémů v prostředí UML se zaměřením na sběr dat a analýzu funkčních a nefunkčních požadavků. Praktická část je zaměřena na návrh informačního systému umožňujícího snadné a přehledné prostředí pro uživatele v prostředí skladovacích prostor. Na základě získaných teoretických a praktických poznatků budou formulovány závěry práce.

Doporučený rozsah práce

50 – 60

Klíčová slova

UML, diagramy, USE CASE, Informační systém, statické modelování, dynamické modelování, funkční a nefunkční požadavky

---

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.

BOOCH, G. – RUMBAUGH, J. – JACOBSON, I. The unified modeling language user guide. Addison-Wesley Professional, 2005. ISBN 978-0321267979

KANISOVÁ, H. – MÜLLER, M. UML srozumitelně. Brno: Computer Press, 2004. ISBN 978-80-251-1083-6

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

2020/21 ZS – PEF (únor 2021)

Vedoucí práce

doc. RNDr. Dana Klimešová, CSc.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 19. 2. 2020

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 19. 2. 2020

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 31. 03. 2021

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Návrh informačního systému pro sklad" jsem vypracoval(a) 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(ka) uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31.3.2021 _____

Poděkování

Rád(a) bych touto cestou poděkoval(a) doc. RNDr. Daně Klimešové, CSc. za její odborné vedení mé diplomové práce, cenné rady a především ochotu a čas, které mi věnovala v průběhu zpracování této závěrečné práce.

Návrh informačního systému pro sklad

Abstrakt

Tato diplomová práce se zabývá návrhem informačního systému pro správu skladu. Cílem práce bylo navrhnout informační systém, který by zohlednil nároky kladené na způsob skladování zboží na skladě a zároveň byl přehledný a jednoduchý při běžném používání. Teoretická část popisuje typy informačních systému dle jejich využití a přístupy k tvorbě těchto systémů. Praktická část se zaměřuje na návrh systému, který je definován na základě funkčních a nefunkčních požadavků. Na základě těchto požadavků byl pomocí jazyka UML vypracován model tříd, stavový model a model interakcí, které podrobněji popisují procesy navrhovaného systému. Na závěr této práce je návrh doplněn o přibližný vizuální vzhled.

Klíčová slova: UML, diagramy, Use Case, Informační systém, statické modelování, dynamické modelování, funkční a nefunkční požadavky, model tříd, stavový model, model interakcí

Design of information system for warehouse

Abstract

This diploma thesis deals with the design of information system for warehouse management. The goal was to design information system that would take into account different ways on how to store goods in the warehouse and still be well-arranged and simple during its use. The theoretical part describes types of information systems according to their use and approaches to create these systems. Practical part focus on design of system, which is defined on functional and non-functional requirements. Based on these requirements with the help of UML language, there was created class model, state model and interaction model, which in more detail describe processes of proposed system. The end of this thesis, system is supplemented with approximate visual concept.

Keywords: UML, diagrams, Use Case, information system, static modelling, dynamic modelling, functional and non-functional requirements, class model, state model, interaction model

Obsah

Seznam obrázků

Seznam tabulek

Seznam použitých zkratek

UML - Unified Modeling Language

IS - Informační systém

PIS - Podnikový Informační systém

BI - Business Inteligence

ERP - Enterprise resource planning

SCM - Supply chain management

CRM - Customer relationship management

XP - Extrémní programování

ERD - Entity relationship diagram

DFD - Data flow diagram

FSD - Function state diagram

STD - State transition diagram

DD - Data dictionary

OOP - Objektově orientovaný přístup

CASE - Computer aided software engineering

FIFO – First in First out

1 Úvod

Informační systémy se staly nedílnou součástí dnešního světa. Téměř každá firma dnes nějaký systém používá a s tímto zmiňovaným systémem denně pracují desítky, někdy až stovky uživatelů, ať už v podobě zaměstnanců, zákazníků či majitelů konkrétních firem. Každý systém je svým způsobem unikátní a vyžaduje pro své uživatele různé funkce a oprávnění pro jeho úspěšné fungování.

Při tvorbě systémů se řada tvůrců potýká s různými problémy při jeho návrhu či při případné implementaci. Nesourodosti vznikají především chybnou komunikací mezi zákazníkem a zadavatelem. Problémy se týkají především špatně či nepřesně stanovených požadavků na systém. Výsledkem výše zmíněných nejasností může být systém, který se zdá být z hlediska stanovených požadavků kompletní, ale budoucí uživatelé, kteří budou se systémem pracovat, mohou později objevit jeho nedostatky, jenž jim brání v rychlém a efektivním provedení svých pracovních úkonů.

K překlenutí těchto překážek slouží Unified modeling language. UML je grafický jazyk pro vizualizaci, návrh a dokumentaci programových systémů. Slouží ke zpracování objektově orientovanému návrhu budoucího systému. UML nabízí grafické a slovní výstupy, které jsou snadno pochopitelné jak pro budoucího programátora či tvůrce systému, tak pro zadavatele, kteří nemají s programováním a tvorbou systémů žádné praktické zkušenosti. Hlavní výhodou je, že při správném použití metodik UML je zákazník schopen přesně určit veškeré požadavky na systém a naopak programátor je schopen přesně dodržet požadavky stanovené zadavatelem.

Při dodržení postupů je výsledkem správně pochopený i naprogramovaný systém, který byl dodán včas, nepotřeboval zdroje ani finance navíc, a funguje dle všech stanovených požadavků. Díky těmto vlastnostem se UML i ostatní modelovací jazyky stali nezbytností při návrhu a tvorbě jakýchkoliv systémů.

Důvodem návrhu informačního systému pro sklad jsou autorovy osobní zkušenosti. Při studiu problematiky logistiky skladu a práci v prostorech skladu bylo nalezeno mnoho procesů, které nebyly dostatečně ošetřeny a docházelo tak ke zbytečným chybám a nepřesnostem mezi informacemi uvedenými v původním systému a skutečností.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem práce je návrh informačního systému pro správu skladovacích prostorů. Návrh systému je vytvořen pomocí jazyka UML a je zaměřen na zjednodušení procesů, které souvisí s příjmem zboží na sklad, jeho uskladněním, kontrolou stavu a množství zboží na skladě a konečným prodejem zákazníkovi. Systém je orientován pro práci s příchozím zbožím, aby se předcházelo chybnému uskladňování zboží. Dále předchází ztrátě zboží způsobenou jeho exspirací a umožňuje efektivně sledovat jakýkoliv pohyb či manipulaci se zbožím.

2.2 Metodika

Metodika diplomové práce je založena na studiu a analýze odborných pramenů. Na základě analýzy procesů při práci v nejmenované firmě byla shromážděna data o nedostatcích v systému, které výrazně omezovaly jeho použití. Na základě dat byly formulovány funkční a nefunkční požadavky kladené na systém, které v původním řešení systému nebyly zohledněny. Systém je navržen v jazyce UML pomocí modelu tříd, stavového modelu a modelu interakcí. V závěru práce je vyhodnocen a posouzen návrh systému a jeho přínos k řešení problematiky včetně grafického návrhu.

3 Teoretická východiska

3.1 Informační systémy

Informační systémy se staly nedílnou součástí dnešního světa počítačů, avšak když pohlédneme do minulosti, informační systémy již existovaly. Mohlo jít například o kartotéky u lékařů či v bankách k uchovávání informací o pacientech, nebo klientech a informací s nimi spojených.

Pro popsání informačních systémů v dnešním pojetí je vhodné použít jejich definici. Pohled na systémy a jejich definice jsou u všech autorů velice podobné jako například u dvou českých autorů:

*„**Podnikový informační systém** vytvářejí lidé, kteří prostřednictvím dostupných technologických prostředků a stanovené metodologie zpracovávají podniková data a vytvářejí z nich informační a znalostní bázi organizace, sloužící k řízení podnikových procesů, manažerského rozhodování a správě podnikové agendy.“* [1]

*„**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.“* [2]

Informační systém lze chápat jako soubor prvků, který je reprezentován subjekty (všichni uživatelé systému), technickými prostředky (hardware nezbytný pro chod systému jako servery a počítače) a metodami (software), které slouží k jeho fungování s využitím současných technologií. [3]

Samozřejmě každý kvalitní systém by měl splňovat jisté vlastnosti, jako například jaké informace uchovává, kde je analyzuje a jakou rychlostí je předává dalším procesům systému. Neméně důležité jsou také informace o konkurenci či světovém trhu, trendech, jenže se neustále mění a vyvíjí. Dále pak moduly pro zjednodušení a urychlení výroby. A v neposlední řadě také umožňovat rychlou komunikaci mezi zaměstnanci firmy, případně mezi jednotlivými odděleními společnosti, nebo komunikaci s okolním světem. [3]

Dělení informačních systémů je rozsáhlé. V historii byla vydána řada publikací snažící se klasifikovat jednotlivé typy systémů, avšak jakákoliv publikace byla vždy ovlivněna soudobými technologiemi (například literatura z 90. let již dnes není aktuální). Dalším faktorem ovlivňujícím kategorizaci IS je fakt, že jeden typ informačního systému nemusí patřit pouze do jedné kategorie, ale může patřit do více skupin najednou. Z tohoto důvodu se dnes systémy dělí do dvou skupin: Veřejné informační systémy a Podnikové informační systémy. [4]

3.1.1 Veřejné informační systémy

Tyto systémy uchovávají informace, které jsou dostupné široké veřejnosti či širší komunitě lidí. Jejich správu a provoz mají na starosti různé organizace nebo instituce. Kromě správy a provozu je musí též financovat například z veřejných rozpočtů, dobročinných příspěvků nebo z příjmů z reklam, ale také pomocí vlastních prostředků. Příkladem jsou webové IS, muzea, galerie nebo systém veřejné knihovny.

3.1.2 Podnikové informační systémy

Jde o infomační systém, který je využíván firmou pro své účely podnikání. V těchto systémech využívají organizace vlastní data, ke kterým má přístup pouze omezený počet uživatelů (zaměstnanců). Pojem podnikových informačních systémů je velice rozsáhlý, neboť může jít o jakýkoliv informační systém, který vylepšuje obchodní podnikové procesy a následně je integruje do struktury podniku. Což obvykle znamená náročné plánování a práce s velkými objemy dat, aby byl výsledný systém schopen podporovat procesy velkých a složitých podniků či organizací.

Podnikové informační systémy musí fungovat ve všech částech podniku i na všech jeho úrovních. Tyto systémy v podstatě poskytují technologickou platformu, která poskytuje organizacím prostředky k integraci a koordinaci všech svých firemních procesů za účelem jejich automatizace a optimalizování. Poskytují jediný centrální systém, který zajišťuje, že informace jsou sdíleny na všech hierarchických úrovních. Obecně mohou podnikové IS zvýšit obchodní produktivitu a zkrátit například vývojové nebo marketingové procesy. [5]

Podnikové IS jsou na rozdíl od svých předchůdců samoobslužné a dokáží se přizpůsobit podmínkám klíčovým pro firmu. Dále pak obsahují jak operační (část zpracovávající dané transakce), tak informační (datová úložiště, reporty). Hlavní předností PIS jsou podsystémy, které jsou zaváděné společně se systémem, a které spolu vzájemně spolupracují.

Tyto podsystémy jsou vybírány organizací na základě jejich preference a existuje jich celá řada. Každý z těchto podsystémů obsahuje široké množství parametrů a funkcí, jež se dají dále měnit dle odvětví firmy, ve kterém se organizace pohybuje. [6]

BI - Business Intelligence

Business Intelligence je termín označující analytické a vykazovací podnikové aplikace. Jedná se o prostředek sloužící pro práci s firemními daty. Business Intelligence označuje paletu softwarových aplikací využívaných k analýze syrových dat organizace a je velmi často využívána firmami k činění kvalifikovanějších rozhodnutí, snižování nákladů či hledání nových obchodních příležitostí. Dále slouží například k analyzování dat v minulosti, ale také simuluje a pomáhá předpovídat budoucí vývoj.

Mnoho firem využívá nepřeberné množství podnikových podsystémů (ERP, CRM, SCM viz níže) a je zahlcena daty z těchto různých zdrojů. A to je právě hlavním úkolem BI. Sjednotit tyto zdroje do jednoho uceleného a přehledného systému. Samozřejmě stále platí, že pro úspěšné zavedení Business Intelligence je nutné mít kvalitní datovou základnu, neboť BI pouze agreguje ostatní podsystémy s jejich aktuálními daty. Pokud jsou tyto data sama o sobě nepřehledná či nevypovídající, zavedením Business Intelligence se jejich váha nebo vliv nevylepší. [7][8]

ERP - Entreprpise resource planning

Jedná se o podsystém, jímž podnik integruje většinu oblastí své činnosti. ERP poskytuje sjednocený celopodnikový pohled na to, co se v jednotlivých divizích podniku odehrává. Též obsahuje databázi obsahující veškerá podniková data. Entreprpise resource planning obvykle pokrývá tyto okruhy:

- Finance – Jde například o správu závazků a pohledávek, řízení hotovosti, přehled o dlouhodobém majetku, ale také analýza ziskovosti a finanční konsolidace.
- Personalistika – Především pokrývá plánování pracovníků a s nimi spojené mzdové funkce, dále cestovní výlohy na pracovní cesty a také evidenci odpracovaných hodin.
- Výroba a logistika – Tento okruh spravuje především sklady a řízení zásob, nákup a příjem zboží. S tím související plánování výroby, kalkulace nákladů na výrobu a řízení jakosti a projektů. V neposlední řadě i hodnocení dodavatelů.
- Marketing a prodej – Poslední z okruhů má na starosti správu zakázek, jejich ceny a také řízení a plánování prodeje.

Aplikace ERP s sebou přináší jak výhody, tak i jistou kritiku. Mezi výhody nezpochybnitelně patří integrace podnikových procesů, které zaručuje přehlednější orientaci mezi jednotlivými daty, jenž mohou být sdílena mezi více odvětvími. Díky sdíleným datům je daleko jednodušší vytvářet přehledné reporty pro manažery, což může vést ke kvalifikovanějším a úspornějším rozhodnutím, které mohou dále zvyšovat celkovou výkonnost podniku. [9][10]

Co se týká kritiky, je neustále opakováno, že integrační proces je složitý pro uživatele na pochopení i na obsluhu, ale je třeba si uvědomit, že školení personálu je proces, jimž si projdou všichni uživatelé při zavádění nového systému. Také je třeba zmínit, že proces přijetí EPR neslouží k usnadnění práce uživatelům, nýbrž má prospět celému podniku. Další zmiňovanou nevýhodou bývá, že uvedení do provozu je až příliš náročné na čerpání časových i finančních prostředků. Je pravda, že integrační proces je náročný, protože než může být systém zcela zaveden, musí se připravit spousta databází a jejich údaje pro vložení do systému. Dále rozhodnout, které funkce budou využívány, a které naopak podstatné nejsou. To může mít vliv jak na čerpání časového fondu, tak na čerpání financí. [11]

SCM - Supply chain management

Jedná se o optimalizaci všech činností a systémů pro zabezpečení dodávky produktů a služeb od dodavatelů surovin přes jejich výrobu nebo vývoj, přes distribuční kanály až ke koncovému spotřebiteli. SCM obecně označuje prostředky či postupy sloužící ke koordinaci výrobků, materiálů, služeb, informací a financí, jenž pochází od dodavatelů přes zpracovatele nebo výrobce, dále prodejce, až k spotřebitelům.

Celý proces počíná zadáním objednávky, jejím posouzením a zpracováním. Pokračuje výrobou a dodaním zboží či služeb a končí zpětnou vazbou spotřebitele. Celý tento proces má pět fází:

- Plánování – První a nejdůležitější fáze pro správu a monitorování zdrojů, která zajistí, že řetězec spotřebovává přiměřené náklady, přičemž disponuje vysokou kvalitou.
- Získávání – Hlavní náplní je nalézt vhodné dodavatele a zároveň nastavit vztahy s nimi včetně smluvení dodacích a platebních podmínek.
- Výroba – Následuje proces transformující suroviny nebo komponenty na produkty nebo služby. Důležitá je kvalita výsledku.
- Dodání – Řídí dopravu zboží od výrobce k spotřebitelům. Bere v úvahu rozmístění skladových prostor, stavy zásob jednotlivých skladů i fakturaci za produkt.
- Vrácení – Jedná se výhradně o situace, kdy zákazníci reklamují zboží. Zahrnuje přijetí zboží včetně zajištění náhrady, příjem vratných obalů a návrat vadných dílů. Dále pak zisk zpětné vazby od spotřebitelů (spokojenost/nespokojenost). [12][13]

CRM - Customer relationship management

Jde o zákaznicky orientovaný management nebo podnikatelský přístup vyznačující se aktivní tvorbou a udržováním dlouhodobě prospěšných vztahů se zákazníky. Zákaznické vztahy jsou velice důležité a je třeba je udržovat.

CRM systémy obecně sbírají data o zákaznické spokojenosti či nespokojenosti. Důvodem jejich použití je obecně pochopit smýšlení a potřeby zákazníka. Toho je dosaženo zajištěním komunikace mezi spotřebiteli a organizací. [14]

3.2 Životní cyklus systému a techniky vývoje informačních systémů

Vývoj informačních systémů je náročná činnost. V první řadě je třeba si uvědomit, zda námi uvažovaný systém (ať už nový či přepracování stávajícího systému) bude užitečný, využívaný nebo zda bude v budoucnu ku prospěchu společnosti z hlediska přívětivosti uživatelům či zpřehlednění firemních dokumentů. Ale to je jen první z mnoha etap, které jsou při tvorbě IS nutné vzít v úvahu. K vytvoření systému je nutné rozložit celek procesů na menší procesy, které by se dali snadněji plánovat a řídit.

Zde však nehovoříme o rozdělení samotného systému, avšak o rozložení procesů vedoucí ke zdárnému dokončení systému a k jeho uvedení do provozu. V takovémto případě hovoříme o životním cyklu. Životní cyklus je popsán fázemi, jež popisují procesy od jeho zrození (prvotní myšlenka či úvaha nad zamýšleným systémem), až po jeho zánik (ukončení používání systému). V počtu těchto fází se mnoho autorů literatury rozchází. Je to způsobeno především díky rozmanitým metodám sloužícím k návrhu informačních systémů (viz níže). Přesto však můžeme obecně charakterizovat tyto fáze životního cyklu:[19][20]

- Předběžná analýza – Je základem celkového návrhu IS nebo úpravy stávajícího systému. Cílem této fáze je shromažďování požadavků (funkčnosti systému, rozsah, návratnost investice), analýza a odhad času a celkové náklady na výrobu systému (zdroje jako personál, hardware, software, licence). Tato fáze má za cíl pouze shromažďování předchozích hledisek nikoliv její analýza, to je úkolem další fáze.
- Analýza systému – Tato část má na starosti rozbor v návaznosti na předchozí fázi. Důležitým úkolem je odhalit všechny chyby ve struktuře dat a systému. Pokud nejsou odhaleny v této fázi, jejich oprava v dalších fázích je velice obtížná.
- Projektová studie – Výsledkem předchozích dvou částí je projektová studie. Jedná se o dokument, jež jsou podkladem pro smluvní podmínky mezi zadavatelem a tvůrcem. Obsahem tohoto dokumentu jsou dále základní informace o zadavatelské organizaci včetně uvedení zaměstnanců spolupracujících na vytvoření projektu, popis současného stavu struktury organizace a v neposlední řadě detailní návrh IS (fyzický datový model).

  Důležité je přitom uvést veškerá fakta dostatečně detailně, aby se předešlo případným nedorozuměním. Je to poslední dokument, který vedení firmy obdrží před konečným rozhodnutím o realizaci.

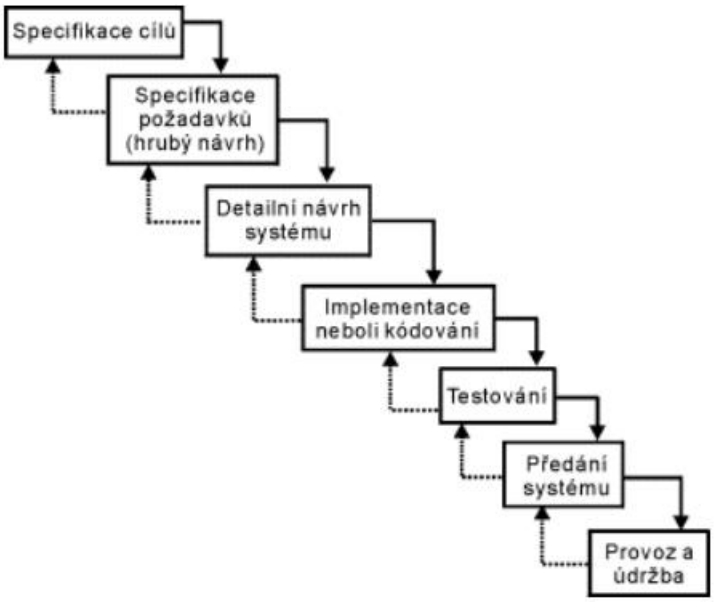
- Implementace – Následuje vlastní tvorba a programování softwaru. Jako podklady pro práci jsou k dispozici všechny dokumenty z předešlých částí (proto by měli bát zpracovány dostatečně detailně, viz výše).
- Testování – Zjišťování funkčnosti systému je důležité. Je to poslední možnost odhalit chyby v systému. Zároveň je produkt testován na veškeré možné vstupy, přičemž samotný systém ještě není zaveden do reálného prostředí (nasazení nepřipraveného systému by mohlo mít fatální následky, například systém pro zdravotnictví či letectví).
- Zavádění systému – Jedná se o instalaci systému a jeho zavedení do provozu. S tím také přichází poskytnutí provozních manuálů zadavateli a následné proškolení budoucích uživatelů.
- Zkušební provoz – Jde spíše o období, kdy je tvůrce systému povinen poskytnout okamžitý servis při výskytu jakékoliv chyby systému, která narušuje firemní procesy, a ihned poskytnout nápravu.
- Provoz a údržba – Finální fáze. Poslední nastavení systému jako přidělování práv. Sledování provozu z hlediska poruchovosti a následné řešení pomocí optimalizace provozu. Střežení systému proti neoprávněnému vniknutí. Celková údržba systému.
- Reengeenering – Fáze kdy je nutné přehodnotit systémové požadavky. Tyto požadavky nelze pouze opravit, neboť závažnost vzniklých chyb je natolik závažná, že nutí tvůrce systému vrátit se k první fázi a započít vývoj IS od začátku. [21]

Výše zmiňované fáze jsou pouze ilustrační. Je to dáno velkým množstvím vývojových technik (viz níže), které tyto fáze člení dle své koncepce (jsou rozdílně agregovány). Z tohoto důvodu může u následujících vývojových konceptů být rozdílné množství takovýchto fází.

3.2.1 Vodopádový model

Vodopádový model je nejstarším modelem životního cyklu při tvorbě informačních systémů. Jedná se o sekvenční model, to znamená, že činnosti na sebe postupně navazují. Na jeho fáze se nahlíží jako na svažující se tok, přesně jako u vodopádu. Poprvé byl popsán autorem Winstonem W. Roycem (1929-1995) v článku „Managing the Development of Large Software System“ v roce 1970.

Původně o tomto modelu hovořil jako o příkladu chybného přístupu k tvorbě systémů, avšak postupem času prošel model různými modifikacemi. Pro tento model platí pravidlo, že pokud si nejsme jistí, že je daná fáze stoprocentně kompletní, nemůžeme se posunout do dalších fází. V současné době tento přístup používá například ministerstvo obrany Spojených států nebo NASA.



Obrázek 1: Vodopádový model (Zdroj: https://www.fi.muni.cz/~smid/mis-zivcyk.htm)

Vodopádový model má i přes své počáteční odmítání několik výhod souvisejících s tvorbou projektů. Pokud je při tvorbě systému věnována dostatečná pozornost prvotním fázím návrhu a tvorbě požadavků, mohou se tím snížit finanční náklady na opravu dané chyby v následujících fázích. Vždy platí, že chyby se nejjednodušeji opravuje v dané fázi, než ve fázích následujících, ze kterých se již těžko vrací do fází předchozích a čerpání finančních prostředků se rapidně zvyšuje. Například pokud je chyba nalezena až ve fázi implementace může být finančně nákladnější jí odstranit, než kdyby byla chyba nalezena a opravena už ve fázích návrhu.

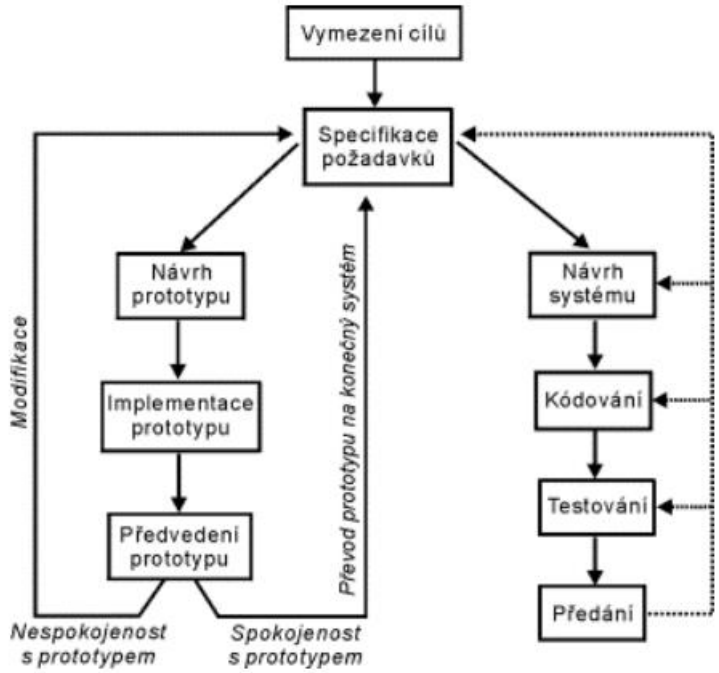
Při tvorbě systémů se také stává, že v průběhu mohou odcházet členové týmu a tím dochází ke ztrátě důležitých vědomostí. To však tak úplně neplatí v případě vodopádového modelu. Klade totiž důraz na jednotlivé fáze včetně počáteční dokumentace. Pokud je tedy kvalitně a správně zpracována dokumentace, nábor nových členů tvůrčího týmu, nebo dokonce celých týmu, nemusí být tak zdlouhavý, protože se členové začlení tím, že si prostudují dokumentaci.

Samozřejmě vodopádový model s sebou nese i nevýhody. Především je nemožné dovézt jednotlivé etapy do konce co se velkých či komplexnějších projektů týká. V druhé řadě jde o samotné klienty. Ti sama si nemusí býti jistí svými původními požadavky, například protože si sami nedokáží představit výsledný produkt. Často se stává, že po předložený hotového systému začnou klienti vyjadřovat k prototypu a měnit své požadavky. To je pro vodopád problém, neboť je nutné opět začít od první fáze a přepracovat návrh. [21][22][23]

3.2.2 Prototypový model

Jedná se o takový přístup, který se spíše využívá u menších projektů. Návrh modelu se objevil kolem roku 1988. Funguje na principu částečné implementace produktu – prototyp. Tento prototyp může charakterizovat celý systém, nebo jeho konkrétní části.

Při tvorbě prototypu se vždy postupuje stejně. Po specifikaci požadavků se prototyp navrhne, implementuje a poté se předvede zákazníkovi. Zákazník se k takovémuto prototypu vyjádří buď jeho přijetím či jakýmikoliv připomínkami. V případě jakéhokoliv nepřijetí systému je na místě návrat ke specifikaci požadavků a celý proces se opakuje, dokud není systém zcela vyladěn. Až po konečném schválení klientem se začíná tvořit finální systém.



Obrázek 2: Prototypový model (Zdroj: https://www.fi.muni.cz/~smid/mis-zivcyk.htm)

Mezi jednoznačné výhody patří bez pochyby zapojení uživatele do celého procesu. Tím se zvyšuje šance na přijetí projektu nebo minimálně přispívá ke zkrácení časového hlediska na projekt, pokud zákazník shledá funkce navrženého prototypu dostačující. Dále snižuje nebezpečí projektových rizik rozdělením projektu na menší části. [24]

3.2.3 Iterativní model

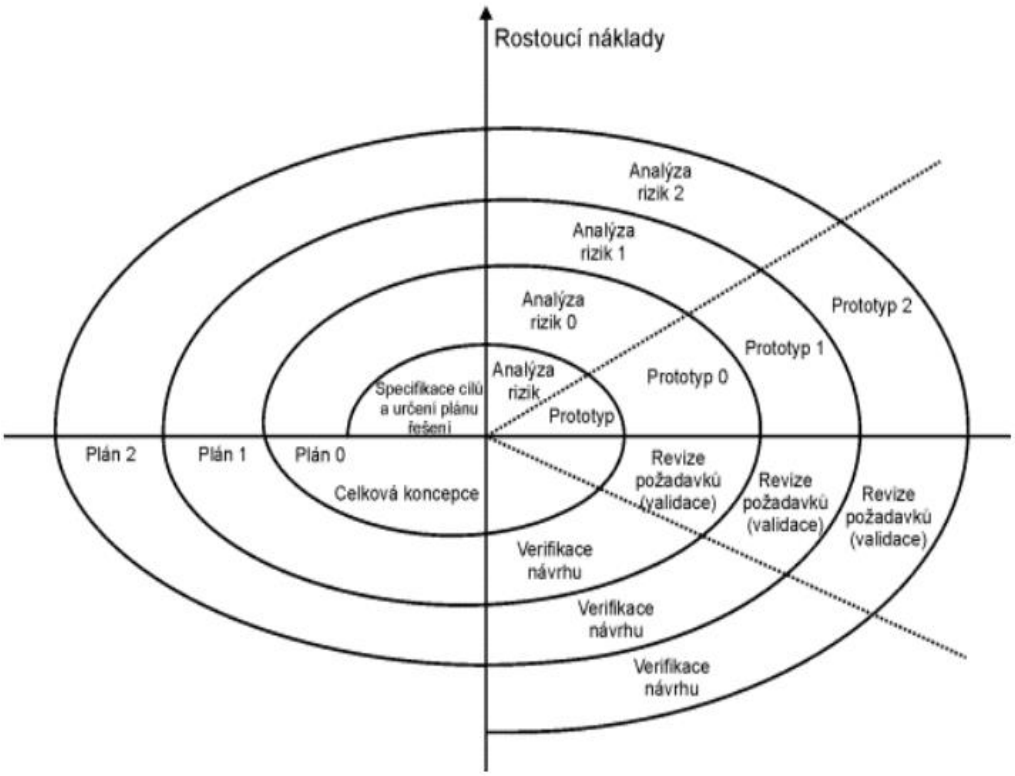
Iterativní přístup byl vytvořen jako reakce na nevýhody a problémy spojené s vodopádovým přístupem. Základem iterativního modelu je vývoj systému za pomocí opakovaných cyklů. Díky tomu mohou vývojáři neustále předkládat klientovi výsledky své práce v menších dávkách a ihned reagovat na klientovi připomínky. Další výhodou je možnost využít zkušeností z předchozích částí projektování.

Iterativní model rozděluje celý projekt na několik fází, přičemž každá fáze se skládá z několika iterací s cyklem typu vodopád. Výsledkem každé iterace je funkční verze systému. Cílem každé iterace je úplná část neúplného produktu. [25]

3.2.4 Spirálový model

První definici spirálového modelu stanovil Barry Boehm v roce 1986 ve svém článku „*A Spiral Model of Software Development and Enhancement*“. Byl vytvořen za účelem pokrytí nedostatků vycházejících z vodopádového modelu, přičemž do svých metodik zahrnuje iterativní přístup (viz výše). Tento model též spadá pod přístupy řízené riziky. To znamená, že postup do další fáze závisí na důsledně provedené analýze všech rizik a možných problémů.

Spirálový model probíhá v neustále se opakujících krocích, dokud se nedosáhne naplnění všech stanovených cílů kladených na příslušný projekt, přičemž dbá na opakovanou analýzu rizik. Počáteční vývoj probíhá pouze na úrovni hrubých specifikací, a až v pozdějších fázích jsou specifikace upřesňovány po konzultaci se zákazníkem. Díky svým iterativním vlastnostem a opakované analýze rizik se tento model daleko lépe vyrovnává s pozdější úpravou požadavků.



Obrázek 3: Spirálový model (Zdroj: https://www.fi.muni.cz/~smid/mis-zivcyk.htm)

Celý proces spirálového modelu je rozdělen do čtyř hlavních částí:

- Určení cílů, alternativ, omezení (Determine objectives, alternatives, constraints)
- Vyhodnocení alternativ, identifikace a řešení rizik (Evaluate alternatives, identify, resolve risks)
- Vývoj a verifikace další úrovně produktu (Develop, verify next-level product)
- Plánování následujících fází (Plan next phases)

Po každé z těchto fází následuje testování současných výsledků, jejich hodnocení a předání dílčích výsledků klientovi ke konzultaci. Produkt je tedy neustále testován, zda vyhovuje všem stanoveným specifikacím a požadavkům. Vždy při případných změnách je nutné upravit testovací případy a testovací scénáře.

Mezi výhody spirálového modelu patří možnost včasně vyloučit nevhodné návrhy řešení problémů. Další výhodou je možnost sledovat už první verze systému a hodnotit jej od jeho počátku přes postupný vývoj. Největším přínosem je možnost konzultovat se zákazníkem v každém kroku vývoje i v prvotních specifikačních fázích a systém ihned formovat dle klientových požadavků. V neposlední řadě je analýza rizik, která pomáhá předcházet chybám.

Na druhé straně jsou se spirálovým modelem spojeny i nevýhody. První z těchto nevýhod paradoxně vychází z výhod, protože neustálá konzultace s klientem vyžaduje jeho přítomnost a zákazník nemusí být vždy schopen se dostavit ke konzultaci a nemá ani dostatečně kompetentní osobu k zastoupení. Dále není zcela možné přesně definovat časový rozvrh, neboť při každém cyklu může být nalezena nová chyba či zákazník se může rozhodnout, že v systému je třeba zcela nová funkce, které dříve nebyla v požadavcích uvedena. Pak je také nutné zmínit, že při provádění analýzy rizik je nutný pečlivý přístup, protože na analýzách rizik jsou stavěny další fáze projektu. [26][27]

3.2.5 Agilní metody

Termín agilní metody se objevil v roce 2001, a byl definován skupinou vývojářů, kteří se snažili nalézt náhradu za vodopádový model. Jedná se o interaktivní metody řízení projektů. Základním stavebním kamenem je aktivní spolupráce mezi zákazníkem a týmem, což pomáhá zajistit rychlé reagování na jakékoliv změny ať už ze strany zákazníka, nebo vlivem okolního prostředí. Pojem agilní znamená čilý, aktivní či horlivý. Pojem agilní metody není označení konkrétního nástroje či metody, ale jedná se o soubor metodik.

Předpokladem agility je, že osobní komunikace a schopnosti každého jedince mají mnohem větší význam, než procesy a nástroje stanovené před započetím projektu samotného. Dokumentace je omezena jen na nejnutnější informace (například jak bude probíhat kontrola a seznam obecných požadavků), snaží se vyhnout zbytečným informacím a dbá na včasnou aktualizaci daných dokumentů.

Agilní metody se nejvíce používají u projektů, které jsou natolik nové nebo inovativní, že neexistují nebo je velice málo podkladů k jeho úspěšnému a pečlivému naplánování. Dalším důvodem užití mohou být nedostatečné informace poskytnuté klientem pro stanovení troj imperativu (odhad nákladů, zdrojů a časového hlediska). Jedním z dalších mnoha důvodů může být i předpoklad, že projekt samotný přinese velké množství změn v projektu.

Jak je zmíněno výše, agilní metody není jedna metoda, ale celý soubor. V následující části budou představeny nejznámější a nejpoužívanější metody agility: [28]

SCRUM

Jedná se o iterativní způsob vývoje softwaru, který patří mezi agilní metodiky. Základní jednotkou tvorby projektu je sám organizační tým, který pracuje jako jednotka za účelem vytvoření produktu. Klíčové na metodě SCRUM je, že klienti mohou kdykoliv měnit názory na to co si ve svém produktu přejí mít. SCRUM zastává názor, že problém nelze zcela pochopit nebo plánovat, a z toho důvodu se spoléhá na schopnosti týmu dodávat části projektu dle priority k nim přiřazené.

Jednotlivé fáze projektování v metodice SCRUM se nazývají události. Jednou z těchto událostí je tzv. sprint, který slouží jako základní časová jednotka sloužící při vývoji systému. Každý takovýto sprint trvá nejčastěji okolo dvou až čtyřech týdnů. Před každým sprintem je definován úkol platící pro celý sprint a po každém sprintu předloží tým výsledek klientovi a takovýto výsledek je zhodnocen.

Tým tvoří tři hlavní role označovány jako SCRUM tým:

- Product owner (vlastník projektu) – Představuje zainteresované subjekty. Pověřená osoba, jejímž úkolem je dohlížet, že jsou splněny požadavky na produkt kladené zákazníkem. Přiřazuje prioritu konkrétním činnostem a hodnotí výsledek.
- Vývojový tým - Úkolem vývojového týmu je být zodpovědný za dodání části produktu na konci každého sprintu. Tým bývá složen z 3-9 členů, přičemž každý má na starost konkrétní rolo (analýze, design, vývoj, dokumentace).
- Scrum master – Jedná se osobu odpovědnou za odstranění překážek v týmu. Není vedoucím týmu, slouží jako prostředník mezi týmem a jakýmikoliv negativními vlivy. [29]

Extrémní programování (XP)

Metoda připisující se dané činnosti všem účastníkům vývojového procesu. Jde o běžné činnosti, jež jsou však dovedeny do extrému. Je nazýváno extrémním, neboť jak již bylo zmíněno, vše co se osvědčí v praxi XP se používá neustále. Například pokud se osvědčilo testování kódu, pak se v XP vše testuje neustále. Nebo pokud je důležitá architektura, pak se budou všichni členové neustále podílet na její úpravě.

Důležité je neustále udržovat komunikaci se všemi zainteresovanými subjekty. Selhání komunikace mezi programátorem a zákazníkem by mělo obrovský vliv na výsledný produkt. Dalším důležitým bodem je jednoduchost. To znamená naprogramovat pouze to, co původní požadavky požadovali. V neposlední řadě je důležité mít odvahu pouštět se do řešení problémů, ač toto řešení může způsobit kaskádu dalších potenciálních problému. [30]

3.3 Analýza a návrh informačních systémů

Analýza a návrh informačního systému je náročný proces. V průběhu vývoje systému se časem vytvořily dva přístupy jak k tvorbě systému přistoupit. Jsou jím strukturovaný přístup a objektově orientovaný přístup.

Oba přístupy sdílí myšlenku rozdělit proces analýzy a návrhu na jednotlivé činnosti a procesy, které nejsou tak velké a tudíž je jednodušší jejich popis a znázornění vazeb mezi nimi. Každá z těchto metod přitom používá odlišné nástroje k dosažení svých cílů.

Strukturovaný přístup

Počátky strukturovaného přístupu jsou zasazeny do 70. let 20. století. Tento přístup pohlíží na systémy (jak již název vypovídá) strukturovaně. Je pro něj charakteristické samostatné zobrazení datových struktur v modelu. Při popisu zobrazuje graficky struktury systému za pomocí stromových diagramů, která díky svému uspořádání umožňují zobrazit příslušné subsystémy a zachytit všechny informační vazby mezi nimi.

Z hlediska abstrakce rozlišujeme u strukturovaného přístupu tři úrovně:

- Konceptuální model – Zajišťuje prvotní rozpoznávání datových objektů, jejich vztahů a vazeb mezi nimi. Takto stanovený návrh není závislý na technologickém prostředí.
- Logický model – Představuje relační schéma, přičemž bere v úvahu integritní omezení kladená na model. Určuje strukturu dat. Většinou vyjádřen formou ERD a DFD diagramů (viz níže).
- Fyzický datový model – Má na starosti samotnou implementaci v konkrétním databázovém prostředí. [15]

Pro popis systému a jeho modelování využívá různé diagramy, které graficky zobrazují vztahy mezi jednotlivými procesy. Každý diagram slouží k popisu jiné části či jiného chování systému:

- Entity Relationship Diagram (ERD) – ERD poskytuje statický pohled na systém. Modeluje strukturu informačního systému z hlediska databází. Mapuje data, která jsou ukládána v jednotlivých databázích a jejich vzájemné vztahy.
- Data Flow Diagram (DFD) – Slouží k modelování datových a řídících toků v systému. Je také užíván jako tzv. kontextový diagram (zvláštní případ DFD, jenž zobrazuje systém jako jediný proces a jeho vztah k okolí).
- Function State Diagram (FSD) – Jeho účelem je zobrazit funkční celky (subsystémy) se zaměřením na hierarchickou strukturu.
- State Transition Diagram (STD) – Tento diagram je důležitý z hlediska pochopení logiky systému. Modeluje stavy, v jakých se systém může nacházet včetně jejich posloupnosti. Zobrazuje podmínky, za kterých systém přejde z jednoho stavu do druhého.
- Data Dictionary (DD) – Datový slovník slouží pro popsání dat v systému z pohledu uživatele. Slovník obsahuje všechny položky databáze. Při kompletním vypracování DD můžeme zjistit, zda se některé položky neobjevují v databázi vícekrát, a tudíž se vyhneme redundanci dat.
- Structure chart – Znázorňuje především hierarchii programových modulů systémů. Graf je ve tvaru stromu, kořenem je hlavní modul a každý uzel značí podřízené moduly volané hlavním systémem.
- Flow chart (vývojový diagram) – Slouží ke grafickému popisu jednotlivých kroků v algoritmu. [15][16]

Objektově orientovaný přístup

Objektově orientované programování (OOP) se začalo rozmáhat především v 90. letech 20. století. Od předchozích přístupů se liší především jejich přístupem. Ten je totiž založen na tzv. objektech.

Objekt je struktura, která má předem definované vlastnosti (atributy) a chování (metody). Jak vlastnosti, tak atributy jsou zapouzdřené v každém objektu, a tyto objekty jsou schopny reagovat na jednotlivé události. V případě OOP pohlížíme na informační systém jako na množinu spolupracujících objektů. Obecně objektový přístup lépe odpovídá chování reálného světa.

Pro každý objekt platí čtyři základní charakteristiky, které ho reprezentují. Mezi tyto charakteristiky patří:

- Identita – Každý objekt je jedinečný. Přesto, že může mít více objektů stejné vlastnosti i metody jsou odlišné. K rozlišení jednotlivých objektů se využívá identifikační číslo.
- Klasifikace – Objekty se stejnou datovou strukturou (atributy) a chováním (metody, operacemi) jsou seskupeny do tříd. Každý objekt je pak výskytem (instancí) své konkrétní třídy. Každá instance třídy má své vlastní hodnoty atributů, ale sdílí strukturu s ostatními instancemi. Příkladem může být kufr. Každý kufr má svou výšku, šířku, hloubku a barvu, ale každý kufr má jiné hodnoty těchto parametrů, a tudíž je každý kufr unikátní instancí.
- Polymorfizmus – Odkazovaný objekt se může chovat podle instance své třídy. Jde o případ objektů, které poskytují stejné rozhraní a navenek se s nimi i stejně pracuje, ale jejich konkrétní chování se liší podle implementace. Jde o případy, kdy mají třídy stejné metody, ale pro každou třídu platí jiné hodnoty. Například v šachu je metoda táhni pro koně i věž stejné, ale výsledný posun figurky se značně liší.
- Dědičnost – Objekty jsou organizovány stromovým způsobem. Díky tomuto mohou objekty určitého druhu dědit vlastnosti svého předchůdce a dále je jistým způsobem rozšiřovat či upřesňovat. Tento princip je hlavní předností OOP a zmenšuje opakování v návrhu. [17][18]

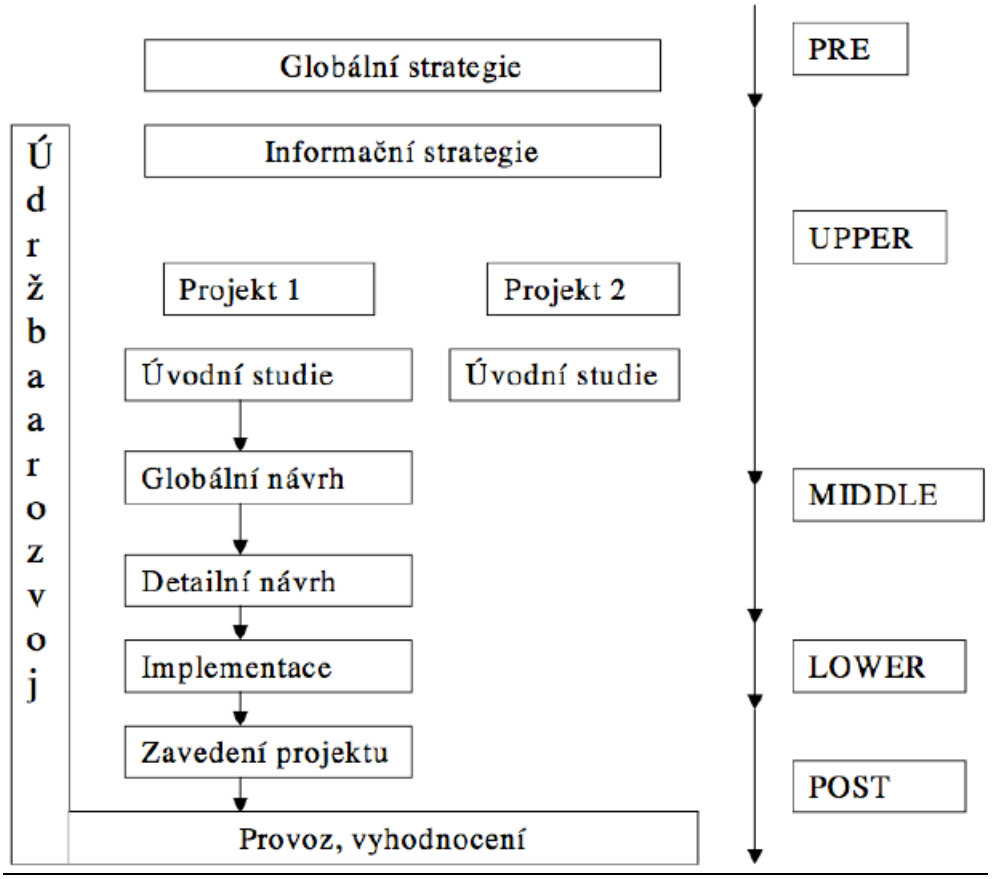
3.4 Nástroje CASE

Zkratka představuje Computer Aided Software Engineering nebo Computer Aided Systems Engineering, což může být přeloženo jako počítačem podporované softwarové inženýrství či vývoj software s využitím počítačové podpory. Využívají software při vývoji a údržbě počítačových programů nebo systémů za účelem dosažení vyšší kvality, bezchybnosti a udržitelnosti software.

Nástroje CASE slouží k modelování systémů pomocí diagramů, neboť člověk obecně lépe chápe obrázek než psané slovo. Dále se používá ke generování kódu podle modelu (reverse engeneering) a v neposlední řadě k vytvoření dokumentace dle modelu. CASE nástroje podporují týmovou práci při vývoji systému, podporují také sdílení rozpracovaných částí systému. Hlídají také dodržování užití zvolené metodiky.

Nástroje CASE můžeme dělit dle fází životního cyklu systému:

- Pre CASE – Slouží při tvorbě globální strategie
- Upper CASE – Slouží pro plánování, specifikaci požadavků a modelování organizace podniku včetně globální analýzy informačních systémů.
- Middle CASE – Slouží k detailní analýze a vlastnímu návrhu informačního systému.
- Lower CASE – Podporuje fyzickou realizaci systému
- Post CASE – Podporuje zavedení systému a jeho následnou údržbu a rozvoj. [31]



Obrázek 4: Nástroje CASE dle životního cyklu IS (Zdroj: https://docplayer.cz/68144164-Case-nastroje-jaroslav-zacek.html)

3.5 Unified modeling language (UML)

Jedná se o grafický jazyk v softwarovém inženýrství sloužící pro specifikaci, vizualizaci, navrhování a dokumentaci programových systémů. Byl navržen v roce 1994 Gradym Boochem, Jimem Rumbaughem a Ivarem Jacobsonem, kteří původně pracovali každý v jiných společnostech. Sjednotila je myšlenka vytvořit metodologii sloužící k analýze a návrhu systémů a programů využívající objektově orientovaný přístup, jež by umožňovalo

přehlednou tvorbu systému a zároveň poskytoval nástroje pro komunikaci mezi klientem a tvůrcem, kterým by rozuměli obě strany.

Jak již bylo zmíněno UML je objektově orientovaný jazyk, který na rozdíl od textově orientovaných jazyků využívá vlastní grafickou syntaxi (definuje pravidla pro sestavování konkrétních elementů do komplexnějších objektů) a sémantiku (pravidla určující konkrétním výrazům jejich význam).

V současné době se UML nejvíce využívá při tvorbě informačních systémů, neboť díky svým metodám zvládá rychle popisovat a charakterizovat složitější aplikace, což je nezbytným předpokladem pro rychlé a kvalitní vytvoření systému. Pro popis systému a jeho funkcí využívá diagramy, přičemž pro každý diagram je jasně definována podoba a funkce pro každý jednotlivý diagram, což značně pomáhá předcházet omylům spojeným se špatnou komunikací například mezi analytiky a vývojáři.

Celý proces tvorby v UML je založen na třech elementech, jež jsou reprezentovány pomocí textových a grafických značek. První z těchto elementů jsou předměty. Jsou reprezentovány dvojrozměrně, převážně jako plošné tvary (elipsy, kvádry, obdélníky), které jsou vždy uzavřené. Předměty dále rozlišujeme na čtyři typy abstrakcí:

- Strukturní abstrakce – Jedná se o programové třídy, aplikační nebo objektové rozhraní, případy užití systému, komponenty nebo uzly.
- Abstrakce chování – Tyto abstrakce reprezentují vzájemné komunikace mezi objekty. Chování umožňuje modelovat stavový stroj, který definuje přechody, události a aktivity.
- Abstrakce seskupení – Slouží k seskupování částí diagramů z nižších úrovní. Vytváří balíčky s popisem dané abstrakce. Seskupení nejsou v některých aplikacích použitelná (tzn. Ne vždy je možné je využít v hierarchických diagramech).
- Poznámky – Používají se k bližšímu specifikování vlastností a chování ostatních elementů UML diagramů.

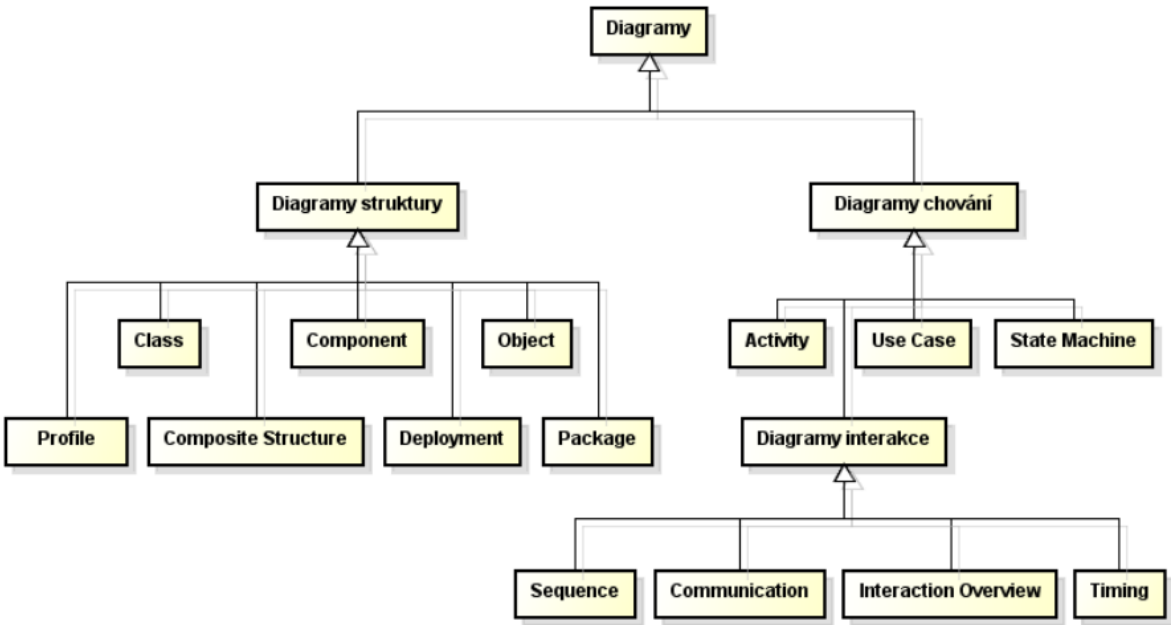
Druhým elementem jsou relace. Jejich úkolem je propojování předmětů v prostředí UML. Určují vztahy mezi různými předměty (objekty) a dělí se dle typu vazby:

- Asociace – Udává obecnou souvislost předmětů, která je dle norem UML přesně definována. Speciální případem asociace je agregace (definuje, že objekt je součástí

jiného objektu, skládá se z více částí) a kompozice (zvláštní případ agregace, zánikem celku zaniká i součást).

- Závislost – Používá se v případě, že změna jednoho předmětu ovlivní druhý předmět nebo v případě, že jeden předmět poskytuje druhému požadované informace.
- Generalizace – Modeluje stav, v němž je jeden předmět specializací druhého. Někdy se také označuje jako dědičnost.
- Realizace – Znázorňuje vztah typu dohoda, za jejíž splnění je odpovědný jiný předmět. Realizuje se pomocí rozhraní. [17][32][33]

Třetím elementem jsou diagramy. Jak již bylo mnohokrát výše zmíněno UML využívá ke znázornění struktury a chování systému diagramy, které se používají jako jednoduchý nákres. Tyto diagramy se mohou kreslit na papír při jednání s klientem, nebo můžeme využít počítačové programy k tomu určené. V současné době se UML skládá ze čtrnácti druhů diagramů.



Obrázek 5: Diagramy UML (zdroj: https://www.itnetwork.cz/navrh/uml/uml-uvod-historie-vyznam-a-diagramy)

Každý z těchto diagramů má v UML své využití, avšak v rámci této práce (tedy návrh informačního systému) budou využity jen diagramy sloužící právě k modelování vnitřních komponent systému, a tudíž zde nebudou rozvedeny všechny.

3.5.1 Modelování systému

Návrh informačního systému je založen na principu třech modelů. Každý z těchto modelů se zabývá jiným pohledem na systém a utváří nám různé prvky a subsystémy:

- Model tříd – Určuje statickou strukturu objektů a jejich vztahů. K modelování tohoto stavu využívá tzv. class diagram (diagram tříd). Objekty jsou reprezentovány uzly a vztahy reprezentují souvislosti.
- Stavový model – Zobrazuje změny aspektů systému v čase. Používá stavový diagram, kde stavy jsou opět reprezentovány uzly a události zobrazují souvislosti.
- Model interakcí – Znázorňuje spolupráci objektů v systému. Využívá k tomu diagramy: Use case, sekvenční diagram a diagram aktivit. Tyto modely se vzájemně doplňují a jsou vzájemně propojeny. [17]

3.5.2 UML diagramy

Jak již bylo výše několikrát zmíněno jazyk UML využívá diagramy pro popis systému, jeho chování a změn v čase. V následující části budou blíže popsány diagramy, které budou následně využity ve vlastní práci.

Diagram tříd

Jedná se o hlavní diagram popisující datovou statickou strukturu navrhovaného softwarového systému. Třída je základním logickým objektem, který zachycuje vlastnosti jednotlivých objektů včetně jejich chování. Při vývoji logického modelu jako je strukturální hierarchie v UML pracujeme výhradně s třídami. „Každá třída reprezentuje skupinu objektů s podobnými vlastnostmi, společným chováním nebo stejnými vztahy k jiným objektům.“

Třída kromě svého názvu obsahuje atributy, které popisují vlastnosti dané třídy. Každý atribut může být doplněn informacemi, o jaký datový typ se jedná nebo jaká je jeho počáteční hodnota. Dále třída obsahuje operace a metody. Jedná se o funkce, které daný objekt provádí. [34]

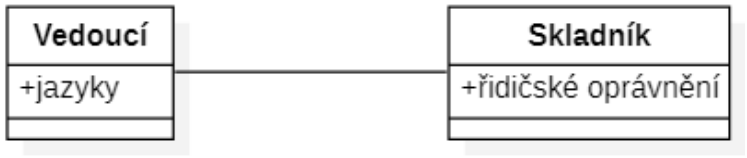
Nedílnou součástí diagramu tří mimo třídy samé jsou vazby mezi nimi tzv. asociace. Asociace (vazba nebo také spojení) zobrazuje vztah mezi dvěma nebo více objekty a může se

vyskytovat v několika formách dle způsobu potřebné vazby. Vazby bývají doplněny o multiplicitu.

Multiplicita je určení kolik výskytů jedné třídy může mít vztah k výskytu druhé třídy. Je označována pomocí následujících symbolů:

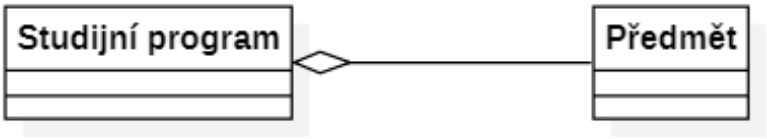
- Pomocí čísla (1) – znamená přesně stanovený počet výskytů (například 1 znamená právě 1)
- Pomocí hvězdičky (*) – symbol říká, že počet výskytů může být libovolně velký. Hvězdička zahrnuje i hodnotu 0.
- Pomocí intervalu (1..5) – počáteční a koncová hodnota udávají začátek a konec intervalu, ve kterém se mohou výskyty pohybovat.

Asociace – základním spojením dvou objektů je asociace. Dle notací UML má následující podobu.



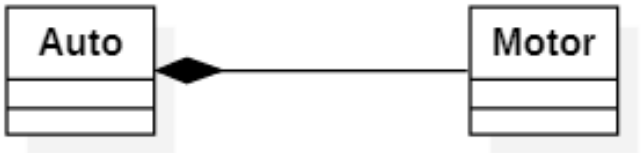
Obrázek 6: Diagram tříd s asociační vazbou (Zdroj: Vlastní tvorba)

Agregace – reprezentuje speciální typ vazby. Určuje vztah typu „součást - celek“. Jedná se o vazbu „je součástí“ nebo „skládá se z“. Agregace je znázorněna symbolem prázdného diamantu.



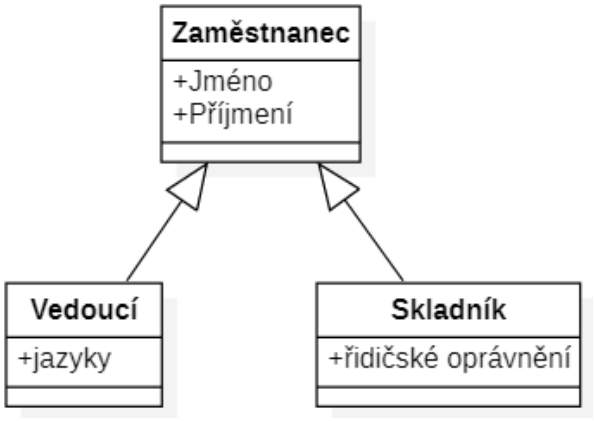
Obrázek 7: Diagram tříd - agregace (Zdroj: Vlastní tvorba)

Kompozice – je speciálním případem agregace, kdy platí 2 omezení: „součást může patřit pouze jednomu celku“ a „zánikem celku zaniká i součást“



Obrázek 8: Diagram tříd - kompozice (Zdroj: Vlastní tvorba)

Dědičnost – představuje účinný nástroj pro sdílení podobnosti mezi třídami, přičemž lze zachovat charakteristické odlišnosti. Jde o vztah „nadtřída - podtřída“, kdy podtřída dědí vlastnosti nadtřídy a zároveň obsahuje své unikátní vlastnosti. [17]

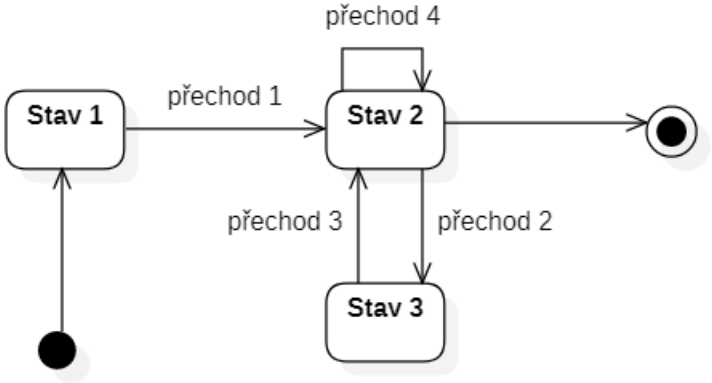


Obrázek 9: Diagram tříd - generalizace (Zdroj: Vlastní tvorba)

Stavový diagram

Stavový diagram se využívá při dynamickém modelování systému a slouží k popisu vývoje v čase. Hlavními stavebními kameny stavového diagramu jsou stav a událost (někdy také přechod). Událost je přechod mezi dvěma stavy, který je vyvolán určitou událostí, podmínkou nebo efektem. Přechody jsou různých povah dle způsobu vyvolání. Může jít o informativní přechody, kdy jde pouze o převedení informace z jednoho stavu na druhý. Dalším druhem přechodu může být splnění či nesplnění definované podmínky. V neposlední řadě může být událost způsobená naplněním časového limitu. [17]

Stav na druhé straně představuje současnou situaci, ve které se objekt může nacházet. Mezi stavy lze pomocí událostí přecházet nebo se do nich i vracet, pokud jsou k tomu stanoveny příslušné přechody. [35]



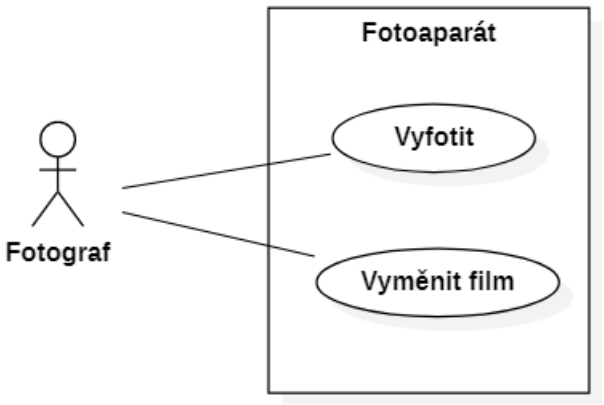
Obrázek 10: Stavový diagram (Zdroj: Vlastní tvorba)

Use Case

Use case, jsou také někdy nazývány diagramy chování. Jsou využívány k popisu sady akcí, které navrhovaný systém nebo systémy mohou poskytovat externím uživatelům (tzv. aktérům). Jedná se o jeden ze tří modelů interakcí, jež slouží jako doplněk k předchozím modelům (objektový a dynamický model). [36]

Use case, popisuje na vrcholné úrovni interakci systému s okolím. Pracuje především s aktéry a samotnými případy užití (jednotlivý use case). Aktéři jsou uživatelé (aktér přitom nemusí být jen osoba), kteří přistupují do systému a přímo s ním komunikují, ale nejsou přímou součástí systému. Na druhé straně use casy, jsou funkční části systému, které jsou poskytovány aktérovi při komunikaci.

Diagram use case má dle notace UML zobrazené hranice systému obdélníkem obsahujícím název systému. Každý jednotlivý use case, je znázorněn elipsou s názvem uvnitř. [17]

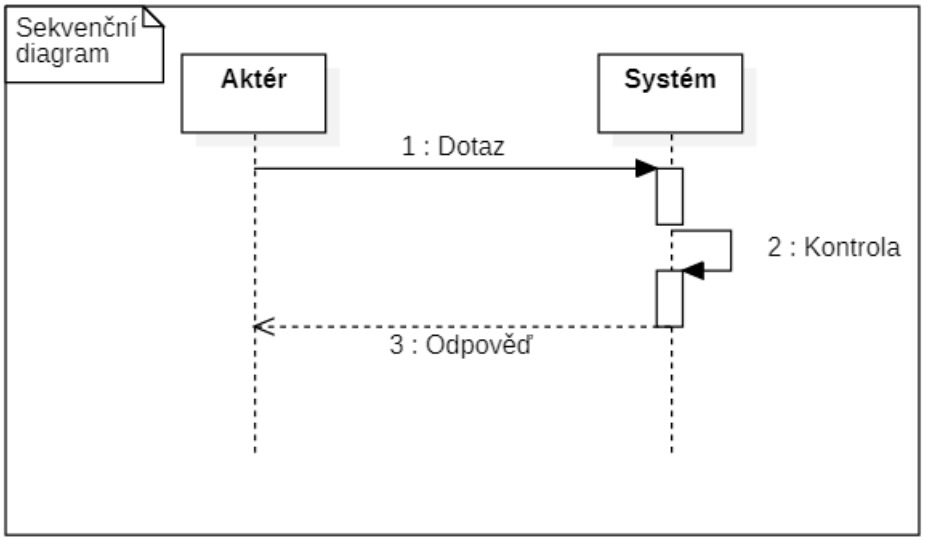


Obrázek 11: Use case diagram (Zdroj: Vlastní tvorba)

Sekvenční diagram

Sekvenční diagramy jsou druhým typem diagramů v modelu interakcí. Existují 2 typy sekvenčních diagramů a to scénáře a sekvenční diagramy samotné. Scénář popisuje posloupnost událostí, které nastávají v průběhu užití systému. Scénáře mohou vyjadřovat chování existujícího systému (chování zachycené v průběhu sledování) nebo chování, které se očekává od nově vyvíjeného systému. [37]

Samotné sekvenční diagramy jsou dle notace UML reprezentovány obdélníkem, který představuje aktéra nebo systém samotný. Z každého takto vytvořeného aktéra je vedena svislá čára (někdy označována jako „pupeční ‚šnůra“), která znázorňuje posloupnost instrukcí. Čas postupuje od shora dolů a řadí zprávy, jak za sebou následují, nikoliv však jejich reálnou délku trvání. [17]

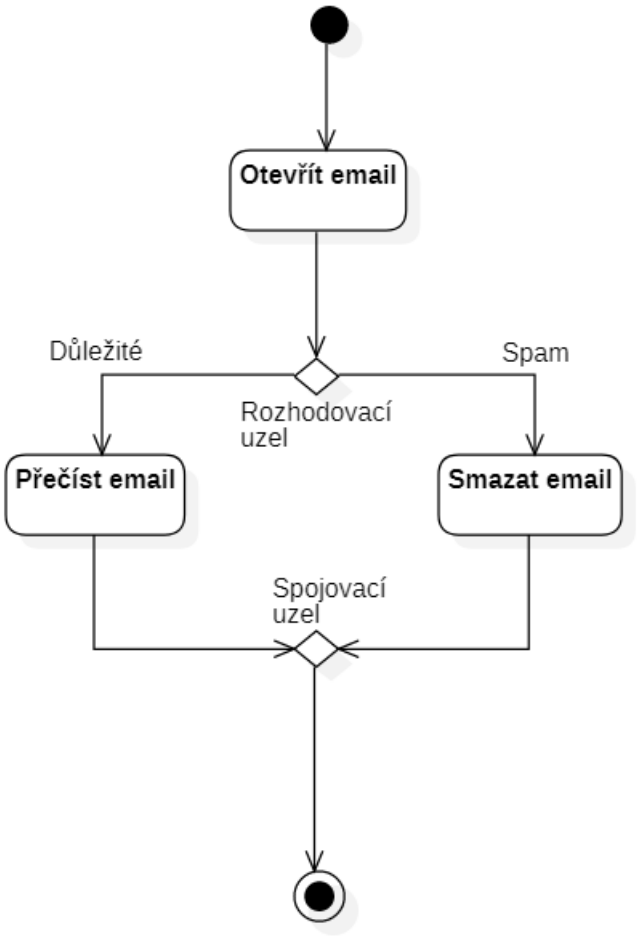


Obrázek 12: Sekvenční diagram (Zdroj: Vlastní diagram)

Diagram aktivit

Je třetím aspektem modelu interakcí. Jeho úkolem je popisovat dynamické aspekty systému. Zobrazuje jednotlivé toky mezi aktivitami. Diagram aktivit je dost podobný diagramu stavů, ať už způsobem vyjádření, tak podobnou notací dle pravidel UML (oba dva typy diagramů zobrazují posloupnost stavů, které nastávají včetně způsobu přechodů mezi nimi). Stavový diagram se ale především zaměřuje na jednotlivé stavy daného objektu, zatímco diagram aktivit ukazuje na stav výpočtu samotného (myšleno stav algoritmu či procesu). [38]

Avšak hlavním rozdílem od stavového diagramu je, že může modelovat i konkurenční kroky a toky řízení. Aktivity běží časově neomezeně a po jejich dokončení může následovat další krok. Aktivity mohou být dle potřeby rozděleny a znázorňovat tak více kroků, které musí proběhnout, aby byla aktivita řádně dokončena. [17]



Obrázek 13: Diagram aktivit (Zdroj: Vlastní tvorba)

3.6 Požadavky na systém

Při tvorbě systému je třeba dbát na potřeby klienta, které stanovuje jisté nároky na jím požadovaný produkt. Takovýmto požadavků se říká požadavky na systém (někdy také analýza požadavků). Obsahuje úkoly, jež vstupují do rozhodování o potřebách a podmínkách kladených na systém. Správná analýza požadavků je podmínkou úspěšného dokončení každého projektu. Každý požadavek se vztahuje ke konkrétnímu požadavku a musí být detailně popsaný.

Požadavky v zásadě rozlišujeme na dva typy. Funkční požadavky a nefunkční požadavky. Funkční určují například co má systém dělat, jak se má chovat nebo jaké informace má evidovat. Může se jednat o výpočty, technické detaily, manipulace s daty a další procesy, které jsou od systému vyžadovány a očekávány. Obecně nám funkční požadavky charakterizují určitý výsledek systému.

Druhým typem jsou požadavky nefunkční. Určují spíše omezení kladená na systém samotný. Definuje příkladem požadavky na řešení, jako jsou podmínky či termíny dodání. Dále zahrnuje vnější požadavky, mezi které patří legislativní omezení, cena výsledného produktu nebo následný servis. [39]

4 Vlastní práce

K návrhu systému vedla autora osobní zkušenost ve firmě v odvětví logistiky skladu. V průběhu práce se současným systémem bylo objeveno několik překážek a nedostatků při zpracovávání jinak jednoduchých procesů nebo úkolů, které byly shledány zaměstnanci firmy nekomfortní. Tyto nedostatky obtěžují především zaměstnance firmy, kteří nemohou svou práci vykonávat jednoduše a efektivně.

Mezi nejčastější chyby patřila nedostatečná kontrola pohybu zboží na jednotlivých pozicích a sledování stavu zboží. Tento přístup způsoboval hned několik nesrovnalostí, které způsobovali nepřehlednost a nekonzistenci údajů mezi systémem a reálným stavem. Jako první nesrovnalost způsobenou dříve zmíněným přístupem bylo rozprostření stejného zboží na více různých pozicích. Systém navíc při odběru nekontroluje data příjmu zboží a není tak možné odebírat zboží dle metod přístupu FIFO (first in, first out – první dovnitř, první ven). Zboží proto dále zastarávalo a stávalo se, že již nemohlo být plně použitelné či prodané.

S tímto problémem dále souvisí přístup systému k informacím o zboží a s jejich zpracováním. Systém vyžaduje malé, nebo žádné podmínky pro upozornění na právě zmíněný problém týkající se zastarávání a neodebíraní zboží. Dochází tak k hromadění starého či zastaralého zboží na pozicích, které nebylo pořádně aktivováno. Je nutné zmínit, že systém vlastní nějaké podmínky, které upozorní zaměstnance, ale tyto podmínky mají široké časové hledisko a v případě, že je zboží z jakéhokoliv důvodu přeskladněno je tato podmínka obnovena. Systém se tedy chystal nahlásit aktivně neodebírané zboží, ale při přeskladnění zboží je systém zmaten a myslí si, že zboží odebráno bylo.

V návaznosti na přístup systému k informacím souvisí i hlídání exspirace. Z hlediska příjmu zboží je samozřejmě složité kontrolovat každý kus zboží a zapisovat jednotlivá data exspirace, ale při hromadném objednávání většinou sedí data exspirace všech kusů uvnitř balení. Navíc cílem systému není upozornit na prošlé zboží, ale naopak předejít jeho znehodnocení a upozornit ve stanovený čas, než k tomuto znehodnocení dojde. Tato funkce není taktéž v současném systému zahrnuta, byť je pro správné fungování nezbytná.

V neposlední řadě, jedním z dalších velkých nedostatků je detailní výstup ze systému ohledně pohybu zboží, na jakých pozicích se současně nachází nebo množství zboží. Tento

pohled současný systém poskytuje, ale bez možnosti zjistit kdo příjem zboží, přeskladnění zboží nebo výdej zahájil či potvrdil. V případě, že se z jakýchkoliv důvodů dohledává zboží, jsou tyto informace nezbytné. Avšak největším problémem je, že systém po odepsání posledního kusu zboží smaže historii o pohybu, a tudíž při zpětné kontrole již zboží pro systém neexistuje.

Poslední problém, který by měl být zmíněn, jsou přístupová práva k jednotlivým úkonům (jako je příjem zboží, výdej zboží atd.) v systému. Současná verze totiž umožňuje značně nesourodá práva vůči jednotlivým typům zaměstnanců (zaměstnanci obchodního oddělení mohou manipulovat se zbožím ve skladu). Ať už byl účel jakýkoliv, každý systém musí mít pevně nastavená práva tak, aby nedocházelo k zásahům do pracovních povinností zaměstnanců rozdílných oddělení.

V důsledku těchto zmiňovaných chyb se autor rozhodl navrhnout nový systém, který by splňoval požadavky pro práci se zbožím, přičemž by vzal v úvahu výše zmiňované chyby starého systému a napravil je již ve fázi návrhu informačního systému.

4.1 Požadavky na systém

Požadavky jsou jednou z podstatných částí návrhu informačních systémů. Rozdělujeme je na Funkční požadavky a Nefunkční požadavky. Ačkoliv nejsou přímo součástí návrhových diagramů UML jsou nedílnou součástí a definují nároky uživatelů (případně majitelů systému) kladené na funkce systému.

Následující požadavky byly stanoveny na základě osobních zkušeností při práci v zadavatelské firmě a také na základě konzultací s pracovníky dané firmy.

4.1.1 Funkční požadavky

Funkční požadavky v zásadě řeší, co bude systém dělat. Jedná se o přímé požadavky zákazníka na funkčnosti budoucího systému, které od něj očekává. Zákazníci tak definují například časté chyby, které mohou způsobovat nebo způsobují překážky při užívání systému. Požadavky jsou vyobrazeny v tabulce a dále jsou popsány níže dle jejich identifikačního čísla.

ID požadavku	Popis požadavku
FP 1	Systému umožňuje uživatelům přihlášení i opětovné odhlášení
FP 2	Systém bude pod správou administrátora
FP 3	Systém bude umožňovat administrátorovi přidělovat práva
FP 4	Systém umožňuje zaměstnancům spravovat položky na skladě
FP 5	Systém umožňuje práci s objednávkami a jejich archivaci
FP 6	Systém bude kontrolovat dostupnost zboží na skladě
FP 7	Systém bude umožňovat vnější přístup uživatelům (uživatelská práva zákazníka)
FP 8	Systém upozorní zaměstnance na minimální zásobu výrobku a jeho vyčerpání
FP 9	Systém upozorní zaměstnance na zboží, které není aktivně odebíráno
FP 10	Systém bude umožňovat správu volných i obsazených pozic
FP 11	Systém umožní zaměstnancům obchodního oddělení nastavovat kategorie zboží
FP 12	Systém uživatele upozorní na blížící se exspiraci zboží

Tabulka 1: Funkční požadavky (Zdroj: Vlastní tvorba)

FP1 – Přihlášení a opětovné odhlášení

Uživatelé se do systému přihlašují pomocí přístupového jména a hesla, které jsou provázány s příslušným uživatelem. Po vyplnění příslušného jména, hesla a stisknutí potvrzovacího tlačítka je uživatel přihlášen a může se systémem dále pracovat dle jeho příslušných práv (viz FP3).

Z hlediska bezpečnosti se uživatelé musí po skončení svých pracovních povinností odhlásit ze systému. Tento úkon je proveden stisknutím odhlašovacího tlačítka a následným potvrzením odhlášení (z důvodů zabránění nechtěného odhlášení).

FP2 – Správa systému administrátorem

Administrátor se do systému přihlašuje jako standartní uživatel, ale s tím rozdílem, že po přihlášení má absolutní přístup do všech ovládacích prvků, které systém poskytuje. Administrátor je znalý uživatel systému a je schopen napravovat základní chyby vycházející ze špatného užívání systému ostatními uživateli (opravy dat v databázi jmen apod.) a provést kroky, které slouží pro odstranění těchto nedostatků. Jednou z jeho hlavních úloh je však

správa uživatelů a jejich práv (viz. FP3). Další jeho povinností je úprava webového sortimentu skladu (viz NP10).

FP3 – Přidělování práv

Administrátor spravuje a přiděluje dva základní typy práv. Těmito typy práv jsou zákaznická a zaměstnanecká oprávnění. Zákaznické oprávnění je vytvářeno automaticky založením nového zákazníka. Administrátor zajišťuje, že všechny účty jsou řádně zadány vzhledem k současným požadavkům a budoucím aktualizacím (přidávání atributů nutných ke správné identifikaci zákazníka). Dále poskytuje zákazníkům pomoc při řešení nestandartních situací (viz FP2).

Druhým typem jsou oprávnění zaměstnanecká. Administrátor může přidělovat několik druhů zaměstnaneckých oprávnění, přičemž každé slouží k provádění jiných pracovních činností (například pro zaměstnance skladu, obchodního oddělení nebo vedení firmy). Zaměstnanec nemusí mít striktně pouze jeden typ oprávnění.

FP4 – Správa položek na skladě

Zaměstnance má po přihlášení k dispozici moduly, které jsou charakterizovány názvem a stručným popisem k čemu daný modul slouží (příjem zboží, přeskladnění, šrotace atd.). Má k dispozici pouze ty moduly, které potřebuje pro výkon svých pracovních povinností. Moduly pro správu položek na skladě slouží pro zadávání či změnu všech atributů, které se zobrazují u konkrétních položek.

Zaměstnanci je umožněno nastavovat u jednotlivých položek základní vlastnosti jako název, ale i mnohem důležitější atributy jako jsou velikost či váha zboží, které dále rozhodují, na jaké pozice může být zboží umístěno z hlediska uzpůsobenosti pozic pro rozdílné váhové a rozměrové hodnoty. V neposlední řadě je umožněno zaměstnanci zadat datum exspirace daného zboží, pokud zboží nějaké má (viz. FP12). Je také umožněno nastavit jakým způsobem se bude zboží ukládat ve vztahu k bezpečnosti práce (například ukládání hořlavých látek do specializovaných skladů, které jsou k tomu určené).

FP5 – Správa objednávek

Objednávky jsou k dispozici jak pro zaměstnance, tak pro zákazníka. Zákazník si jejich prostřednictvím objednává dané zboží a jeho množství. Následně vyplní formulář, ve kterém

upřesní, na jakou osobu bude zpracována konečná faktura a kam bude zboží nakonec dodáno. Všechna pole formuláře jsou neustále kontrolována, zda jsou korektně vyplněna. V případě, že je zákazník zaregistrován, jsou pole obsahující údaje o osobě předem vyplněna. Objednávka vytvářená zaměstnancem je značně zjednodušená, neboť všechny informace (firma a adresa) jsou již předvyplněné. Zaměstnanec pouze zadá zboží, které je třeba dodat do skladu.

Po zadání jsou informace o objednávce daným uživatelem potvrzeny a následně přijde zpětné potvrzení o obdržení objednávky druhou stranou. Před potvrzením je vždy možné pro oba uživatele upravit údaje, avšak po potvrzení tato možnost zaniká. V případě objednávky u zaměstnance je nakonec objednávka archivována, protože stejná objednávka může být pravidelně odesílána (pravidelné zásobování skladu stejný zbožím).

FP6 – Kontrola dostupnosti zboží na skladě

Při tvorbě objednávky je pro zákazníka důležité vědět, kdy může zboží očekávat. Proto je při tvorbě objednávky zákazník vždy po výběru konkrétního typu zboží důrazně upozorněn, že jím požadované zboží není v současné době na skladě. Zároveň s tímto upozorněním je i zákazníkovi sděleno, za jak dlouho může svou objednávku očekávat, což ovšem prodlouží dodací dobu celé objednávky. I na tuto skutečnost musí být zákazník upozorněn.

Z hlediska dostupnosti zboží pro sklad jako takový je nutné dělat pravidelné objednávky (viz FP5 – archivování objednávek). V případě, že zboží, které je skladem poskytováno se přestalo vyrábět, je také nutné tuto informaci zobrazit zákazníkovi.

FP7 – Umožnění vnějšího přístupu

Systém musí zaměstnancům sloužit pro práci se zbožím, ale musí také zajišťovat vnější přístup pro zákazníky. Z toho důvodu musí být systém rozdělen na dva různé uživatelské přístupy. První pro zaměstnance dané firmy, kteří mají přidělená práva a přistupují do různých segmentů systému dle jejich práv (viz FP3).

Druhý typ pro zákazníky je napojen na systém, ale slouží pouze k prohlížení zboží, jež je v současné době nabízeno a jeho objednávání. Zákazník jej využívá k výběru zboží, které si ukládá do košíku. Po potvrzení výběru je zákazník vyzván k vyplnění objednávky a její

potvrzení (viz FP5). Vnější přístup musí být spravován administrátorem, aby bylo možné neustále aktualizovat portfolio.

FP8 - Systém upozorní zaměstnance na minimální zásobu výrobku a jeho vyčerpání.

Zaměstnanci je umožněno nastavovat pro jednotlivé typy zboží jejich minimální zásobu. Minimální zásoba je nastavovaná individuálně na základě množstevní jednotky, ve které je zboží přijímáno a prodáváno (například kusy, metry, litry kilogramy atd.). Zaměstnanec může nastavit minimální zásobu již ve stavu přijímání nového zboží nebo až po jeho naskladnění (pokud zaměstnanec zjistí, že zboží je velice často odebíráno).

Po nastavení minimální zásoby každého zboží bude možno nastavit způsob upozornění (hláška v systému nebo upozorňovací email), který signalizuje kritické množství materiálu nebo snižující se stav. Tyto stavy budou reprezentovány podbarvením položky zboží (červená – kritická zásoba zboží, žlutá – snižující se zásoba, zelená – dostačující zásoba). Ke každému z těchto stavů musí zaměstnanec nastavit takové upozornění, které bude odpovídat situaci.

FP9 - Systém upozorní zaměstnance na zboží, které není aktivně odebíráno

Při práci se zbožím může dojít k situaci, kdy jistá položka není aktivně odebírána zákazníkem. Zaměstnanci již další objednávky tohoto artiklu neprovádějí (viz FP8 - dostačující zásoba), ale zboží zabírá místo, jež by mohlo být využíváno pro skladování žádanějšího zboží. Z toho důvodu budou moci zaměstnanci nastavit u každé položky časovou jednotku (například měsíce nebo roky) a upozornění, které bude signalizovat neaktivitu konkrétního zboží.

Zároveň by jako aktivita nemělo být bráno přeskladnění zboží. Pokud zaměstnanci přeskladní zboží, měl by systém uložit informaci o jeho pohybu, ale nesmí jí evidovat jako odběr materiálu, protože přeskladnění není aktivní odběr zákazníkem a tudíž nemůže charakterizovat spotřebu zboží.

FP10 – Správa volných i obsazených pozic

Při příjmu zboží zadávají zaměstnanci rozměry a váhu nového artiklu do systému (viz FP4). V případě, že se rozměry ukáží nadměrné (o čemž rozhodují fyzické velikosti pozic) musí mít uživatel možnost ověřit si, zda jsou pozice určené pro nadměrné zboží volné, případně zda nejsou obsazené zbožím, které by mohlo být na pozicích pro malé artikly.

Zaměstnanci si budou moci ke každé pozici vytvořit popis charakterizující její rozměry, nosnost, případně speciální ochranné prostředky, které zabraňují poškozování křehkého zboží (například pozice je vybavena šuplíky s ochrannými foliemi pro skladování broušených dílců, keramických nádob nebo skleněných lahví). V případě objednávání musí zaměstnanec zkontrolovat a vyhledat pozice, kam bude nově příchozí zboží založeno, aby nedošlo k poškození vinou špatného sladování.

Zaměstnanci budou též moci vytvářet v systému nové pozice (v případě fyzického rozšíření skladů) a dále u nich charakterizovat jejich vlastnosti pro uchovávání konkrétních artiklů.

FP11 – Nastavení kategorie zboží

Zaměstnanci obchodního oddělení mohou zakládat a případně i editovat či mazat kategorie zboží. Tyto kategorie budou sloužit zaměstnancům příjmu, aby bylo přidělování pozic rychlejší a přesnější. V těchto kategoriích mohou zaměstnanci obchodního oddělení předdefinovat zaměstnancům příjmu skladovací podmínky jednotlivých typů zboží, například hořlavé látky, rozměrné zboží jako je bílá elektronika nebo měkké zboží jakým je oblečení či toaletní potřeby.

FP12 – Systém uživatele upozorní na blížící se exspiraci zboží

Systém upozorní zaměstnance na blížící se datum exspirace zboží (viz. Definice v bodě FP4). Časové rozmezí, ve kterém systém na zboží upozorní, bude volitelné v návaznosti na jeho běžnou trvanlivost při vhodných skladovacích podmínkách.

V případě, že se naplní zmiňované datum, upozorní systém odpovědného zaměstnance, který prověří exspiraci zboží a zajistí jeho rychlé využití (pokud o typ zboží umožňuje).

4.1.2 Nefunkční požadavky

Cílem nefunkčních požadavků je definovat jak bude systém řešit funkční požadavky. Stanovují, jakým způsobem budou v systému funkční požadavky zrealizovány, ať jde o nároky na kvalitu softwaru nebo rychlost dané aplikace. Stejně jako v případě funkčních

požadavků jsou nefunkční požadavky nejdříve shrnuty v tabulce a dále podrobněji popsány dle svých identifikátorů.

ID požadavku	Popis požadavku
NP 1	Systém musí být schopen obsluhovat příslušné zaměstnance celé firmy
NP 2	Systém musí být schopen uložit veškeré informace o stavu zásob, právech uživatelů atd.
NP 3	Systém bude dbát na ustanovení GDPR
NP 4	Systém bude neustále dostupný
NP 5	Systém bude konstruován pro jednoduché a intuitivní ovládání
NP 6	Systém bude obsahovat pro uživatele pouze funkce, které potřebuje pro práci
NP 7	Systém bude přístupný z webu
NP 8	Systém bude požadovat dostatečně silné heslo a změnu hesla každé 3 měsíce
NP 9	Systém bude schopen komunikovat s veškerými potřebnými zařízeními
NP 10	Systém bude automaticky zálohovat data třikrát týdně

Tabulka 2: Nefunkční požadavky (Zdroj: Vlastní tvorba)

NP1 - Schopnost obsluhovat příslušné zaměstnance celé firmy

Navrhovaný systém musí být dostatečně robustní, aby zvládal obsluhovat velké množství uživatelů najednou. Systém bude využíván více odděleními ve firmě, a tudíž jej může využívat mnoho uživatelů najednou.

Nemalou částí uživatelů budou zákazníci, kteří si vybírají a následně objednávají zboží. V případě výpadku serveru by to mohlo znamenat přerušení probíhajících objednávek a v důsledku toho následnou ztrátu aktuálních zákazníků, kteří si zboží objednají u jiného dodavatele. Z tohoto důvodu je nutné zvolit spolehlivého poskytovatele serverů, kteří mohou nabídnout rychlý servis.

NP2 - Uložení informací o stavu zásob, právech uživatelů atd.

Systém bude ukládat velké množství informací a to nejen o zaměstnancích, zákaznících a jejich oprávněních, ale především o zboží na skladech, jejich umístění a množství. Také bude v systému uloženo velké množství objednávek a faktur, které je nutné archivovat. Proto musí být při jeho zavedení k dispozici dostatečně velký úložný prostor. I v tomto případě je na místě volba poskytovatele, který bude schopen poskytnout dostatečnou kapacitu (viz NP1).

V neposlední řadě servery, na kterých bude vše uloženo, musí mít zálohu v případě nenadálého výpadku či napadení.

NP3 - Ustanovení GDPR

Vzhledem k ustanovení GDPR, jehož úkolem je zvýšit ochranu osobních dat občanů, je nutné promyslet, jaké osobní informace bude nutné shromáždit, a to jak pro zaměstnance, tak pro zákazníky. V případě zaměstnanců je nutné zahrnout ustanovení GDPR jak do smlouvy, tak při prvotním vstupu do systému (každý uživatel bude veden pod svým jménem). Pokud se jedná o zákazníky, je nutné je s tímto faktem seznámit v rámci formuláře pro objednání zboží.

NP4 - Systém bude neustále dostupný

V případě nutnosti je v některých situacích (například výdej zboží před vánočními svátky) nezbytná i práce o víkendu. Proto je nutné, aby byl systém dostupný 24 hodin 7 dní v týdnu.

NP5 - Jednoduché a intuitivní ovládání

Při návrhu vzhledu systému je nutné klást důraz na přehlednost jednotlivých ovládacích prvků, neboť nepřehledný či nevkusný vzhled může mít vliv jak na zákazníky, tak na zaměstnance.

Pokud jde o zákazníky, defekt ve vzhledu a funkčnosti může mít neblahý vliv na jejich ochotu nakupovat zboží, a v tomto důsledku přímo ovlivňovat prodeje. V případě zaměstnanců může jít o špatný přehled v aplikaci samotné, načež mohou vznikat zbytečné chyby při manipulaci se zbožím (zboží naskladněno na špatnou pozici), nebo při zpracovávání objednávek (omylem stornovaná platná objednávka).

NP6 - Pouze funkce, které uživatel potřebuje pro práci

Systém rozlišuje mezi zaměstnanci a zákazníky. Zatímco u zákazníků jsou pevně dána omezení pro vstup do systému (přístup k nabízeným produktům, možnost si objednat zboží), u zaměstnanců rozlišuje mnoho typů systémových práv dle jejich zařazení do příslušného oddělení (zaměstnance skladu, obchodního oddělení atd.). U každého oprávnění je nutné přesně stanovit, jaké ovládací prvky musí mít konkrétní typ zaměstnance k dispozici, aby mohl efektivně vykonávat svou práci.

NP7 - Systém bude přístupný z webu

Aby se předešlo tvorbě aplikací zvlášť pro počítače a pro mobilní zařízení (se systémem android) musí být systém přístupný z webového prohlížeče. Používání webové aplikace pro různá zařízení bude řešeno responzivním designem. V návaznosti na tento bod je také nutné, aby byl systém funkční v různých typech webových prohlížečů.

NP8 – Změna hesla

Při prvním přihlášení bude systém požadovat heslo pro přihlášení. Heslo musí být dostatečně silné (heslo musí obsahovat malá a velká písmena, číslo atd.) a musí být pravidelně měněno. Systém bude požadovat, aby uživatele (tedy konkrétně zaměstnanci) každé tři měsíce měnily své heslo, přičemž nové heslo nesmí obsahovat podobnosti s předchozím heslem.

NP9 – Komunikace s veškerými potřebnými zařízeními

Z hlediska pracovních povinností musí být systém schopen komunikovat a pracovat se všemi perifériemi, které budou zaměstnanci využívat. Jedná se o tiskárny, laserové čtečky čárových kódů a další zařízení.

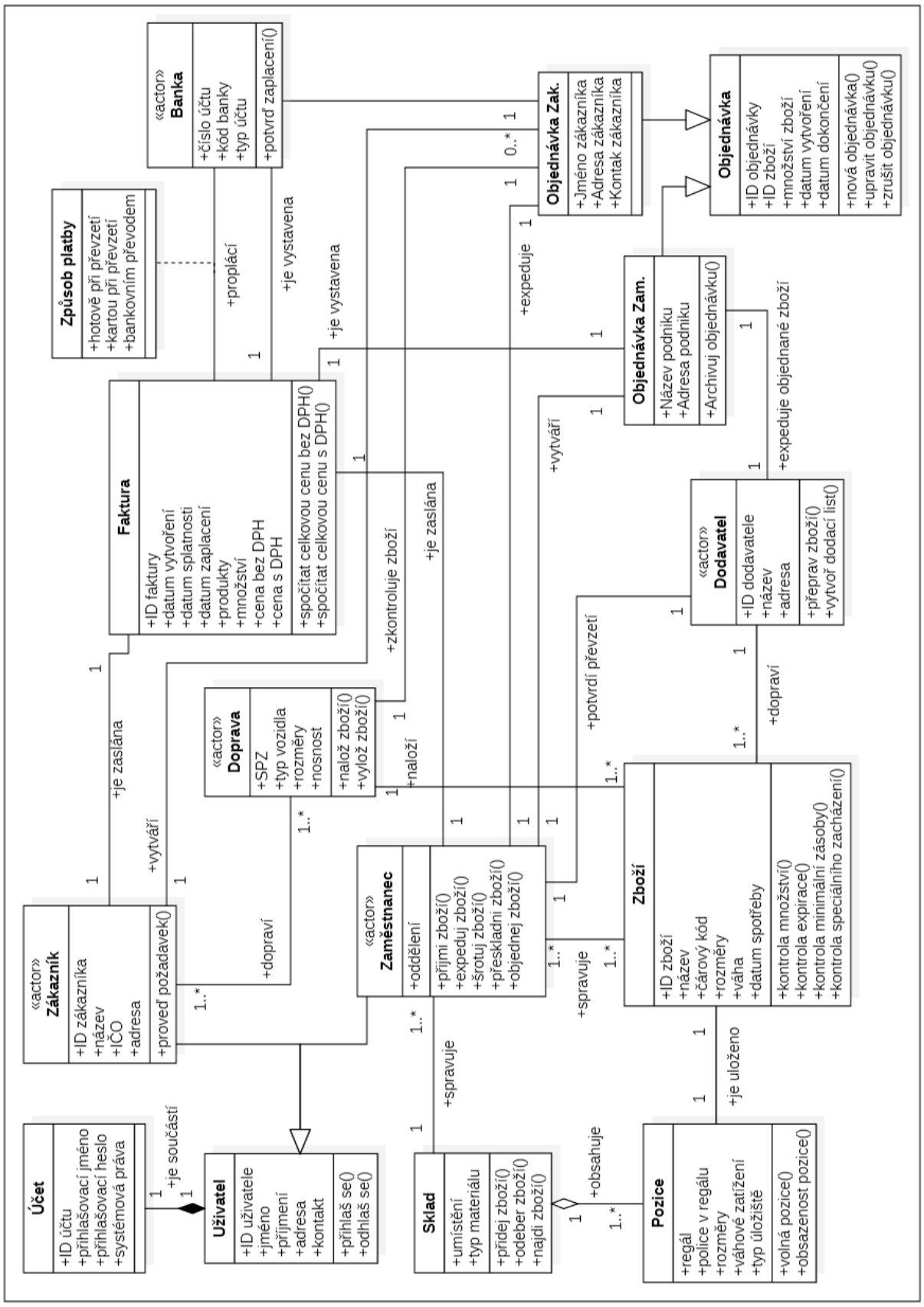
NP10 – Zálohování dat

Systém musí provádět pravidelné automatické zálohy všech informací a to alespoň třikrát týdně. Zálohovací dny musí být uzpůsobeny tak, aby byla záloha provedena vždy po ukončení běžné pracovní směny.

4.2 Model tříd

Prvním ze tří typů modelů, které slouží pro popis budoucího chování systémů je diagram tříd (angl. class diagram). Popisuje statickou strukturu navrhovaného systému. Diagram tříd je nedílnou součástí modelování, neboť jasně vymezuje třídy objektů a jejich vzájemné vazby. Při tvorbě diagramu tříd jsou definovány třídy objektů, nutné pro návrh budoucího systému. Pomáhá graficky vizualizovat a pochopit komplexnost daného systému a komunikace mezi jednotlivými třídami. Každý diagram tříd je následně doplněn o tzv. datový slovník, který popisuje všechny vyobrazené třídy s jejich vazbami.

Je nutné dodat, že diagram tříd uvedený v této práci zahrnuje pouze nezbytné třídy. Vytvoření detailního diagramu tříd a datového slovníku, který by pokryl problematiku takového rozsahu, by vyžadoval větší počet odborných pracovníků, více času a překračoval by rámec rozsahu této práce.



Obrázek 14: Diagram tříd (Zdroj: Vlastní tvorba)

4.2.1 Datový slovník

Uživatel - Třída *Uživatel* reprezentuje skutečného uživatele systému, tedy reálnou osobu, která vstupuje do systému a aktivně s ním pracuje. Tato třída uchovává základní informace jako je jméno, příjmení, adresa a kontakt. Její součástí je třída *Účet*, která dále určuje přístup jednotlivým uživatelům.

Zaměstnanec – *Zaměstnanec* je podtřída třídy *Uživatel* a představuje fyzickou osobu, která přímo pracuje se systémem, jenž slouží pro interní práci ve firmě. Není reprezentací pouze jednoho typu zaměstnance, ale více typů najednou (například zaměstnance skladu, obchodního oddělení nebo managementu). Skladník spravuje skladovací prostory a manipuluje se zbožím (příjem zboží, výdej zboží, přeskladnění a šrotace). Zaměstnanec obchodního oddělení spravuje objednávky sloužící k doplnění zásob.

Zákazník – *Zákazník* (stejně jako zaměstnanec) je podtřídou třídy *Uživatel*. Znázorňuje zákazníka, jenž systém využívá pro vyhledávání jím požadovaného produktu a k jeho následnému objednání.

Účet – Třída *Účet* slouží k přidělování přihlašovacího jména a hesla třídě *Uživatel*. Dále přiděluje práva k prohlížení či manipulaci se systémem (je nutné v systému oddělit typy zaměstnanců a zákazníka).

Sklad – Přestavuje fyzické úložiště, v němž je uloženo *Zboží*. *Sklad* je reprezentován atributy umístění (reprezentace nejen fyzického umístění, ale především vzdálenostní kritéria, nezbytná pro bezpečné skladování nebezpečných látek) a typ materiálu (jedná se o abstraktní vyjádření typů úložných prostor v daném skladu, které slouží pro uchovávání zboží vyžadující si speciální typ zacházení).

Pozice – *Pozice* je agregací třídy Sklad, jenž znázorňuje prostory, které slouží k fyzickému ukládání zboží. Každé zboží je přidělené na pozici, která zohledňuje umístění zboží vzhledem k jeho případné váze (váhové zatížení), výškovému uložení (regál a police v regálu), rozměry zboží (rozměry) nebo speciální podmínky pro uložení jako například nutnost speciálního balení křehkých dílců (typ úložiště).

Zboží – Představuje stěžejní třídu celého diagramu. *Zboží* je třídou reprezentující materiál v jakékoliv formě (polotovar, hotový výrobek), který je firmou přijímán, distribuován či jinak transformován do finální podoby, jenž slouží jako nástroj pro uspokojování potřeb klientů.

Doprava – Třída slouží k přepravě zboží ze skladovacích prostor k zákazníkovi. Atributy třídy představují kapacity konkrétních dopravních prostředků, které využívají při přepravě. Ovlivňují množství a velikost materiálu, který je schopen dopravit k zákazníkovi.

Dodavatel – Třída reprezentující dodávání nového zboží do skladovacích prostor. Primárně slouží k zásobování novým či stávajícím zbožím. Obsahuje základní informace o dodavateli a metody, které představují převoz zboží do skladu a dokumenty sloužící k překontrolování zásob zaměstnancem.

Objednávka – Představuje třídu, jež slouží k objednávání zboží. Je využívána třídou *Zaměstnanec* i *Zákazník*. K rozlišení objednávek mezi těmito dvěma subjekty je použita generalizace, která rozděluje objednávku zvlášť pro zaměstnance a zákazníka.

- Objednávka Zam. – Generalizace vycházející ze třídy objednávka. Udává aktora využívajícího třídu. V tomto případě se jedná o zaměstnance.
- Objednávka Zak. - Generalizace vycházející ze třídy objednávka. Udává aktora využívajícího třídu. V tomto případě se jedná o zákazníka.

Způsob platby – *Způsob platby* je vazební třídou mezi třídami *Faktura* a *Banka*. Určuje možnosti, jakými může být faktura proplacena (hotově při převzetí zboží, platební kartou při převzetí zboží nebo bankovním převodem).

Faktura – Třída, která vychází ze třídy *Objednávka*. Jejím úkolem je vytvořit souhrn zboží, které si *Zákazník* nebo *Zaměstnanec* vytvořil a přidat veškeré náležitosti, které jsou nutné pro jakýkoliv doklad (například data, která jsou nutná pro kontrolu účetních záležitostí nebo celkovou cenu zboží s DPH pro plátce daní či bez DPH pro firmy). Takto vytvořený doklad je dále zasílán zákazníkovi nebo zaměstnanci.

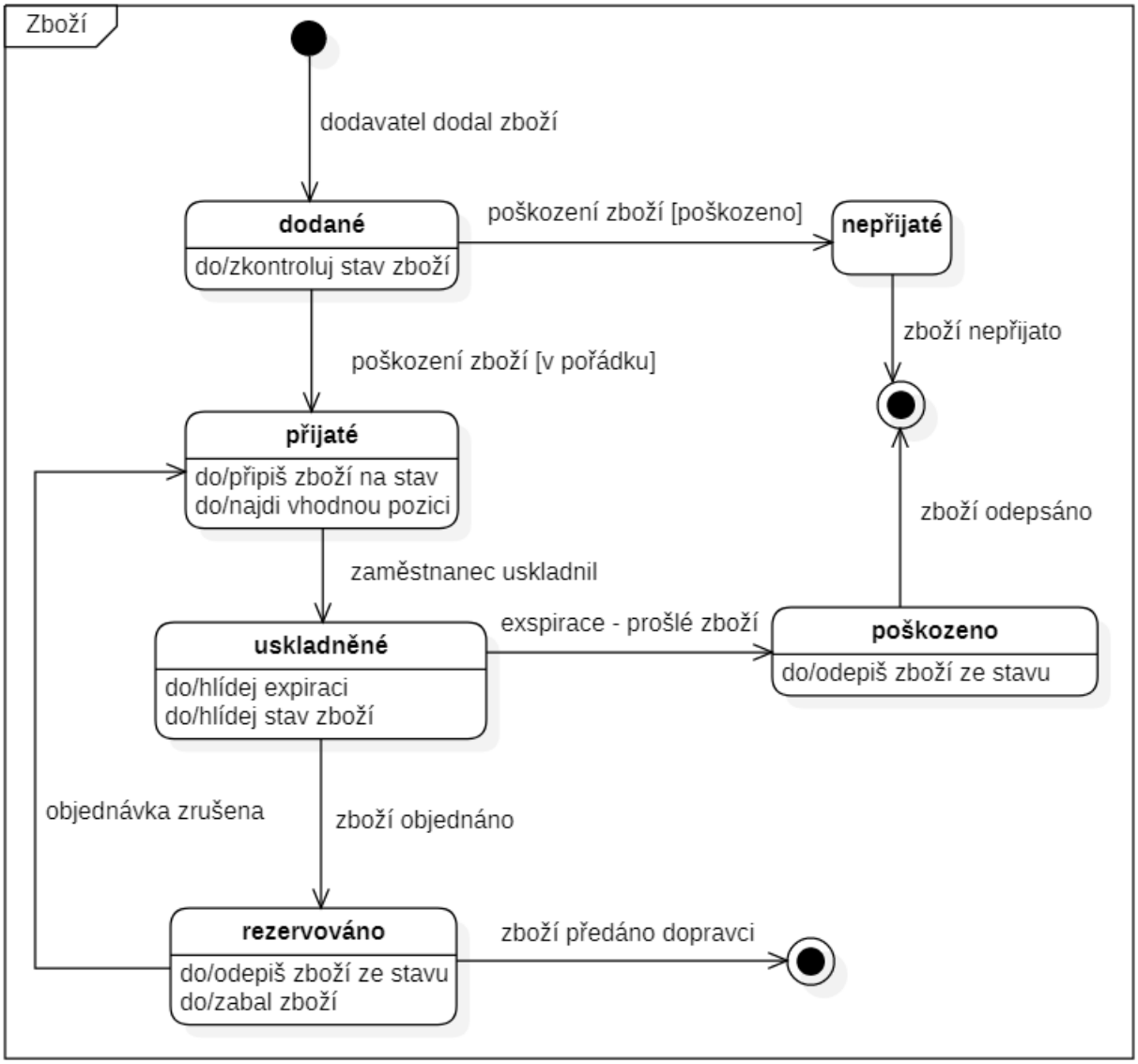
Banka – *Banka* slouží jako konečný milník při zaplacení objednaného zboží (ať už se jedná o objednávku od zákazníka nebo zaměstnance). Zajišťuje proplacení třídy *Faktura* a následně zasílá příslušnému subjektu potvrzení o proběhlých platbách.

4.3 Stavový model

Stavový model se stává z diagramu tříd, které mají dynamické chování. Základem dynamického chování je stav a událost. Stavy charakterizují současný stav třídy, ve kterém se nachází (například vypínač má stavy *zapnuto* a *vypnuto*). Událost určuje podmínku, která musí být splněna, aby třída přešla z jednoho stavu do druhého. Stavový diagram se standardně vypracovává pro každou třídu, ale pro potřeby této práce byly autorem vytvořeny následující diagramy.

Stavový diagram třídy Zboží

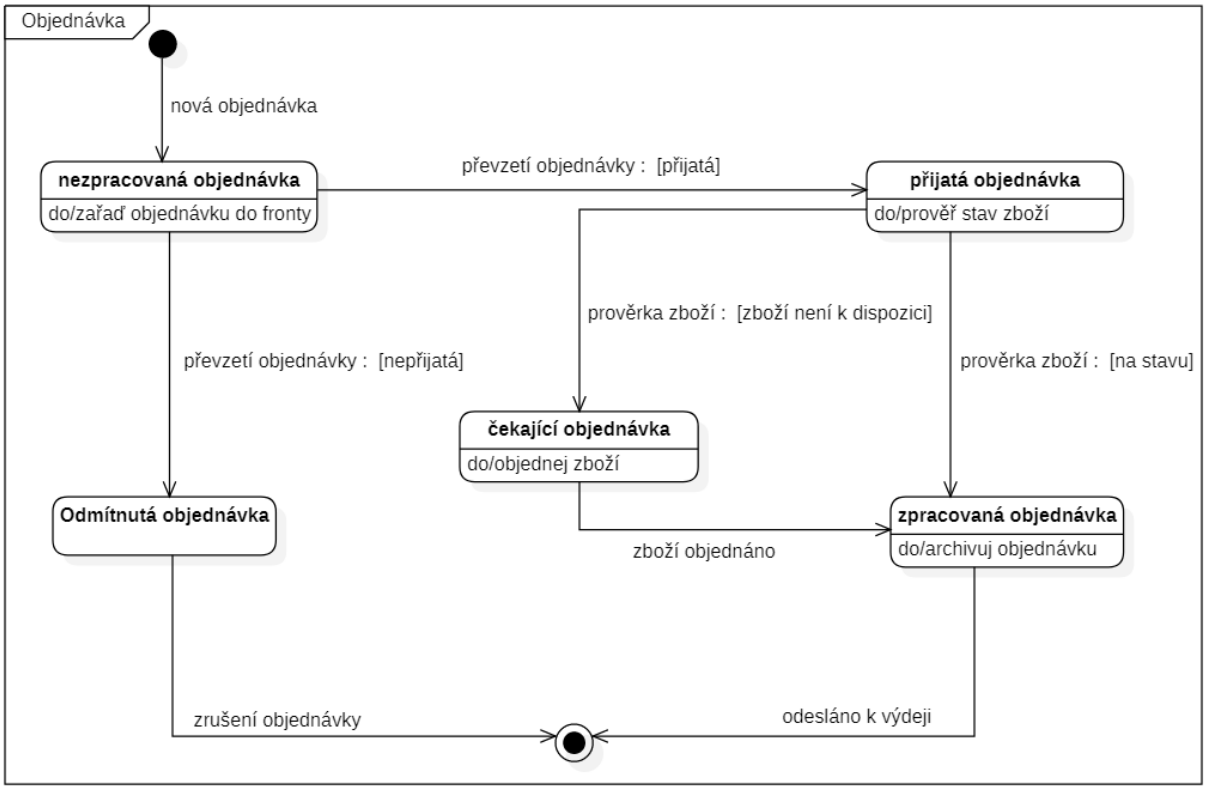
Třída zboží je z hlediska systému jednou z neproměnlivějších tříd. Představuje fyzický materiál, který je potřeba skladovat a hlídat jeho spotřebu. Po dodání zboží dodavatelem do skladu musí zaměstnanec posoudit stav přijatého zboží což je reprezentováno první podmínkou *poškození zboží* (je nesmyslné přijímat zboží poškozené během přepravy, neboť by nebylo prodejné). V případě nepřijetí zboží je z hlediska systému ukončeno. V případě přijetí je zboží přijato a následně uskladněno. Jediným způsobem ukončení je rezervování zboží zákazníkem nebo vyprší exspirace daného zboží.



Obrázek 15: Stavový diagram - zboží (Zdroj: Vlastní tvorba)

Stavový diagram třídy Objednávka

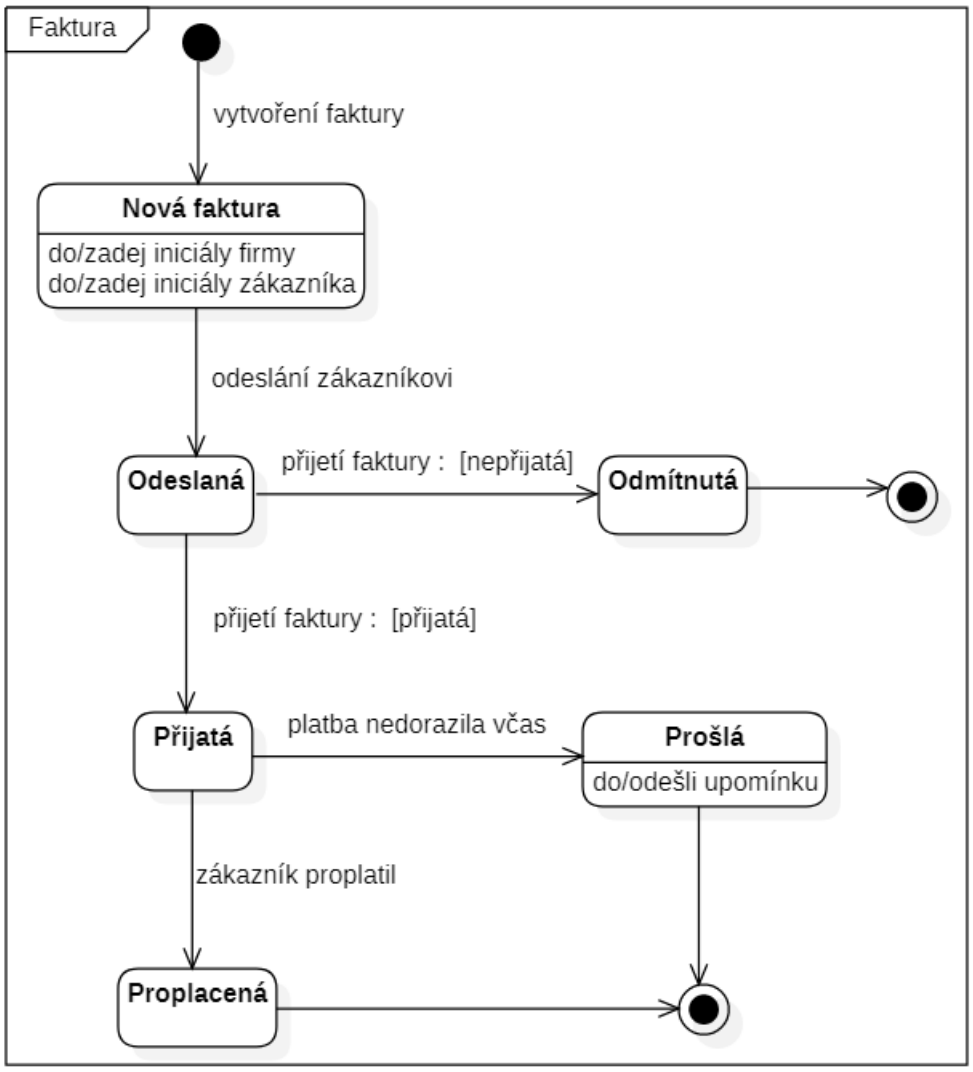
Třída objednávka reprezentuje jakoukoliv objednávku, která je v systému zpracována (objednávky zpracované zákazníky i objednávky zpracované zaměstnanci, které slouží pro doplnění zboží na stav). Objednávka začíná přijetím požadavku, kde čeká, zda bude přijata, nebo odmítnuta poskytovatelem. V případě, že je objednávka přijatá a veškeré zboží dostupné (viz. Podmínka *prověrka zboží*), je zpracována. V případě, že zboží není dostupné je objednávka přesunuta do stavu čekající, kdy musí být potřebné zboží doobjednáno.



Obrázek 16: Stavový diagram - Objednávka (Zdroj: Vlastní tvorba)

Stavový diagram třídy Faktura

Třída faktura vychází ze třídy objednávka. Charakterizuje účet za dodané zboží. Obsahuje popis toho, co se má zaplatit. Vzniká na základě zpracované objednávky, a tím se vytvoří nová faktura. Po odeslání je faktura přijatá nebo nepřijatá (pokud je v ní nalezena chyba či nesrovnalost). Po přijetí se faktura přesune do stavu proplacená nebo prošlá, pokud nebyla proplacena včas.



Obrázek 17: Stavový diagram - Faktura (Zdroj: Vlastní tvorba)

4.4 Model interakcí

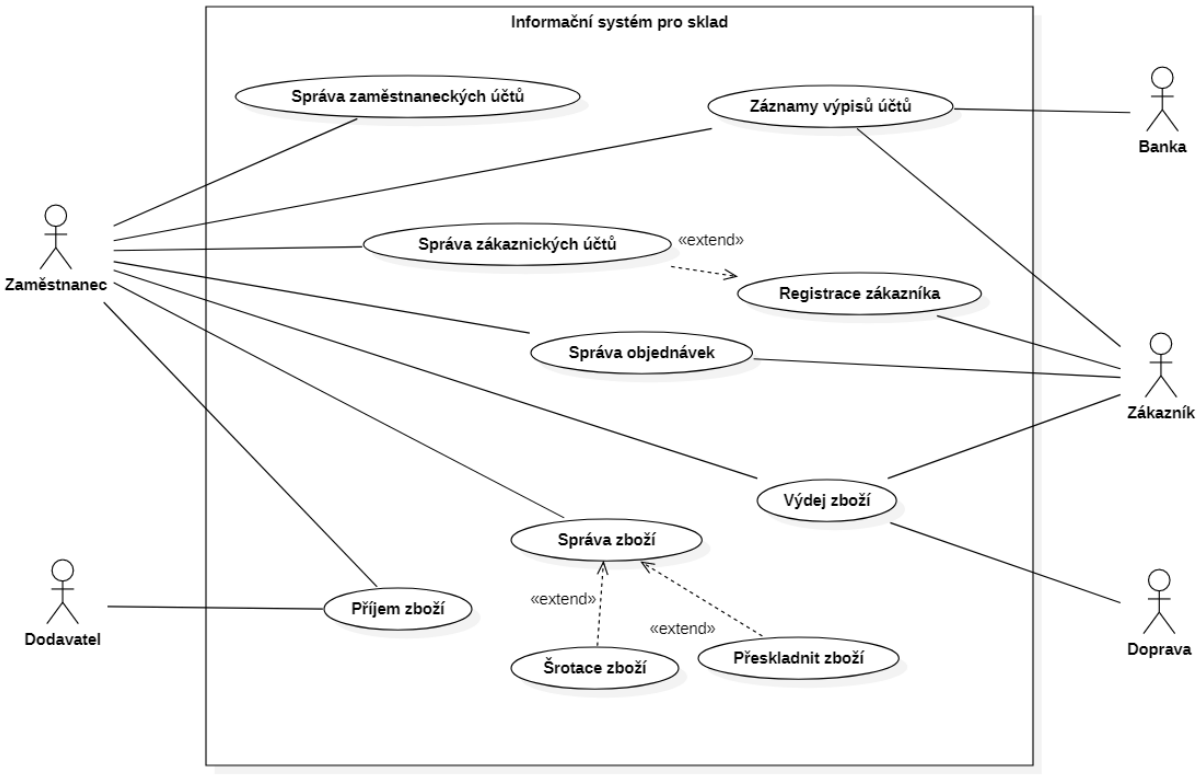
4.4.1 Use case diagram

Use case diagram (nebo také diagram případu užití) je prvním ze tří modelů interakcí a poskytuje obecný pohled na systém a jeho funkčnosti. Popisuje jednotlivé případy užití, které jsou aktivně využívány definovanými aktéry. Use case diagram, vychází především z funkčních požadavků kladených na systém.

V následujícím diagramu jsou uvedeni tito aktéři: *Zaměstnanec*, *Zákazník*, *Dodavatel* a *Doprava*. Aktér *Zaměstnanec* reprezentuje více typů zaměstnanců (skladník, zaměstnanec obchodního oddělení, vedoucí pracovník, administrátor), přičemž plná práva a tudíž přístup do všech částí systému má pouze administrátor. V případě ostatních typů zaměstnanců se mohou přístupy do jednotlivých případů užití lišit.

Zákazník přistupuje do systému především za účelem objednání zboží. V případě, že zákazník objednává zboží často, má možnost využít use case registrace. Registrováním se bude zapsán do databáze zákazníků.

Aktéři *Dodavatel* a *Doprava* vstupují do systému výhradně za účelem dopravy zboží do skladu, nebo jeho převezení k zákazníkovi.



Obrázek 18: Use case diagram (Zdroj: Vlastní tvorba)

4.4.2 Sekvenční model

Sekvenční model je užíván pro podrobný popis chování systému. K popisu používá scénáře a samotné sekvenční diagramy. Scénář zobrazuje použití jednotlivých částí systému. Zobrazuje jednotlivé události, jak následují za sebou, přičemž nebere v úvahu délku jejich trvání. Sekvenční diagram pak popisuje komunikaci jednotlivých systémových komponent či aktérů.

Pro popis pomocí výše zmíněných prostředků sekvenčních modelů byly vybrány některé důležité případy užití a tyto případy byly autorem podrobněji popsány.

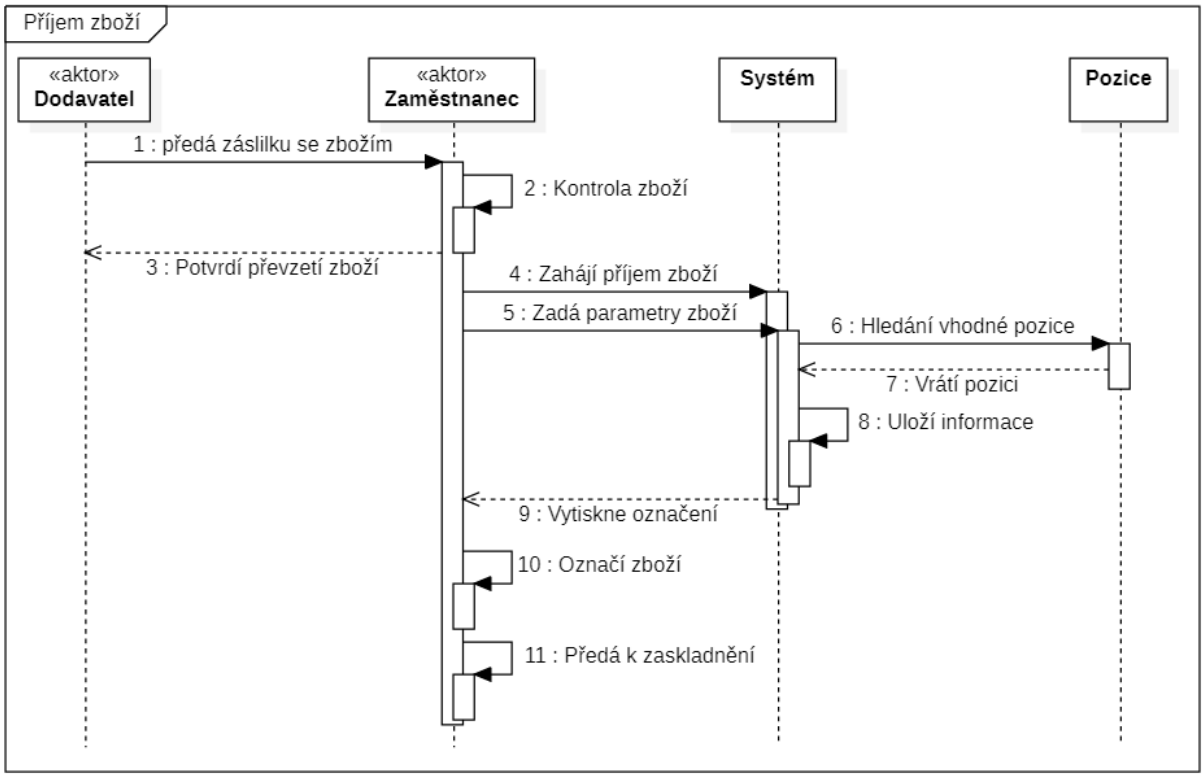
Příjem zboží

Příjem zboží slouží především zaměstnancům skladu (skladníkům nebo vedoucím pracovníkům) přijímat nové zboží na sklad a připisovat jej na stav. V prvním scénáři je nastíněna situace standardního příjmu zboží na sklad, a s tím spojenou komunikaci dodavatele, zákazníka a systému.

Případ užití: Příjem zboží	
Scénář číslo 1	
Popis	Zaměstnanec přijímá zboží na sklad.
Aktér	Zaměstnanec, Dodavatel
Scénář	1) Dodavatel přiveze zásilku se zbožím do skladu. 2) Zaměstnanec zkontroluje zboží. 3) Zaměstnanec zásilku převezme a potvrdí dodavateli převzetí. 4) Zaměstnanec v systému založí nový příjem zboží. 5) Zaměstnanec zadá parametry zboží do systému a potvrdí je. 6) Systém požádá o vhodnou pozici. 7) Databáze pozic odešle fyzické umístění zpět do systému. 8) Systém uloží informaci o zvolené pozici. 9) Systém vytiskne označovací štítek. 10) Zaměstnanec označí zboží štítkem. 11) Zaměstnanec předá zboží k fyzickému uložení.

Tabulka 3: Případ užití - Příjem zboží, Scénář č. 1 (Zdroj: Vlastní tvorba)

Proces začíná předáním zboží zaměstnanci, který potvrdí přijetí dodavateli. Zaměstnanec dále zadá parametry přijatého zboží do systému, který na základě těchto parametrů nalezne místo odpovídající nárokům na skladování konkrétních druhů zboží. Po vyhodnocení vrácení vhodné pozice systémem zaměstnanec označí zboží identifikačním štítkem obsahujícím číslo zboží a jeho umístění.



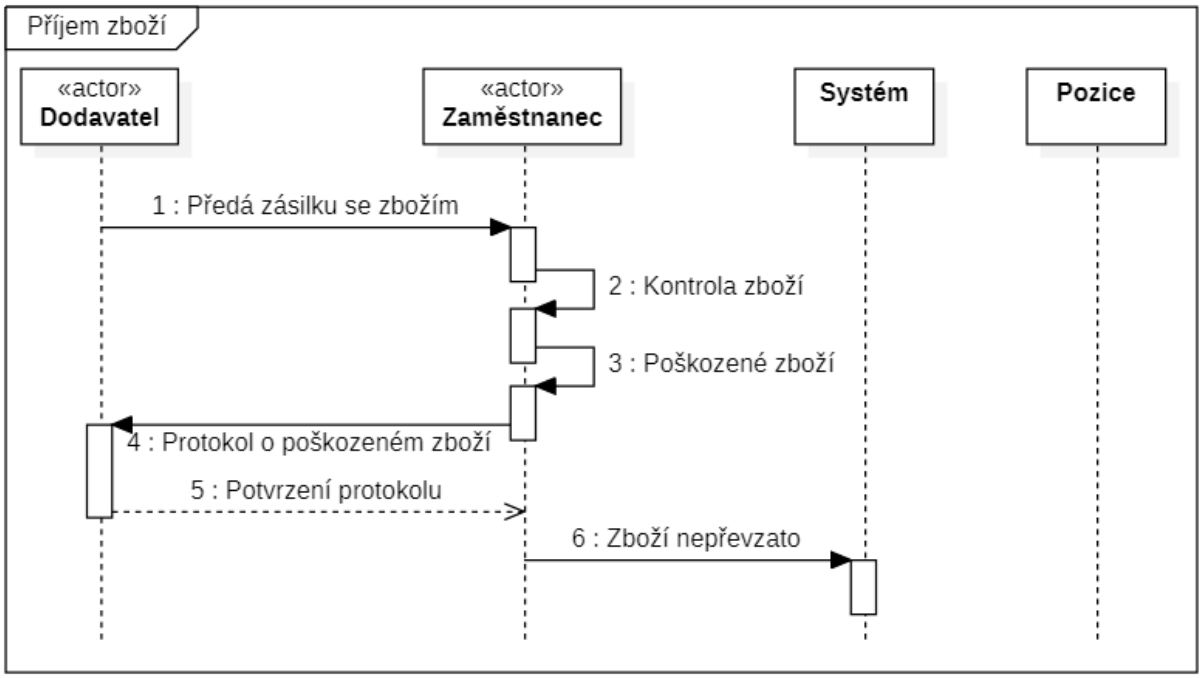
Obrázek 19: Sekvenční diagram - Příjem zboží (Zdroj: Vlastní tvorba)

Druhý scénář příjmu zboží popisuje situaci, kdy dodavatel dopraví zboží do skladu, ale zaměstnanec ho nemůže přijmout, protože zboží je viditelně poškozené, a to by mohlo ovlivnit funkčnost nebo použitelnost zboží.

Případ užití: Příjem zboží	
Scénář číslo 2	
Popis	Zaměstnanec přijímá zboží na sklad.
Aktér	Zaměstnanec, Dodavatel
Scénář	1) Dodavatel přiveze zásilku se zbožím do skladu 2) Zaměstnanec zkontroluje zboží. 3) Zaměstnanec zjistí, že zboží je poškozené a nemůže ho převzít. 4) Zaměstnanec vystaví protokol pro dodavatele o důvodech proč, nebylo zboží převzato. 5) Dodavatel potvrdí zaměstnanci protokol. 6) Zaměstnanec zadá do systému hlášku, proč nebylo zboží převzato.

Tabulka 4: Případ užití - Příjem zboží, Scénář č. 2 (Zdroj: Vlastní tvorba)

Stejně jako v předchozím scénáři dodavatel dopraví zboží do skladu a předá jej zaměstnanci, který si zkontroluje jeho stav. Při kontrole zjistí poškození obalu a zhodnotí, že zboží uvnitř by mohlo být neprodejné. Zaměstnanec vystaví protokoly pro dodavatele a pro sebe, jako podklad pro reklamaci zásilky. Zaměstnanec nakonec musí zadat záznam i do systému.



Obrázek 20: Sekvenční diagram - Příjem zboží (Zdroj: Vlastní tvorba)

Výdej zboží

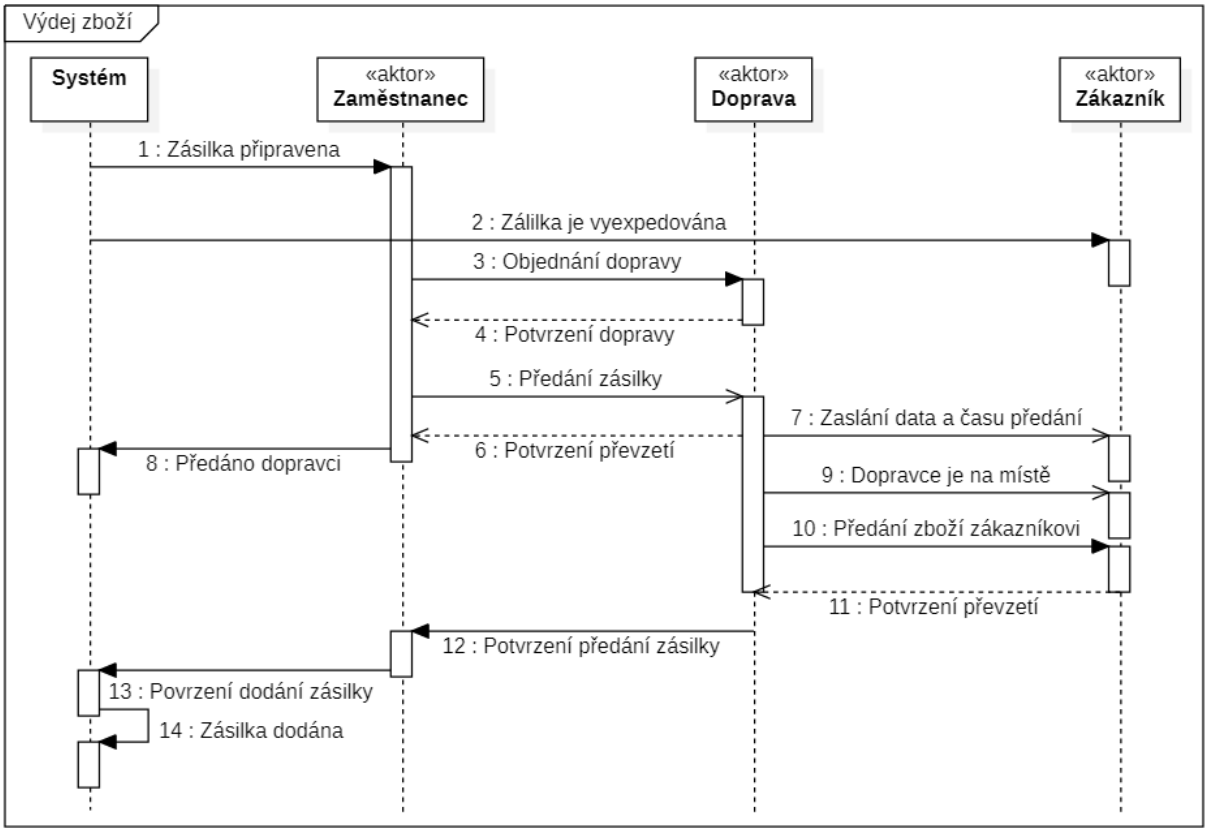
Výdej zboží je nutný pro efektivní a rychlé dodávání zboží zákazníkům. Zboží je dodáno pomocí dopravy (zákazník si zboží může vyzvednout i osobně). Stejně jako v případě příjmu zboží i první scénář pro výdej zboží znázorňuje situaci, kdy doprava úspěšně předala zboží zákazníkovi.

Případ užití: Výdej zboží	
Scénář číslo 1	
Popis	Vyexpedování zboží pro zákazníka
Aktér	Zaměstnanec, Doprava, Zákazník
Scénář	1) Systém informuje zaměstnance expedice o připravené zásilce. 2) Systém informuje zákazníka, že objednávka je připravena k expedici, a kdy může zboží očekávat. 3) Zaměstnanec expedice objedná u dopravce přepravu objednávky k zákazníkovi 4) Doprava potvrdí přepravu zásilky 5) Zaměstnanec expedice předá zboží dopravci 6) Dopravce potvrdí zaměstnanci expedice převzetí zásilky 7) Dopravce zašle zákazníkovi podrobnosti o dni a časovém rozmezí, kdy bude zásilka dopravena. 8) Zaměstnanec expedice zadá do systému, předání zásilky dopravci. 9) Dopravce zašle zprávu zákazníkovi, že dorazil na místo předání. 10) Dopravce předá zásilku zákazníkovi. 11) Zákazník potvrdí převzetí.

	12) Dopravce potvrdí zaměstnanci předání zásilky zákazníkovi 13) Zaměstnanec zadá do systému, že zásilka byla dodána 14) Systém uloží zásilku jako dodanou.

Tabulka 5: Případ užití - Výdej zboží, Scénář č. 1 (Zdroj: Vlastní tvorba)

Prvním předpokladem je připravení a zabalení zboží zaměstnancem, který poté zadá do systému, že je vše připraveno k expedici. V tom okamžiku systém uvědomí zaměstnance expedice o nové zásilce, které je třeba zajistit doprava a zároveň informuje zákazníka, že je zásilka připravena. Poté je zásilka předána dopravci, který jí následně předá zákazníkovi a zadá do systému její úspěšné předání.



Obrázek 21: Sekvenční diagram - Výdej zboží (Zdroj: Vlastní tvorba)

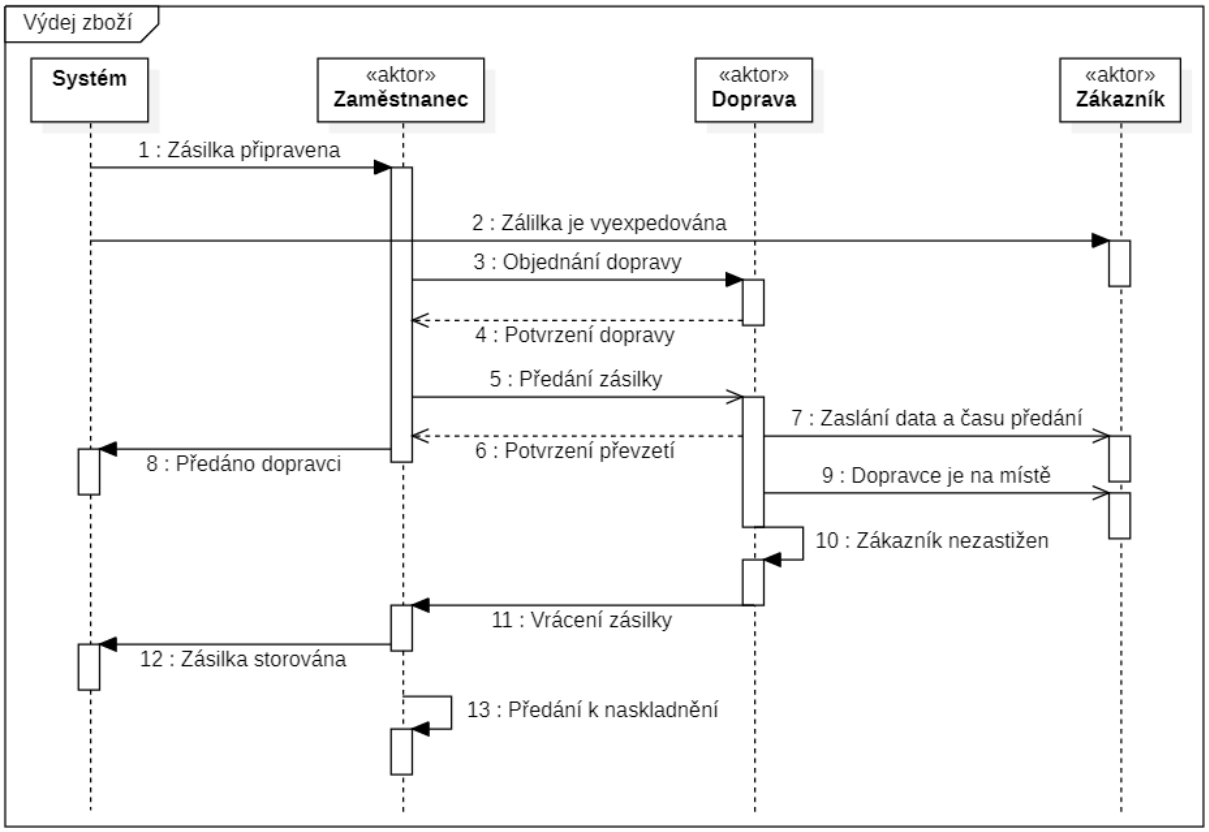
Alternativním scénářem pro výdej zboží je situace, při níž je zboží vyexpedováno, ale zákazník nebyl přes snažení dopravce zastihnut.

Případ užití: Výdej zboží	
Scénář číslo 2	
Popis	Vyexpedování zboží pro zákazníka
Aktér	Zaměstnanec, Doprava, Zákazník
Scénář	1) Systém informuje zaměstnance expedice o připravené zásilce. 2) Systém informuje zákazníka, že objednávka je připravena

	k expedici, a kdy může zboží očekávat. 3) Zaměstnanec expedice objedná u dopravce přepravu objednávky k zákazníkovi 4) Doprava potvrdí přepravu zásilky 5) Zaměstnanec expedice předá zboží dopravci 6) Dopravce potvrdí zaměstnanci expedice převzetí zásilky 7) Dopravce zašle zákazníkovi podrobnosti o dni a časovém rozmezí, kdy bude zásilka dopravena. 8) Zaměstnanec expedice zadá do systému, předání zásilky dopravci. 9) Dopravce zašle zprávu zákazníkovi, že dorazil na místo předání. 10) Dopravci se nepodařilo zkontaktovat zákazníka a předat zásilku. 11) Dopravce vrátí zásilku zaměstnanci expedice. 12) Zaměstnanec expedice zadá do systému, že zásilka nebyla předána a stornuje ji. 13) Zaměstnanec předá zásilku na příjem k opětovnému naskladnění.

Tabulka 6: Případ užití - Výdej zboží, Scénář č. 2 (Zdroj: Vlastní tvorba)

Zboží je zaměstnancem vyexpedováno a předáno dopravci stejně jako ve Scénáři číslo 1 s tím rozdílem, že zákazník nepřevzal zboží od dopravce na smluvené adrese. Po neúspěšném pokuse o kontaktování zákazníka nakonec dopravce odveze zásilku zpět zaměstnanci expedice, který zadá do systému zásilku jako nezaslanou, stornuje ji a předá k příjmu na naskladnění.



Obrázek 22: Sekvenční diagram - Výdej zboží (Zdroj: Vlastní tvorba)

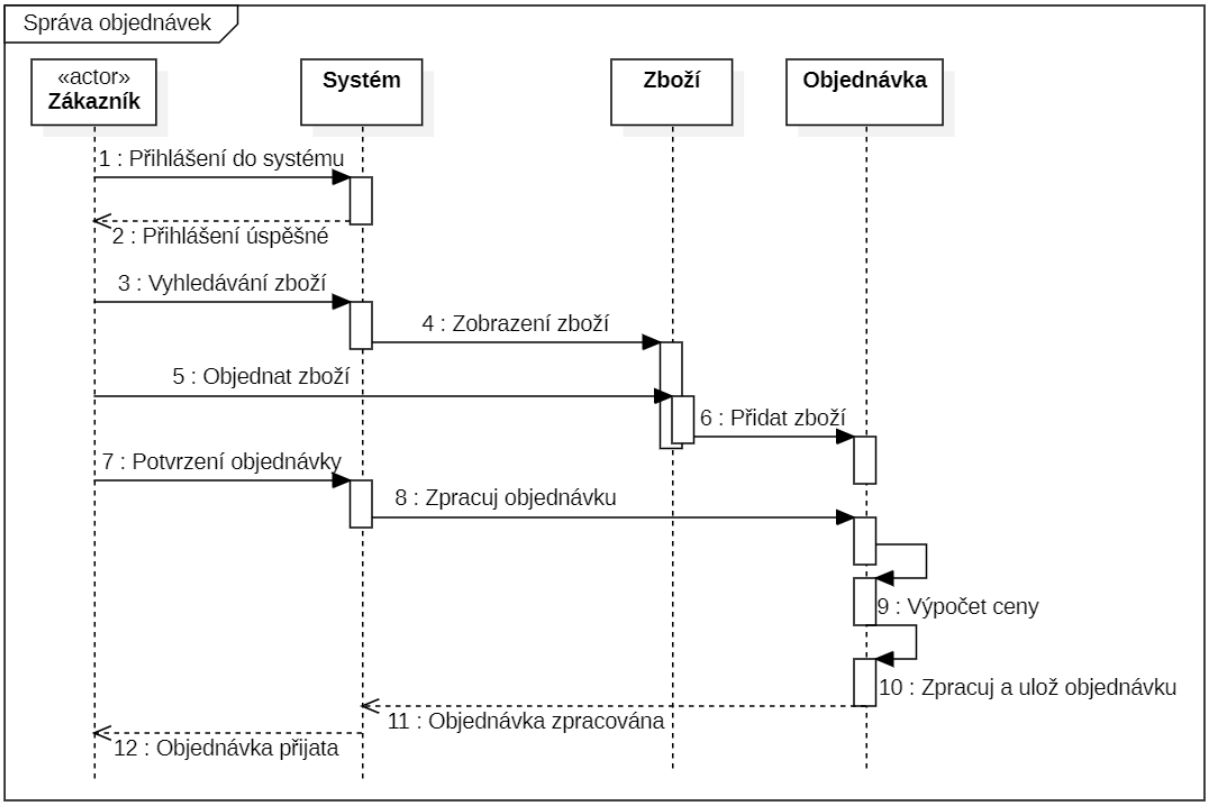
Správa objednávek

Správa objednávek může být využívána jak zaměstnancem, tak zákazníkem. V následujícím sekvenčním diagramu a scénáři, který jej doplňuje je znázorněna objednávka vytvořená registrovaným zákazníkem.

Případ užití: Správa Objednávek	
Scénář číslo 1	
Popis	Registrovaný zákazník objednává zboží
Aktér	Zákazník
Scénář	1) Zákazník se přihlásí do systému 2) Systém potvrdí úspěšné přihlášení 3) Zákazník zadá do vyhledávače systému požadované zboží 4) Systém vyhledá požadované zboží 5) Zákazník objedná zobrazené zboží 6) Zboží je přidáno do objednávky 7) Zákazník systému potvrdí objednávku 8) Systém nechá zpracovat objednávku 9) Objednávka přepočítá celkovou cenu 10) Objednávka je zpracována a uložena 11) Objednávka systému potvrdí zpracování a uložení 12) Systém informuje zákazníka o přijetí objednávky

Tabulka 7: Případ užití - Správa objednávek, Scénář č. 1 (Zdroj: Vlastní tvorba)

Zákazník přistupuje do systému a přihlásí se. Zákazník zadá do systému požadované zboží a systém ho nechá třídu zboží zobrazit. Následně je zboží objednáno a přidáno do objednávky. Následně zákazník potvrdí objednávku, objednávka přepočítá celkovou cenu, zpracuje objednávku a uloží ji. Nakonec objednávka informuje systém, že vše bylo úspěšně zpracováno a uloženo do fronty. Systém nakonec informuje zákazníka o úspěšně přijaté objednávce.



Obrázek 23: Sekvenční diagram - Správa objednávek (Zdroj: Vlastní tvorba)

Správa zboží

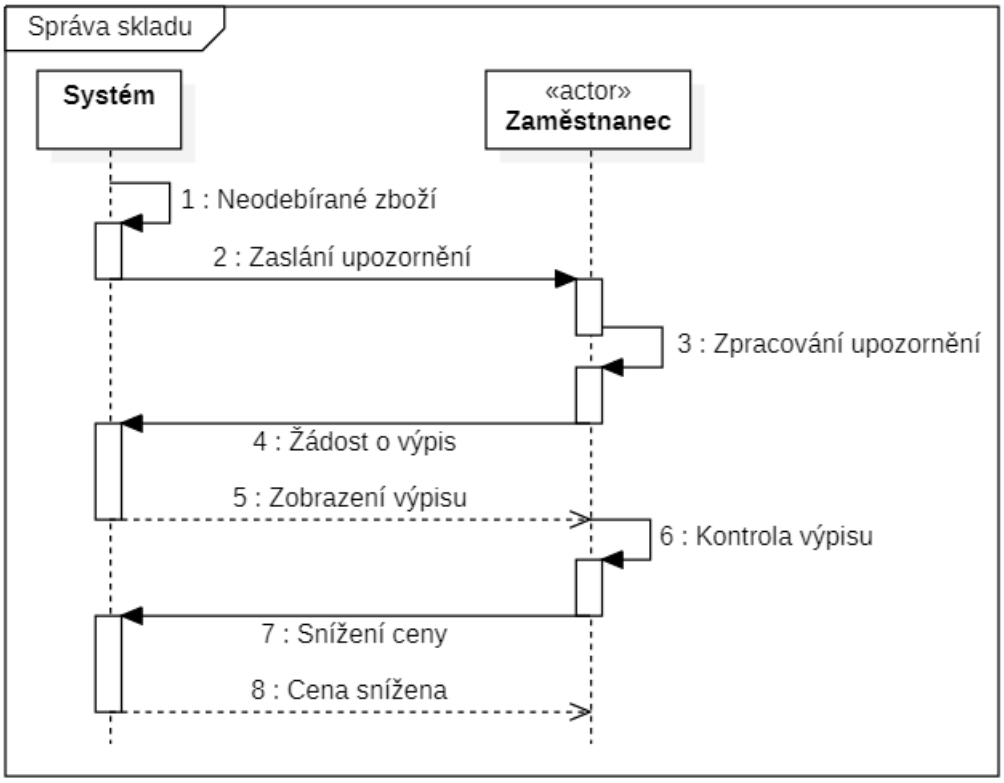
Správa zboží slouží zaměstnancům k manipulaci se zbožím na pozicích, změně skladovacích podmínek nebo kontrolování stavu zboží na pozici. Dále také slouží systému pro předávání upozornění zaměstnancům ohledně stavu zboží.

Případ užití: Správa zboží	
Scénář číslo 1	
Popis	Systém upozorní zaměstnance na neodebírané zboží
Aktér	Zaměstnanec
Scénář	1) Systém nalezne zboží, které nebylo delší dobu aktivně odebíráno. 2) Systém upozorní zaměstnance na neodebírané zboží. 3) Zaměstnanec si převezme upozornění od systému.

	4) Zaměstnanec systém požádá o výpis pohybu zboží. 5) Systém zobrazí výpis pohybu zboží. 6) Zaměstnanec zkontroluje historické příjmy a výdeje zboží. 7) Zaměstnanec na základě kontroly příjmů a výdejů sníží cenu zboží v systému. 8) Systém potvrdí změnu ceny.

Tabulka 8: Případ užití - Správa zboží, Scénář č. 1 (Zdroj: Vlastní tvorba)

Systém zasílá zaměstnancům různá upozornění dle nastavení při příjmu. V předchozím scénáři a následujícím diagramu je popsána situace, kdy systém upozorní odpovědného zaměstnance na zboží, které již delší dobu nebylo aktivně odebíráno. Zaměstnanec si převezme požadavek a systém tak může vyhodnotit, že požadavek je zpracován. Zaměstnanec si ze systému nechá zobrazit výpis všech informací o zboží a jeho pohybech na pozicích. Na základě výpisu se rozhodl snížit cenu zboží. Systém potvrdí provedenou změnu.



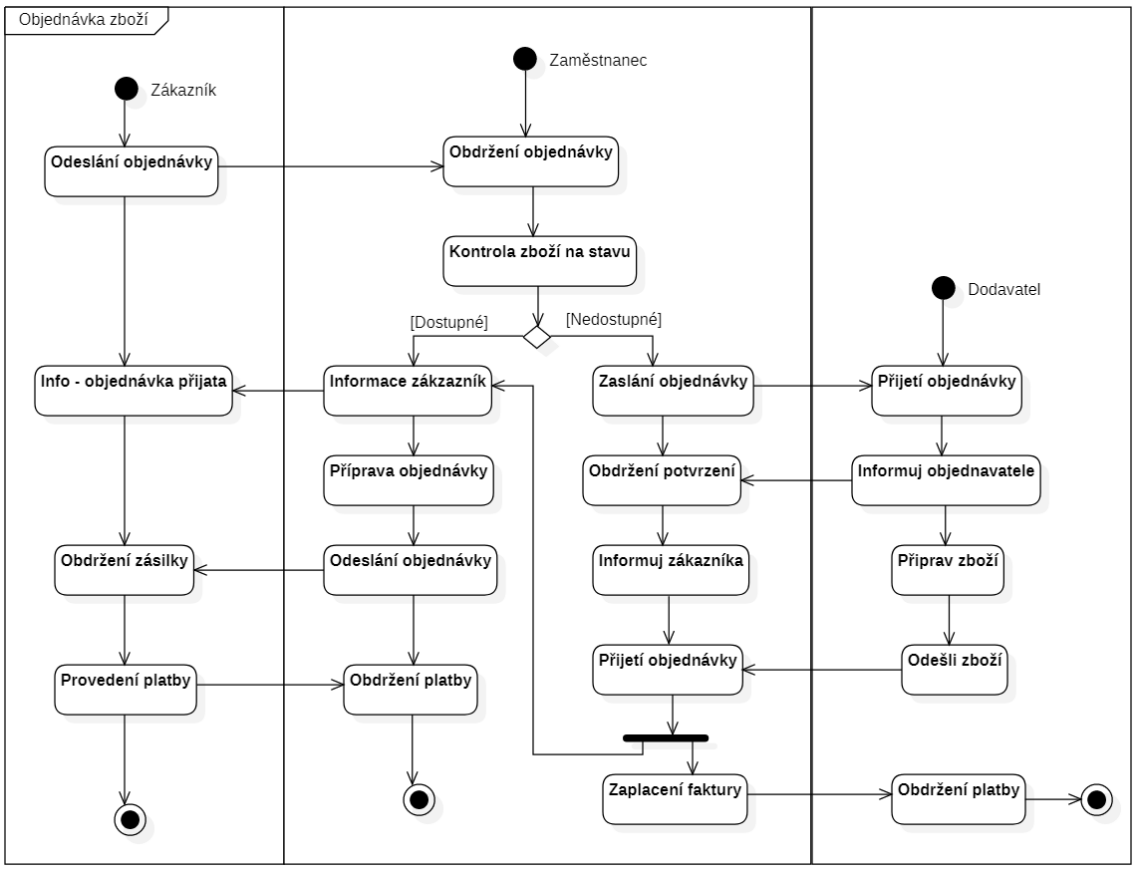
Obrázek 24: Sekvenční diagram - Správa skladu (Zdroj: Vlastní tvorba)

4.4.3 Diagram aktivit - objednávka

Diagram aktivit reprezentuje nejdetailnější popis systému. Slouží k vyobrazení a popisu složitého algoritmu nebo procesu, který je z hlediska systému náročný a může se v rámci chodu systému často opakovat.

Následující diagram popisuje objednávku vytvořenou zákazníkem. Tato objednávka se může zdát z hlediska běžného uživatele jako marginální, avšak z hlediska systému může jít o náročný proces. A k popisu těchto procesů slouží diagram aktivit, díky němuž lze znázornit celý proces s jednotlivými posloupnostmi kroků včetně konkurenčních či souběžných toků.

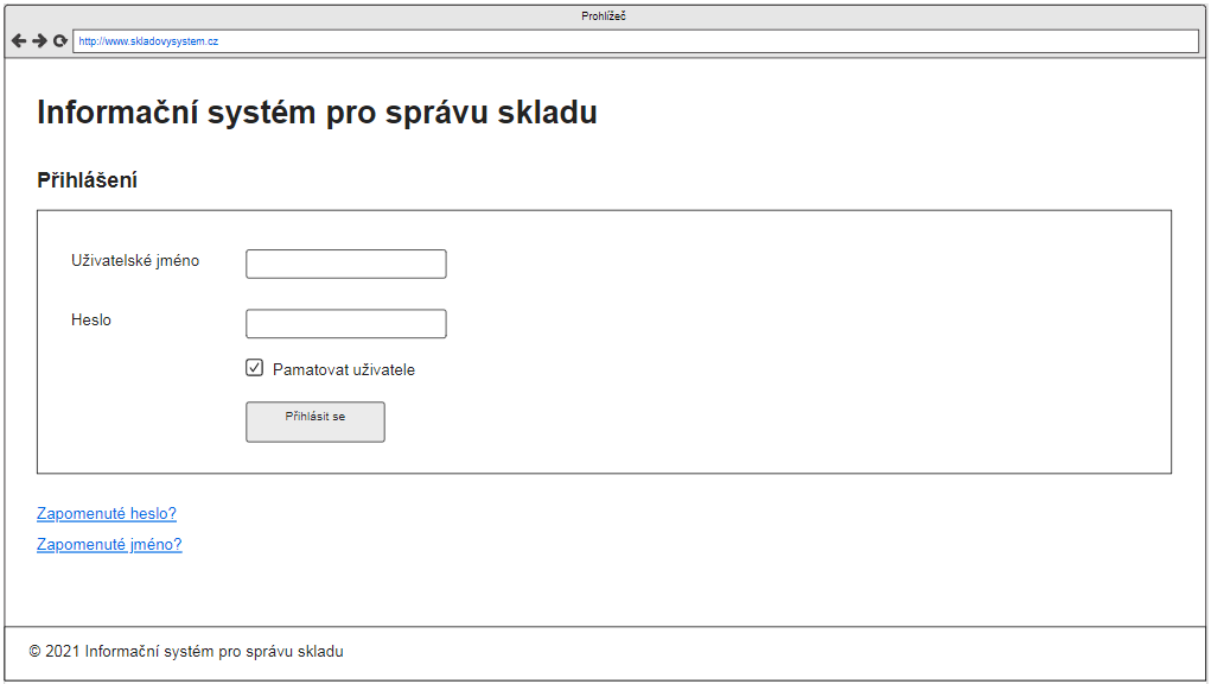
Diagram aktivit popisuje proces, kdy si zákazník objedná zboží, dostane potvrzení a po obdržení zásilky za ní zaplatí. Diagram však dále zohledňuje situaci, kdy zaměstnanec při zpracování objednávky některé ze žádaného zboží není na skladě a musí být doobjednáno od dodavatele. Procesy dodavatele jsou taktéž zohledněny v diagramu. V neposlední řadě lze následující diagram aplikovat i v případě předobjednávek.



Obrázek 25: Diagram aktivit - Objednávka zboží (Zdroj: Vlastní tvorba)

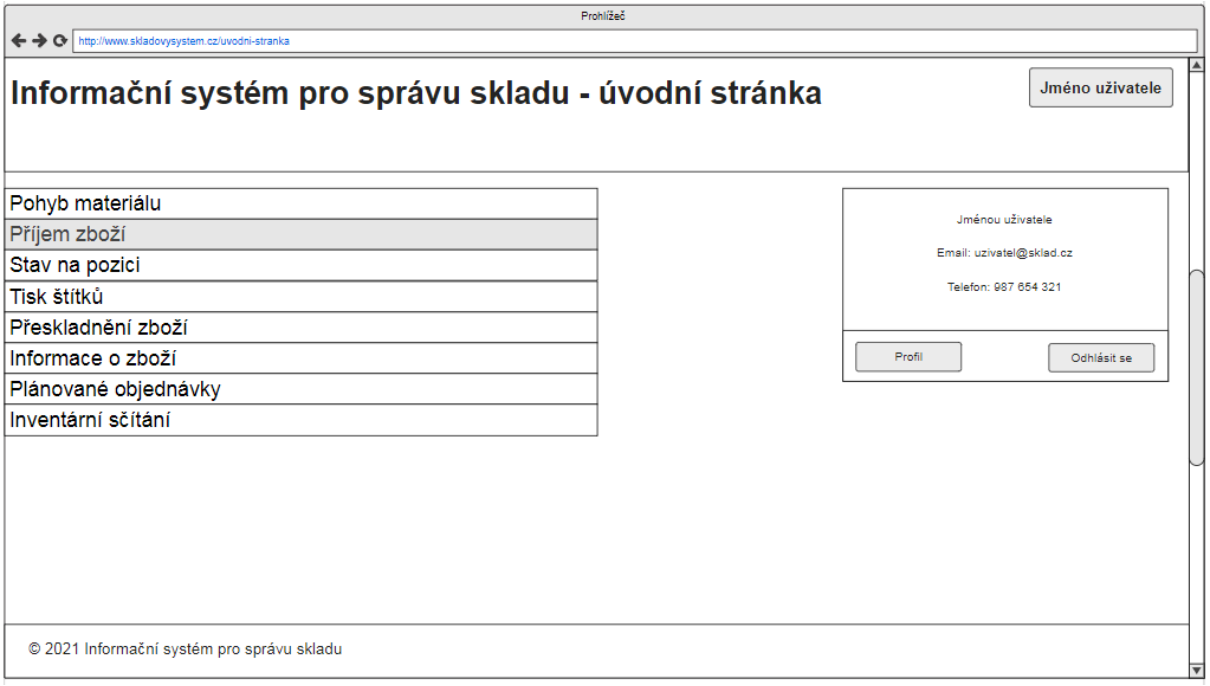
4.5 Návrh vzhledu systému

Dříve použité modely UML slouží pro popis struktury a funkčnosti systému. Pro návrh vzhledu systému se autor rozhodl použít „wireframe“ (jedná se o grafický návrh, který definuje funkce a obsah stránek webové aplikace). Wireframe pomáhá osobám, které neovládají jazyk UML lépe pochopit funkce a přínos navrhovaného systému. Následující wireframy zobrazují vzhled systému pro zaměstnance, kteří s ním budou pracovat.



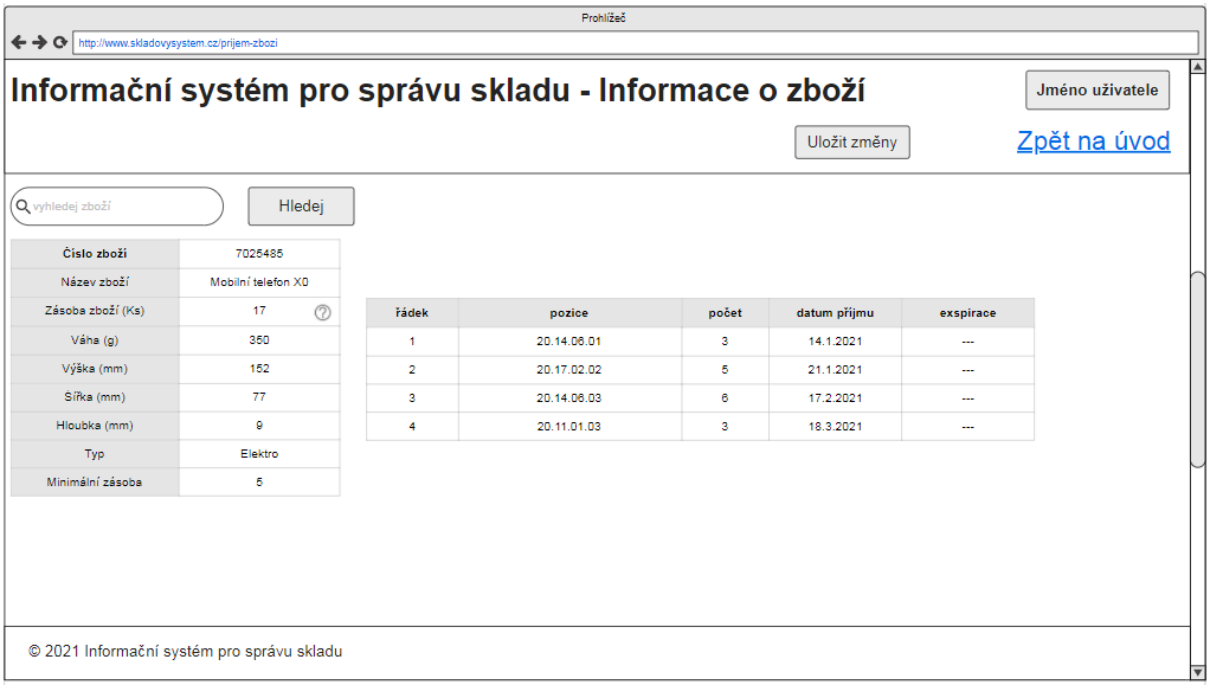
Obrázek 26: Wireframe - Přihlášení (Zdroj: Vlastní tvorba)

První grafický návrh vyobrazuje přihlašovací bránu pro jakéhokoliv zaměstnance. Záhlaví a zápatí stránky bude fixní pro všechna dialogová okna systému. Tedy v záhlaví je název systému, případně může být doplněno o logo firmy, která jej bude využívat. Přihlašovací stránka obsahuje textová pole pro zadání jména a hesla, jež umožní zaměstnanci se přihlásit a využívat jemu přidělené funkce. V případě, že uživatel používá pravidelně jednu pracovní stanici, může využít možnost „Pamatovat si uživatele“. Tato možnost způsobí, že uživatelské jméno bude stále vepsaná v poli „Uživatelské jméno“ a zaměstnanec, tak zadává pouze heslo. Přihlašovací okno je doplněno o odkazy v případě zapomenutého hesla či jména. Tyto odkazy slouží pro vygenerování nového přístupu na emailovou adresu, která je spárována s uživatelským přístupem.



Obrázek 27: Wireframe - Úvodní stránka (Zdroj: Vlastní tvorba)

Druhý wireframe zobrazuje systém po úspěšném přihlášení. Uživateli je vyobrazena jeho hlavní pracovní nabídka. Tato nabídka je charakterizována dle jeho přístupových práv, která mu byla přidělena administrátorem při vytvoření účtu. Návrh ukazuje hlavní nabídku zaměstnance skladu na příjmu zboží. Jak bylo zmíněno výše, záhlaví a zápatí stránky jsou fixní pro všechna okna. Záhlaví se však při přihlášení změní a obsahuje navíc jméno přihlášeného uživatele, které se nachází v pravém horním rohu. Při kliknutí na zmíněné jméno se objeví dialogové okno, obsahující informace o uživateli, jeho přiřazenou emailovou adresu, případně i telefonní číslo. Dále je zde možnost spravovat svůj profil. Mezi možnosti správy účtu patří především změna hesla. Zmíněné dialogové okno též nabízí možnost odhlásit se ze systému.



Obrázek 28: Wireframe - Informace o zboží (Zdroj: Vlastní tvorba)

Poslední wireframe ukazuje pohled zaměstnance na detail zboží. Zaměstnanec může použít vyhledávač k nalezení jakéhokoliv zboží, které je zaregistrované v systému. Detail zboží se objeví i v případě, že žádné zboží není na stavu. Systém zobrazí detailní popis zboží, jako je počet kusů na skladě, rozměry, váha nebo typ, dle kterého je zařazen na systémem definovanou pozici. V případě, že chce zaměstnanec vědět, kde se nacházejí jednotlivé kusy, může k tomu využít ikonu lupy vedle záznamu „Zásoba zboží“, přičemž systém ukáže dodatečnou tabulku s informacemi o jednotlivých pozicích, na kterých se zboží nalézá. Detail umožňuje měnit číslo zboží, název zboží nebo i parametry. Na konci jakýchkoliv změn musí uživatel tyto změny potvrdit stiskem tlačítka „Uložit změny“. Pokud zaměstnanec dokončil práci a chce se vrátit na úvodní stránku, použije „Zpět na úvod“ v pravém horním rohu obrazovky.

5 Výsledky a diskuse

Návrh informačního systému pro správu skladu byl vytvořen za účelem zjednodušení procesů souvisejících se zprávou zboží, jako je příjem zboží na sklad, výdej zboží, uskladňování nebo kontrolou zboží na stavu.

Samotný návrh vychází z praktických zkušeností a sběru požadavků kladených na systém. Tyto požadavky byly formulovány na základě faktických nedostatků, které původní systém nebral v úvahu. Požadavky jsou rozděleny dle notace na funkční a nefunkční. Funkční požadavky definují nedostatky systému, které souvisí s přímým užíváním. Nefunkční požadavky pak definují zákonné a bezpečností prvky, které jsou pro funkčnost systému nutné.

Navrhovaný systém řeší problémy se skladováním zboží a přiřazováním zboží na vhodné pozice. Systém umožňuje u každého zboží definovat typ, rozměry a váhu zboží a na tomto základě dále systém přiřazuje pozice, na kterých bude zboží uloženo. Dále systém aktivně upozorňuje zaměstnance na exspiraci nebo na zboží, které není zákazníky aktivně odebíráno, přičemž rozlišuje, zda byl pohyb materiálu způsoben nákupem zboží či přeskladněním. Upozornění lze také nastavit na jednotlivé typy zboží, a to na minimální zásobu, kterou je nutné dodržovat vzhledem k četnosti odběru materiálu. Tato práce je aplikovatelná na většinu typů skladovacích prostor, které se zabývají dlouhodobějším skladováním materiálů. V případě krátkodobého skladování například potravin by tento systém zcela nevyhovoval. Při tvorbě takto specifických systémů je v praxi velice důležité formulovat požadavky nejen na základě manažerského ohledu, ale také z pohledu zaměstnanců, kteří budou systém každý den využívat.

Při tvorbě návrhu systému bylo objeveno mnoho prací a odborných pramenů, které se zabývaly tvorbou informačních systému. Většina těchto prací se zabývala systémy pro správu internetových obchodů, a tudíž nelze tuto práci přímo porovnat s jinými autory. Nelze však zároveň vyloučit, že prvky, které tato práce obsahuje, mohou sdílet jistou podobnost s jinými odbornými prameny, které se zabývali výše zmíněnými systémy pro správu internetových obchodů, jako je například proces tvorby objednávky, faktury nebo způsob jakým je odesíláno zboží zákazníkovi.

Navrhovaný systém pokrývá mnoho chyb, které se v praxi běžně vyskytují. Je jich však mnoho a rozsah této práce neumožňuje je všechny efektivně reprezentovat a navrhnout účinné řešení problematiky. V případě většího časového horizontu by mohl návrh obsahovat řešení na problém opakovaného příjmu zboží, kdy systém odebíral zboží náhodně a nikoliv dle metody FIFO. Řešením je datum příjmu jednotlivých kusů zboží. Systém by pak dále musel obsahovat pojistku, která by si pamatovala pozici, na které je zboží uloženo a neumožnilo by novému příjmu uložení na té stejné pozici.

Při shrnutí celého postupu tvorby tohoto informačního systému je jasné, že problematika tohoto rozsahu by při větším časovém fondu jistě mohla být obsáhlejší. Zahrnuté diagramy a popisy funkcí jsou základním stavebním kamenem pro návrh, avšak existují další metody, které by mohli pochopení problematiky prohloubit a přinést lepší řešení pro budoucí vývoj takovýchto specifických systémů. V případě, že by celý tento návrh měl být formulován od začátku, tak by autor postupoval stejně, avšak kladl by větší důraz na formulování požadavků a na jejich detailnější vykreslení pomocí vhodných nástrojů, jako jsou diagramy jazyka UML.

6 Závěr

Cílem této závěrečné práce bylo navrhnout informační systém pro sklad pomocí jazyka UML, který by splňoval nároky kladené na takovýto systém. Práce je rozdělena na teoretickou a praktickou část. V teoretické části jsou popsány definice systému dle různých autorů, a dále pak historické i současné přístupy pro tvorbu informačních systémů. Praktická část popisuje navrhovaný systém.

Práce staví na jednom z nejdůležitějších faktorů každého návrhu, na charakteristice funkčních a nefunkčních požadavků. Charakteristika požadavků definuje jednotlivé nároky zákazníka a kritéria, která musí systém splňovat. Správné definování požadavků je klíčové pro tvorbu jakéhokoliv systému, neboť špatně definované požadavky vedou k nesprávně fungujícímu výsledku, který následně musí být přepracován. Toto přepracování výrazně přečerpává časové i finanční zdroje. Požadavky na systém byly sestaveny na základě osobních zkušeností autora při práci v logistice skladu. Definované požadavky slouží k odstranění nedostatků již při návrhu systému. Tyto nedostatky komplikovaly práci a způsobovaly rozdílné informace mezi reálným stavem a informacemi v systému.

Požadavky byly rozděleny na funkční a nefunkční, a dále podrobněji popsány pomocí diagramu tříd, stavového diagramu a modelu interakcí. Diagram tříd v rámci rozsahu této práce vyobrazil důležité třídy, které mají vliv na pohyb a manipulaci se zbožím. Pomocí stavového diagramu byly popsány třídy s nejvíce dynamickým chováním, popisující jednotlivé stavy tříd a přechodové události mezi stavy. Model interakcí doplnil návrh o diagram užití, který představil základní funkcionality navrhovaného systému a tyto funkcionality byly detailně popsány pomocí sekvenčních diagramů a scénářů. Scénáře byly využity k charakterizování nejen standardních i alternativních situací. K popisu řízení informačních toků byl použit diagram interakcí. Závěr práce je doplněn o grafický návrh systému, který slouží k lepší představě o praktickém fungování systému.

Pro nasazení systému tohoto rozsahu do jakékoliv firmy by bylo nutné definovat další sadu požadavků, neboť každá firma pracující s uskladněním zboží, může mít více nároků a požadavků na systém, jako je například doba uložení zboží na pozici, nebo častější kontrola stavu zboží. Avšak tento návrh lze použít jako dobrý základ pro vytvoření systému pro správu skladu.

7 Seznam použitých zdrojů

1. Sodomka, Petr. *Informační systémy v podnikové praxi.* místo neznámé : COMPUTER PRESS, 2011. str. 504. 978-80-251-2878-7.

2. Molnár, Zdeněk. *Podnikové informační systémy.* místo neznámé : Česká technika - nakladatelství ČVUT, 2009. 978-80-01-04380-6.

3. Pojem informačního systému. *fi.minu.cz.* [Online] [Citace: 3. 2 2020.] https://www.fi.muni.cz/~smid/mis-infsys.htm.

4. Aalst, Wil M.P. van der. *Modeling Business Processes.* místo neznámé : The MIT Press, 2011. 978-0262015387.

5. Ezenwa, Joseph. linkedin. *ENTERPRISE INFORMATION SYSTEM AND ENTERPRISE RESOURCE PLANNING.* [Online] 21. 4 2016. https://www.linkedin.com/pulse/enterprise-information-systemeis-resource-planning-erp-why-ezenwa/.

6. FOSSLOOK. *Enterprise Information Systems Definition.* [Online] 2001. https://fosslook.com/articles/enterprise-information-systems.

7. management mania. *Business Intelligence.* [Online] 2011. https://managementmania.com/cs/business-intelligence.

8. Businessworld. *Co je to business intelligence.* [Online] 1999. https://businessworld.cz/ostatni/co-je-to-business-intelligence-7157.

9. Bluedynamic. *Co je ERP.* [Online] 20. 11 2018. https://bluedynamic.cz/blog/co-je-erp-enterprise-resource-planning/.

10. Oracle. *Co je ERP?* [Online] 2020. https://www.oracle.com/cz/applications/erp/what-is-erp.html.

11. Web archive. *Enterprise resource planning: Implementation procedures.* [Online] 2003. https://web.archive.org/web/20150218161616/http://www.student.oulu.fi/~jolahti/accinfo/9%20ERP%20Implementation%20procedures.pdf.

12. Křižko, Ivo. systemonline. *SCM: Supply Chain Management.* [Online] 10 2002. https://www.systemonline.cz/clanky/scm-supply-chain-management.htm.

13. altaxo. *Supply Chain Management.* [Online] 2019. https://www.altaxo.cz/provoz-firmy/management/rady-pro-manazery/supply-chain-management.

14. Žebrák, Miroslav. bussinessworld. *Integrace CRM se systémy třetích stran.* [Online] 4. 2 2009. https://businessworld.cz/crm-lidske-zdroje/Integrace-CRM-se-systemy-tretich-stran-4214.

15. Danel, Roman. Homel. *Strukturovaný přístup k analýze a návrhu IS.* [Online] http://homel.vsb.cz/~dan11/aps/texty/Danel%20-%20APS%20-%20Strukturovany%20pristup%20k%20navrhu%20systemu.pdf.

16. is.muni. *Modelování IS Strukturovaný a objektově orientovaný přístup.* [Online] 2015. https://is.muni.cz/el/1421/podzim2015/VIKBA18/um/VIKBA18_-_2._blok_ZS_2015.pdf.

17. Vrana, Ivan. *Projektování informačních systému s UML.* Praha : Česká Zemědělská Univerzita v Praze, 2014. 978-80-213-1817-5.

18. Merunka, Vojtěch, Pergl, Robert a Pícka, Marek. *Objektově orientovaný přístup v projektování informačních systémů.* Praha : Česká zemědělská univerzita v Praze, 2005. 80-213-1352-8.

19. Businessinfo. *Životní cyklus a fáze projektů.* [Online] 29. 6 2011. https://www.businessinfo.cz/navody/zivotni-cyklus-a-faze-projektu/.

20. Bruckner, Tomáš, Voříšek, Jiří a Buchalcevová, Alena. *Tvorba informačních systémů.* Praha : Grada Publishing a.s., 2012. 978-247-7902-7.

21. Šmíd. Fi.muni. *Životní cyklus informačního systému.* [Online] 24. 4 2002. https://www.fi.muni.cz/~smid/mis-zivcyk.htm.

22. Rerych, Markus. Cartoon. *Softwareentwicklungsmodelle.* [Online] 2002. http://cartoon.iguw.tuwien.ac.at/fit/fit01/wasserfall/entstehung.html.

23. Testovanisoftwaru. *Vodopádový model.* [Online] 2016. http://testovanisoftwaru.cz/manualni-testovani/modely-zivotniho-cyklu-softwaru/vodopadovy-model/.

24. Web.archive. *SELECTING A DEVELOPMENT APPROACH.* [Online] 27. 3 2008. https://web.archive.org/web/20100331173931/http://www.cms.hhs.gov/SystemLifecycleFramework/Downloads/SelectingDevelopmentApproach.pdf.

25. Zcu.arcao. *Iterativní přístup k vývoji software, výhody, nevýhody.* [Online] 30. 5 2011. https://zcu.arcao.com/_statnice_ing_fav/nis/wiki_zvesela/03.htm.

26. Testovanisoftwaru. *Spirálový model.* [Online] 2016. http://testovanisoftwaru.cz/manualni-testovani/modely-zivotniho-cyklu-softwaru/spiralovy-model/.

27. Uzlabina2. *Softwarové inženýrství.* [Online] 2012. http://uzlabina2.aspone.cz/softwaroveinzenyrstvi.aspx.

28. Stoica, Marian, a další. Revistaie. *Analyzing Agile Development – from Waterfall Style to Scrumban.* [Online] 2016. http://revistaie.ase.ro/content/80/01%20-%20Stoica,%20Ghilic,%20Mircea,%20Uscatu.pdf.

29. Web.archive. *The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework.* [Online] 2. 4 2012. https://web.archive.org/web/20150814201800/http://jeffsutherland.com/ScrumPapers.pdf.

30. Is.muni. *Metodiky vývoje - Agilní a extrémní programování.* [Online] 2006. https://is.muni.cz/el/1433/podzim2006/PA165/um/02/printable.pdf.

31. Žáček, Jaroslav. Docplayer. *CASE nástroje.* [Online] 2018. https://docplayer.cz/68144164-Case-nastroje-jaroslav-zacek.html.

32. Tišnovský, Tomáš. Root. *Nástroje pro tvorbu UML diagramů.* [Online] 4. 7 2005. https://www.root.cz/clanky/nastroje-pro-tvorbu-uml-diagramu/.

33. Managementmania. *UML.* [Online] 2016. https://managementmania.com/cs/unified-modeling-language.

34. Methodsandtools. *Database Modelling in UML.* [Online] 1999. http://www.methodsandtools.com/archive/archive.php?id=9.

35. Mpavus. *State machine diagram - diagram stavového stroje.* [Online] http://mpavus.wz.cz/uml/uml-b-state-3-2-2.php.

36. Uml-diagrams. *UML Use Case Diagrams.* [Online] 2009. https://www.uml-diagrams.org/use-case-diagrams.html.

37. Mpavus. *Sequence diagram.* [Online] http://mpavus.wz.cz/uml/uml-b-sequence-3-2-4.php.

38. Mpavus. *Activity diagram.* [Online] http://mpavus.wz.cz/uml/uml-b-activity-3-2-3.php.

39. Ecom.ef.jcu. *Příklady a návody - Požadavky, specifikace.* [Online] http://ecom.ef.jcu.cz/web2/download/podklady/pozadavky-specifikace.pdf.