

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačního inženýrství**



**Diplomová práce**

**Návrh informačního systému v UML**

**Bc. Michal Gruber**

© 2021 ČZU v Praze



## ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Michal Gruber

Systémové inženýrství a informatika  
Informatika

Název práce

**Návrh informačního systému v UML**

Název anglicky

**Design of Information System in UML**

---

### Cíle práce

Cílem diplomové práce je návrh informačního systému pro správu a zpracování elektronických exekučních příkazů. Informační systém je navržen pro bankovní sektor tak, aby respektoval platné legislativní požadavky na zpracování zaslaných elektronických dokumentů. Systém je navržen s důrazem na automatizované zpracování. Informační systém dále umožňuje uživatelům provádět správu a kontrolu zpracovaných dokumentů.

### Metodika

Metodika práce je založena na studiu legislativních požadavků, odborných literárních a internetových zdrojů. Dále je autorem provedena analýza současného stavu informačních systémů ve vybrané finanční instituci. Následně je proveden rozbor a syntéza legislativních a funkčních požadavků. Na základě získaných poznatků je navržen informační systém, který je v souladu s definovanými požadavky.

## Doporučený rozsah práce

60 – 80 stran

## Klíčová slova

informační systém, návrh, UML, analýza, elektronický exekuční příkaz

---

## 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.

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

SCHMULLER, J. *Myslíme v jazyku UML : knihovna programátora*. Praha: Grada, 2001. ISBN 80-247-0029-8.

VRANA, I. – ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE. PROVOZNĚ EKONOMICKÁ FAKULTA, – ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE. KATEDRA INFORMAČNÍHO INŽENÝRSTVÍ. *Projektování informačních systémů s UML*. V Praze: Česká zemědělská univerzita, Provozně ekonomická fakulta, 2008. ISBN 978-80-213-1817-5.

---

## Předběžný termín obhajoby

2020/21 LS – PEF

## Vedoucí práce

doc. RNDr. Dana Klimešová, CSc.

## Garantující pracoviště

Katedra informačního inženýrství

---

Elektronicky schváleno dne 19. 11. 2020

**Ing. Martin Pelikán, Ph.D.**

Vedoucí katedry

---

Elektronicky schváleno dne 19. 11. 2020

**Ing. Martin Pelikán, Ph.D.**

Děkan

V Praze dne 31. 03. 2021

### **Čestné prohlášení**

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

V Praze dne 31.3.2021

---

## **Poděkování**

Rád bych touto cestou poděkoval doc. RNDr. Daně Klimešové, CSc. za vedení mé diplomové práce, odborný dohled a cenné rady. Dále děkuji všem, kteří mi v průběhu zpracování poskytovali konzultace.

# Návrh informačního systému v UML

## Abstrakt

Diplomová práce se zabývá návrhem informačního systému pro zpracování elektronických exekučních příkazů (EEP). Návrh je vytvořen za využití modelovacího jazyka UML. První část diplomové práce sdružuje poznatky o informačních systémech, jejich životních cyklech, UML a systémovém modelování.

V praktické části diplomové práce je představen současný bankovní systém pro zpracování EEP a princip jeho fungování. Dále jsou prezentovány vybrané legislativní pojmy, které ovlivňují podmínky pro zpracování EEP. Následně jsou vypracovány funkční a nefunkční požadavky na navrhovaný systém. S ohledem na tyto požadavky je prostřednictvím UML navržen nový informační systém. Návrh doplňuje vytvořený webový prototyp.

**Klíčová slova:** informační systém, exekuce, elektronický exekuční příkaz (EEP), návrh, software, UML, prototyp

# Design of Information System in UML

## **Abstract**

The diploma thesis deals with the design of an information system for the processing of electronic execution orders (EEP). The design is created by using the UML modeling language. The first part of the diploma thesis combines knowledge about information systems, their life cycles, UML and systems modeling.

The practical part of the diploma thesis presents the current banking system for processing EEP and the principle of its operation. Furthermore, selected legislative concepts that affect the conditions for processing EEP are presented. Subsequently, functional and non-functional requirements for the proposed system are developed. With regard to these requirements, a new information system is designed through UML. The design is complemented by the created web prototype.

**Keywords:** information system, execution, electronic execution order (EEP), design, software, UML, prototype



# Obsah

|   |           |
|---|-----------|
| <b>1 Úvod.....</b>                        | <b>12</b> |
| <b>2 Cíl práce a metodika .....</b>       | <b>14</b> |
| 2.1 Cíl práce .....                       | 14        |
| 2.2 Metodika .....                        | 14        |
| <b>3 Teoretická východiska .....</b>      | <b>15</b> |
| 3.1 Informační systém.....                | 15        |
| 3.1.1 Softwarové inženýrství .....        | 17        |
| 3.1.2 Návrh požadavků na IS.....          | 18        |
| 3.1.3 Životní cyklus IS.....              | 22        |
| 3.1.4 Modely životního cyklu systému..... | 28        |
| 3.2 UML.....                              | 31        |
| 3.2.1 Základní charakteristiky UML.....   | 31        |
| 3.3 Systémové modelování .....            | 34        |
| 3.3.1 Model tříd .....                    | 34        |
| 3.3.2 Model stavů.....                    | 37        |
| 3.3.3 Model interakcí .....               | 38        |
| <b>4 Vlastní práce .....</b>              | <b>40</b> |
| 4.1 Bankovní instituce.....               | 40        |
| 4.1.1 Stávající systém .....              | 40        |
| 4.1.2 Zpracovatelské oddělení .....       | 40        |
| 4.2 Legislativní pojmy .....              | 41        |
| 4.3 Požadavky na nový systém .....        | 42        |
| 4.3.1 Funkční požadavky .....             | 42        |
| 4.3.2 Nefunkční požadavky .....           | 43        |
| 4.4 Objektový model.....                  | 43        |
| 4.5 Stavový model.....                    | 47        |
| 4.6 Systémové interakce.....              | 50        |
| 4.6.1 Případy užití.....                  | 50        |
| 4.6.2 Sekvenční diagramy.....             | 51        |
| 4.6.3 Diagramy aktivit .....              | 61        |
| 4.7 Prototyp.....                         | 72        |
| <b>5 Výsledky a diskuse .....</b>         | <b>75</b> |
| <b>6 Závěr.....</b>                       | <b>76</b> |
| <b>7 Seznam použitých zdrojů .....</b>    | <b>77</b> |

|   |           |
|---|-----------|
| <b>8 Přílohy .....</b>                                    | <b>79</b> |
| 8.1 Ukázka zdrojového kódu obrazovky vyhledávání DZ ..... | 79        |

## Seznam obrázků

|  |    |
|--|----|
| Obrázek 1: Cyklus transformace dat na informace a znalost (Zdroj: [9] – upraveno).....   | 17 |
| Obrázek 2: Náklady na opravu v jednotlivých fázích projektu (Zdroj: [13] – upraveno) ... | 19 |
| Obrázek 3: Jednotlivé fáze vodopádového modelu (Zdroj: [16]) .....                       | 28 |
| Obrázek 4: Fáze prototypového modelu (Zdroj: [16]) .....                                 | 29 |
| Obrázek 5: Fáze spirálového modelu (Zdroj: [16]) .....                                   | 30 |
| Obrázek 6: Diagramy v UML a jejich dělení (Zdroj: Vlastní tvorba) .....                  | 32 |
| Obrázek 7: Rozložení pohledů v "4+1" (Zdroj: [10] – upraveno) .....                      | 33 |
| Obrázek 8: Příklad diagramu tříd (Zdroj: Vlastní tvorba) .....                           | 34 |
| Obrázek 9: Příklad multiplicity (Zdroj: Vlastní tvorba) .....                            | 35 |
| Obrázek 10: Příklad agregace (Zdroj: Vlastní tvorba) .....                               | 35 |
| Obrázek 11: Příklad kompozice (Zdroj: Vlastní tvorba) .....                              | 36 |
| Obrázek 12: Příklad generalizace (Zdroj: Vlastní tvorba) .....                           | 36 |
| Obrázek 13: Příklad stavového diagramu (Zdroj: Vlastní tvorba).....                      | 37 |
| Obrázek 14: Příklad diagramu případu užití (Zdroj: [30]).....                            | 39 |
| Obrázek 15: Diagram tříd (Zdroj: Vlastní tvorba) .....                                   | 44 |
| Obrázek 16: Stavový diagram: Datová zpráva (Zdroj: Vlastní tvorba) .....                 | 48 |
| Obrázek 17: Stavový diagram: Dokument (Zdroj: Vlastní tvorba) .....                      | 49 |
| Obrázek 18: Stavový diagram: Případ (Zdroj: Vlastní tvorba).....                         | 49 |
| Obrázek 19: Diagram případů užití (Zdroj: Vlastní tvorba) .....                          | 50 |
| Obrázek 20: Sekvenční diagram: Kontrola DZ (Zdroj: Vlastní tvorba).....                  | 56 |
| Obrázek 21: Sekvenční diagram: Supervize (Zdroj: Vlastní tvorba).....                    | 60 |
| Obrázek 22: Aktivitní diagram: Obecný proces zpracování EEP (Zdroj: Vlastní tvorba) ...  | 61 |
| Obrázek 23: Aktivitní diagram: Stažení DZ (Zdroj: Vlastní tvorba).....                   | 63 |
| Obrázek 24: Aktivitní diagram: Zpracování DZ (Zdroj: Vlastní tvorba) .....               | 64 |
| Obrázek 25: Aktivitní diagram: Vyhodnocení dat z xml (Zdroj: Vlastní tvorba).....        | 66 |
| Obrázek 26: Aktivitní diagram: Validace vyhledaných dat (Zdroj: Vlastní tvorba) .....    | 67 |
| Obrázek 27: Aktivitní diagram: Přiřazení DZ (Zdroj: Vlastní tvorba) .....                | 67 |
| Obrázek 28: Aktivitní diagram: Historie změn (Zdroj: Vlastní tvorba) .....               | 68 |
| Obrázek 29: Aktivitní diagram: Kontrola DZ (Zdroj: Vlastní tvorba) .....                 | 69 |
| Obrázek 30: Aktivitní diagram: Supervize (Zdroj: Vlastní tvorba).....                    | 71 |
| Obrázek 31: Prototyp stránky vyhledávání (Zdroj: Vlastní tvorba) .....                   | 73 |
| Obrázek 32: Prototyp stránky kontrola DZ (Zdroj: Vlastní tvorba) .....                   | 73 |
| Obrázek 33: Prototyp stránky kontrola dokumentu (Zdroj: Vlastní tvorba).....             | 74 |

## Seznam tabulek

|  |    |
|--|----|
| Tabulka 1: Funkční požadavky (Zdroj: Vlastní tvorba).....  | 43 |
| Tabulka 2: Nefunkční požadavky (Zdroj: Vlastní tvorba).....  | 43 |
| Tabulka 3: Scénář případu užití: Úspěšné provedení kontroly (Zdroj: Vlastní tvorba).....                 | 53 |
| Tabulka 4: Scénář případu užití: Předání z kontroly na supervizi (Zdroj: Vlastní tvorba)..               | 54 |
| Tabulka 5: Scénář případu užití: Předání z kontroly na ruční zpracování<br>(Zdroj: Vlastní tvorba) ..... | 55 |
| Tabulka 6: Scénář případu užití: Předání ze supervize ke kontrole (Zdroj: Vlastní tvorba)                | 58 |

|   |    |
|---|----|
| Tabulka 7: Scénář případu užití: Předání ze supervize na ruční zpracování<br>(Zdroj: Vlastní tvorba)..... | 59 |
| Tabulka 8: Možná časová úspora po zavedení systému (Zdroj: Vlastní tvorba).....                           | 75 |

# 1 Úvod

Uchování, správa a efektivní zpracování dat jsou dnes velmi aktuálním a závažným problémem. Jeho strukturované řešení nabízí správně navržený informační systém (IS). Z toho důvodu je pro konkurenceschopný subjekt kvalitní IS nutností. Podniky ve všech odvětvích investují značné prostředky do návrhu a vývoje takového systému, který jim umožní snížení nákladů, přilákání nových zákazníků anebo zefektivnění používaných procesů.

Výjimkou není bankovní sektor, který využívá množství informačních systémů pro interakci se zákazníky, správu a prodej jejich produktů a k vnitřním provozním účelům. Bankovní instituce jsou povinny dodržovat platná legislativní nařízení. Jedním z takových nařízení je povinnost součinnosti v exekučním řízení. Proto se tato práce se zaměřuje na možnosti návrhu interního informačního systému pro zpracování elektronických exekučních příkazů (EEP).

Průběh, omezení a vlastnosti exekucí definuje zákon č. 120/2001 Sb. o soudních exekutorech a exekuční činnosti. Tento zákon specifikuje povinnost bankovních institucí spolupracovat ve věci poskytování informací o klientech, zadržování prostředků na určených účtech a vyplacení prostředků z exekucí postižených účtů.

Tato povinnost představuje pro banky dodatečnou administrativní zátěž. Je tedy očekávané, že jednotlivé bankovní ústavy usilují o vytvoření co nejefektivnějších procesů pro zpracování dokumentů souvisejících s exekučním řízením. Na podporu nastavených procesů vznikají interní systémy, které zajišťují splnění všech legislativních požadavků.

Výraznou změnu do zavedených procesů vnesl v roce 2009 začátek platnosti zákona č. 300/2008 Sb. o elektronických úkonech a autorizované konverzi dokumentů. Ten ukládá povinnost zřízení datové schránky pro všechny orgány veřejné moci a právnické osoby. Povaha dokumentů spojených s exekučním řízením umožňuje jejich zasílání elektronickou formou, tedy pomocí datových zpráv. Bankovní instituce tedy musely upravit své stávající systémy tak, aby dokázaly přijímat, zpracovávat a uchovávat dokumenty v čistě elektronické podobě.

Posledním výraznějším zásahem do principů správy dokumentů spojených s výkonem exekučního řízení je změna §49 Zákona č. 120/2001, která nabyla platnosti v roce 2015. Dochází zde k definici elektronického exekučního příkazu (EEP). Jedná se o xml soubor obsahující data nezbytná ke strojovému zadání vybraných elektronických dokumentů.

Bankovním institucím se zde tedy otevírá prostor k využití automatizace. V roce 2019 bylo v České republice vedeno 4 679 200 pravomocných exekucí proti fyzickým osobám [1], z toho 508 062 bylo nově zahájeno [2]. Proto má automatizace zpracovatelského procesu i pro menší množinu obdržených dokumentů má rozsáhlé dopady na rychlost a výši nákladů na zpracování.

## **2 Cíl práce a metodika**

### **2.1 Cíl práce**

Cílem této diplomové práce je návrh informačního systému pro automatizované zpracování příchozích elektronických exekučních příkazů v bankovní instituci. K realizaci návrhu je využit modelovací jazyk UML. Navržený IS je zamýšlen jako doplněk stávajícího systému, který je schopen elektronické exekuční příkazy zpracovávat, ovšem pouze manuálně.

Nový IS je navržen tak, aby navázal na strukturu systému stávajícího, zefektivnil využití lidských zdrojů a splnil nutné legislativní požadavky. Návrh čerpá z autorových zkušeností s bankovním exekučním prostředím. Zaměření IS je hlavně na strojové zpracování příchozích dokumentů, jejich následnou kontrolu a supervizi. Návrh je dále doplněn o vytvořený prototyp klientské části systému.

### **2.2 Metodika**

Metodika této závěrečné práce je založena na studiu odborné literatury a internetových zdrojů. Získané poznatky spolu se zkušenostmi autora a získanými daty staly po vyhodnocení nástrojem pro stanovení funkčních a nefunkčních požadavků na systém. Dále je provedena analýza a návrh systému zaměřeného na zpracování příchozích elektronických exekučních příkazů. Následně je implementován prototyp webové aplikace k demonstrování možného vzhledu a základních principů systému. K tvorbě jsou využity technologie HTML, CSS, Bootstrap, jQuery. V diskusi je vyhodnocen přínos navrženého informačního systému a předložen návrh na další možné rozšíření.

## 3 Teoretická východiska

### 3.1 Informační systém

Informační systém je ve společnosti již zažitý slovní pojem, pod kterým si laik představí kompletní technologické zázemí firmy. Odborně jej lze definovat mnoha způsoby:

*„Informační systém organizace je systém informačních technologií, dat a lidí, jehož cílem je efektivní podpora informačních a rozhodovacích procesů na všech úrovních řízení organizace (firmy).“ [3]*

*„Informační systém – primárním účelem tohoto systému je spravovat databázi informací a poskytovat k ní přístup. Informační systémy se zabývají otázkami bezpečnosti, použitelnosti, soukromí a zachování integrity dat.“ [4]*

*„Informační systém je soubor technických prostředků (hardware, případně software), které udržují a poskytují data (informace) pro požadovaný účel jeho uživatelů, tedy lidem. Základní účel informačního systému je tedy udržování a poskytování informace (respektive dat) jeho uživatelům.“ [5]*

*„IS jsou komplexní systémy, jejichž působením jsou vytvářeny a vznikají interakce, které jsou prováděny samy o sobě a se svým prostředím. Momentka, snímek systému zachycuje jeho statické aspekty, takové jako databázové schéma, jeho obsah, okamžitou interakční historii. Některé z těchto aspektů jsou fixní pro daný systém, zatímco jiné se mění, vyvíjí v čase. Systémový snímek nějakého bodu v čase poskytuje informaci o systémovém stavu. Dynamika systému není zachycena „snímkovým“ modelem, který zachycuje pouze jeho statické aspekty. Dynamiky jsou procesy, které ovládají systémový stav.“ [6]*

Ačkoliv jsou jednotlivé definice IS rozdílné, vyskytují se v nich pojmy, které je zapotřebí více vysvětlit. Konkrétně se jedná o pojmy systém, informace, data a znalost.

## **System**

System lze obecně charakterizovat jako množinu prvků, které propojují jednosměrné či obousměrné vazby. Nedílnou součástí systému jsou dále vstupní a výstupní vazby, které slouží k přijímání a předávání dat okolí. Pro pochopení systému je nutné jeho pozorování a zkoumání. Při pozorování je kladen důraz na komunikaci systému s jeho okolím a na jeho výsledné chování [7].

## **Data, Informace, Znalost**

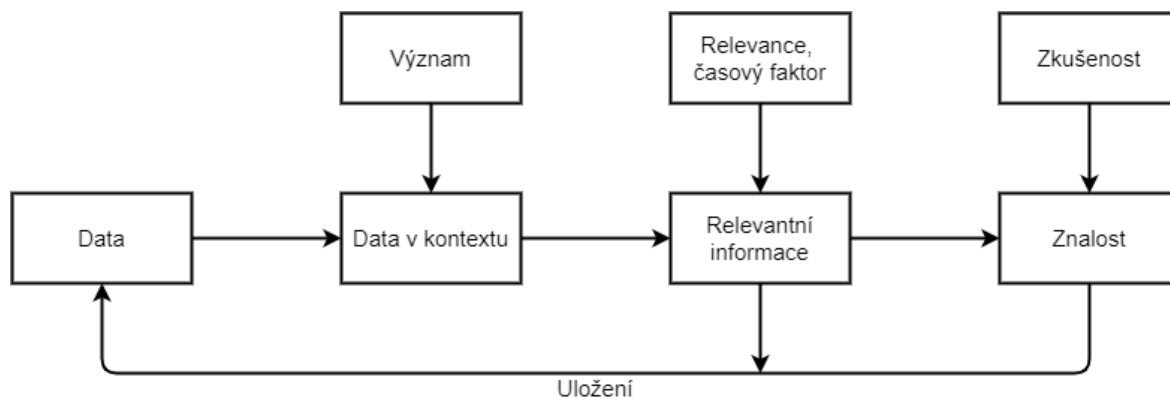
Data jsou objektivní reprezentací věcí a atributů. Bez jejich náležité interpretace nelze jejich obsah a potenciál využít. Data mohou být reprezentována jakoukoliv fyzikální či virtuální formou. Bez kontextu může být hodnota 299 792 458 považována za náhodný řetězec. Pokud tuto hodnotu doplníme o kontext a řekneme, že k ní v dnešní době náleží jednotka m/s, tak víme, že se bude pravděpodobně jednat o rychlost. Získáme tedy informaci [8], [9].

Informaci lze definovat jako data doplněná o kontext chápaný určenou osobou v daný čas. Data i kontext musí být vzájemně relevantní a vztažené k určitému času. Častým názorem je, že data doplněná o kontext tvoří informaci. Nejedná se o pravdivý výrok. Data spojená s významem jsou pouze data vložená do kontextu. Z takových mohou vzniknout relevantní a nerelevantní informace. Ovšem pouze relevantní informace může přejít ve znalost [8].

Znalost je schopnost spojit novou informaci s informacemi již známými či podobnými, které byly získány zkušeností nebo asociací. Jedná se o informaci, která byla uchována s porozuměním o jejím významu. Znalosti jsou získávány zkušeností, studiem, poznáním podobnosti, asociací, povědomím anebo porozuměním [9].

Znalost lze dělit na implicitní a explicitní. Implicitní znalost je ta, kterou si daná osoba uchovává ve své paměti. Je obtížné ji předávat ostatním a šířit je globálně. Oproti tomu explicitní znalost je znalost převedená do unifikovaného tvaru a uložena na médium. Příkladem takového média může být papír, nahrávka, disk. Taková znalost také odkazuje na příbuzná témata a tím tvoří síť křížových odkazů nutných pro jejich pochopení [9]. Cyklus transformace dat na informace a znalost je zobrazen na diagramu níže.





Obrázek 1: Cyklus transformace dat na informace a znalost (Zdroj: [9] – upraveno)

### 3.1.1 Softwarové inženýrství

Softwarové inženýrství lze označit za technickou disciplínu zaměřenou na komplexní produkci informačního systému. Rozsahem softwarové inženýrství pokrývá vše od rané fáze specifikace systému po jeho údržbu a reengineering. Samotný pojem „softwarové inženýrství“ zazněl poprvé na konferenci NATO v roce 1968. Zde proběhly diskuse o škálovatelnosti individuálního přístupu k vývoji velkých a složitých IS. Na základě těchto diskusí byl sjednocen názor, že takové řešení není vhodné. Takto produkované systémy často překračovaly plánovaný rozpočet, byly nekvalitní a docházelo ke znatelným odchylkám vůči plánovanému harmonogramu [4].

Hodnocení kvality IS se zaměřuje na několik klíčových aspektů. Jedná se hlavně o efektivitu, přijatelnost, spolehlivost, bezpečnost a schopnost údržby.

Efektivita je hodnocena na základě práce s poskytovanými prostředky. Jedná se o prostředky časové a hardwarové. Je tedy hodnoceno, zda systém efektivně využívá přidělený výpočetní výkon. Zároveň systém musí umožňovat uživatelům efektivně zpracovávat požadované úkoly [10].

Hodnocení přijatelnosti vychází zejména z uživatelské práce. Systém musí být přívětivý, srozumitelný, použitelný. Často opomíjeným kritériem bývá integrace s okolím. Systém musí být kompatibilní s okolními systémy, které uživatelé při své práci využívají [4].

Spolehlivost sdružuje kritéria dostupnosti a stability. Systém by neměl obsahovat vady a v případě selhání by nemělo dojít ke ztrátě dat, fyzickému poškození nebo ekonomické škodě. Pod spolehlivost se určitě řadí i kritérium stabilní a dlouhodobě neklesající rychlosti odezvy [3].

Bezpečnost se zaměřuje na vnější a vnitřní hrozby. Systém musí jasně definovat oblasti, které jeho uživatelé s určeným oprávněním mohou využívat či spravovat. Zároveň musí být systém odolný vůči vnějším útokům, a tedy zabránit vzniku ekonomických či technických škod [3].

Hodnocení v oblasti schopnosti údržby jsou z velké části založena na návrhu IS. Systém musí být vytvořen tak, aby v průběhu času mohl být modifikován na základě nových uživatelských požadavků. Systém, který nemůže být, anebo není v průběhu času upravován dle aktuálních potřeb uživatelů, přestane být použitelným. Systémové inženýrství využívá systematický přístup, který je také označován jako softwarový proces. Takový proces se skládá z posloupných aktivit, které vedou k produkci IS [4].

Hlavními částmi softwarového procesu jsou specifikace, vývoj, kontrola a evoluce. Každý proces tedy musí být jednoznačně definován, vytvořen, podroben revizi a po dobu svého produkčního cyklu inovován. Na základě zvoleného modelu životního cyklu IS může docházet ke změně pořadí těchto kroků [11], [10].

### **3.1.2 Návrh požadavků na IS**

Sběr, kompletace a smysluplná interpretace požadavků na nový systém jsou klíčovým procesem. Realizace nového systému je rozsáhlý úkol, který vyžaduje kooperaci vývojového, analytického, testovacího, manažerského a uživatelského týmu. Všichni členové týmu musí mít přístup k přesně definovanému zadání, které je tvořeno ve spolupráci všech výše zmíněných týmů [12].

Na základě zadání jsou definovány nutné rozsahy práce. Dále je určen dopad na okolní systémy. V neposlední řadě ze zadání vyplývají omezení, která musí systém vynucovat. Ucelené a správně navržené zadání je kritické pro úspěšné pokračování projektu a pro snížení budoucích nákladů. S každou další fází vývoje projektu cena opravy chyb exponenciálně stoupá. Je tedy nutné eliminovat maximum chyb, omylů a nejasností v co nejdřívější části projektu [13].



Obrázek 2: Náklady na opravu v jednotlivých fázích projektu (Zdroj: [13] – upraveno)

Sběr požadavků je výchozím bodem celé analýzy. Musí být dobře rozvržen. Velmi často se používá kombinace 2 a více různých metod pro získání detailního a dostatečně širokého náhledu řešené problematiky. Typicky používanými metodami pro sběr informací jsou [12]:

- Dotazníkové šetření.
- Interview.
- Uživatelské scénáře.
- Rozbor stávající dokumentace.
- Pozorování zavedených procesů.
- Aktivní zapojení do pracovního procesu.
- Studium legislativních omezení.
- Prototypování.

### Dotazníkové šetření

Dotazníkové šetření patří mezi kvantitativní metody. Jednou sestavený dotazník může být prezentován širokému okruhu respondentů a jeho vyhodnocení není tak časově náročné, jako například interview a může být do značné míry automatizováno. Kvalita výstupu dotazníkového šetření je přímo závislá na kvalitě zvolených otázek. Běžnou praxí je volba uzavřených otázek, které jsou následně doplněny otázkami otevřenými. Dalším typem užívaných otázek je použití škály, která umožňuje respondentům vyjádřit kombinaci preferencí v dané oblasti. Potenciální problém této metody tvoří nepochopení problematiky

zadavatele anebo špatná formulace otázek, kdy respondenti mohou celý dotazník vyplnit a odlišný výklad tématiky nemusí být z odpovědí zřejmý [14].

### **Interview**

Interview je moderovaný strukturovaný rozhovor, který je veden s předem vybranou osobou. Otázky jsou připravené tak, aby odpovídaly řešenému tématu se zaměřením na oblast či roli, kterou dotazovaná osoba v řešené problematice zastává. Zavedenou praxí bývá provedení dotazníkového šetření, které je následně detailněji diskutováno v rámci interview. Tímto způsobem lze odhalit nedorozumění v pochopení zadání a případné formální nedostatky dotazníku. Interview je časově a administračně náročnou metodou. Její vyhodnocení je vždy individuální, ale poskytuje nejinteraktivnější přístup k získávání informací od účastníka procesu [14].

### **Uživatelské scénáře**

Scénáře popisují konkrétní průchod či interakci uživatele se systémem. Scénáře jsou připravovány ve spolupráci se zadavateli. Každý scénář musí ve své struktuře obsahovat informaci o entitě, která akci provádí, přesný popis provedené akce a očekávaný výsledek nebo reakci systému. Uživatelské scénáře následně slouží jako podklad pro stanovení seznamu entit, které systém bude podporovat a jejich jednotlivé role v rámci celého procesu. Scénáře jsou vždy součástí modelu interakcí [11].

### **Rozbor stávající dokumentace**

Jestliže probíhá analýza s účelem nahrazení či rozvinutí stávajícího systému, je vždy proveden rozbor stávající dokumentace. Vnitropodniková dokumentace je zdrojem cenných informací, které definují vnitřní procesy systému, datové toky a principy napojení na okolní systémy. V případech, kdy se nové a staré systémy po delší časové úseky budou na produkčním prostředí překrývat, se vytváří rozdílová dokumentace, která zachycuje přesné implementační rozdíly. Rozdílová dokumentace není zadáním pro nový systém [4].

### **Pozorování zavedených procesů**

V případech, kdy probíhá nahrazení nebo rozšíření stávajícího systému, jsou členové analytického a vývojového týmu v kontaktu s účastníky měněného procesu. Pozorují průběh procesů, zpracovávají informace o užívaných vstupech a výstupech. Při použití této metody členové analytického a vývojového týmu nejsou uživateli pracovního procesu [15].

### **Aktivní zapojení do pracovního procesu**

Jedná se o obdobnou metodu k pozorování zavedených procesů. V tomto případě jsou členové analytického a vývojového týmu zapojeni do pracovního procesu. Účelem je zajistit jim reálnou zkušenost s již používaným systémem tak, aby byli schopni nutné procesy navrhnout a realizovat v novém zadání [12].

### **Studium legislativních omezení**

Ve vybraných odvětvích může být navrhovaný systém omezen legislativními podmínkami. Návrh systému, který zcela odpovídá aktuálně platné legislativě, je hlavní prioritou. V běžné praxi je aplikace procesů odpovídajících legislativním požadavkům konzultována s odborníky na výklad práva. Za nedodržení těchto nařízení jsou tvrdé sankce a je vyžadováno zjednání téměř okamžité nápravy [15].

### **Prototypování**

Pro jednoznačné pochopení složitějšího rozložení prvků v zobrazení aplikace jsou využívány prototypy. Zadavatel má pak možnost jasně definovat požadované rozložení a chování jednotlivých prvků. Vytváří se buďto papírové prototypy anebo v případě rozsáhlejších projektů se prototypy škálují od drátových modelů až po graficky funkční prototypy, které dokáží jednodušeji prezentovat funkční prvky na jednotlivých stránkách, přechody mezi jednotlivými stavy a návaznosti funkčních celků. Prototyp pracuje se statickými daty a neobsahuje žádnou výpočetní logiku na pozadí [12].

### 3.1.3 Životní cyklus IS

Definice životního cyklu IS vznikla z praxí ověřených metodik, které si vývojové týmy se stoupající praxí osvojily. V literatuře lze nalézt několik různých přístupů a definic životního cyklu IS. Jednotlivé fáze dělí následovně [16]:

- Prvotní analýza a určení cílů.
- Analýza systému a sjednocení požadavků.
- Návrh systému.
- Implementace.
- Testování.
- Uvedení do produkčního prostředí.
- Zkušební provoz.
- Produkční období.
- Reengineering.

#### **Prvotní analýza a určení cílů**

Výchozí část celého projektu. V prvé řadě dochází ke sběru požadavků od zadavatelů a organizace. Následně jsou analyzována funkční, nefunkční a legislativní omezení na požadovaný systém. Cílem této části je vytvoření základního přehledu o uvažovaném projektu. Není přípustné, aby docházelo k detailní analýze a tvorbě uceleného podrobného zadání [12].

Výstupem této části je stručný dokument, který musí obsahovat časový plán projektu a rozpis nutných zdrojů. V rozpisu jsou zahrnuty všechny dotčené oblasti, tedy náklady na vývoj, následný provoz, cena použitého SW i HW. Dále je sepsán rozsah systému, předpokládané funkčnosti, odhadovaná návratnost a uvažovaný přínos. V případě, že dochází k rozšiřování či změně původní funkčnosti, musí být uveden rozdíl a případný dopad vůči stávajícímu řešení. Poslední část prvotní analýzy tvoří soupis známých problémů a předpokládaných komplikací [15].

Dokument, který vzejde z prvotní analýzy, bývá předkládán k posouzení zadavateli nebo investorovi projektu. Zadavatel na základě tohoto dokumentu schvaluje uvedené rozsahy plánovaného projektu. Je možné, že nastane situace, kdy plánované náklady,

předpokládaná rizika anebo výše přínosu nejsou pro zadavatele akceptovatelné. Tehdy dochází buďto k opakování prvotní analýzy s přihlédnutím ke zpětné vazbě od uživatele anebo k ukončení projektu [15].

### **Analýza systému a sjednocení požadavků**

Na základě dokumentu z prvotní analýzy je provedena podrobnější studie. Cílem je provedení analýzy okolního prostředí a možností jeho integrace. Na základě získaných poznatků je proveden návrh datové i systémové struktury a rozdělení systému do logických, na sebe navazujících celků. Ucelené zadání přispívá k eliminaci návrhových chyb na vyšší úrovni detailu [17].

### **Návrh systému**

Návrh systému navazuje na předchozí analýzu systému. Často je tento krok nazýván také Projektová studie. Výstupem této aktivity je zadávací dokument pro realizační tým. Dokument tedy musí obsahovat informace o zodpovědnostech tvůrců systému. V případě spolupráce s externí firmou musí být jednoznačně definováno, kdo, kdy, v jakém rozsahu a za jakých finančních podmínek systém dodá. Může nastat situace, kdy je systém dělen na jednotlivé logické celky a každý celek bude dodáván jiným partnerem. Pak musí být každý samostatný celek jasně přiřazen a průvodní dokumenty musí jednoznačně určit hranice jednotlivých celků, způsob vzájemné komunikace a určení zodpovědnosti za sladění jednotlivých částí [18].

Dále je uveden objednatel služby a jmenovány konkrétní osoby ze strany objednatele, které budou s externí firmou spolupracovat v jednotlivých oblastech. V dokumentu je přiložena analýza stávajícího stavu, globální návrh nového IS, datová analýza, logický model, fyzický model, návrh funkcí a datových toků [17].

Vedle analýz dokument určuje podmínky vývoje, testování a nasazení do produkčního prostředí. Přejechod nového IS do produkčního prostředí je rizikový a komplikovaný proces. Je proto nutné předem specifikovat neopomenutelné kroky k zajištění hladkého přechodu. Do této oblasti spadají i požadavky na HW, SW a provedení studií souvisejících s nasazením. Specifika testovacího provozu a poskytování pozáručního servisu nesmí být vynechány [15].

Z organizačních témat je v dokumentu zastoupen harmonogram spolupráce, který obsahuje časový harmonogram dodávek, plateb a servisních zásahů. K harmonogramu spolupráce se přidávají také podmínky dodání, podmínky servisů a celková cena. Často spornou, přesto nezbytnou částí je definice podmínek pro změnu zadání a smluvních pokut či náhradních řešení v případě nedodání včas nebo v požadované jakosti [17].

## **Implementace**

V rámci implementace přicházejí ke slovu vývojové týmy, které ve spolupráci s analytiky začnou zamýšlený systém vytvářet. Již v tomto kroku je často navázána spolupráce s testovacími týmy pro konzultace či tvorbu testovacích scénářů. Implementace probíhá dle dodané dokumentace [19].

## **Testování**

Po ukončení implementace probíhá intenzivní testování. První jsou prováděny unit testy. Jedná se o atomické scénáře, které píše vývojáři. Typicky bývají součástí kompilace buildu, tedy jsou automaticky provedeny vždy před předáním nové verze testerům. Testeři se zaměřují na funkční, integrační a akceptační testy [16].

Funkční testování se zaměřuje na testy jednotlivých funkcionalit. Nejčastěji se dělí na smoke, sanity a regresní testy. Sanity testy se zaměřují na jednoduché funkční testy. Účelem je rychle ověřit základní funkční scénáře. Smoke testy prochází funkčností skrz celou aplikaci nebo vybraný celek. Důraz je kladen na testy propojení a komunikace mezi jednotlivými funkčnostmi a aplikacemi. Při testování je běžnou praxí opakování testů v rámci jedné oblasti ve více instancích. Je možné, že vlivem postupného vývoje, oprav chyb anebo změnou podkladových dat dojde k zavlečení chyb do již otestované oblasti. Sady testů pro opakované použití se nazývají regresní [20].

Akceptační testy jsou realizovány ve spolupráci se zákazníkem. V závislosti na formě projektu je buďto zákazníky nadefinován požadovaný mix vstupních dat, který následně pro transformaci systémem kontrolují anebo je vytvořeno dedikované prostředí. V takovém prostředí probíhá co nejuvěrnější simulace produkčních procedur a vstupních dat. Zákazníci se v tomto prostředí pohybují v souladu s pokyny, které budou vydány pro produkční prostředí. Akceptační testy jsou potvrzením, že dodaná část systému splňuje požadavky zákazníka [20].



V závislosti na použitých metodách analýzy a dokumentace jsou testy dělené na White box a Black box. White box testy reprezentují otevřenou analýzu. Tedy test analytik zná systém a ví, jaké principy jsou při transformaci dat uplatněny. Black box testování naopak představuje práci se systémem, jehož vnitřní fungování není známo. V takovém případě jsou dodány kombinace vstupních a k nim náležících výstupních dat. Tester tedy nezná principy transformace dat [21].

Posledním dělením, které začíná být velmi diskutované a užívané zejména v posledních letech, je manuální a automatizované testování. Manuální testování spočívá v ručním průchodu zvolené oblasti testerem. Automatizované testy sjednocují široký okruh testů různých stupňů automatizace. Společným specifickým je jednotný proces automatické přípravy jedinečných dat pro požadovaný test, provedení testu, vyhodnocení testu a odstranění použitých dat [22].

Automatizované testy jsou často děleny na aplikační vrstvy. Velmi oblíbeným zástupcem frontendových testů je Selenium, které umožňuje v prohlížeči simulovat interakce uživatele. Mezi backendové testy je často řazeno rozšíření unit testů o uživatelské scénáře. Unit testy jsou psané ve stejném jazyce jako hlavní kód, neboť pracují se jednotlivými třídami a metodami. Integrované testy ověřují dostupnost a odpověď online volání. Volání online služeb nejčastěji zahrnuje SOAP nebo REST služby. Jejich automatizované testování může být realizováno přímo v kódu anebo za pomoci dedikovaných nástrojů. Mezi nejznámější lze zařadit ReadyAPI. Poslední oblastí jsou databázové testy. Ovšem různá databázová řešení se vnitřní konfigurací liší natolik, že zde nelze hovořit o jednotně užívané metodice [22], [23].

### **Uvedení do produkčního prostředí**

Jeden z nejkritičtějších momentů v životním cyklu IS je proces nasazení do produkčního prostředí. Předchází mu důkladné plánování k zajištění bezchybného průběhu nasazení. V závislosti na zvolené strategii jsou provedeny testovací simulace ve vývojovém prostředí. Nejčastěji používanými strategiemi jsou souběžná, pilotní, postupná a nárazová [16].

Souběžná strategie podporuje souběžný provoz obou systémů. Veškeré pracovní aspekty musí být pokryty oběma systémy. Dochází tak k duplikaci práce, ale zároveň je nový systém podroben obvyklé činnosti. Nedostatky jsou tedy jednoduše odhalitelné. Současný

provoz starého systému zajišťuje bezpečný provoz i v případě výpadku či zjištění kritických nedostatků systému nového. Oba systémy jsou na produkčním prostředí nasazeny více pracovních cyklů. Tedy do doby, nežli je potvrzeno, že nový systém funguje stabilně a je připraven na samostatný produkční provoz v plném rozsahu [24], [16].

Pilotní strategie pracuje s omezenou množinou produkčních uživatelů. Cílem je vybrat dostatečně variačně rozsáhlý soubor uživatelů, kteří začnou nový systém využívat. V případě problémů tedy nedostatky nepostihnou všechny uživatele, ale pouze vybranou skupinu. Jakmile jsou hlavní nedostatky odstraněny, je pilotní skupina rozšířena o nově vybrané skupiny uživatelů. Pokud pilotní provoz projde i rozšířenou skupinou uživatelů, následuje zvolení ostrého nasazení aplikace do produkčního prostředí pro všechny uživatele.

Postupná strategie bývá aplikována u rozsáhlých, složitých či modulárních systémů. Do produkčního prostředí jsou tedy nejdříve uvedeny primární části IS. Následně je provedeno jejich testování v produkčním prostředí. Po odstranění všech nedostatků probíhá nasazení další části či modulu. Zároveň s postupným uvolňováním nových částí do produkčního prostředí v případě starého IS dochází naopak k postupnému vypínání. [20]

Nárazová strategie je založena na jednorázovém přechodu ze starého systému na nový. V rámci této strategie nedochází ke zvyšování pracovních nároků na uživatele. Nový systém kompletně nahradí starý. Z hlediska nákladů na zavádění systémů se jedná o nejlevnější variantu. Je zde ovšem vysoké riziko problémů. Jednorázové přechody se často setkávají s nepřipraveností uživatelů a teprve po provedení přechodu bývají objeveny koncepční nedostatky [16].

### **Zkušební provoz**

V předem definovaném období probíhá zkušební provoz. Tedy poskytovatel systému provádí aktivní vyhledávání chyb, které mohou vznikat běžným provozem, dále provádí prioritní opravu nahlášených chyb. Po dobu zkušebního provozu často bývají zavedena dočasná pracoviště podpory, kam se uživatelé obracejí s procesními dotazy. Je jim tedy poskytována podpora pro práci s aplikací. Dále bývá vytvořen anebo rozšířen jednotný komunikační kanál s uživateli pro případ rozsáhlejších výpadků či omezení nového systému. V rámci zkušebního provozu bývají velmi často nalezeny procesní nedostatky, které jsou následně dodatečně implementovány. Rozsah, délka a míra zkušebního provozu je smluvně definována již během návrhu systému [25].

### **Produkční období**

Jedná se o nejdelší etapu životního cyklu IS. Během produkčního cyklu je zajišťována běžná údržba systému a drobný vývoj. Podněty pro vývoj během produkčního cyklu mohou být podněty uživatelů, změna firemní politiky, legislativní požadavky anebo přechody mezi verzemi využívaného HW a SW. Mimo udržování systému v provozuschopném stavu je do této části také zařazeno nové i opětovné školení uživatelů [25].

### **Reengineering**

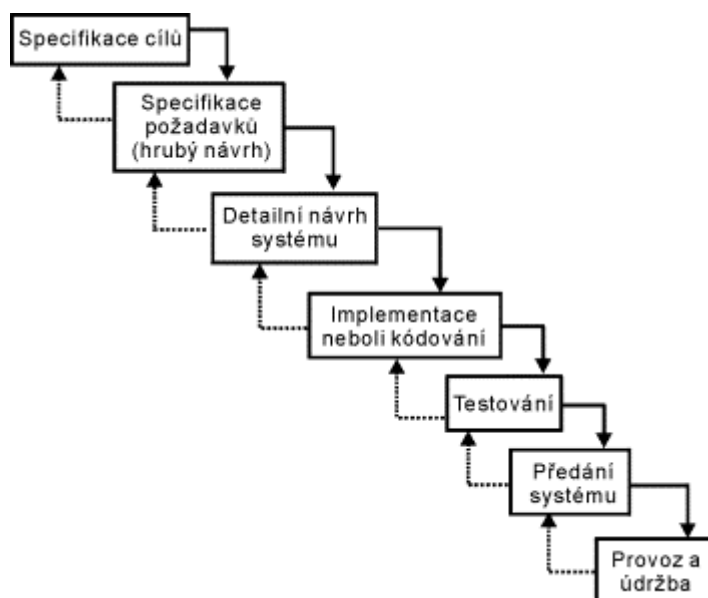
Při reengineeringu dochází k přehodnocení přínosů stávajícího systému. Systémy s časem zastarávají a je nutné je neustále přizpůsobovat aktuálním požadavkům. Pokud je úprava stávajícího systému pro splnění aktuálních požadavků příliš nákladná anebo technologicky nemožná, je přistoupeno k zahájení procesu vyřazení stávajícího systému. Vyřazení aktuálního systému je zároveň prvním krokem životního cyklu IS, tedy prvotní analýza a určení cílů nového informačního systému [16].

### 3.1.4 Modely životního cyklu systému

Jednotlivé části životního cyklu IS byly představeny v předchozí kapitole. Na základě časového uspořádání a návaznosti jednotlivých kroků existuje několik modelů. Podle [16] lze základní modely rozdělit na vodopádový, prototypový a spirálový.

#### Vodopádový model

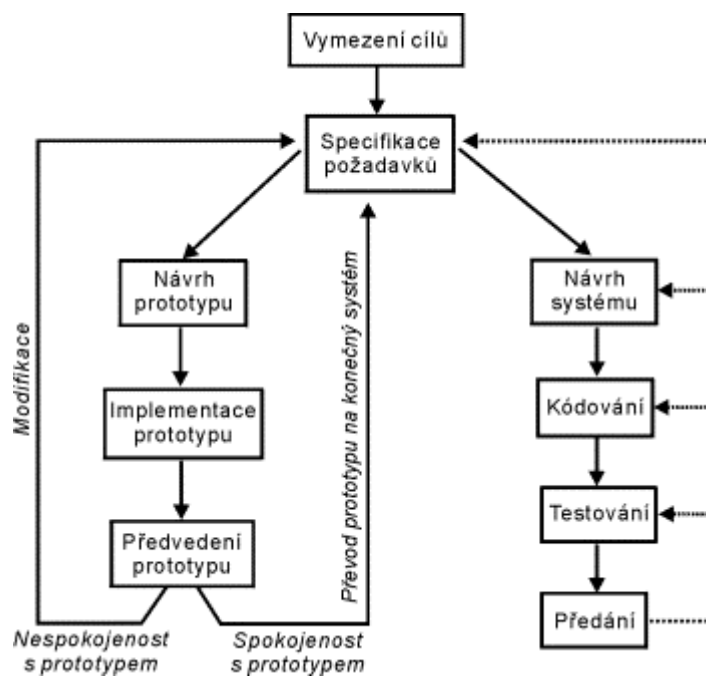
Při využití vodopádového modelu je vytvořen přesný harmonogram kroků, které na sebe vzájemně navazují a nikdy se nepřekrývají. Dokončený krok je vstupem pro další aktivitu. Návrat k předchozí aktivitě není možný. Vodopádový model je relativně levné a rychlé řešení v situacích, kde předem podrobně známe sledovaný problém a způsob jeho řešení. Mezi hlavní nevýhody vodopádového modelu patří dodání první verze až po ukončení všech kroků. Pro zákazníka tedy není možnost hodnotit navržené části dříve a odhalovat návrhové nedostatky před dodáním celého systému. Tyto nedostatky odstraňuje například prototypový model [16], [26].



Obrázek 3: Jednotlivé fáze vodopádového modelu (Zdroj: [16])

## Prototypový model

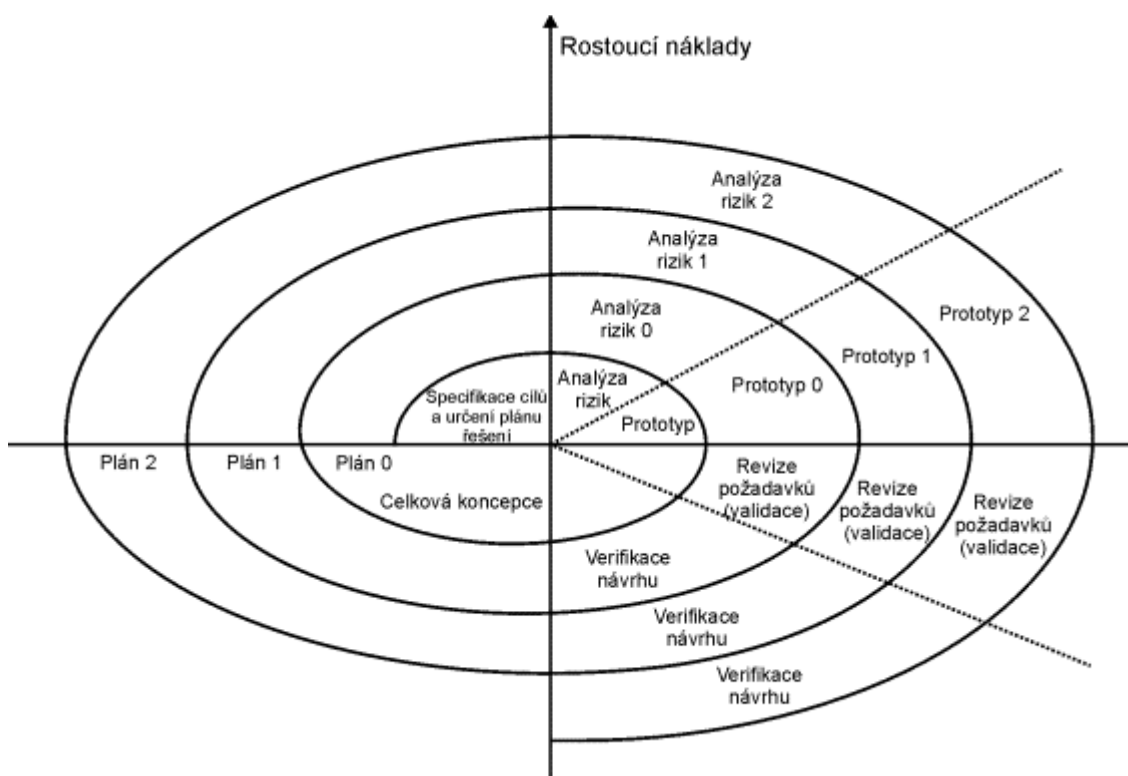
V rámci prototypového modelu dochází k opakované iteraci s uživateli. Dodavatel v co nejkratší době připraví prototyp nebo část implementovaného systému. Následně proběhne konzultace se zákazníky. Požadavky na změny oproti již konkrétnímu prototypu jsou zanalyzovány, zapracovány a opět konzultovány se zákazníky. Tento proces je prováděn do té doby, nežli je zákazník s výsledkem spokojen. Hlavní výhodou tohoto modelu je intenzivní spolupráce s klienty již během vývoje. Dochází tedy k odstranění závažných nedostatků v brzké části vývoje. Nevýhodou může být vleklost vývoje. Pokud uživatelé neustále mění své požadavky, může začít vývoj stagnovat. Proto se u velkých projektů předem stanovují limity na počet opakování iterace a maximální délku trvání celého vývoje [16], [26].



Obrázek 4: Fáze prototypového modelu (Zdroj: [16])

## Spirálový model

Spirálový model kombinuje prototypový přístup s opakovanou analýzou rizik. Jednotlivé vývojové kroky jsou periodicky opakovány. Nejdříve dochází k implementaci jádra systému. Následně jsou postupně přidávány části na vyšší úrovni. Jednotlivé kroky jsou shodné s prototypovým modelem. S opakujícími cykly narůstá náročnost časová i nákladová. Mezi hlavní výhody patří možnost konzultace s uživateli v průběhu vývoje, využití ověřených kroků a opakovaná aplikace analýzy rizik, což významně snižuje riziko chyb. Tento model s sebou nese i nevýhody. Není možné na začátku projektu pevně stanovit termíny a cenové náklady. Součinnost zákazníků je nutná po celou dobu projektu. Proto tento model není vhodný pro systémy vyvíjené bez zapojení uživatelů [16], [26].



Obrázek 5: Fáze spirálového modelu (Zdroj: [16])

## 3.2 UML

V roce 1997 byl jazyk UML přijat sdružením OMG (Object Management Group) jako první průmyslový standard objektově orientovaného jazyka. Pod zkratkou UML se skrývá Unified Modeling Language, neboli unifikovaný modelovací jazyk. Původním záměrem bylo sjednocení všech dříve používaných metodik. UML je otevřený standard. Jedná se o jazyk pro vizuální modelování systémů. Nejčastěji je spojován s návrhy objektově orientovaných softwarových systémů. Jeho využití jsou ovšem mnohem širší. UML je nositelem vizuální syntaxe pro sestavování modelů [10].

### 3.2.1 Základní charakteristiky UML

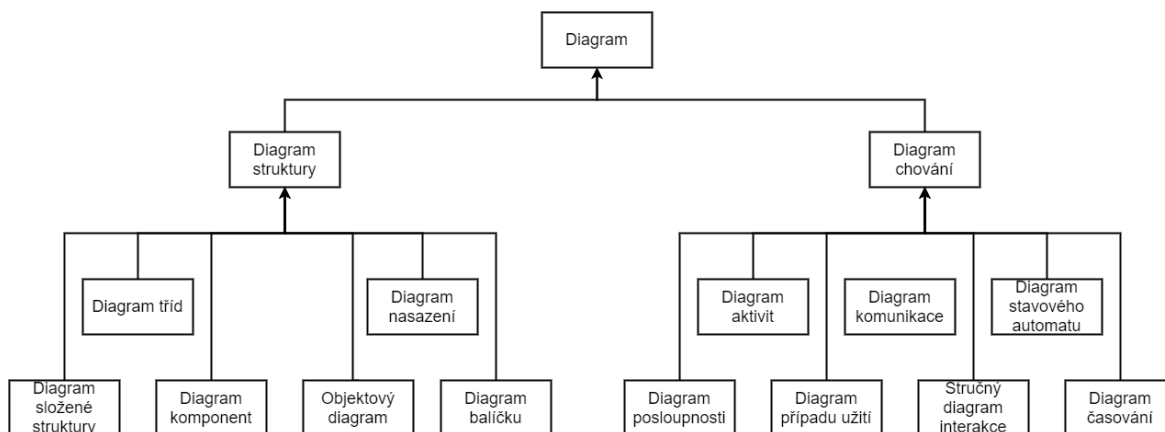
Strukturu UML lze rozdělit do 3 hlavních částí. Jedná se o stavební bloky, obecné mechanismy a architekturu [10].

#### Stavební bloky

Do této skupiny patří předměty, relace a diagramy. Předměty jsou strukturní abstrakce, které jsou vyjádřeny podstatnými jmény. Jejich chování je vyjádřené slovesy. Pokud se v modelu vyskytuje více významově souvisejících prvků, jsou seskupeny. Předměty mohou také obsahovat poznámky k zachycení dodatečných informací [11].

Relace umožňují zachytit povahu vztahu dvou rozdílných předmětů. Existuje množství typů relací, které UML definuje [10]. Jejich bližší popis a grafická reprezentace jsou uvedeny v kapitole Model tříd na straně 34.

Diagramy jsou náhledy modelu. Diagram tedy není modelem. V UML je definováno 13 různých typů diagramů. Je možné je rozdělit do 2 skupin na diagramy struktury a diagramy chování. Diagramy struktury zachycují statický model. Tedy předměty a asociace mezi nimi. Diagramy chování zachycují dynamický model. Zaměřují se na vzájemné ovlivňování jednotlivých předmětů [10]. Výčet všech diagramů a jejich rozdělení je zobrazeno na obrázku na straně 32.



Obrázek 6: Diagramy v UML a jejich dělení (Zdroj: Vlastní tvorba)

## Obecné mechanismy

V celém UML se konzistentně vyskytují 4 strategie. Jedná se o specifikace, ornamenty, podskupiny a mechanismy rozšiřitelnosti [11].

Specifikace poskytují sémantický základ modelu. Jedná se o textový popis jednotlivých prvků, bez kterých by samotná symbolika nebyla dostatečným nositelem informace. UML umožňuje, aby modely byly proškrtané, neúplné nebo nekonzistentní [27].

Ornamenty představují sadu jednoduchých symbolů k vyjádření vlastností prvku. Použití ornamentů přispívá ke zjednodušení komplikovaných struktur. Počet použitých ornamentů není omezen a je možné je přidávat do té doby, dokud vybraný prvek neobsahuje dostatečné množství podrobností [11].

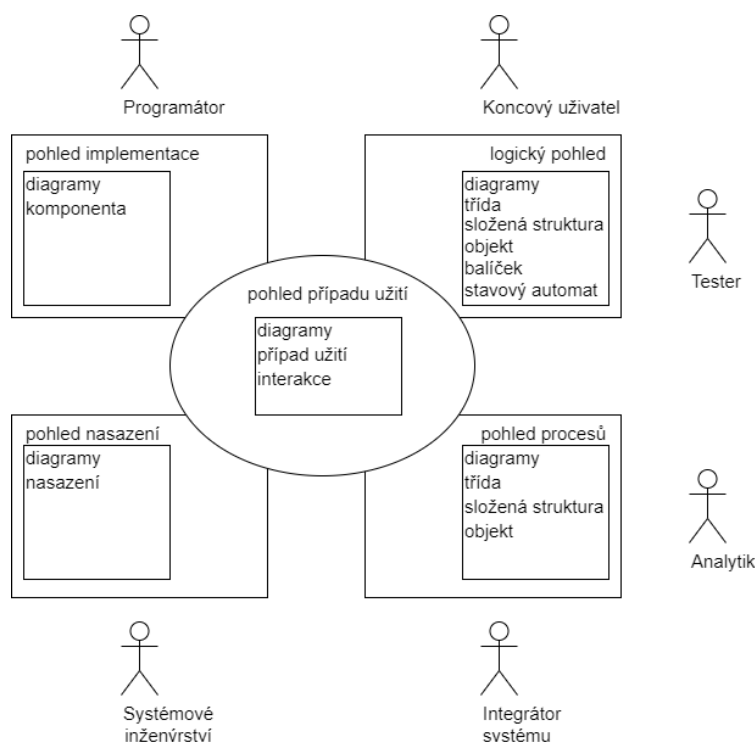
Podskupiny reprezentují možnosti různých náhledů řešené problematiky. UML rozlišuje 2 podskupiny. Jedná se o klasifikátory a instance a skupinu rozhraní a implementací. Klasifikátor představuje abstraktní představu třídy. Například obecný pojem auto. Instance už ovšem představuje konkrétní auto. Rozhraní reprezentuje vykonávanou činnost, co předmět provádí. Implementace uvádí, jakým způsobem bylo rozhraní fyzicky vyřešeno. Příkladem rozhraní může být dálkový ovladač od televize. Implementace by v tomto případě uváděla, jaký byl použit materiál, rozložení tlačítek anebo jednotlivých funkce tlačítek [10].



Mechanismy rozšiřitelnosti představují možnosti rozšíření UML. Základní mechanismy jsou 3. Jedná se o omezení, stereotypy a označené hodnoty. Omezení je vložená podmínka určená pouze pro jeden zamýšlený prvek, která musí být vyhodnocena kladně. Graficky jí reprezentují složené závorky. Pod označením stereotyp se skrývá definice nových prvků modelu za využití prvků existujících. Jejich grafickou reprezentací jsou dvojité lomené závorky. Posledním ze zmiňovaných mechanismů rozšiřitelnosti jsou označené hodnoty. Ty umožňují stávající prvky modelu rozšířit o vlastní parametry [11].

## Architektura

V UML diagramech je zachycen systém z různých úhlů pohledu. Pokud zkombinujeme několik různých pohledů, získáme architekturu. Nejčastěji je využíván pohled „4+1“. Ten tvoří čtyři pohledy a jeden, který je všechny integruje. Pohled případů užití do sebe integruje pohled logický, implementace, procesů a nasazení. Jednotlivé pohledy jsou složeny z UML diagramů. Každý pohled je svými vlastnostmi zajímavý pro jiného účastníka vývoje IS [10]. Rozložení pohledů, diagramů a rolí je zachycuje Obrázek 7 níže.



Obrázek 7: Rozložení pohledů v "4+1" (Zdroj: [10] – upraveno)

### 3.3 Systémové modelování

Při návrhu informačního systému lze využít abstraktní modely systémů. Principy tvorby a použití takových modelů označujeme za systémové modelování. V současné době se systémové modelování téměř vždy pojí s UML. Užité modely se dělí na třídní, stavové a modely interakcí. Jednotlivé druhy se navzájem doplňují tak, aby komplexně zachytily celý systém a jeho chování v čase [10].

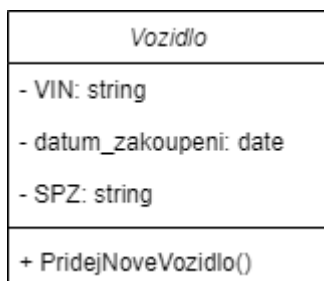
#### 3.3.1 Model tříd

Pro zachycení statické struktury systému se užívá model tříd. Model tříd neobsahuje časovou složku, proto je označován za statický. Díky svým parametrům je k objektovému modelování zpravidla využíván diagram tříd [28].

#### Diagram tříd

Diagram tříd je reprezentací grafu. Tvoří ho množství spojených uzlů a hran. Nejedná se ovšem o graf orientovaný. Jednotlivé třídy jsou reprezentovány uzly a vazby mezi třídami tvoří hrany. V diagramu tříd se mohou vyskytovat abstraktní třídy. Taková třída reprezentuje skupiny objektů, které mají podobné či stejné parametry [11].

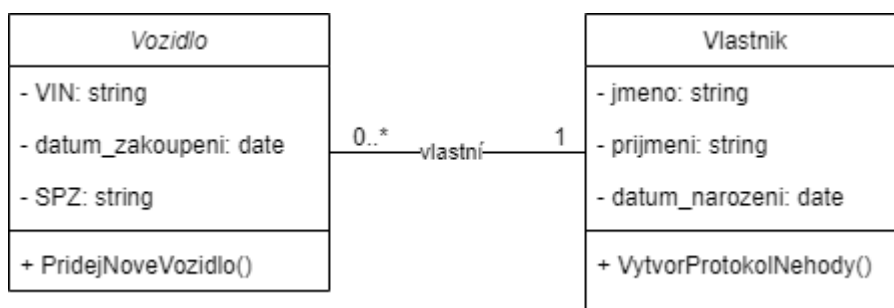
Grafická reprezentace třídy je čtverec. Nadpis určuje jméno třídy. Ve střední části jsou uvedeny všechny atributy třídy spolu s datovým typem, který budou využívat. Ve spodní části jsou uvedeny metody. Protože se jedná o metodu, je uvedený datový typ, kterým bude metoda volána a výstupní datový typ [11]. Uvedení datového typu u metod není povinné. Zobrazení diagramu tříd najdeme na obrázku níže.



Obrázek 8: Příklad diagramu tříd (Zdroj: Vlastní tvorba)

Jednotlivé třídy jsou propojeny vazbami. U vazeb vzniká kombinace parametrů, které je nutné definovat. V první řadě je to kardinalita a multiplicita. Kardinalita popisuje možný

současný počet instancí objektů jedné třídy. Multiplicita udává možný rozsah kardinalit. Často je uváděna jako násobnost. V diagramech tříd je vždy uváděna horní a spodní mez rozsahu anebo přesná hodnota. Pokud se v rozsahu vyskytuje nula, představuje nepovinný nebo žádný výskyt. Znakem hvězdy je označováno libovolné přirozené číslo. Poslední možností je výskyt konkrétní hodnoty vypsání číslem [29]. Příklad kombinací můžeme vidět na diagramu níže. První vazba ukazuje, že ke každému vozu může být žádný až neomezeně záznamů nehody. Naopak každý záznam musí mít přesně jedno přiřazené vozidlo.

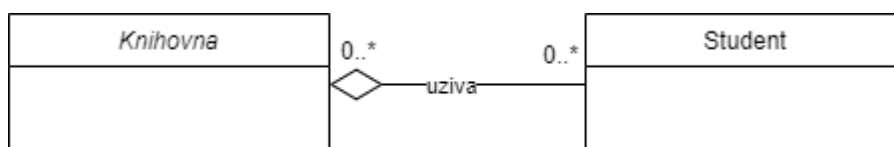


Obrázek 9: Příklad multiplicity (Zdroj: Vlastní tvorba)

Další z parametrů, který popisuje vazbu mezi třídami objektů je druh vazby. Druh vazby je graficky reprezentován symbolem v místě spojení hrany s uzlem. V diagramech tříd rozlišujeme asociace, agregace, kompozice a generalizace [11].

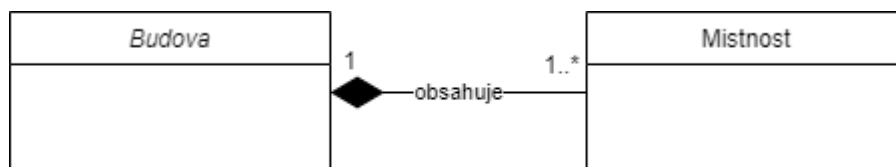
Asociace reprezentuje základní typ vztahu. Instance jednotlivých tříd mezi sebou mají přímou vazbu. Graficky je tento typ vztahu reprezentován čarou. Vazba musí obsahovat název a může být doplněna o pojmenování rolí [10]. Obrázek 9 prezentuje příklad asociace.

Agregace reprezentuje vztah entity a její části. Jedná se o volnou vazbu celku a jeho součástí. Jeden celek může využívat více částí. Součást může volně existovat i bez celku. Grafickou reprezentací agregace je prázdný kosočtverec [10]. Příklad agregace je zobrazen na diagramu níže.



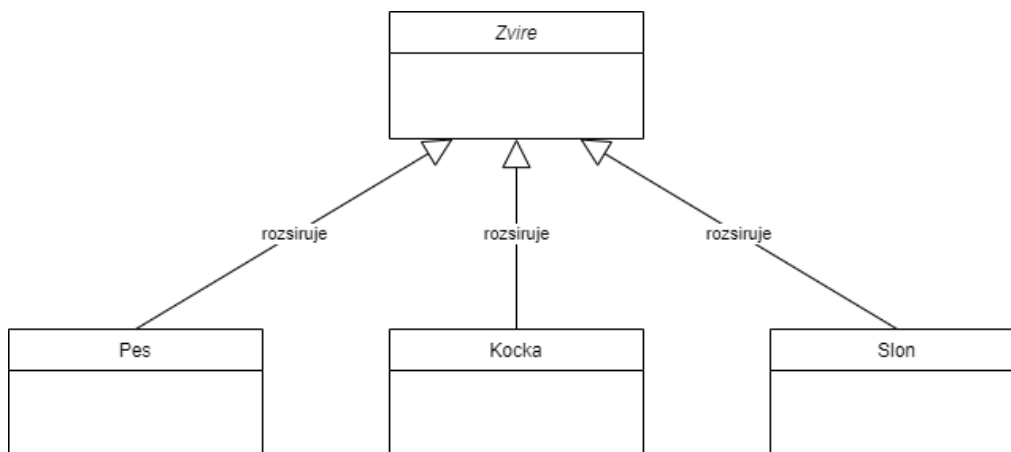
Obrázek 10: Příklad agregace (Zdroj: Vlastní tvorba)

Kompozice představuje silnou formu agregace. Kompozice je vztah mezi celkem a částí, kde část nemůže samostatně existovat. Grafickou reprezentací kompozice je plný kosočtverec [28]. Na diagramu níže je zobrazen příklad kompozice. Budova jako celek obsahuje místnosti. Místnost jako část nemůže samostatně existovat bez budovy.



Obrázek 11: Příklad kompozice (Zdroj: Vlastní tvorba)

Generalizace reprezentuje dědičnost. Jedná se o hierarchický vztah mezi třídami. Podtřída dědí vybrané atributy a metody od nadtřídy a přidává své vlastní. Generalizaci graficky reprezentuje trojúhelník [11]. Na diagramu níže je zobrazen příklad generalizace.



Obrázek 12: Příklad generalizace (Zdroj: Vlastní tvorba)

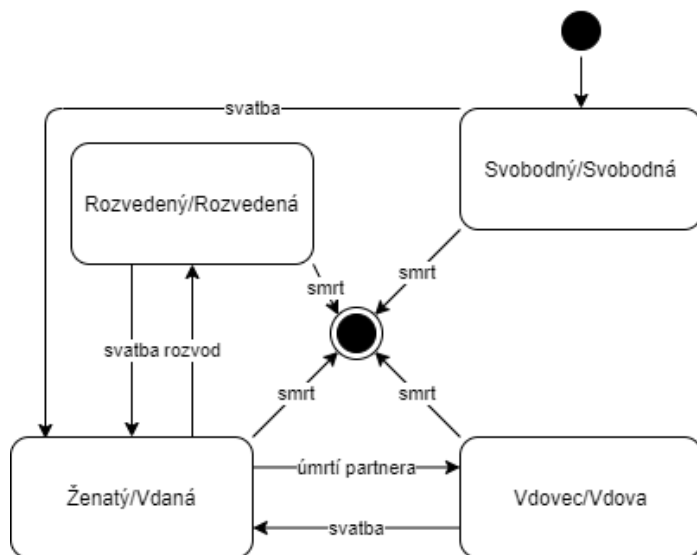
### 3.3.2 Model stavů

Model stavů zachycuje chování systému v čase. Účelem tohoto modelu je zachycení reakcí systému na rozličné vstupy z okolního prostředí. Součástí modelu stavů je také definice reakce na jednotlivé vstupy. Model stavů využívá stavové diagramy, které jsou vytvářeny pro všechny v čase proměnné třídy [28].

#### Stavový diagram

Diagram stavů zachycuje vnitřní chování jednoho objektu nebo části systému. Ve stavovém diagramu je vždy jasně definován začátek a jeden či více konců. Začátek diagramu reprezentuje výchozí stav objektu, který se následně vlivem příchozích podnětů přesouvá do stavu dalšího. Možné jsou jen ty stavy, které jsou definovány v diagramu [29].

Základními prvky stavového diagramu jsou stavy, události a přechody. Stav představuje okamžitý stav objektu a jeho atributů. Událost je podnětem, který vyvolá změnu stavu. Přejít je změna stavu objektu vyvolaná událostí [29]. Na diagramu níže jsou zobrazeny možné rodinné stavy a přechody mezi nimi. Smrt je koncovým bodem všech stavů. Svatbou je možné přejít mezi stavy „Svobodný“, „Rozvedený“ a „Ženatý“. Dalšími možnými událostmi jsou „rozvod“ a „úmrtí partnera“.



Obrázek 13: Příklad stavového diagramu (Zdroj: Vlastní tvorba)

### 3.3.3 Model interakcí

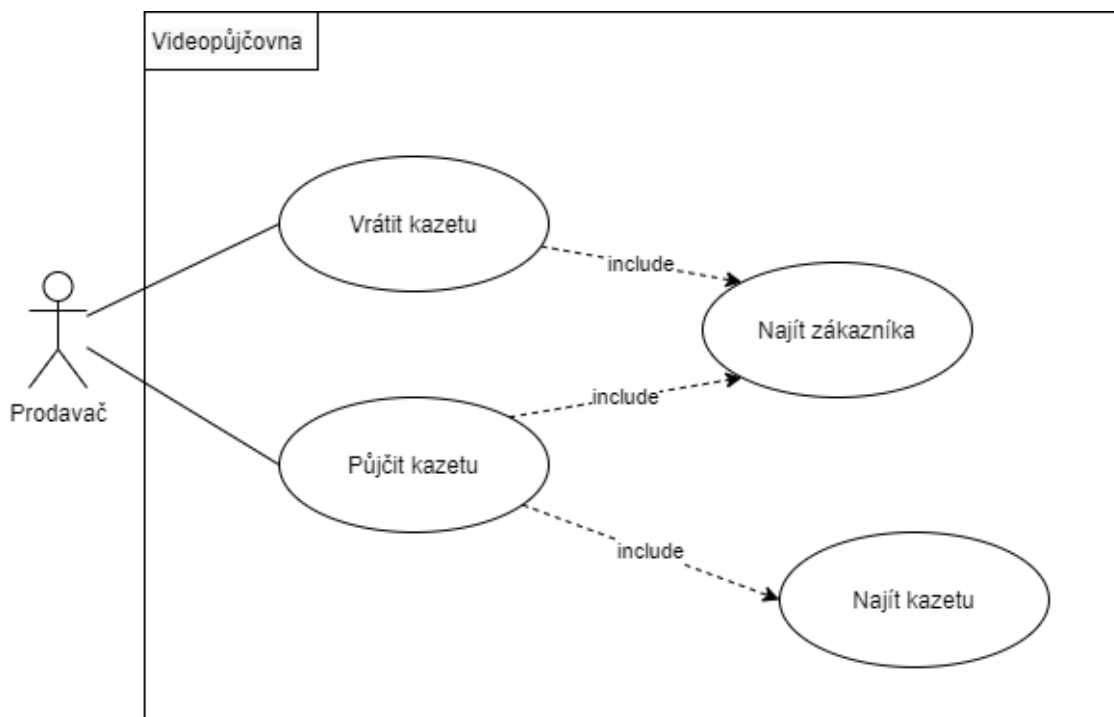
Model interakcí zachycuje probíhající uživatelské i systémové interakce. Výchozím bodem pro tyto interakce bývá uživatelský vstup. Popisovaný umožňuje prezentovat uživatelům interakce se systémem, možnosti uživatelských průchodů a důsledky jednotlivých kroků. Model interakcí tvoří především diagramy případů užití, sekvenční diagramy a diagramy aktivit [11].

#### Diagram případů užití

Pro zachycení chování systému z pohledu uživatele využíváme diagram případů užití. Diagram zachycuje všechny typy uživatelů systému a případy užití, které jsou s nimi asociovány. Případy užití bývají nazývány Use Case. Uživatelé, kteří do procesu zasahují jsou nazýváni aktéry. Vztah mezi aktérem a případem užití, tedy tok informací, reprezentuje souvislá čára. Jednotlivé případy užití mezi sebou mají také vztahy [10].

V Use Case diagramu rozlišujeme 3 typy vztahů, zahrnutí, rozšíření a generalizaci. *Zahrnutí* označované jako „include“ reprezentuje vztah povinnosti, kdy v situaci užití jednoho případu je užit i případ druhý. *Zahrnutí* je graficky znázorněno přerušovanou čarou s textovým označením „include“ a směřováním k případu podřízenému. *Rozšíření* označované jako „extend“ představuje vztah rozšiřitelnosti, kdy jeden případ užití může, ale nemusí zahrnovat i případ užití druhý. *Rozšíření* je graficky reprezentováno přerušovanou čarou s označením „extend“ a šipkou mířící k nadřazenému případu užití. *Generalizace* se dělí na 2 typy. Jeden je využíván ke znázornění dědičnosti mezi aktéry. Druhý typ reprezentuje speciální vazby mezi jednotlivými případy užití [10].

Obrázek 14 zobrazuje příklad diagramu případů užití. Videopůjčovna, kde prodavač provádí 2 činnosti. Tou první je vrácení kazety, které zahrnuje vyhledání zákazníka. Druhým uvedeným případem užití je půjčení kazety. Zde je nutné nejen vyhledat zákazníka, ale také samotnou kazetu.



Obrázek 14: Příklad diagramu případu užití (Zdroj: [30])

### Sekvenční diagram

Reprezentaci následnosti činností mezi jednotlivými objekty či aktéry tvoří sekvenční diagram. Tento diagram zobrazuje dílčí kroky, které systém provede v následnosti na akci uživatele. Sekvenční diagramy se pojí se scénáři jednotlivých případů užití systému. Scénář je strukturovaný sled událostí s jasnou definicí rolí, alternativních scénářů, vstupních a výstupních podmínek [28].

### Diagram aktivit

K popsání aktivit v modelu interakcí na nejnižší úrovni slouží diagram aktivit. Jeho účelem je zejména zachycení lineárních a paralelních toků řízení. Každý diagram aktivit obsahuje začátek reprezentovaný černým kruhem a konec reprezentovaný kruhem s opsanou kružnicí. Kosočtverce reprezentují rozhodování a tlustá čára paralelizaci nebo sjednocení procesů [29].

## **4 Vlastní práce**

### **4.1 Bankovní instituce**

Povinnost zpracovávat exekuční příkazy a s nimi související dokumenty je dána zákonem. Pro účely této práce byla zvolena nejmenovaná bankovní instituce spravující finance více jak 1,5 milionu uživatelů. Jedná se tedy o institut vyžadující robustní systém pro správu a archivaci velkého množství exekučních příkazů. Protože práce s legislativními dokumenty vyžaduje znalost právních dokumentů a může mít v případě pochybení právní důsledky, vyčlenil bankovní institut specializované pracoviště, které se zabývá pouze těmito činnostmi.

#### **4.1.1 Stávající systém**

Zpracování legislativních dokumentů, nejen tedy pouze exekučních příkazů, v současné době probíhá manuálně. Zprávy jsou do systému staženy z datové schránky. Následně DZ získá uživatel ke zpracování. Manuálně přepíše hodnoty ze zobrazeného dokumentu do systému. Po dokončení zpracování je datová zpráva předána dalšímu uživateli, který provede kontrolu. Pokud se v zadání vyskytuje chyba, je zpráva vrácena zadavateli, který nedostatky odstraní. Všichni uživatelé zastávají více rolí, tedy mohou provádět kontrolu i zpracování. Důležitou podmínkou zpracování je zaručení kontroly čtyř očí. Tedy zpracování a kontrolu nesmí provádět ten samý uživatel.

#### **4.1.2 Zpracovatelské oddělení**

Zpracovatelské oddělení tvoří 20 zaměstnanců, kteří jsou rozděleni do 2 týmů. Každý tým má svého vedoucího, který působí jako supervizor. První tým se zaměřuje na zpracování a kontrolu příchozích DZ. Druhý tým je dedikován pro následující správu exekucí. Stará se o správné blokace položek na účtech, provedení výplat z jednotlivých účtů a další úkony související se správou zadaných dat.



## 4.2 Legislativní pojmy

Pro návrh informačního systému pro správu a zpracování legislativních požadavků je nutné nejdříve jednotlivé pojmy a mechaniky definovat.

### Účastníci

Exekuční příkaz zahrnuje více stran. Pro potřeby této práce je nutné uvést 3 a to sice povinné, oprávněné a právní subjekt exekuci vykonávající, tedy výstavce. Povinný je osobou, proti které je exekuční řízení vedeno. Může se jednat o fyzickou osobu, fyzickou osobu podnikající a právnickou osobu. Exekuční příkaz vždy uvádí alespoň jednoho povinného. Dále je povinně uváděn vztah povinného vůči exekučnímu řízení. Zde lze rozlišovat 3 vztahy. Dlužník, spoludlužník a manžel/manželka. Oprávněný je osobou nárokující prostředky, které jsou předmětem exekučního řízení. Posledním účastníkem, jak je uvedeno výše, je výstavce. Z informačního systému datových schránek (ISDS) lze získat seznam držitelů datových schránek (SDS). Tento seznam určuje subjekty, které mohou odesílat datové zprávy. Výstavci jsou děleni do skupin. Příkladem lze uvést exekutory, celní anebo finanční úřady [31], [32], [33], [34], [35].

### Exekuce

Exekuce je státem definovaný proces získávání prostředků k vyrovnání soudem uznaných dluhů. Samotnému provedení exekuce předchází minimálně 2 soudní řízení. V prvním soud zkoumá existenci a výši potencionálního dluhu. Na základě nálezu soud určí do kdy má dlužník povinnost dluh splatit. Pokud není stanovená lhůta splněna, může se věřitel obrátit na soud a v rámci druhého řízení požádat o nařízení exekuce. Následně soud pověří exekutora provedením exekuce [34].

Povinností bankovní instituce je okamžité zadržetí prostředků povinného. Banka tak koná na základě exekučního příkazu. Jedná se o legislativní dokument, který jednoznačně specifikuje oprávněné, povinné, účty povinného a částky, které jsou součástí exekučního příkazu. Tyto položky jsou v bankovních aplikacích označeny jako položky platebních předpisů. Součástí těchto předpisů jsou i účty výstavce, kam mají být vymožené prostředky vyplaceny [36].

Na základě exekučního příkazu je založen exekuční případ. Entita, která uchovává všechny informace spojené s konkrétním řízením. Exekuční řízení má mnoho dalších atributů a stavů, které nejsou pro účely této práce podstatné, proto nejsou popsány [36].

### **Datové zprávy**

Legislativní dokumenty jsou do banky zasílány pomocí systému datových zpráv. Datová zpráva (DZ) je složena z obálky a přílohy. Obálka obsahuje předmět zprávy, informace o odesílateli a přílohy. Ty jsou tvořeny jednotlivými dokumenty nesoucími sdělení [32].

### **Elektronický exekuční příkaz**

Od roku 2015 je možné zasílat elektronické exekuční příkazy (EEP). Tyto příkazy jsou zasílány ve strojově čitelném formátu xml. Legislativa definuje strukturu a obsah dat odesílaných těmito soubory. Spolu s xml souborem datová zpráva obvykle obsahuje také pdf, které je nositelem totožné informace, ale je pro lidského uživatele lépe čitelné [32], [34].

## **4.3 Požadavky na nový systém**

Tato kapitola se zabývá definicí požadavků na doplňkový systém pro zpracování EEP. Hlavními omezujícími faktory pro nový IS jsou principy stávajícího systému, které musí být respektovány. Dalším důležitým faktorem jsou legislativní nařízení, která musí být za každou cenu dodržena.

### **4.3.1 Funkční požadavky**

Uvedené funkční požadavky mají společný základ s požadavky stávajícího systému. Zde budou uvedeny pouze ty, které jsou relevantní pro systém nový. Bezpečnostní politika banky definuje metodické požadavky na všechny interně užívané aplikace. Proto požadavky na přihlašování a odhlašování uživatele uváděny nejsou.

| Aktér      | Identifikátor | Požadavek  |
|------------|---------------|--|
| Kontrolor  | K01           | Systém umožňuje vyhledání příchozí datové zprávy         |
|            | K02           | Systém umožňuje získání volné DZ ke kontrole             |
|            | K03           | Systém umožňuje kontrolu každého dokumentu datové zprávy |
|            | K04           | Systém umožňuje předání DZ na supervizi                  |
|            | K05           | Systém umožňuje předání DZ k ručnímu zpracování          |
| Supervizor | S01           | Systém umožňuje vyhledání příchozí datové zprávy         |
|            | S02           | Systém umožňuje získání volné DZ k supervizi             |
|            | S03           | Systém umožňuje úpravu každého dokumentu datové zprávy   |
|            | S04           | Systém umožňuje předání DZ na supervizi                  |
|            | S05           | Systém umožňuje předání DZ k ručnímu zpracování          |

Tabulka 1: Funkční požadavky (Zdroj: Vlastní tvorba)

### 4.3.2 Nefunkční požadavky

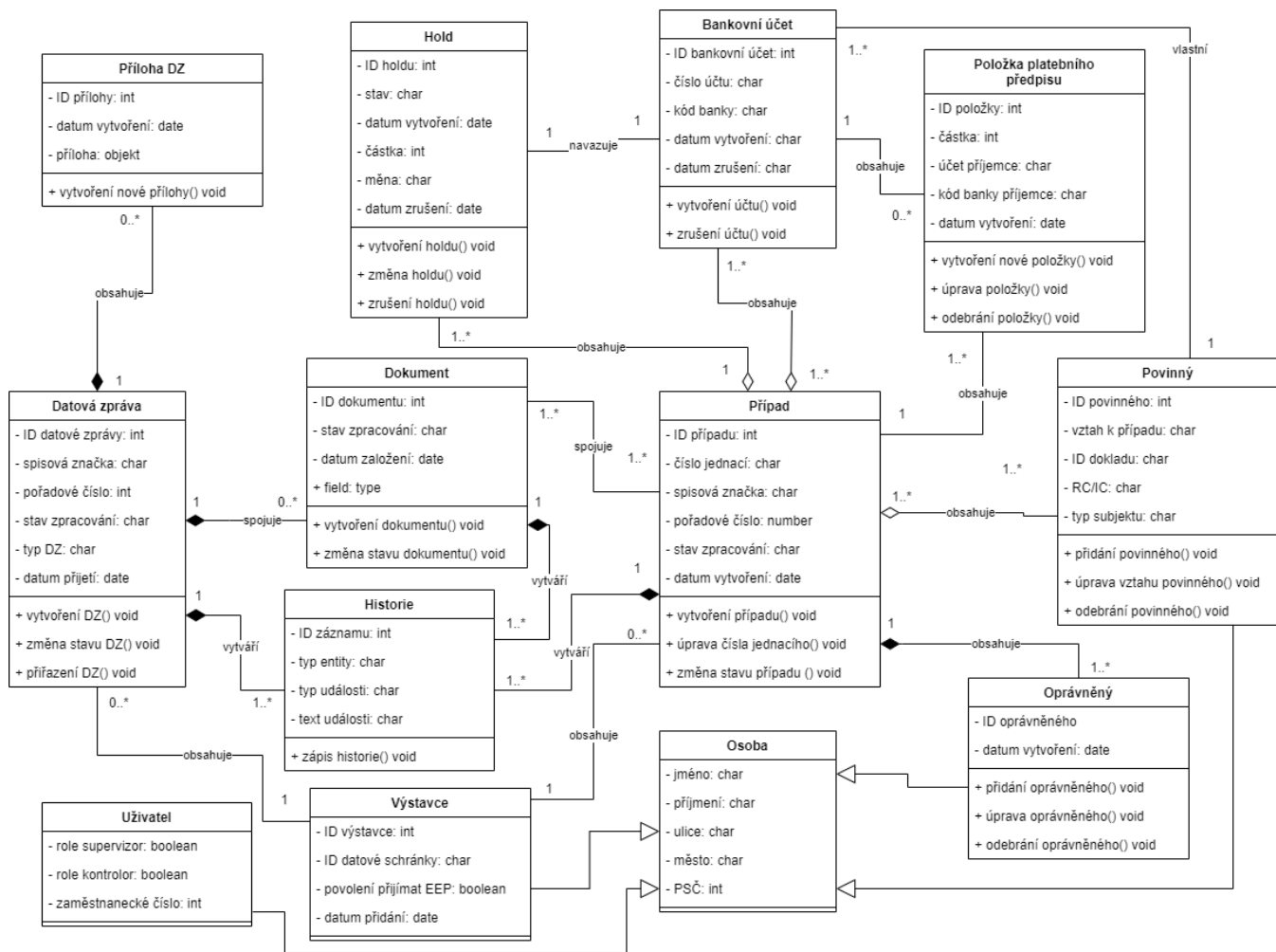
Uživatelé pracují na bankou přiřazeném počítači. Jedná se o typizovanou sestavu. Je tedy nutné systém těmito omezením přizpůsobit. Dostupnost systému je rozdílná pro uživatele a zpracování DZ z toho důvodu, že je nutné co nejdříve zajistit prostředky na účtech. Zbytek úkonů není natolik prioritní. Uživatelé nesmí být umožněn pohyb po aplikaci mimo navržené průchody.

| Identifikátor | Požadavek   |
|---------------|---|
| N01           | Systém musí být pro uživatele dostupný minimálně 99 % času pracovní doby. |
| N02           | Systém musí být kompatibilní s bankou podporovanou verzí prohlížeče.      |
| N03           | Systém musí být kompatibilní se standardní bankovní PC sestavou.          |
| N04           | Systém nedovolí práci s DZ, která není uživateli aktuálně přiřazena.      |
| N05           | Systém nedovolí pohyb uživatele pomocí URL injection.                     |
| N06           | Systém musí být schopný zpracování datové zprávy 99 % času.               |
| N07           | Systém musí být schopen zpracovat DZ v objemu tisíců.                     |
| N08           | Systém musí být technologicky kompatibilní se stávajícím systémem.        |

Tabulka 2: Nefunkční požadavky (Zdroj: Vlastní tvorba)

## 4.4 Objektový model

Pro potřeby specifikace pohledu na hlavní třídy systému, jejich atributy, metody a vzájemné působení bude využit model tříd. Prezentovaný rozsah tříd není nikterak rozsáhlý, ale přesto některé části vyžadují dodatečné komentáře. Proto za diagramem tříd následuje datový slovník, který dodatečně informace sdružuje.



Obrázek 15: Diagram tříd (Zdroj: Vlastní tvorba)

## Datový slovník

- *Datová zpráva* – obsahuje informace o datových zprávách. Nové instance vznikají v rámci stahování DZ z datové schránky. Během svého životního cyklu datová zpráva mění své stavy v závislosti na jednotlivých krocích zpracování, kontroly a supervize. Datové zprávy jsou i po přechodu do konečného stavu v systému uchovávány.
- *Příloha DZ* – uchovává přílohy DZ a vztah k nim. Nové přílohy jsou tvořeny v rámci stahování datových zpráv z datové schránky. Přílohy po zpracování datové zprávy zůstávají v systému uchovány. Jejich pozdější archivaci se tento systém nezabývá.
- *Osoba* – abstraktní třída, která funguje jako nadtřída pro třídy *Uživatel*, *Výstavce*, *Oprávněný*, *Povinný*. *Osoba* sdružuje informace o osobách a jejich adresách.
- *Uživatel* – slouží jako nositel role uživatele a identifikátor pracovníka, který provedl změny a akce nad dalšími třídami.
- *Výstavce* – podtřída třídy *Osoba*. *Výstavce* sdružuje všechny subjekty, které mohou zasílat legislativní dokumenty spojené s exekčním řízením. U jednotlivých subjektů dále evidujeme informaci, zda mají povoleno zasílat elektronickou verzi dokumentů.
- *Oprávněný* – uchovává informace o všech oprávněných subjektech jednotlivých exekčních příkazů. Informace o oprávněném nejsou spojeny s dalšími částmi bankovních systémů. Vznik instancí oprávněného zajišťuje automat během zpracování DZ. Jejich možnou editaci a odebrání provádí *Supervizor* v rámci supervize.
- *Povinný* – obsahuje informace o povinných subjektech v exekčních případech. Jelikož je povinný klientem banky, jsou jeho data v okolních systémech již dostupná. Proto vytvoření instancí předchází vyhledání dostupných dat v okolních systémech a v případě shody jejich uložení. Odebrání povinného může provádět *Supervizor* v rámci supervize.

- *Bankovní účet* – uchovává informace o exekucích postižených účtech povinných. Účty jsou přidávány stejným způsobem jako povinné. Tedy vyhledáním v okolních systémech banky. Jejich editace není možná a odebrat je od případu může pouze *Supervizor* v rámci supervize.
- *Položka platebního předpisu* – obsahuje informace o jednotlivých finančních položkách exekučních příkazů. Modelovaný systém se zabývá pouze založením a základními editačními prvky jednotlivých položek. Další části životního cyklu, denní přepočty výše úroků a další nezbytné úkony jsou úmyslně vynechány.
- *Hold* – entita sdružující položky umožňující blokace prostředků na jednotlivých účtech. Po zadání případu jsou automaticky vytvořeny holdy ve výši součtu jednotlivých položek platebních předpisů daného exekučního příkazu na všechny postižené účty povinného. Změna a zrušení holdu vyplývá ze změny položek platebních předpisů či jednotlivých účtů v rámci provedené supervize.
- *Dokument* – realizuje vazbu mezi třídami *Datová zpráva* a *Případ*. Ostatní dokumenty exekučního řízení umožňují propojení více případů z jedné přílohy datové zprávy. Životní cyklus dokumentů je řízený jednotlivými kroky zpracování, kontroly a supervize.
- *Případ* – třída, která sdružuje informace o exekučním případě. Jsou k ní navázány instance tříd *Povinný*, *Bankovní účet*, *Oprávněný*, *Položka platebního předpisu*, *Hold*, *Výstavce*. Případy jsou vytvářeny na základě dokumentů. Stavů případů se mění na základě probíhajících kroků zpracování, kontroly a supervize.
- *Historie* – sdružuje všechny provedené změny ve třídách *Datová zpráva*, *Dokument*, *Případ*.

## 4.5 Stavový model

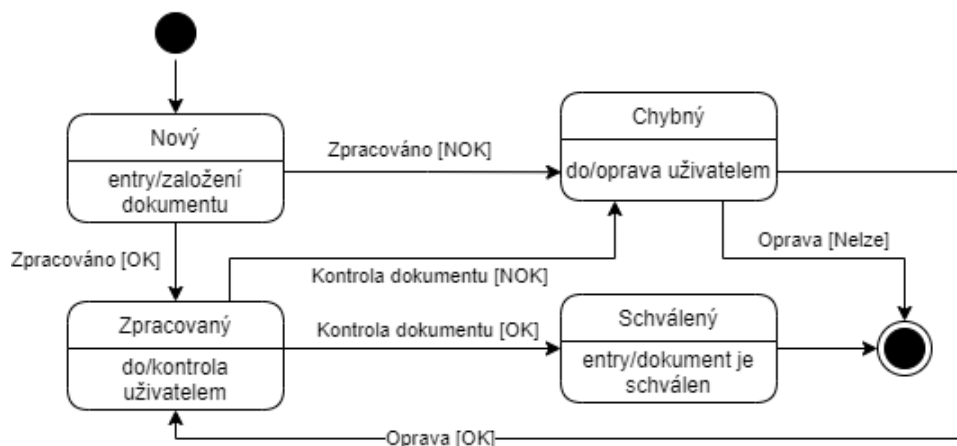
Pro popsání fungování navrhovaného řešení byly vybrány 3 třídy, u kterých budou pomocí stavového diagramu popsány možné stavy a přechody mezi nimi. Zvolené třídy jsou *Datová zpráva*, *Dokument*, *Případ*. Celý systém obsahuje značné množství prvků, které mohou nabývat více stavů, ale pro účely této práce nebudou zmíněny.

Jakmile je stažena nová DZ určená k elektronickému zpracování, nabývá stavu **Stažená XML**. Následně probíhá validace DZ a navázaných příloh. Pokud validace nedopadne úspěšně, tak DZ nabývá stavu **Nezpracovatelná**. Pokud validace dopadne úspěšně, je změněn stav datové zprávy na **Validovaná**. Všechny DZ následně vstoupí do automatizovaného zpracování.

Výsledkem zpracování je buďto chyba a změna stavu na **Nezpracovatelná** anebo je zpracování úspěšné a stav je změněn na **Kontrolovaná**. Datové zprávy ve stavu **Kontrolovaná** jsou postupně přiřazovány uživatelům ke kontrole. Jestliže je vše validně zpracováno, *Kontrolor* označí navázané dokumenty **Schválený**. Následkem toho je stav DZ nastaven na **Uzavřená**, což je jeden z koncových stavů. Pokud datová zpráva kontrolou neprojde s pozitivním výsledkem, je převedena do stavu **Nezpracovatelná**, což je druhý koncový stav. Reprezentuje datové zprávy, které není možné automatizovaně zpracovat. Po označení tímto stavem je DZ předána stávající aplikaci, kde uživatelé provedou kompletní ruční zadání.

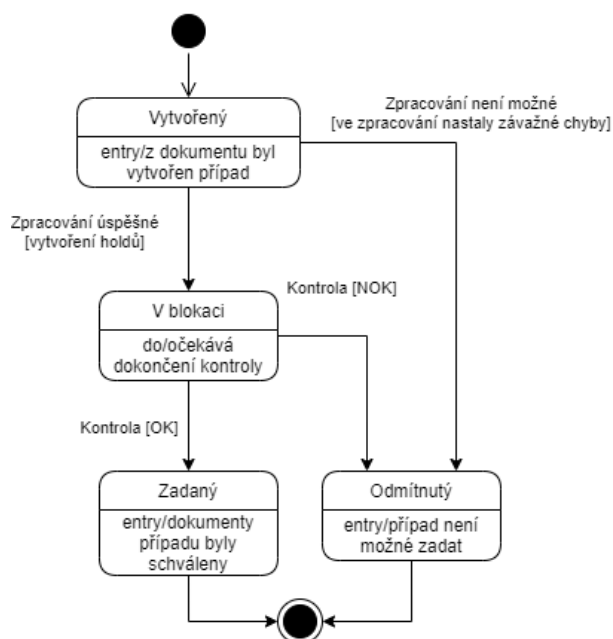






Obrázek 17: Stavový diagram: Dokument (Zdroj: Vlastní tvorba)

Pokud je během zpracování vytvořen dokument, je zároveň vytvořen i případ ve stavu **Vytvořený**. Po vytvoření holdů na prostředky je změněn stav případu na **V blokaci**. Jakmile DZ, na základě které vznikl dokument, a tedy i případ úspěšně projde kontrolou, je změněn stav na **Zadaný**, což je v tomto případě konečný stav. Poslední užívaný stav je **Odmítnutý**. Do tohoto stavu přechází případ v situaci, kdy není možné zpracovat. Ve chvíli, kdy bude případ předán stávajícímu systému na pokračující zpracování, musí obsahovat shodné stavy, které vypovídají o stavu zpracování. Případ v rámci svého životního cyklu nabývá dalších stavů, které ale vzhledem k rozsahu prezentovaného návrhu nejsou zobrazeny.



Obrázek 18: Stavový diagram: Případ (Zdroj: Vlastní tvorba)

## 4.6 Systémové interakce

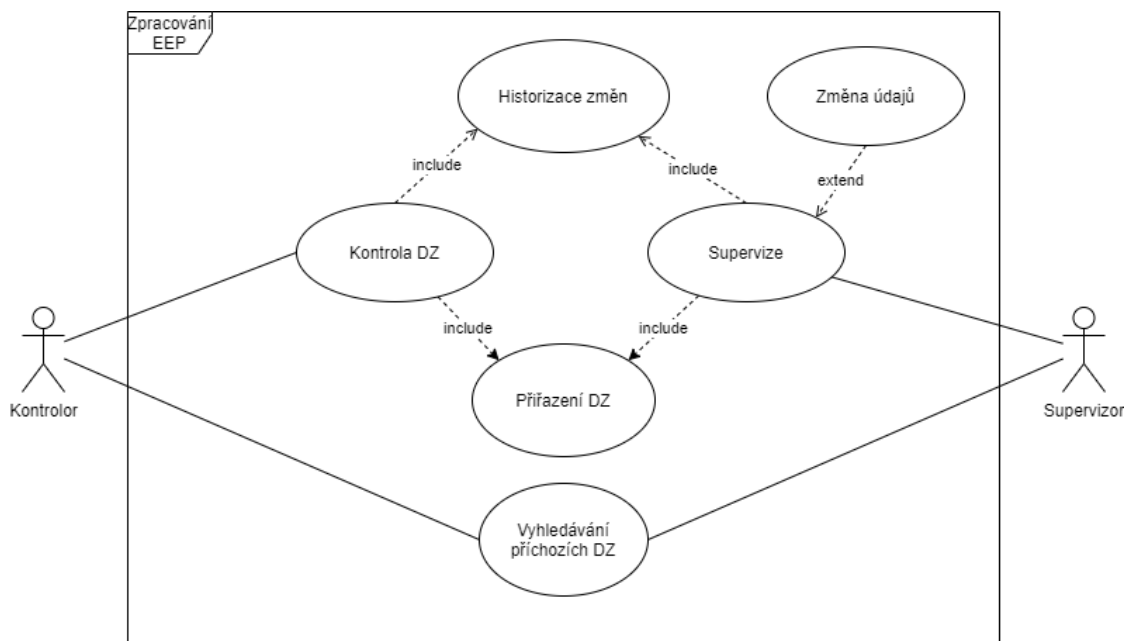
Modely interakcí znázorňují vzájemné působení a ovlivňování jednotlivých tříd systému. Nejvyšší úroveň představuje diagram případů užití. Ten definuje interakce jednotlivých aktérů se systémem. Následně jednotlivé interakce podrobněji rozvádí sekvenční diagramy, které zachycují volání jednotlivých metod v čase. Nejdetailejší pohled na sekvence úkonů a rozhodování přinášejí diagramy aktivit.

### 4.6.1 Případy užití

Základní složkou diagramu případů užití jsou aktéři. V popisovaném systému působí 2 aktéři. Jedná se o supervizora a kontrolora. Oba aktéry mohou tvořit uživatelé současné aplikace.

Prvním aktérem je *kontrolor*. Tohoto uživatele reprezentují zaměstnanci s nižší kvalifikací, kteří mají za úkol hlavně zkontrolovat, zda systém správně rozpoznal a zpracoval zasláná data. V případě, že naleznou nesoulad, rozhodnou, zda se jedná o zcela špatné zpracování, případně předají kontrolovanou DZ supervizorovi, který rozhodne, co se bude dít dále a pokračují v kontrolách.

Druhým aktérem je *supervizor*. Jedná se o pracovně a kvalifikačně zkušeného pracovníka, který má vyšší úroveň zodpovědnosti. Supervizor může přiřazenou DZ upravovat a opravit tím drobné nedostatky automatického zpracování.



Obrázek 19: Diagram případů užití (Zdroj: Vlastní tvorba)

Diagram případů užití, jednotlivé aktéry a k nim navázané příslušné případy užití zobrazuje Obrázek 19 na straně 50. Oba aktéři mají přístup k *Vyhledávání příchozích DZ*.

*Kontrolor* využívá činnost *Kontrola DZ* což je jeho hlavní činnost. Na ní jsou navázány *Přirazení DZ* a *Historizace změn*. Tyto případy jsou vždy při kontrole využity. Tedy kontrolorovi je přiřazena DZ ke kontrole a veškeré jím provedené změny jsou historizovány.

Supervizor využívá činnost *Supervize*. Na tento případ užití jsou navázány *Přirazení DZ* a *Historizace změn*. Tedy shodně jako v případě kontroly je aktérovi přiřazena DZ k supervizi a veškeré provedené úkony jsou historizovány. Supervizor může dále využít *Změna údajů*, kdy ručně zasáhne do systémem zpracovaných dat.

#### 4.6.2 Sekvenční diagramy

Sekvenční diagramy uvedené v této kapitole jsou svázané se scénáři, které popisují jednotlivé kroky a dodatečné informace. Z diagramu případů užití na straně 50 vycházejí jako vhodné *Kontrola DZ* a *Supervize*. Pro případ užití *Kontrola DZ* jsou níže popsány 3 scénáře. Pro případ užití *Supervize* jsou připraveny scénáře 2.

##### **Kontrola DZ**

Prvním uvedeným scénářem je úspěšné provedení kontroly. Uživatel s rolí *Kontrolor* otevře webovou stránku informačního systému. Systém zobrazí hlavní menu, přes které *Kontrolor* zahájí kontrolu volných zpracovaných DZ. Systém vybere volnou DZ ke kontrole, přiřadí ji uživateli a zobrazí její obsah na obrazovce. Pro každý dokument navázaný k DZ je nabídnuta možnost provedení kontroly, která umožní uživateli provést ověření zadání daného dokumentu. Zobrazené informace kontrolovaného dokumentu jsou v režimu čtení, tedy nejsou editovatelné. Pokud je vše správně, označí uživatel dokument jako schválený. Tím ukončí proces kontroly nad daným dokumentem a pokud je k dispozici další, zatím nezkontrolovaný dokument, pokračuje uživatel v kontrolách.

Když jsou kontroly všech dokumentů dokončeny, vyhodnotí systém stav všech navázaných dokumentů. Pokud jsou všechny ve stavu **Schválený**, systém nabídne možnost ukončit kontrolu datové zprávy. Tato volba znamená změnu stavu DZ na **Uzavřená**, uložení historie změn, ukončení kontroly DZ a pokud je k dispozici další DZ ke kontrole, je uživateli načtena. Popisovaný scénář je zachycen v Tabulka 3 na straně 53.

Druhý uvedený scénář se zabývá provedením kontroly, kdy zadané dokumenty mají drobné nedostatky a jejich doplnění je nutné svěřit supervizorovi. Začátek scénáře je shodný s přechozím, tedy je otevřena aplikace, načtena DZ, provedena kontrola dokumentu, ale zpracování dat není možné označit za správné. Uživatel tedy označí dokument jako chybný. Systém zobrazí uživateli pole pro vložení komentáře. Uživatel zde vyplní, jaké měl důvody pro předání supervizorovi. Aplikace tedy označí dokument stavem **Chybný**. Uživatel pokračuje kontrolou ostatních navázaných dokumentů. Jakmile jsou kontroly dokončeny a alespoň jeden dokument není ve stavu **Schválený**, nabídne systém uživateli volbu, zda převést datovou zprávu do ručního zpracování anebo na supervizi. Uživatel zvolí supervizi. Systém změní stav datové zprávy na **Posuzovaná**, zanesse provedené změny do historie, ukončí kontrolu DZ a pokud je k dispozici další DZ ke kontrole, je uživateli načtena. Tento scénář je popsán v Tabulka 4 na straně 54.

Poslední uvedený scénář kontroly uvádí práci s datovou zprávou, která je zadána kompletně špatně, její náprava není v tomto režimu možná, a proto musí být předána na ruční zpracování. Začátek scénáře je tedy stejný jako u předchozích. *Kontrolor* projde až k provedení kontroly dokumentu a označí dokument jako chybný. Systém zobrazí pole pro vložení komentáře. Uživatel zde vyplní, jaké měl důvody pro označení dokumentu jako chybného. Následně je systémem změněn stav dokumentu na **Chybný** a uživatel má možnost pokračovat v kontrolách dalších navázaných dokumentů. Po dokončení dílčích kontrol systém nabídne volbu, zda převést datovou zprávu do ručního zpracování anebo na supervizi. Uživatel zvolí ruční zpracování. Systém provede zápis provedených změn do historie, ukončí kontrolu DZ a pokud je dostupná další datová zpráva, tak je uživateli přiřazena a zobrazena. Tento scénář zachycuje Tabulka 5 na straně 55.

Obrázek 20 na straně 56 zachycuje sekvenční diagram *Kontrola DZ*. Tuto aktivitu provádí množství uživatelů najednou. Proto je jednou z prvních aktivit provolání metody na přiřazení DZ. Ty nejsou přiřazovány uživatelům předem proto, aby docházelo k maximálnímu využití dostupných kapacit *Kontrolorů*.

| <b>Případ užití: Kontrola DZ</b> |  |
|----------------------------------|--|
| <b>ID</b>                        | Kon1   |
| <b>Nadpis</b>                    | Úspěšné provedení kontroly   |
| <b>Hlavní aktér</b>              | Kontrolor  |
| <b>Vstupní podmínky</b>          | Je dostupná DZ ke kontrole<br>Kontrolovaná DZ je správně zadána<br>Uživatel je v roli kontrolora   |
| <b>Kroky scénáře</b>             | <ol style="list-style-type: none"> <li>1. <i>Kontrolor</i> zahájí kontrolu</li> <li>2. Systém uživateli přidělí dostupnou DZ</li> <li>3. <ol style="list-style-type: none"> <li>a) Systém zobrazí detail datové zprávy</li> <li>b) Systém zobrazí navázané případy</li> <li>c) Systém zobrazí náhled na zdrojový soubor</li> <li>d) Systém pro každý dokument ve stavu <b>Zpracovaný</b> zobrazí rozhraní pro zahájení kontroly</li> </ol> </li> <li>4. <i>Kontrolor</i> zahájí kontrolu dokumentu</li> <li>5. <ol style="list-style-type: none"> <li>a) Systém zobrazí informace o zadaném případě v režimu čtení</li> <li>b) Systém zobrazí rozhraní pro schválení či odmítnutí dokumentu</li> </ol> </li> <li>6. <i>Kontrolor</i> schválí dokument</li> <li>7. <ol style="list-style-type: none"> <li>a) Systém označí dokument jako schválený</li> <li>b) Pokud je dostupný další dokument ke kontrole, následuje návrat k 4. kroku</li> </ol> </li> <li>8. Uživatel uzavře datovou zprávu</li> <li>9. Systém označí DZ jako uzavřenou</li> <li>10. Veškeré provedené úkony jsou uloženy do historie</li> <li>11. <ol style="list-style-type: none"> <li>a) Pokud je dostupná další DZ ke kontrole, následuje návrat k 2. kroku</li> <li>b) Pokud není dostupná další DZ ke kontrole, následuje návrat do hlavního menu</li> </ol> </li> </ol> |
| <b>Výstupní podmínky</b>         | DZ má zkontrolované všechny navázané dokumenty   |
| <b>Alternativní scénář</b>       | Alespoň jeden dokument je chybně zpracovaný.<br>Zpracování je kompletně špatné<br>V systému není dostupná DZ ke kontrole   |

Tabulka 3: Scénář případu užití: Úspěšné provedení kontroly (Zdroj: Vlastní tvorba)

| <b>Případ užití: Kontrola DZ</b> |  |
|----------------------------------|--|
| <b>ID</b>                        | Kon2   |
| <b>Nadpis</b>                    | Převedení DZ do Supervize  |
| <b>Hlavní aktér</b>              | Kontrolor  |
| <b>Vstupní podmínky</b>          | Je dostupná DZ ke kontrole<br>Kontrolovaná DZ má drobné nedostatky<br>Uživatel je v roli kontrolora  |
| <b>Kroky scénáře</b>             | <ol style="list-style-type: none"> <li>1. <i>Kontrolor</i> zahájí kontrolu</li> <li>2. Systém uživateli přidělí dostupnou DZ</li> <li>3. <ol style="list-style-type: none"> <li>a) Systém zobrazí detail datové zprávy</li> <li>b) Systém zobrazí navázané případy</li> <li>c) Systém zobrazí náhled na zdrojový soubor</li> <li>d) Systém pro každý dokument ve stavu <b>Zpracovaný</b> zobrazí rozhraní pro zahájení kontroly</li> </ol> </li> <li>4. <i>Kontrolor</i> zahájí kontrolu dokumentu</li> <li>5. <ol style="list-style-type: none"> <li>a) Systém zobrazí informace o zadaném případě v režimu čtení</li> <li>b) Systém zobrazí rozhraní pro schválení či odmítnutí dokumentu</li> </ol> </li> <li>6. <i>Kontrolor dokument</i> neschválí</li> <li>7. <ol style="list-style-type: none"> <li>a) Systém označí dokument jako schválený</li> <li>b) Pokud je dostupný další dokument ke kontrole, následuje návrat k 4. kroku</li> </ol> </li> <li>8. Uživatel předá datovou zprávu na supervizi</li> <li>9. Systém označí DZ jako posuzovanou</li> <li>10. Veškeré provedené úkony jsou uloženy do historie</li> <li>11. <ol style="list-style-type: none"> <li>a) Pokud je dostupná další DZ ke kontrole, následuje návrat k 2. kroku</li> <li>b) Pokud není dostupná další DZ ke kontrole, následuje návrat do hlavního menu</li> </ol> </li> </ol> |
| <b>Výstupní podmínky</b>         | DZ má zkontrolované všechny navázané dokumenty   |
| <b>Alternativní scénář</b>       | Všechny dokumenty jsou korektně zadány<br>Zpracování je kompletně špatné<br>V systému není dostupná DZ ke kontrole   |

Tabulka 4: Scénář případu užití: Předání z kontroly na supervizi (Zdroj: Vlastní tvorba)

| <b>Případ užití: Kontrola DZ</b> |  |
|----------------------------------|--|
| <b>ID</b>                        | Kon3   |
| <b>Nadpis</b>                    | Převedení DZ do ručního zpracování   |
| <b>Hlavní aktér</b>              | Kontrolor  |
| <b>Vstupní podmínky</b>          | Je dostupná DZ ke kontrole<br>Kontrolovaná DZ je kompletně špatně zpracována<br>Uživatel je v roli kontrolora  |
| <b>Kroky scénáře</b>             | <ol style="list-style-type: none"> <li>1. <i>Kontrolor</i> zahájí kontrolu</li> <li>2. Systém uživateli přidělí dostupnou DZ</li> <li>3. <ol style="list-style-type: none"> <li>a) Systém zobrazí detail datové zprávy</li> <li>b) Systém zobrazí navázané případy</li> <li>c) Systém zobrazí náhled na zdrojový soubor</li> <li>d) Systém pro každý dokument ve stavu <b>Zpracovaný</b> zobrazí rozhraní pro zahájení kontroly</li> </ol> </li> <li>4. <i>Kontrolor</i> zahájí kontrolu dokumentu</li> <li>5. <ol style="list-style-type: none"> <li>a) Systém zobrazí informace o zadaném případě v režimu čtení</li> <li>b) Systém zobrazí rozhraní pro schválení či odmítnutí dokumentu</li> </ol> </li> <li>6. <i>Kontrolor dokument</i> neschválí</li> <li>7. <ol style="list-style-type: none"> <li>a) Systém označí dokument jako schválený</li> <li>b) Pokud je dostupný další dokument ke kontrole, následuje návrat k 4. kroku</li> </ol> </li> <li>8. Uživatel předá datovou zprávu k ručnímu zpracování</li> <li>9. Systém označí DZ jako posuzovanou</li> <li>10. Veškeré provedené úkony jsou uloženy do historie</li> <li>11. <ol style="list-style-type: none"> <li>a) Pokud je dostupná další DZ ke kontrole, následuje návrat k 2. kroku</li> <li>b) Pokud není dostupná další DZ ke kontrole, následuje návrat do hlavního menu</li> </ol> </li> </ol> |
| <b>Výstupní podmínky</b>         | DZ má zkontrolované všechny navázané dokumenty   |
| <b>Alternativní scénář</b>       | Všechny dokumenty jsou korektně zadány<br>Alespoň jeden dokument je chybně zpracovaný.<br>V systému není dostupná DZ ke kontrole   |

Tabulka 5: Scénář případu užití: Předání z kontroly na ruční zpracování (Zdroj: Vlastní tvorba)





## Supervize

Prvním scénářem supervize je úprava DZ s drobnými nedostatky. Uživatel s rolí *Supervisor* vstoupí do procesu supervize. Systém vybere volnou DZ k supervizi, přiřadí ji uživateli a načte její obsah. Pro každý dokument, připojený k DZ, může *Supervisor* provést supervizi dle svého uvážení. Pro dokumenty ve stavu **Chybný** je supervize provedena vždy. V rámci supervize jsou zobrazeny všechny údaje příslušného dokumentu v editovatelné formě a rozhraní k potvrzení provedených změn anebo označení dokumentu za neopravitelný.

*Supervisor* může provést změny na každé ze zobrazených položek. Po dokončení úprav uživatel potvrdí provedené změny. Systém změny provedené na dokumentu uloží a nastaví jeho stav na **Zpracovaný**. *Supervisor* pokračuje v práci s ostatními připojenými dokumenty. Pokud jsou všechny navázané dokumenty ve stavu **Zpracovaný**, zobrazí systém rozhraní k předání datové zprávy zpět ke kontrole. Uživatel DZ předá. Systém následně změní stav datové zprávy na **Kontrolovaná**, uloží provedené změny do historie, ukončí supervizi a načte uživateli obsah další dostupné DZ k supervizi. Tento scénář je popsán Tabulka 6 na straně 58.

Druhým scénářem supervize je předání DZ na ruční zpracování. *Supervisor* stejně jako v předchozím scénáři otevře aplikaci, započne supervizi a označí aktuální přílohu za chybnou. Systém změní stav dokumentu na **Chybný**. *Supervisor* dokončí práci s ostatními dokumenty. Pokud je alespoň jeden dokument po provedení supervize označen stavem **Chybný** systém nabídne rozhraní k ukončení supervize datové zprávy. Po jeho potvrzení je uživatel vyzván ke vložení komentáře. *Supervisor* zde uvede důvody svých úprav a označení dokumentu stavem **Chybný**. Systém změní stav datové zprávy na **Nezpracovatelná**, ukončí supervizi a pokud je k dispozici další volná DZ k supervizi, je přiřazena uživateli a načtena. Tabulka 7 na straně 59 tento scénář zachycuje.

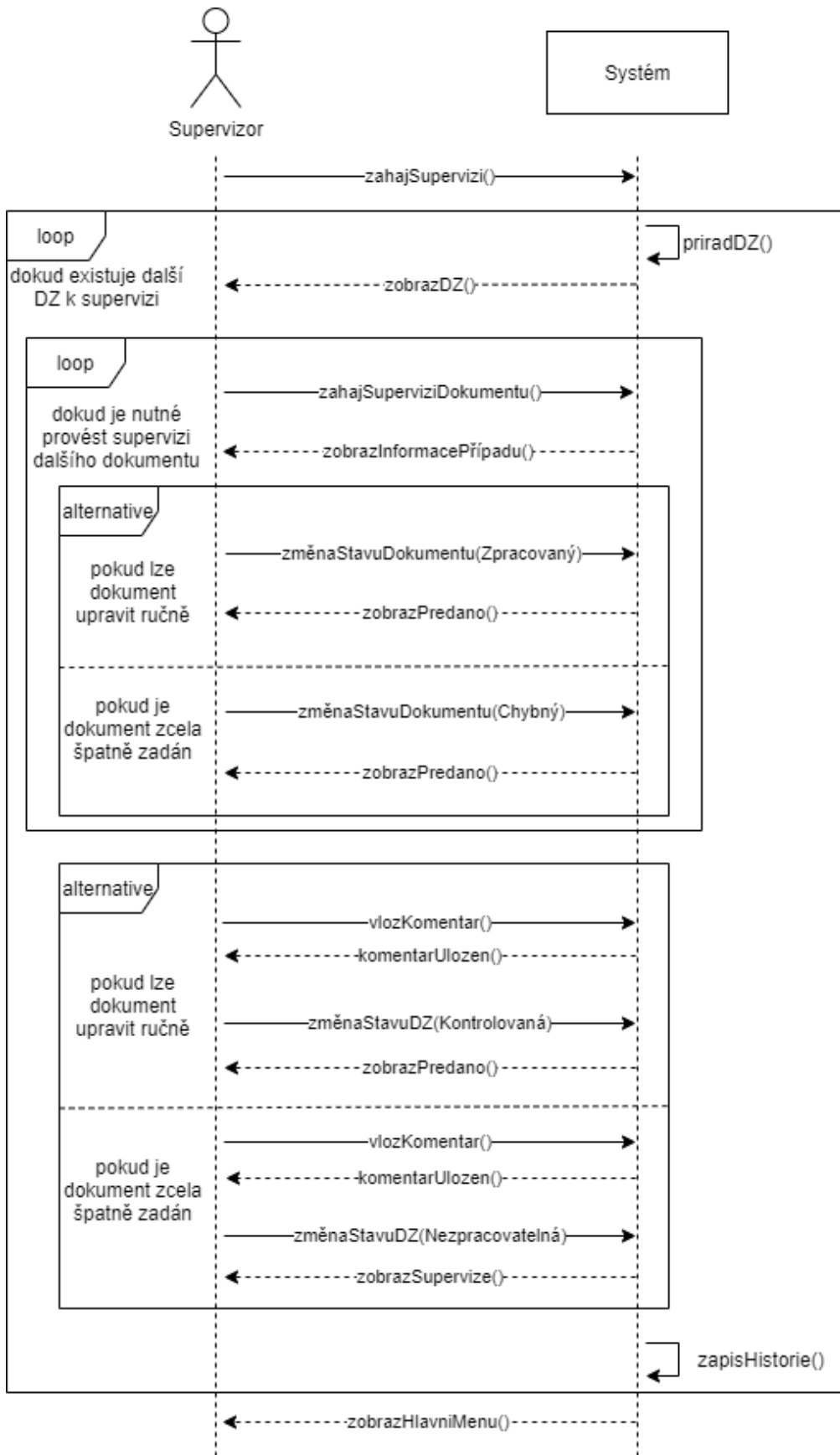
Obrázek 21 obsahuje sekvenční diagram *Supervize*. Každá úprava zadání musí být zkontrolována uživatelem, který změny neprováděl a má roli *Kontrolor*.

| <b>Případ užití: Supervize DZ</b> |   |
|-----------------------------------|---|
| <b>ID</b>                         | Sup1  |
| <b>Nadpis</b>                     | Úspěšné provedení supervize   |
| <b>Hlavní aktér</b>               | Supervizor  |
| <b>Vstupní podmínky</b>           | Je dostupná DZ k supervizi<br>Supervizovaná DZ má drobné nedostatky<br>Uživatel je v roli supervizora   |
| <b>Kroky scénáře</b>              | <ol style="list-style-type: none"> <li>1. <i>Supervizor</i> zahájí supervizi</li> <li>2. Systém uživateli přidělí dostupnou DZ</li> <li>3. <ol style="list-style-type: none"> <li>a) Systém zobrazí detail datové zprávy</li> <li>b) Systém zobrazí navázané případy</li> <li>c) Systém zobrazí náhled na zdrojový soubor</li> <li>d) Systém pro každý dokument ve stavu <b>Posuzovaný</b> zobrazí rozhraní pro zahájení supervize</li> </ol> </li> <li>4. Uživatel zahájí supervizi dokumentu</li> <li>5. <ol style="list-style-type: none"> <li>a) Systém zobrazí informace o zadaném případě v editačním režimu</li> <li>b) Systém zobrazí rozhraní pro schválení či odmítnutí dokumentu</li> </ol> </li> <li>6. <i>Supervizor</i> schválí dokument</li> <li>7. <ol style="list-style-type: none"> <li>a) Systém označí dokument jako zpracovaný</li> <li>b) Pokud je dostupný další dokument k supervizi, následuje návrat k 4. kroku</li> </ol> </li> <li>8. Uživatel uzavře datovou zprávu</li> <li>9. Systém vrátí DZ zpět ke kontrole</li> <li>10. Veškeré provedené úkony jsou uloženy do historie</li> <li>11. <ol style="list-style-type: none"> <li>a) Pokud je dostupná další DZ k supervizi, následuje návrat k 2. kroku</li> <li>b) Pokud není dostupná další DZ k supervizi, následuje návrat do hlavního menu</li> </ol> </li> </ol> |
| <b>Výstupní podmínky</b>          | Všechny navázané dokumenty prošly supervizí   |
| <b>Alternativní scénář</b>        | Zpracování je kompletně špatné<br>V systému není dostupná DZ k supervizi  |

Tabulka 6: Scénář případu užití: Předání ze supervize ke kontrole (Zdroj: Vlastní tvorba)

| <b>Případ užití: Supervize DZ</b> |   |
|-----------------------------------|---|
| <b>ID</b>                         | Sup2  |
| <b>Nadpis</b>                     | Převedení DZ do ručního zpracování  |
| <b>Hlavní aktér</b>               | Supervizor  |
| <b>Vstupní podmínky</b>           | Je dostupná DZ k supervizi<br>Supervizovaná DZ je kompletně špatně zpracována<br>Uživatel je v roli supervizora   |
| <b>Kroky scénáře</b>              | <ol style="list-style-type: none"> <li>1. <i>Supervizor</i> zahájí supervizi</li> <li>2. Systém uživateli přidělí dostupnou DZ</li> <li>3. <ol style="list-style-type: none"> <li>a) Systém zobrazí detail datové zprávy</li> <li>b) Systém zobrazí navázané případy</li> <li>c) Systém zobrazí náhled na zdrojový soubor</li> <li>d) Systém pro každý dokument ve stavu <b>Posuzovaný</b> zobrazí rozhraní pro zahájení supervize</li> </ol> </li> <li>4. Uživatel zahájí supervizi dokumentu</li> <li>5. <ol style="list-style-type: none"> <li>a) Systém zobrazí informace o zadaném případě v editačním režimu</li> <li>b) Systém zobrazí rozhraní pro schválení či odmítnutí dokumentu</li> </ol> </li> <li>6. <i>Supervizor</i> schválí dokument</li> <li>7. <ol style="list-style-type: none"> <li>a) Systém označí dokument jako zpracovaný</li> <li>b) Pokud je dostupný další dokument k supervizi, následuje návrat k 4. kroku</li> </ol> </li> <li>8. Uživatel uzavře datovou zprávu</li> <li>9. Systém vrátí DZ zpět ke kontrole</li> <li>10. Veškeré provedené úkony jsou uloženy do historie</li> <li>11. <ol style="list-style-type: none"> <li>a) Pokud je dostupná další DZ k supervizi, následuje návrat k 2. kroku</li> <li>b) Pokud není dostupná další DZ k supervizi, následuje návrat do hlavního menu</li> </ol> </li> </ol> |
| <b>Výstupní podmínky</b>          | Všechny navázané dokumenty prošly supervizí   |
| <b>Alternativní scénář</b>        | Zpracování má drobné nedostatky<br>V systému není dostupná DZ k supervizi   |

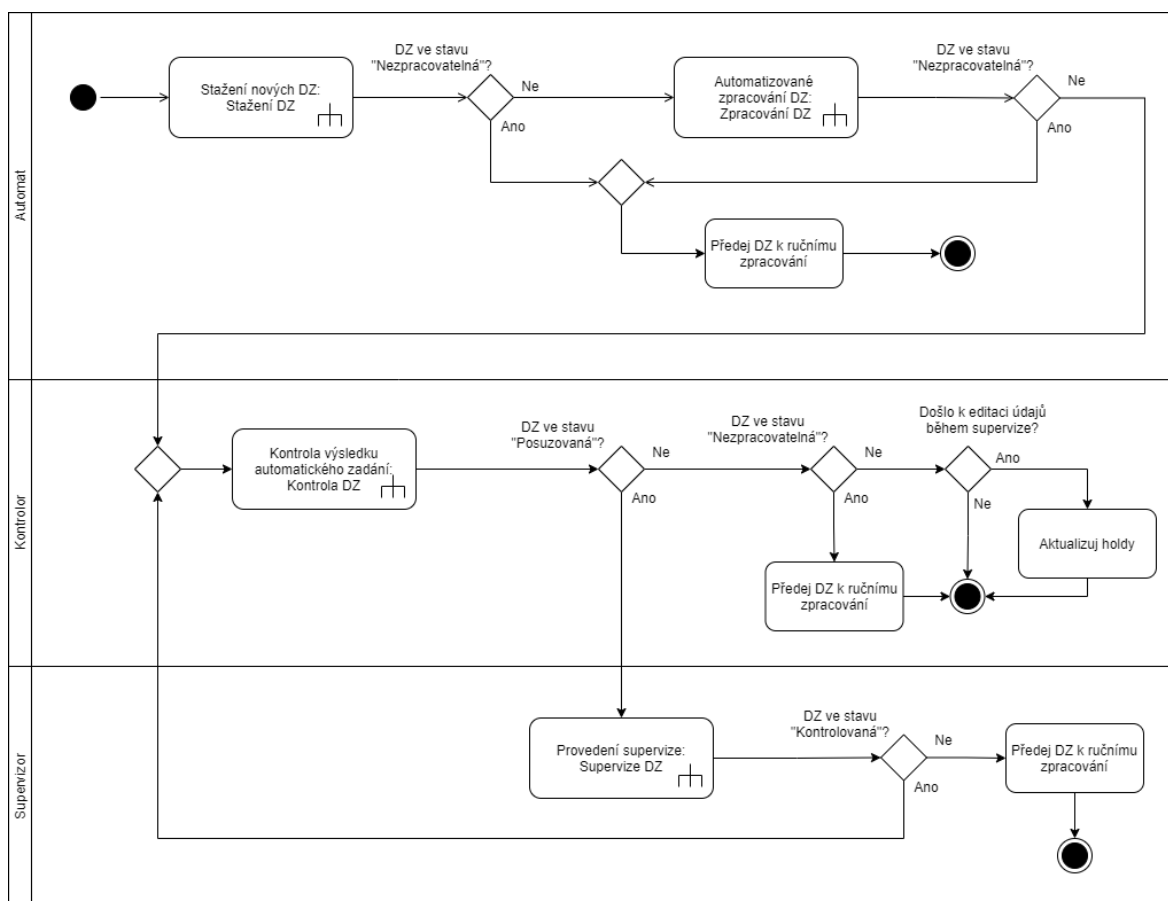
Tabulka 7: Scénář případu užití: Předání ze supervize na ruční zpracování (Zdroj: Vlastní tvorba)



Obrázek 21: Sekvenční diagram: Supervize (Zdroj: Vlastní tvorba)

### 4.6.3 Diagramy aktivit

Významná část návrhu IS se zabývá automatizovaným zpracováním dat. Proto i aktivity diagramy níže popsané jsou rozděleny do několika částí, které jsou postupně specifikovány. Nejdříve je celý proces popsán velmi obecně. Celé zpracování lze rozdělit do 4 hlavních částí. Nejdříve dojde ke stažení a validaci datových zpráv. Následně je provedeno jejich zpracování. Dále jsou výsledky zpracování DZ prezentovány uživatelům, který v rámci kontroly ověří, že informace uvedené v xml jsou korektně přeneseny do bankovních systémů. Pokud kontrolor nepotvrdí správnost výsledků, je možné předat DZ supervizorovi, který může provádět drobné editace a následně DZ vrátit k opětovné kontrole. Pokud je již vše v pořádku, kontrolor potvrdí správnost vytvořených dokumentů a případů. Tím celý modelovaný proces končí. Tento obecný proces znázorňuje Obrázek 22 s diagramem aktivit.



Obrázek 22: Aktivitní diagram: Obecný proces zpracování EEP (Zdroj: Vlastní tvorba)

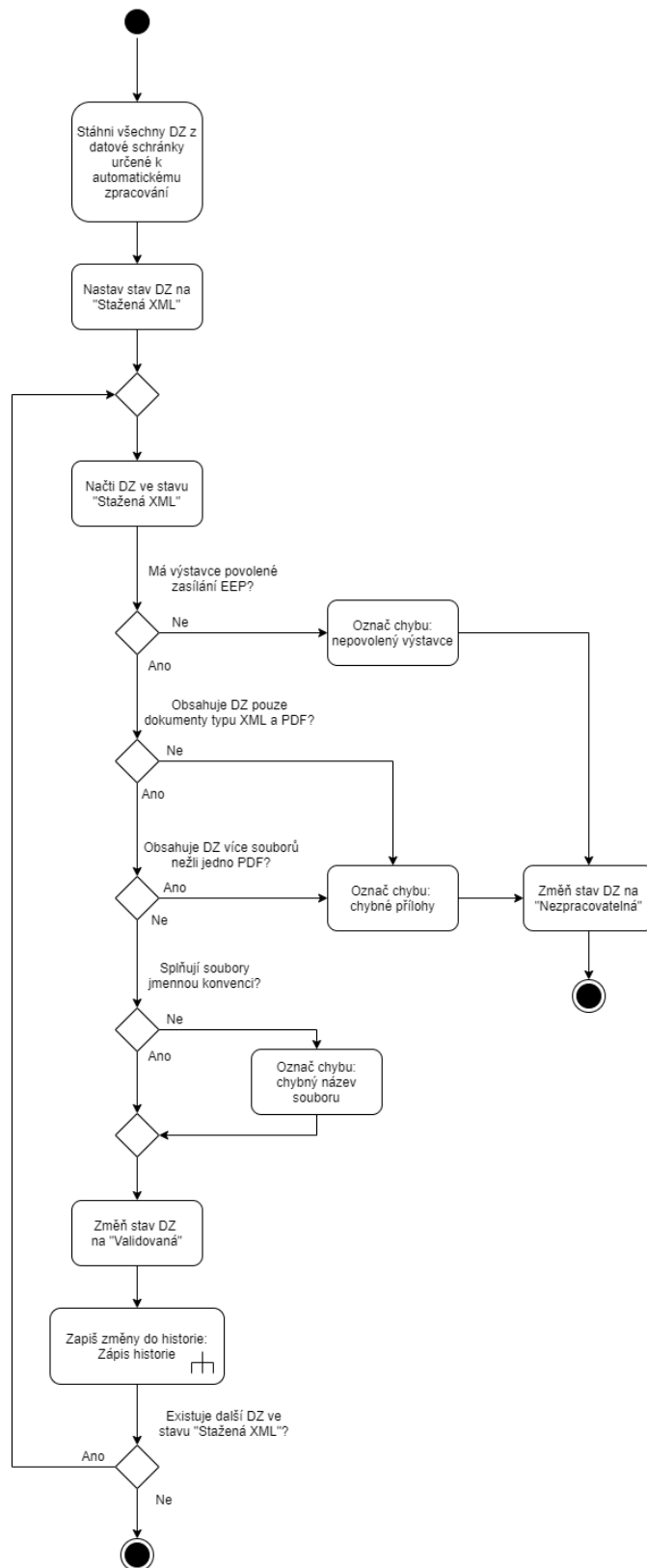
## Stažení nových DZ

Datové zprávy je nutné nejdříve získat z datové schránky bankovní instituce. Tento proces je řešen na obecné úrovni pro všechny bankovní aplikace. Proto zde není tato aktivita více rozvedena. Následně je nastaven výchozí stav DZ. Automat poté provede několik kontrol zaměřených na splnění konvencí stanovených legislativou. Vybrané kontroly jsou zaměřeny na datovou zprávu, její vlastnosti a vlastnosti jejích příloh. Obsah příloh je validován v kroku *Zpracování DZ*. Pokud datová zpráva projde všemi kontrolami, změní automat její stav na **Validovaná** a uloží provedené akce do historie. Pokud DZ validacemi neprojde, je označena k ručnímu zpracování a dále s ní není pracováno. Aktivita diagram *Stažení DZ* je zachycuje Obrázek 23 na straně 63.

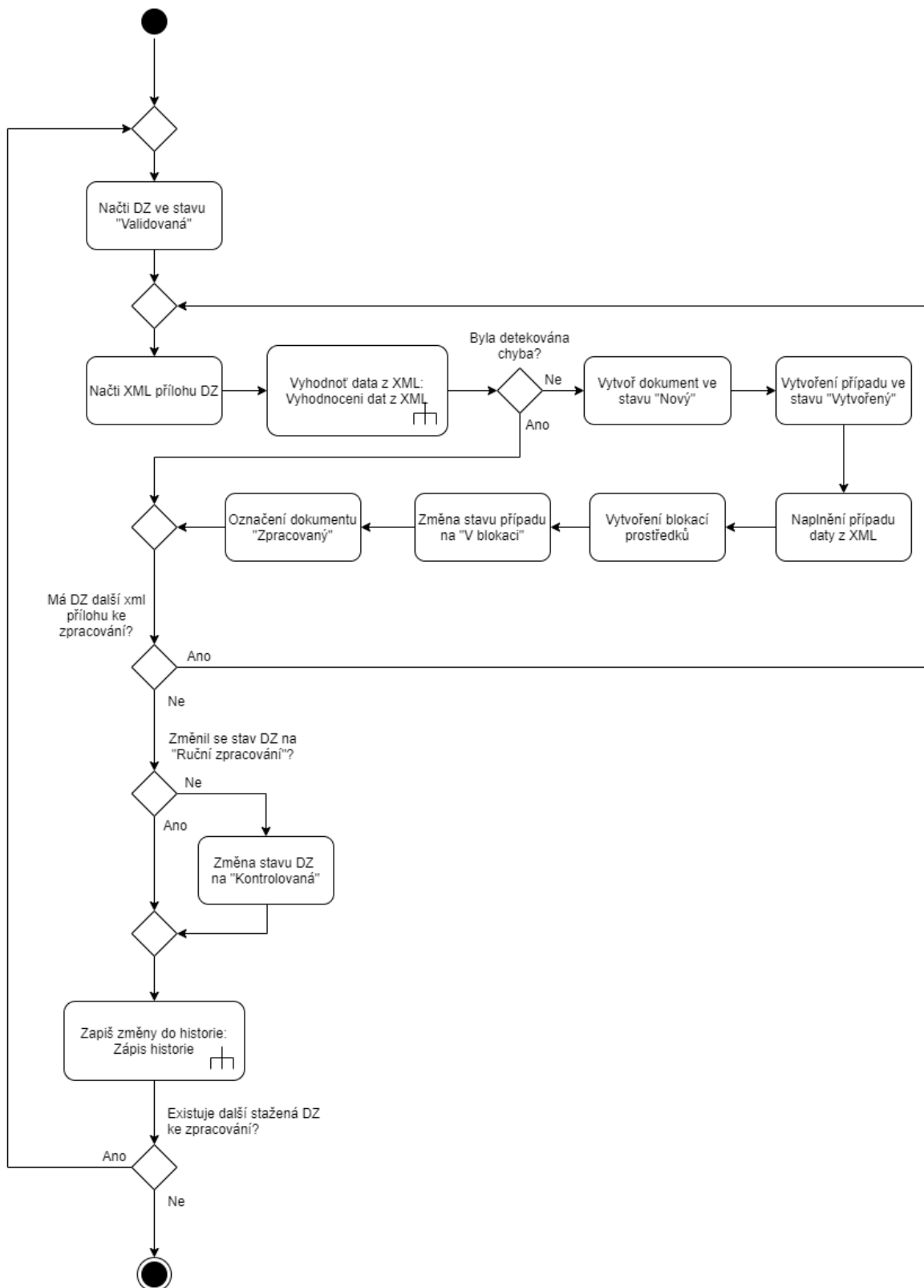
## Zpracování DZ

Stažené zprávy, které prošly prvotní validací a jejich stav nebyl označen **Nezpracovatelná** jsou po jedné načítány ke zpracování. Zpracování probíhá po jednotlivých xml přílohách. Nejdříve je provedeno vyhodnocení obsahu přílohy, které je podrobně ilustrováno na straně 66. Pokud nedošlo k detekování chyby, je následně vytvořen dokument navázaný na přílohu DZ. Dokument je ve stavu **Nový**. Na základě dokumentu vznikne případ, který je naplněn údaji z xml.

Po vyplnění údajů případu jsou založeny holdy, tedy blokace prostředků na účtech přiřazených k případu. Jakmile jsou zadány blokace prostředků na účtech, je změněn stav případu na **V blokaci**. Tímto končí zpracování vybrané přílohy. Stav navázaného dokumentu je změněn na „Zpracovaný“ a pokud je ke zpracovávané datové zprávě navázána další nezpracovaná příloha typu xml, automat ji začne zpracovávat. V opačném případě dojde k ověření, že při zpracování všech příloh nedošlo k chybám. Pokud bylo zpracování bez nahlášených chyb, stav datové zprávy je nastaven na **Kontrolovaná**. Následuje krok zápisu informací do historie. Následně automat hledá další DZ ke zpracování. Pokud se již taková v systému nenachází, proces končí. Aktivita diagram *Zpracování DZ* zachycuje Obrázek 24 na straně 64.



Obrázek 23: Aktivita diagram: Stažení DZ (Zdroj: Vlastní tvorba)



Obrázek 24: Aktivita diagram: Zpracování DZ (Zdroj: Vlastní tvorba)



## Vyhodnocení dat z xml

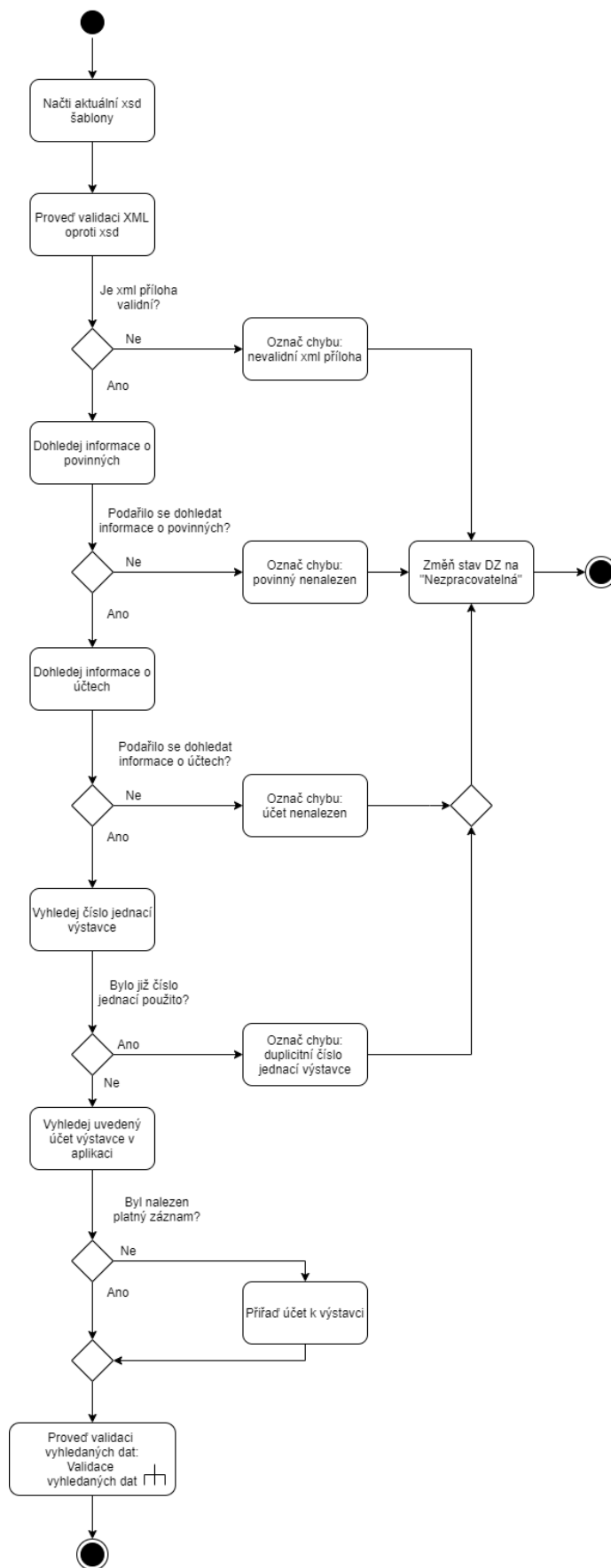
Pro každou xml přílohu je nutné provést kontrolu správnosti obsahu. Nejdříve jsou načteny aktuální xsd soubory, které definují požadovanou strukturu přijímaných xml. Následně je provedena strojová validace těchto souborů. Pokud příloha není validní, je zapsán typ chyby, změněn stav DZ na „Nezpracovatelná“ a vyhodnocení přílohy dále nepokračuje. Následně v xml automat dohledá všechny povinné a pomocí identifikátoru IČO/RČ dohledává jejich záznamy v bankovních systémech.

Pokud není alespoň jeden povinný dohledatelný, je uložena příslušná chyba, změněn stav DZ a zpracování této přílohy dále nepokračuje. Jestliže se povinné dohledat podařilo, jsou k nim hledány příslušné účty. V případě, že alespoň jeden účet nebyl nalezen, uloží automat chybu, změní stav DZ na **Nezpracovatelná** a dále ve zpracování této přílohy nepokračuje. V dalším kroku se ověřuje duplicita čísla jednacího mezi již stávajícími platnými případy. Pokud je nalezena shoda, je zanesena chyba, změněn stav DZ a vyhodnocení přílohy ukončeno. Tímto končí výčet podmínek, které by mohly zapříčinit vyřazení DZ ze zpracování.

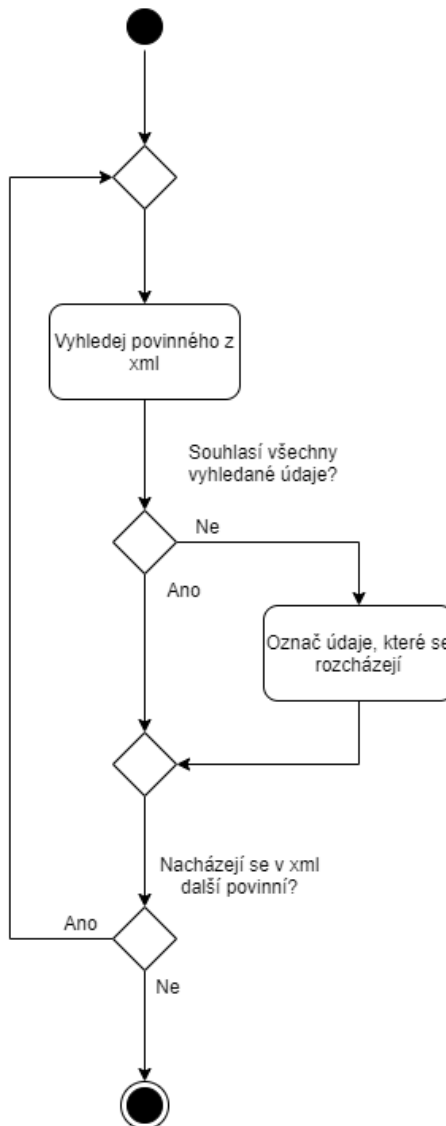
Dále se ověří, že číslo účtu výstavce, tedy účtu, kam je požadováno odeslání jednotlivých finančních položek, je již v aplikaci zanesené. Pokud nikoliv, je příslušný záznam vytvořen. Zbytek *vyhodnocení dat z xml* tvoří validace vyhledaných dat, která je popsána níže. Obrázek 25 s diagramem aktivit na straně 66 zachycuje popisované chování.

## Validace vyhledaných dat

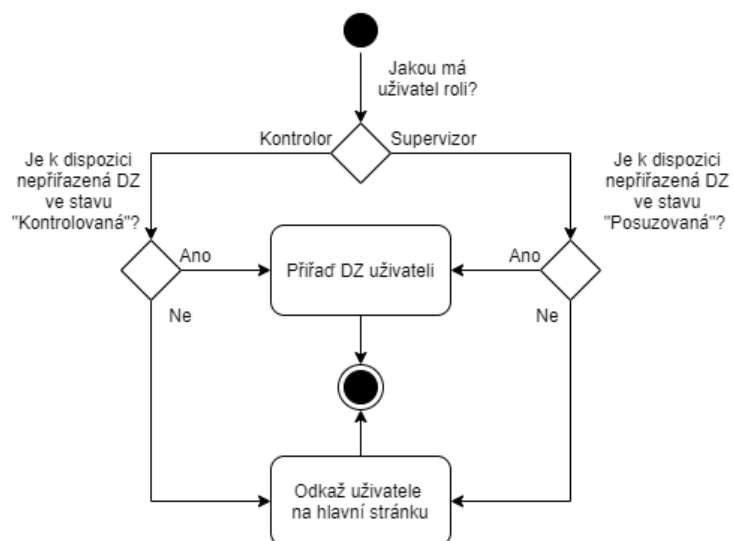
Porovnání příchozích informací v xml a bankou aktuálně vlastněných dat může vykazovat drobné nedostatky. Přijaté xml obsahuje informace o dvou subjektech, povinném a oprávněném. Z nich je pouze povinný porovnáván s bankovními daty. Proto se validace oprávněného netýká. Pokud je podle identifikátoru IČO/RČ možné povinného vyhledat, což je ověřeno v předchozích krocích, je záznam přidán i přes neshodu dat. Protože nesoulad dat může být způsoben velkým množstvím faktorů, je konečné posouzení provedeno *Kontrolorem*. Ten bude následně v rámci kontroly upozorněn, že některé parametry povinného nemusí odpovídat. *Validace vyhledaných dat* zobrazuje Obrázek 26 aktivity diagramem na straně 67.



Obrázek 25: Aktivita diagram: Vyhodnocení dat z xml (Zdroj: Vlastní tvorba)



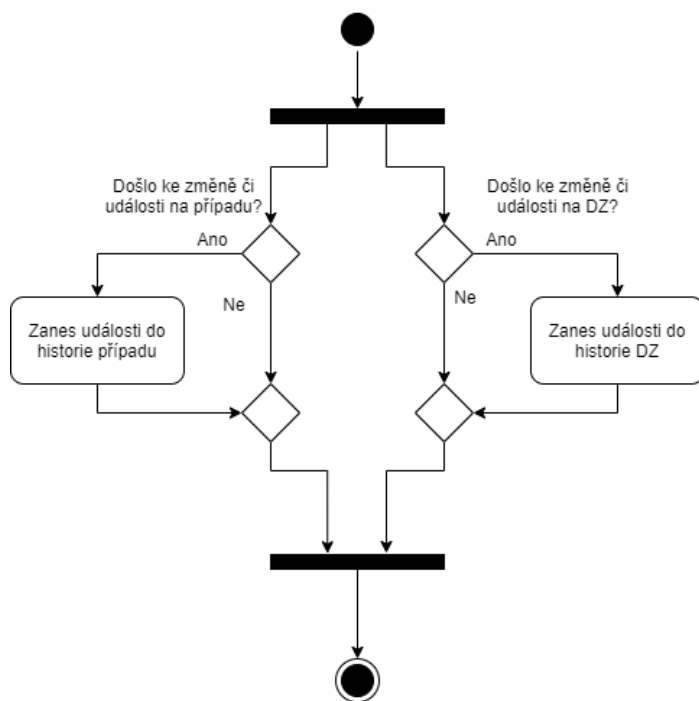
Obrázek 26: Aktivita diagram: Validace vyhledaných dat (Zdroj: Vlastní tvorba)



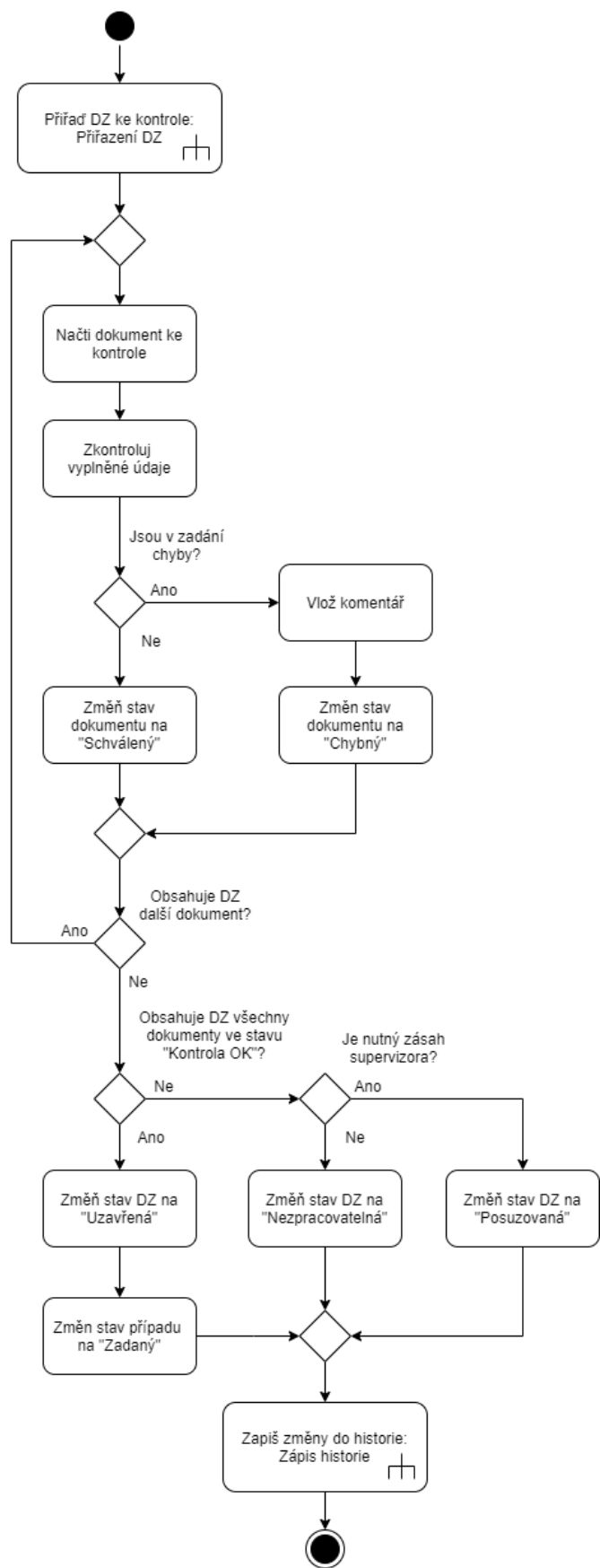
Obrázek 27: Aktivita diagram: Přiřazení DZ (Zdroj: Vlastní tvorba)

## Kontrola DZ

Na počátku kontroly je uživateli přiřazena DZ. Způsob jejího přiřazení zachycuje Obrázek 27 na straně 67. Následně je přiřazená DZ zobrazena a uživatel postupně provádí kontrolu jednotlivých příloh a dokumentů. Pokud je v zadání dokumentu chyba, označí uživatel dokument jako chybný, vloží komentář a pokračuje s kontrolou. Pokud v zadání chyba není, je dokument označen **Schválený**. Následně kontrolor pokračuje v kontrole dalších příloh. Pokud jsou všechny dokumenty ve stavu **Schválený**, je výsledek zpracování správný. Uživatel takovou DZ převede do stavu **Uzavřená** a všechny navázané případy jsou převedeny do stavu **Zadaný**. Pokud zpracování správné není, je uživateli nabídnuta možnost předat DZ supervizorovi a nebo k ruční kontrole. Na základě rozhodnutí *kontrolora* je DZ změněna na stav **Nezpracovatelná** a nebo **Posuzovaná**. Posledním krokem kontroly je zanesení všech provedených změn do historie. Aktivitní diagram zachycující *kontrolu DZ* je Obrázek 29 na straně 69. Historizaci zachycuje diagram na obrázku níže.



Obrázek 28: Aktivitní diagram: Historie změn (Zdroj: Vlastní tvorba)



Obrázek 29: Aktivita diagram: Kontrola DZ (Zdroj: Vlastní tvorba)

## **Supervize**

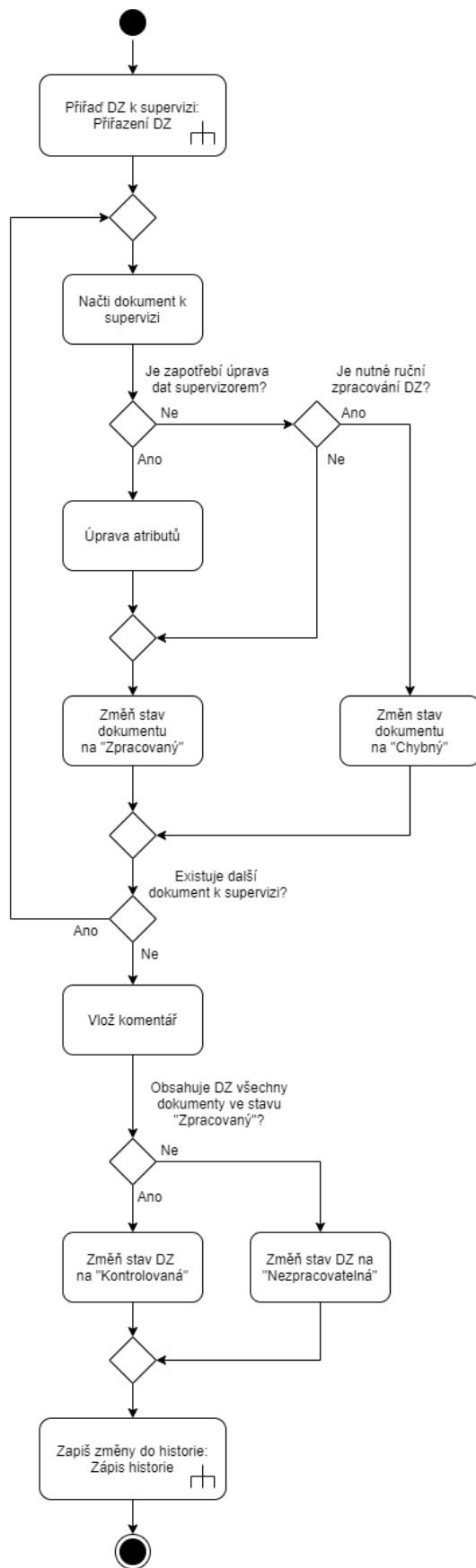
Supervize začíná přiřazením volné DZ ve stavu „Posuzovaná“ uživateli. Následně je DZ zobrazena a probíhá supervize obdobným způsobem jako kontrola datové zprávy. *Supervisor* prochází jednotlivé dokumenty. Pokud nalezne nesoulad v systému vyplněnými daty, může jednotlivá pole editovat. Dále *Supervisor* může jednotlivé dokumenty označit k další kontrole nebo jako chybné. Jakmile je supervize jednotlivých dokumentů dokončena, je na základě jejich stavů rozhodnuto, jestli bude DZ vrácena ke kontrole anebo předána na ruční zpracování. Uživatel je zároveň vyzván k zadání komentáře k provedeným akcím. Poslední aktivitou je zapsání provedených úkonů do historie. Diagram aktivit *Supervize* zachycuje Obrázek 30 na straně 71.

## **Aktualizuj holdy**

Jedná se o aktivitu zobrazenou na diagramu aktivit nejvyšší úrovně na straně 61. Pokud v rámci procesu zpracování EEP dojde k editaci údajů supervizorem, může nastat situace, kdy zadané holdy již nemají validní parametry. Pokud tedy došlo k editaci alespoň jedné z položek s povinnými, účty a položkami platebních předmětů, provede systém kontrolu a případnou úpravu holdů dle aktuálních parametrů.

## **Předej DZ k ručnímu zpracování**

Datové zprávy, které obsahují vady nebo parametry, které nejsou pro automatizované zpracování vhodné, zpracují uživatelé v původní aplikaci. Kromě změny stavu datové zprávy na **Nezpracovatelná**, dojde k nastavení parametru Typ DZ třídy *Datová zpráva* na „manuální“.



Obrázek 30: Aktivita diagram: Supervize (Zdroj: Vlastní tvorba)

## 4.7 Prototyp

Vytvořený návrh IS je doplněn prototypem webové aplikace. Prototyp byl použit na rozdíl od techniky wireframe z toho důvodu, že v rámci uživatelské aktivity dochází k drobným změnám v rámci jedné obrazovky. Toto chování by bylo obtížně zachytitelné nízkým počtem drátěných modelů. Prototyp umožňuje prezentovat uživatelům chování jednotlivých komponent obrazovek. K realizaci byly využity technologie HTML5, CSS, Bootstrap v4 a jQuery. Interpretovány jsou 3 základní obrazovky. Vyhledávání příchozích datových zpráv, provedení kontroly a provedení supervize.

Na uvedených obrazovkách jsou zobrazeny základní průchody, které demonstrují principy fungování navrhovaného řešení. Prototyp zobrazuje pojetí IS pro *kontrolora* a *supervizora* odděleně, protože právě rozdílnost zobrazení a přístupnosti prvků na základě role je klíčovým rozdílem v jejich přístupu k IS.

Prototyp zde slouží pouze jako doplněk návrhu IS. Proto nebude celý podrobně popisován. Ukázka kódu je dostupná v příloze a živá verze prototypu je dostupná na adrese [prototyp.michalgruber.cz](http://prototyp.michalgruber.cz)

Pro *kontrolora* i *supervizora* je po otevření webové stránky IS je jako výchozí obrazovka načten formulář pro vyhledávání příchozích DZ. Při samotném načtení stránky je vyhledáno 100 nejnovějších záznamů bez omezení některým z parametrů. Uživatel má dále možnost vyhledat požadované DZ pomocí čísla jednacního výstavce, datumu doručení, přiřazeného kontrolora a stavu datové zprávy. Každý nalezený záznam nabízí možnost zobrazit detail datové zprávy tlačítkem „Více“. Podobu této části prototypu zachycuje Obrázek 31.



Exekuce Vyhledávání Kontrola 1

Vyhledávání datových zpráv

Číslo jednací:

Datum doručení:

Stav DZ:

Kontrolor:

Vyhledáno 5 záznamů.

| # | Číslo jednací    | Výstavce      | Datum doručení | Stav DZ     | Kontrolor                                       |
|---|------------------|---------------|----------------|-------------|---|
| 1 | 846 EX 1871-5    | Kateřina Nová | 2.3.2021       | Stažená DZ  | <input type="button" value="Více"/>             |
| 2 | 1234 fer 48/1-35 | Josef Roháč   | 20.3.2021      | Supervize   | David Pechl <input type="button" value="Více"/> |
| 3 | 48 ro 11-5       | Karel Krechl  | 17.2.2021      | Ke kontrole | Jiří Kalp <input type="button" value="Více"/>   |
| 4 | 1rt48/187-7      | Josef Roháč   | 2.1.2021       | Uzavřená    | David Pechl <input type="button" value="Více"/> |
| 5 | 987c7-1          | Karel Krechl  | 2.1.2021       | Uzavřená    | Jiří Kalp <input type="button" value="Více"/>   |

Obrázek 31: Prototyp stránky vyhledávání (Zdroj: Vlastní tvorba)

Pro kontrolora se v pravém horním rohu obrazovky se nachází tlačítko s popisem „Kontrola“ a číselným údajem. Číslo uvádí počet DZ, které jsou volné ke kontrole. Toto číslo je aktuální i v případě, že tlačítko je v rámci kontroly neaktivní. Po kliknutí na toto tlačítko kontrolor zahájí scénář kontroly DZ. Aplikace tedy na pozadí přiřadí volnou DZ ke kontrole a načte dialog kontroly. Obrázek 32 zachycuje jeho grafický návrh.

Exekuce Vyhledávání Kontrola 0

Výstavce a datová zpráva

**Kateřina Nová** Stav DZ: **Kontrolovaná**  
 Dlouhá ulice 23/4 Číslo Jednací: 846 EX 1871-5  
 Pardubice Datum doručení: 2.3.2021  
 412 00

Přílohy

navez\_prilohy1.xml   
 Exekuční příkaz 357/14 Chybný

navez\_prilohy2.xml   
 Exekuční příkaz 47/6 Zpracovaný

navez\_prilohy3.xml   
 Exekuční příkaz 47/6 Schvázený

dokument.pdf

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Příklad jednoduchého exekučního příkazu -->
<epelektronickyPrikaz
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ep="https://www.czech-ba.cz/eExekuce/ep"
xsi:schemaLocation="https://www.czech-ba.cz/eExekuce/ep ./xsd/EP.xsd">
<epidentifikace>
<epspisovaZnacka>067 EX 1243/13</epspisovaZnacka>
<epporadoveCislo>1</epporadoveCislo>
</epidentifikace>
<epdatumVydani>2014-04-05</epdatumVydani>
<epobecneUdaje>
<epzakladniMena>CZK</epzakladniMena>
</epobecneUdaje>
<epvystavce>
<epjC>41069684</epjC>
<epnavev>Exekutorský úřad Jeseník - JUDr. Antonín Dohnal, soudní exekutor</epnavev>
<epsidlo>
<epulice>Otakara Březiny</epulice>
<epbudova>
<epcisloOrientacni>229</epcisloOrientacni>
</epbudova>
<epobec>Jeseník</epobec>
<eppsc>79001</eppsc>
<epstat>CZ</epstat>
</epsidlo>
<epkontakt>
<epjmenoPrijmeni>
<epjmeno>Anna</epjmeno>
<eprijmeni>Novotná</eprijmeni>
```

Obrázek 32: Prototyp stránky kontrola DZ (Zdroj: Vlastní tvorba)

Každá příloha je zobrazena zvlášť a po kliknutí na její název se načte její obsah do pravé části aplikace. Pokud byl z přílohy vytvořen dokument a navázaný případ, tak je zobrazený typ případu, jeho bankovní číselné označení a stav zpracování dokumentu. Pokud je

dokument ve stavu **Zpracovaný** je zobrazeno tlačítko „Kontrola“. Jinak jsou tlačítka nepřístupná.

Obrázek 33 ukazuje vzhled systému po zahájení kontroly dokumentu. Pro entity, které jsou dohledávány v interních systémech banky, *kontrolor* vidí prvky, které byly zpracovány z EEP a vedle nich data dohledatelná v KB. Pokud systém považuje data za shodná, označí je zeleným tagem „OK“. V opačném případě je červeně označeno „Nesoulad“. Následuje zbytek zpracovaných informací, historie zpracování a zadané komentáře. Uživatel má na konci možnost dokument schválit či nikoliv. Protože vybraná DZ obsahuje alespoň jeden dokument ve stavu **Chybný**, musí být datová zpráva předána k supervizi anebo na ruční zpracování. Protože již není žádná další DZ ke zpracování, je *kontrolor* po předání převeden na výchozí stránku s vyhledáváním, kde může celý scénář opakovat.

Obdobný scénář je připravený také pro *supervizora*, který pracuje se systémem velmi podobně. Hlavní rozdíl je v možnosti editovat zobrazované hodnoty. Pro entity, které jsou dohledávány v interních systémech banky nemůže *supervizor* provádět přímou editaci, ale chybná data od případu odpojit.

The screenshot shows a web application interface for document control. At the top, there is a header with the text "navez\_prilohy2.xml" and a button labeled "Kontrola". Below the header, there is a section for "Obecné" (General) information, including "Číslo jednací: 067 EX 1243/13-1" and "Název přílohy: navez\_prilohy2.xml". The main part of the interface is a table with columns "Zadané" (Given) and "Dohledané" (Searched). The table is divided into two sections: "Povinný" (Mandatory) and "Účty" (Accounts). The "Povinný" section has two rows. The first row is marked "OK" and shows data for Jan Novák. The second row is marked "Nesoulad" (Discrepancy) and shows data for Petr Nov. The "Účty" section has one row marked "OK" for Jan Novák. To the right of the table is a code window displaying XML data, including tags like "ep:elektronickyPriraz", "ep:identifikace", "ep:datumVydani", "ep:obecneUdaje", "ep:vystavce", "ep:kontakty", and "ep:epikodVystavce".

Obrázek 33: Prototyp stránky kontrola dokumentu (Zdroj: Vlastní tvorba)

## 5 Výsledky a diskuse

Vytvořený návrh svými parametry umožňuje okamžitou implementaci do zavedených bankovních systémů bez nutnosti jejich úprav. Je tedy možné začít postupně automatizovat zpracovatelský proces tak, aby nedošlo k omezení stávajícího provozu. Navrhovaný systém poskytuje alternativní zpracovatelskou cestu za snížení nároků na pracovní obsluhu.

Dle interních bankovních statistik přicházelo v roce 2019 průměrně měsíčně 2 505 datových zpráv s exekčním příkazem. Z toho 1 865 obsahovalo EEP. Pokud by tyto zprávy byly zpracovány navrhovaným řešením, je možné dosáhnout úspory až 404 hodiny měsíčně. Pokud do odhadu započítáme průměrnou 7% chybovost vstupních dat, získáme měsíční úsporu ve výši až 373,8 hodin.

| Možná časová úspora                        |               |
|--|---------------|
| Počet DZ (měsíční průměr)                  | 1865          |
| Průměrná doba zpracování jedné DZ (minuty) | 13            |
| Maximální možný ušetřený čas (hodiny)      | 404,08        |
| Průměrná chybovost vstupních dat (%)       | 7             |
| <b>Ušetřený čas (hodiny)</b>               | <b>375,80</b> |

Tabulka 8: Možná časová úspora po zavedení systému (Zdroj: Vlastní tvorba)

Navržený systém zpracovává příchozí exekční příkazy do fáze, kdy je dokument zadán do systémové struktury, vytvořen exekční případ, zadrženy prostředky na účtech a všechna zpracovaná data potvrzena kontrolorem. Nejedná se tedy o systém, který by pokrýval celý proces exekčního případu, pouze jeho zadání. Proto nejsou v systému řešeny souběhy a dopady dalších událostí, které vyžadují adekvátní odezvu, jako jsou například insolvenční řízení, trestní zajištění anebo souběh více exekucí. Tyto události jsou řešeny v dalších fázích exekčního případu původním systémem.

Budoucí rozšíření systému by se mělo zaměřit na zpracování dalších typů dokumentů, které mohou být zasílány ve formátu xml. Prvním typem dokumentu by byla jistě právní moc, která je součástí téměř každého exekčního řízení. Její definice a struktura umožňuje připojení k jednomu či více případům a kompletně automatizované zadání od zpracování až po kontrolu.

## 6 Závěr

Cílem této diplomové práce bylo za využití modelovacího jazyka UML navrhnout informační systém, který je efektivním doplňkem stávajícího bankovního systému pro zpracování dokumentů souvisejících s exekčním řízením. Navrhovaný informační systém je zaměřen na automatizované zpracování elektronických exekčních příkazů, jejich následnou kontrolu a případnou supervizi.

Stanovený rozsah nepokrývá úkony, které nastávají po zpracování exekčního příkazu. Ty jsou prováděny stávajícím systémem. Navrhovaný systém též nepředpokládá využití jiných nežli stávajících uživatelů. Proto jsou jednotlivé uživateli řízené procesy strukturovány podobně, jako procesy stávající.

Na základě analýzy funkčních a nefunkčních požadavků s přihlédnutím k praktickým zkušenostem autora byla vytvořena sada 3 modelů pro zachycení uceleného pohledu na navrhovaný IS. Modely jsou tvořeny dynamickými i statickými UML diagramy, které souhrnně definují požadované funkce a vlastnosti.

Pro lepší pochopení navrhovaného systému vznikl webový prototyp založený na technologiích HTML5, CSS, Bootstrap a jQuery. Prototyp prezentuje vyhledávání datové zprávy, kontrolu datové zprávy a supervizi.

Výsledkem je návrh IS, který má předpoklady k výraznému snížení nákladů nutných ke zpracování příchozích exekčních příkazů, zjednodušení administrativních procesů a zkrácení doby nutné k zajištění prostředků na účtech povinných.

## 7 Seznam použitých zdrojů

- [1] *Exekutorská komora - ČR Celkový počet pravomocných exekucí* [online]. In: . s. 1 [cit. 2021-03-26]. Dostupné z: <https://statistiky.ekcr.info/docs/statisticky-list-I.pdf>
- [2] *Exekutorská komora - ČR Počty zahájených exekucí:* [online]. , 2 [cit. 2021-03-26]. Dostupné z: <https://statistiky.ekcr.info/docs/statisticky-list-IX.pdf>
- [3] PROCHÁZKA, JAROSLAV a CYRIL KLIMEŠ. *SOFTWAREVÉ INŽENÝRSTVÍ*. 2., aktualiz. a dopl. vyd. Ostrava: Ostravská univerzita v Ostravě, 2009.
- [4] SOMMERVILLE, Ian. *Softwarové inženýrství*. 1. vyd. Přeložil Jakub GONER. Brno: Computer Press, 2013. ISBN 978-80-251-3826-7.
- [5] Informační systém. *Management mania* [online]. 2020 [cit. 2021-02-22]. Dostupné z: <https://managementmania.com/cs/informacni-system>
- [6] BURIAN, Pavel. *Webové a agentové technologie*. Vyd. 1. Praha: Grada, 2012. Průvodce (Grada). ISBN 978-80-247-4376-9.
- [7] VYMĚTAL, Dominik. *Informační systémy v podnicích: teorie a praxe projektování*. 1. vyd. Praha: Grada, 2009. Průvodce (Grada). ISBN 9788024730462.
- [8] SKLENÁK, Vilém. *Data, informace, znalosti a Internet*. Praha: C.H. Beck, 2001. C.H. Beck pro praxi. ISBN 80-717-9409-0.
- [9] KEMPE, Shannon. The Data – Information – Knowledge Cycle. *Dataversity* [online]. 2013 [cit. 2021-02-23]. Dostupné z: <https://www.dataversity.net/the-data-information-knowledge-cycle/#>
- [10] ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
- [11] AMBLER, Scott. *The elements of UML 2.0 style*. Cambridge: Cambridge University Press, 2005. ISBN 05-216-1678-6.
- [12] ŘEPA, Václav. *Procesně řízená organizace*. 1. vyd. Praha: Grada, 2012. Management v informační společnosti. ISBN 9788024741284.
- [13] The exponential cost of fixing bugs. *Deep Source* [online]. [cit. 2021-02-06]. Dostupné z: <https://deepsources.io/blog/exponential-cost-of-fixing-bugs/>
- [14] Postupy pro sběr požadavků. *Zvesela* [online]. 2010 [cit. 2021-03-01]. Dostupné z: [http://zcu.arcao.com/\\_statnice\\_ing\\_fav/nis/wiki\\_zvesela/09.htm](http://zcu.arcao.com/_statnice_ing_fav/nis/wiki_zvesela/09.htm)
- [15] BENEŠ, Miroslav. *Principy objektově orientovaného programování* [online]. Katedra informatiky FEI VŠB-TU Ostrava [cit. 2021-02-28]. Dostupné z: <http://www.cs.vsb.cz/benes/vyuka/upr/texty/objekty/index.html>
- [16] Management informačního systému. *FI MUNI* [online]. [cit. 2021-02-11]. Dostupné z: <http://www.fi.muni.cz/~smid/managis.html>
- [17] TILLEY, Scott a Harry J. ROSENBLATT. *Systems Analysis and Design*. Boston, MA, United States: Cengage Learning, Inc, 2016. ISBN 9781305494602.
- [18] BASL, Josef a Roman BLAŽÍČEK. *Podnikové informační systémy: podnik v informační společnosti*. 2., výrazně přeprac. a rozš. vyd. Praha: Grada, 2008. Management v informační společnosti. ISBN 978-80-247-2279-5.

- [19] BRUCKNER, Tomáš. *Tvorba informačních systémů: principy, metodiky, architektury*. 1. vyd. Praha: Grada, 2012. Management v informační společnosti. ISBN 9788024741536.
- [20] BUREŠ, Miroslav, Miroslav RENDA, Michal DOLEŽEL, Peter SVOBODA, Zdeněk GRÖSSL, Martin KOMÁREK, Ondřej MACEK a Radoslav MLYNÁŘ. *Efektivní testování softwaru: klíčové otázky pro efektivitu testovacího procesu*. Praha: Grada, 2016. Profesionál. ISBN 978-80-247-5594-6.
- [21] Differences between Black Box Testing vs White Box Testing. *Geeks for Geeks* [online]. 2020 [cit. 2021-03-15]. Dostupné z: <https://www.geeksforgeeks.org/differences-between-black-box-testing-vs-white-box-testing/>
- [22] Automation Testing Vs. Manual Testing: What's the Difference?. *Guru 99* [online]. 2021 [cit. 2021-03-15]. Dostupné z: <https://www.guru99.com/difference-automated-vs-manual-testing.html>
- [23] SOAP vs REST. What's the Difference?. *Smartbear* [online]. 2021 [cit. 2021-03-17]. Dostupné z: <https://smartbear.com/blog/soap-vs-rest-whats-the-difference/>
- [24] VRANA, Ivan a Karel RICHTA. *Zásady a postupy zavádění podnikových informačních systémů: praktická příručka pro podnikové manažery*. 1. vyd. Praha: Grada, 2005. Management v informační společnosti. ISBN 8024711036.
- [25] KAJZAR, RNDr. *Projektování informačních systémů II: Výukové materiály předmětu, který přednáší RNDr. Dušan Kajzar* [online]. [cit. 2021-03-14]. Dostupné z: <http://zdenek2.euweb.cz/prois2.html>
- [26] BOEHM, Barry. *Software engineering: Barry W. Boehm's lifetime contributions to software*. Wiley-IEEE Computer Society Press., 2007. ISBN 978-0-470-14873-0.
- [27] *The UML and the Rational Unified Process*.
- [28] FOWLER, Martin. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. 3rd edition. Addison-Wesley Professional, 2003. ISBN 9780321193681.
- [29] LANO, Kevin. *UML 2 Semantics and Applications*. 2009. ISBN 9780470409084.
- [30] Use case diagram - diagram případů užití. *Mpavus* [online]. [cit. 2021-03-20]. Dostupné z: <http://mpavus.wz.cz/uml/uml-b-use-case-3-2-1.php>
- [31] *Zákon o bankách | Zákon č. 21/1992 Sb.* In: . Praha, 2021, ročník 2021, číslo 21.
- [32] *Vyhláška Ministerstva spravedlnosti ze dne 19. listopadu 2001 o postupech při výkonu exekuční a další činnosti 418/2001 Sb. znění účinné od 1. 1. 2015.* In: . Praha, 2015, ročník 2015, číslo 418.
- [33] O projektu. *Centrální evidence exekucí* [online]. [cit. 2021-03-01]. Dostupné z: [https://www.ceecr.cz/?utm\\_source=ekcr&utm\\_medium=banner&utm\\_campaign=banner%20ekcr](https://www.ceecr.cz/?utm_source=ekcr&utm_medium=banner&utm_campaign=banner%20ekcr)
- [34] *Exekuční řád | Zákon č. 120/2001 Sb. o soudních exekutorech a exekuční činnosti: Předpis č. 120/2001 Sb.* In: . Znění od 1. 6. 2019.
- [35] TESAŘ, Ing. *Provozní řád ISDS* [online]. 2019, , 25 [cit. 2021-03-31]. Dostupné z: [https://www.datoveschranky.info/documents/1744842/1746058/provozni\\_rad\\_isds.pdf/a49d0691-d02d-44fd-8068-c158599de574](https://www.datoveschranky.info/documents/1744842/1746058/provozni_rad_isds.pdf/a49d0691-d02d-44fd-8068-c158599de574)
- [36] Co je to exekuce?. *EXEKUTORSKÁ KOMORA ČESKÉ REPUBLIKY* [online]. [cit. 2021-02-04]. Dostupné z: <https://www.ekcr.cz/1/exekutori-radi/794-6-co-je-to-exekuce?w=>

## 8 Přílohy

### 8.1 Ukázka zdrojového kódu obrazovky vyhledávání DZ

```
<!doctype html>
<html>
  <head>
    <link rel="stylesheet" href="style.css">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css">
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <div class="navMenu">
        <a class="navbar-brand" href="#">Exekuce</a>
      </div>
      <ul class="navbar-nav mr-auto">
        <li class="nav-item active">
          <a class="nav-link" href="#">Vyhledávání <span class="sr-
only">(current)</span></a>
        </li>
      </ul>
      <div class="navButton">
        <a href="kontrola.html" class="btn btn-info" role="button">Kontrola
        <span class="badge badge-pill badge-warning">1</span></a>
      </div>
    </nav>
    <div class="vyhledavani">
      <div class="card w-100">
        <div class="card-body">
          <h5 class="card-title">Vyhledávání datových zpráv</h5>
          <div class="container">
            <div class="row">
              <div class="col">
                Číslo jednací: <input class="form-control" type="text">
                Datum doručení: <input class="form-control" type="date">
              </div>
              <div class="col">
                Stav DZ:
                <select class="form-control">
                  <option>---vše---</option>
                  <option>Stažená DZ</option>
                  <option>Ke zpracování</option>
                  <option>Ke kontrole</option>
                </select>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```

        <option>Supervize</option>
        <option>Uzavřená</option>
    </select>
    Kontrolor:
    <select class="form-control">
        <option>---vše---</option>
        <option>Jiří Kalp</option>
        <option>David Pechl</option>
    </select>
</div>
</div>
<div class="vyhledatBtn">
    <a href="#" class="btn btn-primary">Vyhledat</a>
</div>
</div>
</div>
</div>
<div class="warning">
    <div class="alert alert-warning" role="alert">
        Vyhledáno 5 záznamů.
    </div>
</div>
<div class="tableBrowse">
    <table class="table">
        <thead class="thead-light">
            <tr>
                <th scope="col">#</th>
                <th scope="col">Číslo jednací</th>
                <th scope="col">Výstavce</th>
                <th scope="col">Datum doručení</th>
                <th scope="col">Stav DZ</th>
                <th scope="col">Kontrolor</th>
                <th scope="col"></th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <th scope="row">1</th>
                <td>846 EX 1871-5</td>
                <td>Kateřina Nová</td>
                <td>2.3.2021</td>
                <td>Stažená DZ</td>
                <td></td>
                <td><button type="button" class="btn btn-outline-
secondary">Více</button></td>
            </tr>
            <tr>
                <th scope="row">2</th>

```



```

        <td>1234 fer 48/1-35</td>
        <td>Josef Roháč</td>
        <td>20.3.2021</td>
        <td>Supervize</td>
        <td>David Pechl</td>
        <td><button type="button" class="btn btn-outline-
secondary">Více</button></td>
    </tr>
    <tr>
        <th scope="row">3</th>
        <td>48 ro 11-5</td>
        <td>Karel Krechl</td>
        <td>17.2.2021</td>
        <td>Ke kontrole</td>
        <td>Jiří Kalp</td>
        <td><button type="button" class="btn btn-outline-
secondary">Více</button></td>
    </tr>
    <tr>
        <th scope="row">4</th>
        <td>1rt48/187-7</td>
        <td>Josef Roháč</td>
        <td>2.1.2021</td>
        <td>Uzavřená</td>
        <td>David Pechl</td>
        <td><button type="button" class="btn btn-outline-
secondary">Více</button></td>
    </tr>
    <tr>
        <th scope="row">5</th>
        <td>987c7-1</td>
        <td>Karel Krechl</td>
        <td>2.1.2021</td>
        <td>Uzavřená</td>
        <td>Jiří Kalp</td>
        <td><button type="button" class="btn btn-outline-
secondary">Více</button></td>
    </tr>
</tbody>
</table>
</div>
</body>
</html>

```