

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačního inženýrství**



## **Diplomová práce**

**Návrh informačního systému pro správu bytového domu  
s využitím UML**

**Pavel Kovář**

© 2024 ČZU v Praze



## ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Pavel Kovář

Systémové inženýrství a informatika  
Informatika

Název práce

**Návrh informačního systému pro správu bytového domu s využitím UML**

Název anglicky

**Design of Information System in UML**

---

### Cíle práce

Cílem práce je návrh informačního systému pomocí jazyka UML. Informační systém je navržen pro správu bytového domu. Poskytuje uživatelům přehled o stavu domu, aktuálně řešených záležitostech a veškeré informace, které se týkají správy domu. Poskytuje uživatelům nástroje pro efektivní komunikaci a podporu komunitního bydlení. Přínosem je snadný přístup k informacím, efektivní komunikace, efektivní řešení problémů a podpora komunitního soužití v rámci bytového domu.

### Metodika

Teoretická část práce je založena na studiu odborných a vědeckých dokumentů (zejména monografií), které se zabývají vývojem informačních systémů a jazykem UML. Součástí diplomové práce je sběr požadavků a jejich rozbor. Na základě rozboru získaných požadavků je provedena identifikace klíčových procesů, návrh scénářů a syntéza funkčních požadavků. Samotný návrh informačního systému je pak proveden s využitím UML.

## Doporučený rozsah práce

50 – 70

## Klíčová slova

informační systém, UML, objektově orientovaný přístup, funkční analýza, bytový dům, svj

---

## Doporučené zdroje informací

BASL, Josef a BLAŽÍČEK, Roman. Podnikové informační systémy. 3. aktualizované a doplněné vydání. Praha : Grada Publishing, a. s., 2012. ISBN 978-80-247-4307-3

BLAHA, Michael; RUMBAUGH, James. *Object-oriented modeling and design with UML*. Upper Saddle River, NJ: Pearson Education, 2005. ISBN 0130159204.

BRUCKNER, Tomáš, a kol. Tvorba informačních systémů. Praha : Grada Publishing, a. s., 2012. ISBN 978-80-247-7903-4

FOWLER, M. Destilované UML. Praha: Grada Publishing, a. s., 2009. ISBN 978-80-271-2846-4

ŘEPA, Václav. Podnikové procesy. 2. aktualizované vydání. Praha : Grada Publishing, a. s., 2007. ISBN 978-80-247-6722-2

---

## Předběžný termín obhajoby

2022/23 ZS – PEF

## Vedoucí práce

Ing. Jakub Konopásek, Ph.D.

## Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 31. 10. 2022

**Ing. Martin Pelikán, Ph.D.**

Vedoucí katedry

Elektronicky schváleno dne 28. 11. 2022

**doc. Ing. Tomáš Šubrt, Ph.D.**

Děkan

V Praze dne 17. 09. 2023

### **Čestné prohlášení**

Prohlašuji, že svou diplomovou práci "Návrh informačního systému pro správu bytového domu s využitím 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. března 2024

---

## **Poděkování**

Rád bych touto cestou poděkoval Ing. Jakubovi Konopáskovi, Ph.D. za úsilí a čas, které věnoval odbornému vedení mé závěrečné práce.

# Návrh informačního systému pro správu bytového domu s využitím UML

## Abstrakt

Diplomová práce se zabývá návrhem specifického informačního systému pomocí modelovacího jazyka UML. Jde o systém pro správu bytového domu v majetku společenství vlastníků jednotek. Systém je určen zejména pro společenství spravovaná vlastními členy. Přínosy navrženého systému jsou snadný přístup k informacím, efektivní komunikace členů společenství, efektivní řešení každodenních záležitostí i různých problémů a podpora komunitního soužití v rámci bytového domu.

Teoretická východiska jsou řešerši zejména odborných literárních zdrojů. Tato kapitola je obsahově rozdělena do dvou částí. První část se zabývá návrhem informačních systémů. Obsahuje řešerši o návrhu informačních systémů a modelovacím jazyku UML. Druhá část se věnuje problematice společenství vlastníků jednotek.

Vlastní práce se zabývá analýzou systému pomocí objektově orientovaného přístupu k modelování systému s využitím modelovacího jazyka UML. Analýza nahlíží na systém z několika různých pohledů – z pohledu statické struktury objektů v systému a souvislostí mezi nimi, z pohledu chování systému v průběhu času a z pohledu vzájemné spolupráce objektů systému. Výsledkem je podklad pro mnohem podrobnější návrh pro realizaci s možnostmi dalšího rozvoje.

**Klíčová slova:** informační systém, UML, objektově orientovaný přístup, funkční analýza, bytový dům, svj

# Design of Information System in UML

## Abstract

The thesis addresses the design of a specific information system using the UML. It concerns a system for managing an apartment buildings owned by a homeowner associations. The system is particularly intended for communities managed by their own members. The system's benefits are easy access to information, easy communication among community members, effective handling of everyday issues and tasks, and supporting the whole community.

The theoretical background is research of a specialized literature. The chapter is divided into two sections content-wise. The first part is about with the design of information systems, including a review of information system design and the UML. The second part if focused on the issues of homeowners' associations.

The actual work involves analysis of the system through object-oriented approach to modeling of the system using the UML. The analysis approaches the system from several different perspectives – in terms of the static structure of objects in the system and their relations, the dynamic behavior inside the system, and interactions of objects. The result is a foundation for a much more detailed design for implementation with wide possibilities for further development.

**Keywords:** information system, UML, object-oriented analysis, requirement analysis, apartment building, homeowner association



# Obsah

<b>1 Úvod.....</b>	<b>11</b>
<b>2 Cíl práce a metodika.....</b>	<b>12</b>
2.1 Cíl práce .....	12
2.2 Metodika .....	12
<b>3 Teoretická východiska .....</b>	<b>13</b>
3.1 Tvorba informačních systémů.....	13
3.1.1 Vymezení systému a jeho účelu.....	13
3.1.2 Životní cyklus informačního systému .....	13
3.1.2.1 Vodopádový vývoj .....	14
3.1.2.2 Iterativní vývoj .....	15
3.1.2.3 Inkrementální vývoj .....	16
3.1.2.4 Agilní metodiky vývoje.....	17
3.1.3 Přístupy k modelování informačních systémů.....	18
3.1.3.1 Procesní modelování .....	19
3.1.3.2 Datové modelování.....	20
3.1.3.3 Objektově orientovaný přístup k modelování systémů .....	21
3.1.4 Modelovací jazyk Unified Modeling Language .....	21
3.1.4.1 Diagramy UML k modelování struktury .....	23
3.1.4.2 Diagramy UML k modelování chování.....	24
3.2 Problematika společenství vlastníků jednotek .....	28
<b>4 Vlastní práce.....</b>	<b>29</b>
4.1 Sběr a vyhodnocení požadavků.....	29
4.2 Koncepce systému.....	31
4.3 Objektový model .....	31
4.3.1 Diagram tříd.....	32
4.3.2 Datový slovník.....	33
4.4 Stavový model.....	37
4.5 Model interakcí .....	43
4.5.1 Případy užití.....	44
4.5.1.1 Administrace .....	45
4.5.1.2 Vytvoření a správa uživatelského účtu.....	45
4.5.1.3 Požadavky k řešení .....	47
4.5.1.4 Zasedání shromáždění .....	49
4.5.1.5 Komunikace mezi členy SVJ.....	57
4.5.1.6 Bilance.....	58
4.5.1.7 Ukládání a čtení dokumentů.....	59

4.5.1.8	Rezervace společných částí .....	61
4.5.2	Sekvenční diagramy .....	63
4.5.3	Diagramy aktivit .....	73
<b>5</b>	<b>Výsledky a diskuse .....</b>	<b>75</b>
<b>6</b>	<b>Závěr.....</b>	<b>77</b>
<b>7</b>	<b>Seznam použitých zdrojů.....</b>	<b>79</b>
<b>8</b>	<b>Seznam obrázků .....</b>	<b>81</b>
<b>Přílohy</b>	<b>.....</b>	<b>82</b>

# 1 Úvod

V ČR existuje přibližně 65 tisíc společenství vlastníků jednotek, právnických osob vznikajících podle platné legislativy za účelem vykonávání správy domů rozdělených na jednotky. Jak už z názvu právní formy vyplývá, jedná se o lidská společenství. Jsou to skupiny lidí, které sdílí minimálně společné místo žití a majetek. Jsou to skupiny lidí, které si potřebují vzájemně sdílet informace během každodenního života, spolupracovat při řešení běžných záležitostí i nejrůznějších problémů, a komunikovat také na venek. Svým charakterem jsou to dlouhodobá společenství s nízkou fluktuací.

Velký podíl těchto společenství je spravován vlastními členy bez cizí pomoci a jen malý podíl společenství svěřuje svoji správu specializovaným firmám. Profesionální správcovské firmy obvykle ke správě bytových domů používají software, který jim správu portfolia bytových domů umožňuje. Profesionální správci také většinou nabízí klientské zóny, což bývá prostředí, kde členové společenství obvykle nachází některé informace o společenství, jeho majetku a správě. Klientské zóny většinou poskytují jen omezené možnosti komunikace se správcem a žádné možnosti komunikace s ostatními členy společenství. Společenství spravovaná vlastními členy nedisponují ani takovými nástroji.

System, jehož návrhu se tato práce věnuje, by měl zefektivnit a usnadnit vzájemnou komunikaci členů společenství vlastníků jednotek a jeho orgánů, zvýšit transparentnost správy společenství a podpořit komunitní bydlení. Neměl by to být například systém pro vedení účetnictví, ale nástroj pro lidi, kteří spolu žijí na společném místě a tvoří tak společenství nejen z pohledu legislativy. System by měl nabídnout i mechanismy k obsluze významných procesů opět s důrazem na vzájemnou komunikaci všech osob ve společenství. Informace související se společenstvím by pro jeho členy měly být díky systému dostupné snadno na jednom místě.

Vlastní práce se zabývá analýzou systému pomocí objektového modelování systému s využitím modelovacího jazyka UML. Analýza nahlíží na systém z několika různých pohledů – z pohledu statické struktury objektů v systému a souvislostí mezi nimi, z pohledu chování systému v průběhu času a z pohledu vzájemné spolupráce objektů systému. Výsledkem je podklad pro mnohem podrobnější návrh pro realizaci se širokými možnostmi dalšího rozvoje.

## **2 Cíl práce a metodika**

### **2.1 Cíl práce**

Cílem práce je návrh informačního systému pro správu bytového domu pomocí jazyka UML. Obecně popsáný cíl diplomové práce ve schváleném zadání byl na základě součinnosti s vedoucím práce blíže specifikován na vytvoření podkladu pro budoucí mnohem podrobnější implementační návrh systému, který je určen zejména pro společenství vlastníků jednotek spravovaná vlastními členy.

### **2.2 Metodika**

Teoretická východiska jsou řešerší odborných literárních zdrojů. Tato kapitola byla obsahově rozdělena do dvou částí. Do první části byla shrnuta základní teorie o návrhu informačních systémů a modelovacím jazyku UML. Druhá část je věnována problematice společenství vlastníků jednotek.

Sběr požadavků byl proveden během nestrukturovaných rozhovorů s osobami z řad potenciálních uživatelů, tedy členů společenství vlastníků jednotek a odborníků z oboru správy společenství vlastníků jednotek. Na základě těchto požadavků byla formulována koncepce systému shrnující významné procesy a oblasti, které by měl systém pokrýt. Následná analýza provedená s využitím modelovacího jazyka UML nahlíží na systém z několika různých pohledů – z pohledu statické struktury objektů v systému a souvislosti mezi nimi, z pohledu chování systému v průběhu času a z pohledu vzájemné spolupráce objektů systému. Analýza byla koncipována jako podklad pro mnohem podrobnější návrh pro realizaci se širokými možnostmi dalšího rozvoje. K vytvoření diagramů byl použit CASE nástroj StarUML.

## **3 Teoretická východiska**

### **3.1 Tvorba informačních systémů**

#### **3.1.1 Vymezení systému a jeho účelu**

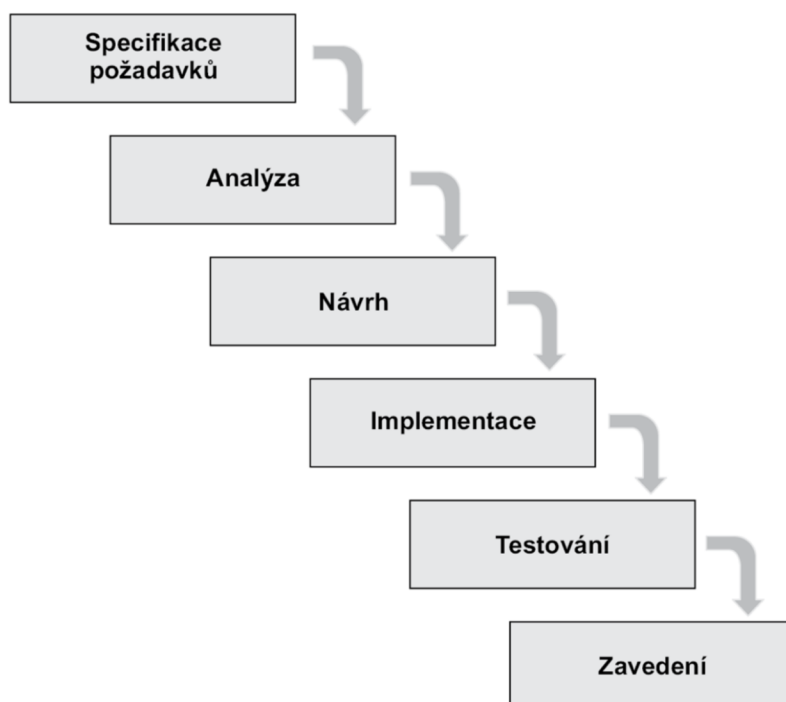
Pro pojem systém existuje nespočetné množství definic. Jejich znění se odvíjí od vědecké disciplíny, ve které se objevují, a autorech, kteří je formulují. Jde o nejednotnost, která vyplývá z interdisciplinarity systémové vědy. Při definici pojmu „systém“ jde tedy především o úhel pohledu. Ten je vždy potřeba přizpůsobit řešené problematice. Bruckner ve své práci (1) vychází z definice původní mezinárodní normy ISO/IEC 15288, která obecně definovala systém jako celek vytvořený a používaný lidmi, poskytující nějaký produkt nebo službu v určeném prostředí za účelem uspokojení potřeb uživatelů nebo jiných zainteresovaných stran. Následně upřesňuje systém jako celek s jasně stanoveným účelem, souhrnem komponent a jejich vzájemných interakcí. Účel informačního systému pak konečně specifikuje (1) jako zajišťování správných informací na správném místě a ve správný čas. Přestože se tato diplomová práce zaměřuje na návrh informačního systému pro společenství vlastníků jednotek, která jsou neziskovou organizací podle § 1194 odst. 1 občanského zákoníku (2), je možné říci, že zmíněné definice systému a účel informačního systému jsou pro její účely vyhovující a dá se na ně při řešení této problematiky nahlížet stejným úhlem pohledu.

#### **3.1.2 Životní cyklus informačního systému**

Životní cyklus informačního systému je obecně rozšířený a velmi často používaný pojem vyskytující se v ICT projektech. Můžeme jej chápat jako časový úsek ohraničený úmyslem nějaký systém vytvořit na jedné straně a ukončením jeho používání na straně druhé. Model životního cyklu je souhrn procesů a aktivit obvykle uspořádaných do jednotlivých etap. (1)

V průběhu času byla zformulována řada modelů životních cyklů, které jsou více či méně frekventovaně využívány. Podle Buchalcevové (1) jsou nejčastěji uplatňovány modely životního cyklu vodopádový, iterativní a inkrementální.

### 3.1.2.1 Vodopádový vývoj



Obrázek 1: schéma vodopádového životního cyklu; zdroj: (1)

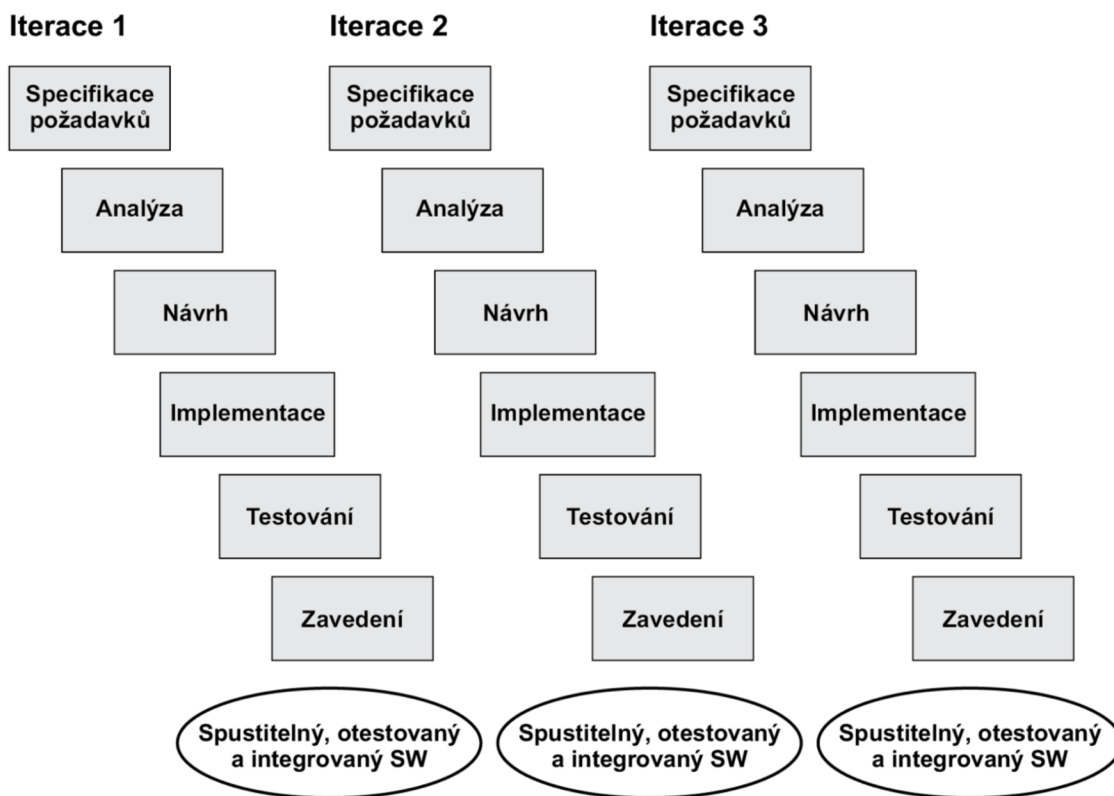
Buchalcevodá uvádí (1), že k rozšíření vodopádového modelu vývoje softwaru došlo zejména v 70. a 80. letech minulého století a dodnes je frekventovaně používán. Jeho princip spočívá v rozdělení vývoje softwaru do jednotlivých etap prováděných postupně za sebou. V pořadí etap se na prvním místě nachází specifikace požadavků, následuje analýza, pokračuje návrhem, který je pak implementován, testován a nakonec zaváděn.

Fowler správně zmiňuje (3), že přestože bývají jednotlivé fáze formálně odděleny, ve skutečnosti nezřídka dochází ke krokům zpět do předchozích fází. Například i během implementace může být nutné vrátit se k činnosti, která se týká návrhu. Podle Fowlera jde o nevyhnutelnou situaci, ve které dochází k revizi dříve uskutečněných kroků například v analýze, nebo návrhu. Upozorňuje přitom, že by během projektu měla být vyvíjena snaha o minimalizaci takových situací, protože časté návraty do předcházejících etap mohou indikovat, že nebyly provedeny úplně správně. (3)

Pokud jde o výhody a nevýhody, autoři se shodují (1) (3), že použití vodopádového životního cyklu je vhodné, pokud se požadavky na výsledné řešení během vývoje příliš nemění a je možné je dobře definovat během úvodní specifikace. Zároveň je obecně výhodné předem znát rozsah řešení. Na druhou stranu je potřeba zmínit i nevýhody, mezi které jistě

patří zapojení zákazníka hlavně v počáteční fázi a pak až na konci během integrace, což může vést ke změnovým požadavkům a zvyšováním nákladů na projekt. (1)

### 3.1.2.2 Iterativní vývoj



Obrázek 2: schéma iterativního životního cyklu; zdroj: (1)

Zmíněné nevýhody vodopádového způsobu vývoje softwaru do určité míry řeší iterativní vývoj. Jeho princip spočívá v rozložení celého projektu na menší celky a vývoj tak probíhá po menších částech v iteracích. (1)

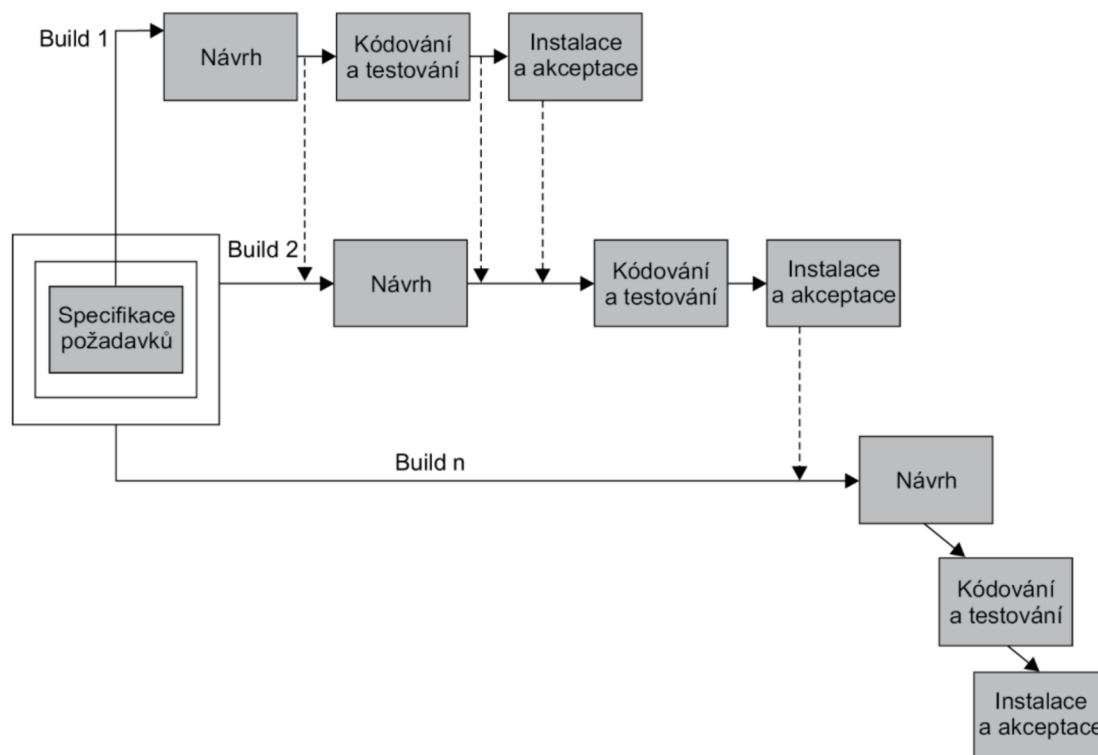
Fowler v souvislosti s iterativním způsobem vývoje podotýká, že mezi dalšími, často zmiňovanými modely, jako je spirálový, evoluční, nebo jacuzzi, není podle něj podstatný rozdíl. Upozorňuje na to, že na rozdílnostech mezi nimi není všeobecná shoda. (3)

Podle literatury není možné zcela jednoznačně určit, který model byl první, nebo který vychází ze kterého. Podle Buchalceové (1) se v souvislosti s modelem vývoje vyskytl pojem iterace poprvé, když Barry Boehm definoval svůj spirálový model ve své práci “A spiral model of software development and enhancement” napsané v r. 1988. Na jeho práci pak reagovali například A. Nilsson a T. Wilson, kteří ve svém článku (4) upozorňují na problémy se spirálovým modelem a upozorňují, že implementace toho přístupu je vágní.

Pokud jde o výhody iterativního modelu vývoje, stojí za zmínku, že každá iterace v sobě zahrnuje podobné fáze jako vodopádový model, jen v menším měřítku, díky čemuž častěji dochází k dodávkám funkčních částí, které jsou následně testovány a integrovány do systému zákazníka. (1)

Fowler doplňuje (3), že komunita objektivě orientované metodologie dlouhodobě preferuje iterativní styl vývoje z toho důvodu, že při vývoji softwaru v iteracích existuje vyšší šance na odhalení problematických situací, které by mohly vést k tomu, že se projekt nebude ubírat správným směrem. Nevýhodou vývoje v iteracích může být riziko nutnosti přepisovat existujících subsystémů. (3)

### 3.1.2.3 Inkrementální vývoj



Obrázek 3: schéma inkrementálního životního cyklu; zdroj: (1)

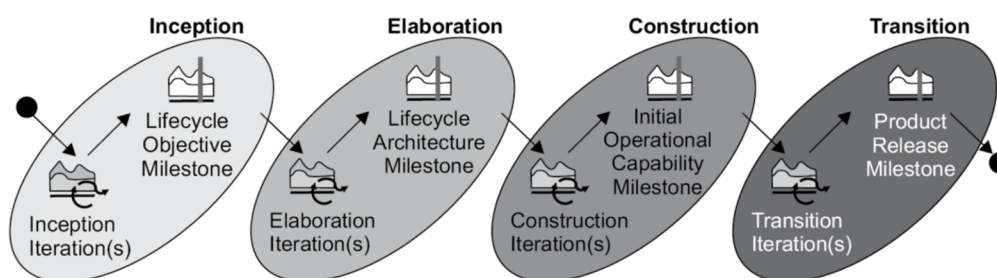
Inkrementální model se od iterativního odlišuje tím, že požadavky jsou formulovány na začátku projektu pro všechny vývojové dílčí části – přírůstky, i když jen na hrubé úrovni. Následně je projekt rozdělen na samostatně proveditelné části, které pak mohou být realizovány samostatně. Přírůstky je možné během vývoje synchronizovat do novějších verzí. Z uvedeného vyplývá, že při použití tohoto způsobu vývoje je potřeba myslet na



návaznosti jednotlivých přírůstků i na jejich velikost. Výhodou jsou opět postupné dodávky a tím i dřívější zpětná vazba od zákazníka. (1)

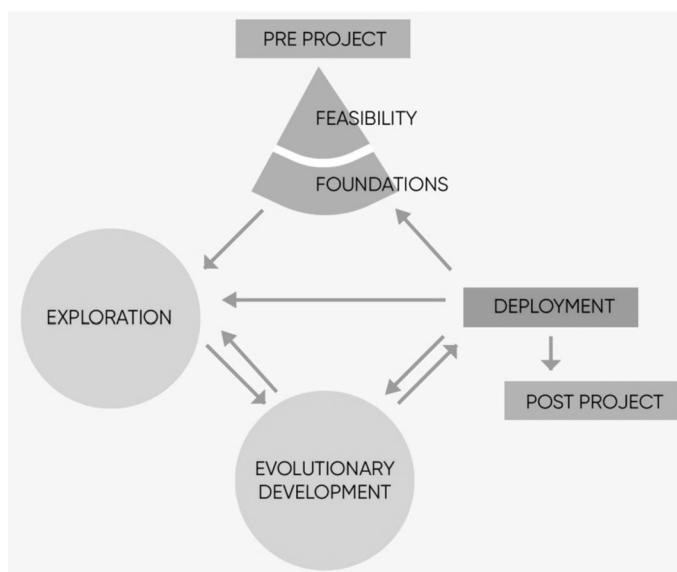
### 3.1.2.4 Agilní metodiky vývoje

Agilní metodiky vývoje jsou adaptivní techniky, které se silně spoléhají na kvalitu členů vývojového týmu, jejich komunikaci a spolupráci. Až na druhé místě je výběr procesu a nástrojů (3). Agilních metodik existuje celá řada a jsou postaveny na souboru principů definovaných v Manifestu agilního vývoje (5). Jako příklady takových metodik je možné uvést Dynamic System Development Method, Scrum, Feature Driven Development.



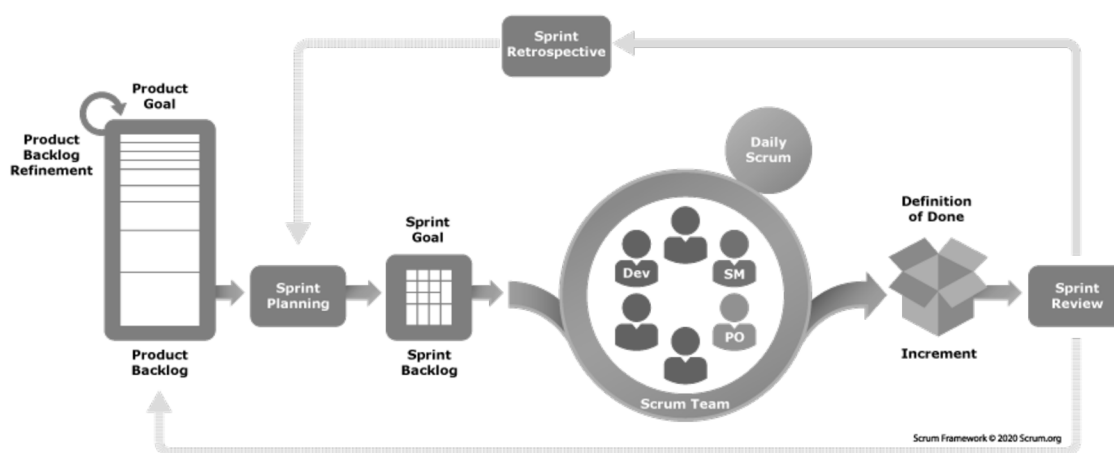
Obrázek 4: schéma procesu metodiky OpenUP; zdroj: (1)

Buchalcevodá konstatuje (1), že tyto metodiky se stále vyvíjí a také vznikají nové. Jako příklady uvádí MSF for Agile Software Development, nebo metodiku OpenUP. Metodika OpenUP je hezkým příkladem takového vývoje. Buchalcevodá uvádí (1), že vznikla z metodiky Unified Process a je postavena na podobných principech jako metodika RUP (Rational Unified Process), což znamená, že je založena na iterativním a inkrementálním modelu, případech užití, řízení rizik a architektuře. (1)



Obrázek 5: schéma etap metodiky DSDM; zdroj: (6)

Obecně se dá říci, že v agilních metodikách vývoje je preferován nižší stupeň formalismu, kde členové týmů nejsou zatěžováni komplikovanými procesy s velkým množstvím různých výstupů, zaznamenáváním kontrolních bodů a dalšími činnostmi, které by v principu omezovaly adaptabilitu, na které agilní metodiky stojí. (3)



Obrázek 6: schéma agilní metodiky Scrum; zdroj: (7)

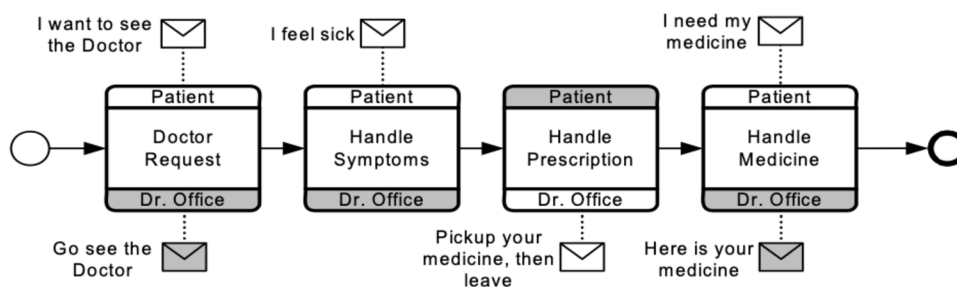
Scrum je metodika stojící na třech hlavních pilířích: transparentnosti, kontrole a adaptaci (7). Vývoj v rámci této metodiky probíhá v tzv. sprintech, což jsou iterace ohraničené krátkým časovým obdobím. Každý sprint je ukončen prostřednictvím sprint review, kde se tým schází nad shrnutím provedené práce a plánováním činností na další sprint. Stejně jako v ostatních agilních metodikách je i v metodice Scrum velmi důležitý vývojový tým a jeho spolupráce. Principy metodiky Scrum zakotvili jeho autoři Sutherland a Schwaber v uceleném dokumentu The Scrum Guide. (8)

### 3.1.3 Přístupy k modelování informačních systémů

V předchozí podkapitole 3.1.2 bylo zmíněno několik metodik vývoje sloužících ke standardizaci vývoje informačních systémů. Metodik existuje samozřejmě mnohem více a metody v nich obsažené nejsou omezeny jen na ně. Metody i techniky mohou být používány i mimo popsané metodiky vývoje. Zatímco metody určují, co se má v určitých etapách životního cyklu stát, techniky poskytují postupy, jak toho dosáhnout. Metody jsou tedy vždy spojeny s konkrétním přístupem, pomocí kterého je, s použitím technik a nástrojů, možné k požadovaným cílům postupovat. (1)

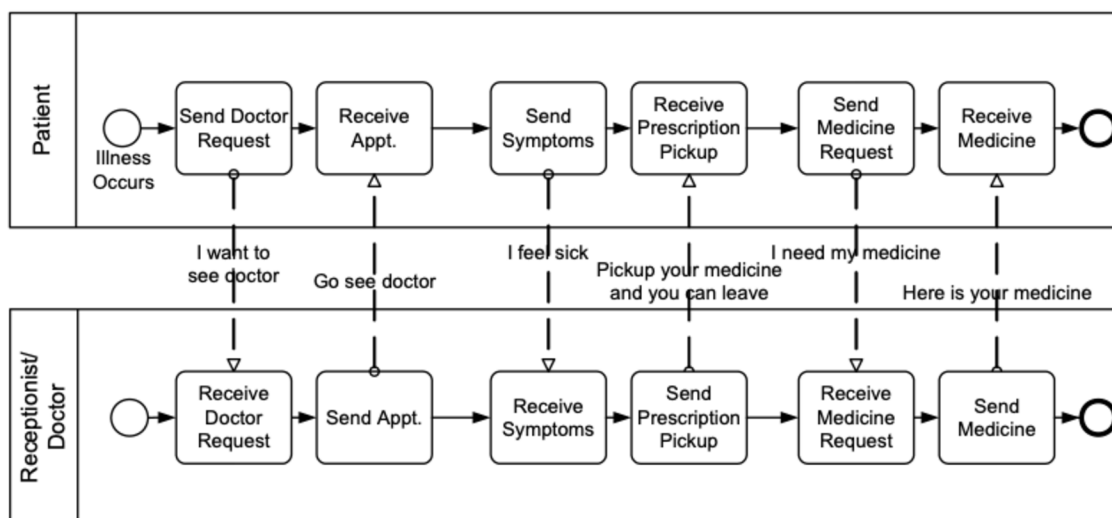
### 3.1.3.1 Procesní modelování

Procesní modelování slouží ke zkoumání podnikových procesů jako vzájemně provázaných činností. Modelování procesů předpokládá, že procesy neprobíhají náhodně, ale na základě vnějších, nebo vnitřních podnětů. Vnější podněty jsou v procesním modelování označovány jako události. Za vnitřní podněty je označován stav, ve kterém se činnost nachází. Jak už bylo zmíněno, činnosti jsou provázány pomocí vazeb, které ze systému činí definovanou strukturu. (9)



Obrázek 7: příklad choreografie v BPMN; zdroj: (10)

Jako příklad jednoho z novějších standardů pro modelování procesů je Business Process Model and Notation, spravovaný konsorciem Business Process Management Initiative, později Object Management Group. Standard zahrnuje grafickou reprezentaci podnikových procesů v přehledných diagramech a je doplněn jazykem Business Process Modeling Language. Základem jazyka BPML je Extensible Markup Language (modely jsou v definovaném prostředí spustitelné) (9). Výhodou notace BPMN je, že diagramy jsou relativně snadno čitelné například i pro zákazníky.

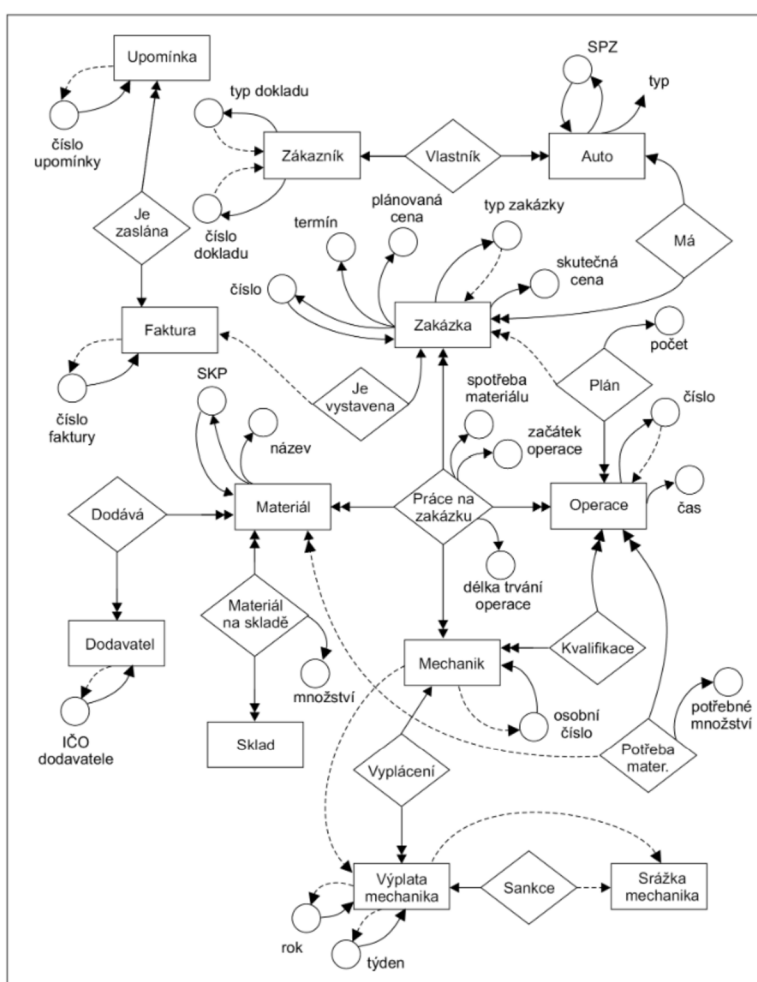


Obrázek 8: příklad kolaborativního procesu v BPMN; zdroj: (10)

### 3.1.3.2 Datové modelování

Při tvorbě datových modelů je vhodné rozlišovat několik úrovní detailu. Konceptuální model popisuje obsah dat v systému nezávisle na implementačním a technologickém prostředí. Logický model jako další úroveň popisuje realizaci systému a respektuje architekturu systému. Fyzický model je třetí úroveň detailu a popisuje vlastní realizaci databáze v konkrétním implementačním prostředí. (1)

Popsaný koncept s využitím abstrakce se v literatuře objevuje také jako princip „tří architektury“ (9).



Obrázek 9: příklad Entity Relationship diagramu; zdroj: (1)

Základními prvky konceptuálního datového modelu jsou entita a vztah. Jejich přesné vymezení je opět o úhlu pohledu. Co je někým považováno za entitu, může být někým jiným považováno za vztah a opačně. Některé přístupy považují entitu za jednu konkrétní věc, jiné za skupinu věcí jako entitní množinu. U vztahů mezi entitami se rozlišuje multiplicita (násobnost) a volitelnost (zda vztah povinný, či není). (1)

### 3.1.3.3 Objektově orientovaný přístup k modelování systémů

Původ objektově orientovaného přístupu k modelování uvádí Booch ve své práci (11) do souvislosti s objektově orientovanými programovacími jazyky. Myšlenku zdůvodňuje tak, že možnosti objektově orientovaných programovacích jazyků vedly k tomu, že se jejich principy snažili lidé aplikovat i na řešení problémů skutečného světa. Objektově orientovaný přístup byl postaven na pevných základech již existujících technologií. Tím, jak software nabýval na komplexitě, vznikla potřeba posunout výzkum v oblasti vývoje softwaru směrem k abstrakci a dekompozici (11). Objektově orientovaný přístup k analýze systémů skutečně vychází z principů objektově orientovaného programování. Principy uplatňované v tomto typu přístupu jsou následující. (12)

- Abstrakce. Princip kladoucí důraz na důležitá hlediska a oproštění od nepodstatných detailů. Princip, jehož smyslem je usnadnit modelování komplexních systémů.
- Zapouzdření. Princip oddělování vnějších hledisek od implementačních detailů.
- Sdílení. Princip umožňující dědičnost datové struktury. Princip důležitý z hlediska opakované použitelnosti a efektivity.

### 3.1.4 Modelovací jazyk Unified Modeling Language

#### Metodika vývoje

Buchalcevo<sup>á</sup> (1) jako výhodu jazyka Unified Modeling Language (UML) uvádí, že je nezávislý na procesu vývoje a není svázán s žádnou konkrétní metodikou. Fowler naopak říká (3), že komunita objektově orientovaného modelování dlouhodobě preferuje iterativní způsob vývoje, či jeho obdoby. Iterativní způsob vývoje byl stručně popsán v podkapitole 3.1.2.2 včetně jeho výhod a nevýhod. Při vývoji softwaru pomocí jazyka UML je vždy minimálně vhodné, spíše nutné, volit a přizpůsobovat metodiku vývoje tak, aby nejvíce vyhovovala konkrétnímu projektu. (3)

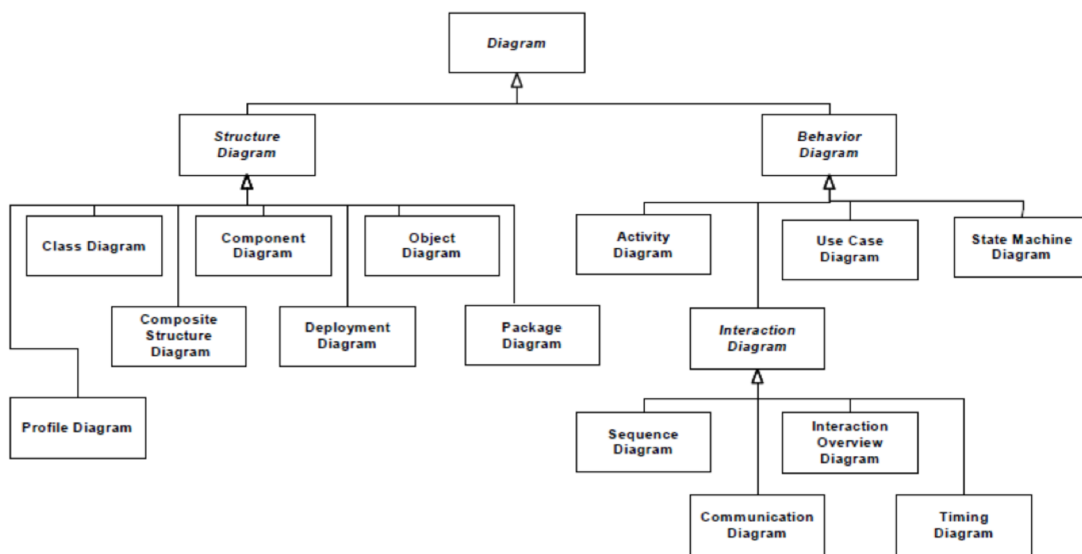
#### Specifikace Unified Modeling Language

O rozvoj specifikace se stará neziskové konsorcium The Object Management Group® Standards Development Organization. Konsorcium má v současnosti více než 220 členů a spravuje přes 260 specifikací (13). Aktuální specifikace OMG® Unified Modeling Language® (OMG UML®) je dostupná ve verzi 2.5.1 (12).

## Významové oblasti UML

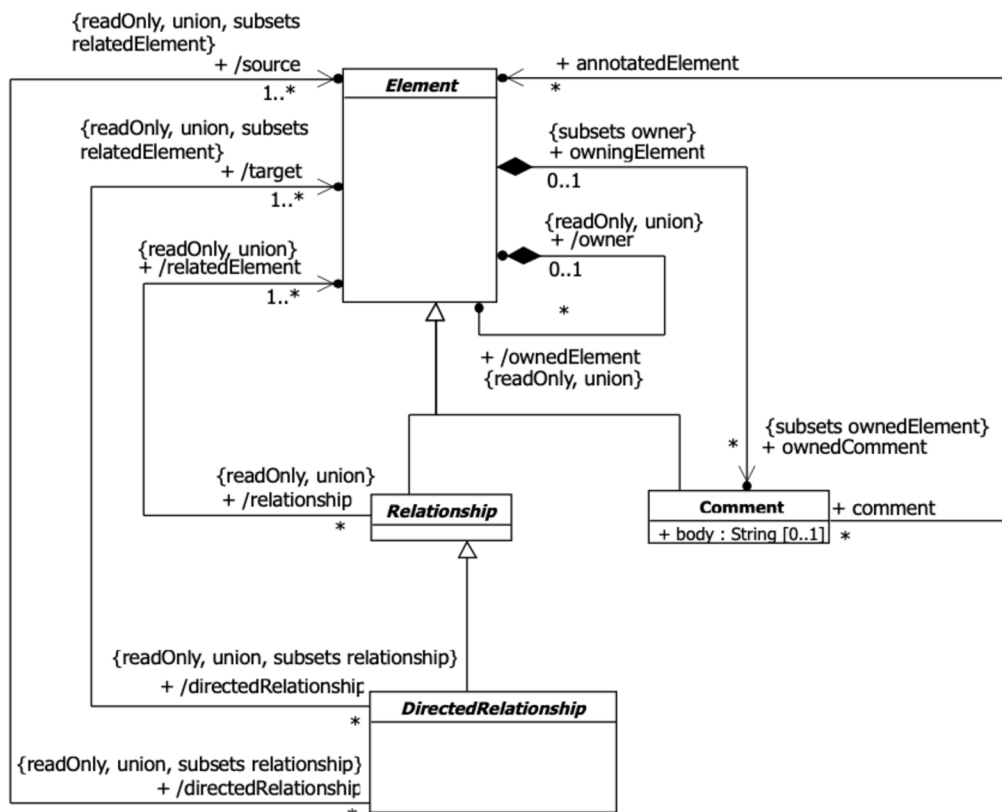
Jazyk UML umožňuje využít různých diagramů k modelování systémů pod různými úhly pohledu v různé úrovni abstrakce. Specifikace UML umožňuje v rámci speciálního frameworku vytvořit diagramy abstraktně, aby ve fázi analýzy pomohly k modelování problému. Na druhou stranu je možné diagramy (ne všechny) vytvořit jako spustitelné v definovaném prostředí. (12)

Specifikace člení diagramy po významové stránce na diagramy zachycující strukturu a diagramy zachycující chování (12).



Obrázek 10: rozdělení UML diagramů podle významu; zdroj: (12)

### 3.1.4.1 Diagramy UML k modelování struktury



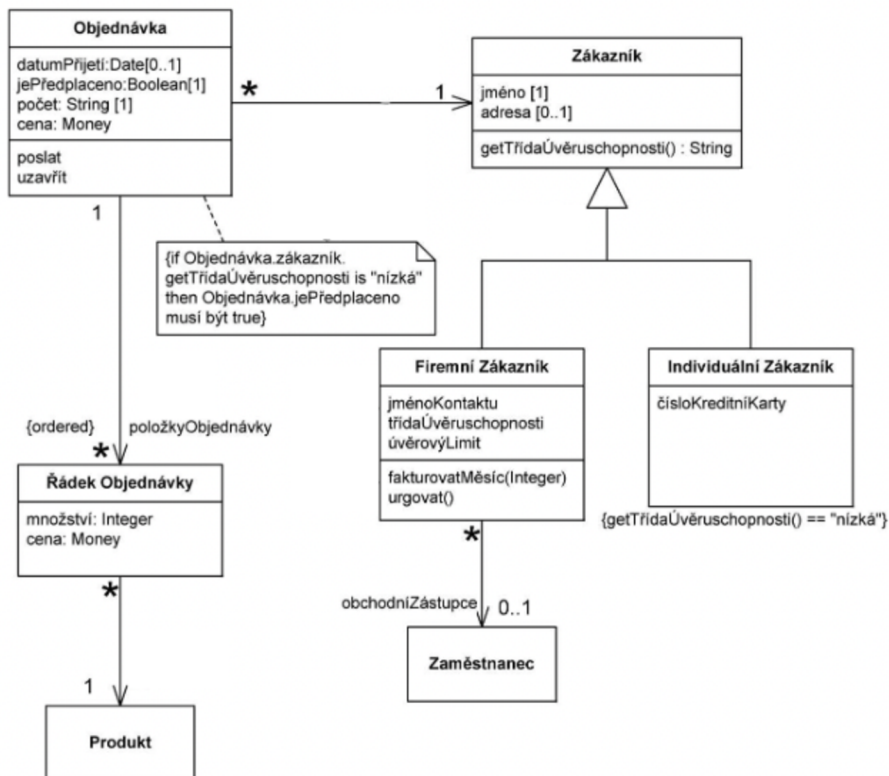
Obrázek 11: základní koncept elementů a vztahů v UML; zdroj: (12)

Specifikace UML řadí do této kategorie například diagramy tříd, diagramy balíčků, diagramy komponent a další. Základním konceptem, na kterém stojí všechny diagramy UML zachycující statickou strukturu systému, jsou elementy a vazby mezi nimi. (12)

#### Diagram tříd

Pomocí diagramů tříd (class diagrams) jsou popisovány typy objektů (třídy), které se v systému vyskytují, a jaké vztahy (asociace) mezi nimi existují. Diagramy tříd zobrazují také vlastnosti objektů a operace, které objekty třídy mohou vykonat. (14)

Vztahy mezi třídami objektů jsou modelovány pomocí asociací. Asociace znamená, že v systému existuje vazba objektu jedné třídy na objekt třídy jiné. Diagram neříká, na základě které události vazba vzniká, nebo v jakou chvíli k vytvoření vazby dochází. Proto jde v diagramu tříd o strukturu statickou. (1)



Obrázek 12: příklad jednoduchého diagramu tříd; zdroj: (3)

Generalizace je typ asociace podle principu dědičnosti. Pomocí generalizace je možné modelovat sdílení vlastností a operací mezi třídami tak, že jedna třída je podtypem třídy jiné, pokud má stejné vlastnosti a operace jako její nadtyp a zároveň má nějaké vlastnosti nebo operace navíc. Z toho vyplývá, že tam, kde je možné použít objekt nadtřídy, by mělo být možné bez problému použít i objekt podtřídy (princip nahraditelnosti). (3) Obrázek 12 zobrazuje generalizaci tak, že Individuální Zákazník a Firemní Zákazník jsou podtypy třídy Zákazník.

Kompozice je typ asociace, která využívá koncept vlastníka objektu. Taková asociace znamená, že každá instance určité třídy může mít právě jednoho vlastníka. Zrušení objektu vlastníka má za následek zrušení i vlastněného objektu. Vlastněný objekt v této asociaci nemůže existovat v systému samostatně bez svého vlastníka. (12)

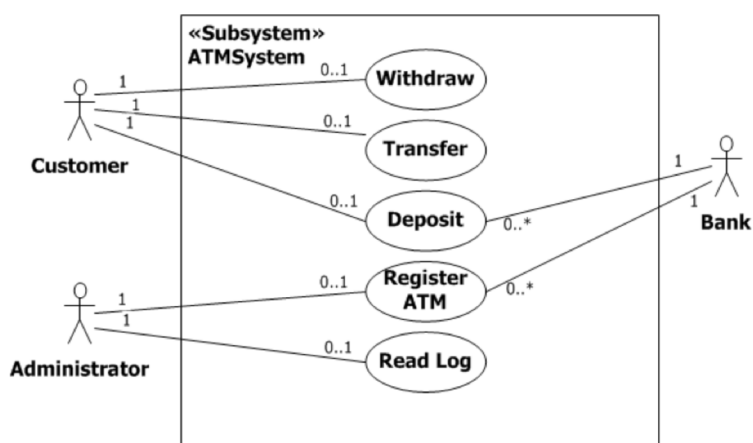
### 3.1.4.2 Diagramy UML k modelování chování

Diagramy chování slouží k modelování chování objektů v systému, včetně metod, interakcí a změn stavů. Dynamické chování je zde považováno za řetězec změn v systému v průběhu času. Mezi diagramy chování jsou specifikací řazeny diagramy užití, diagramy aktivit, sekvenční diagramy a mnoho dalších. (14)



## Diagram případů užití

Podstatu případů užití výstižně popisuje Fowler (3), který říká: „*Případ užití je sada scénářů, které jsou dohromady svázané společným cílem uživatele.*“ Je to trefný popis, protože aby mohly být výstižně popsány interakce účastníků se systémem, je užitečné formulovat konkrétní scénáře, podle kterých interakce probíhají. Samotný diagram případů užití (use case diagram) toho o činnosti aktérů v systému bez popisu (nebo scénářů) mnoho neříká. Všechny scénáře jednoho případu užití by měly mít společný cíl.



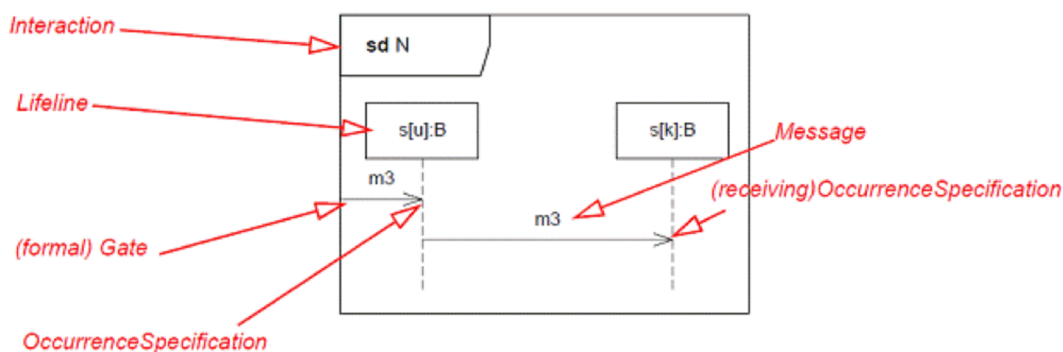
Obrázek 13: příklad use case diagramu; zdroj: (12)

Diagramy případů užití (use case diagrams) jsou řazeny specifikací pod diagramy chování, i když nezachycují přímo chování v čase, zachycují ale činnosti, které účastníci (role, aktéři) provádí v systému. Účastníci nemusí být jen uživatelé, ale také jiné externí systémy, služby atd. Aktér je v diagramu znázorněn piktogramem postavy. Aktér může být znázorněn také obdélníkem s klíčovým slovem «actor». Subjekt sady případů užití (systém) je zakreslen velkým obdélníkem s popisem. Uvnitř tohoto obdélníku jsou zakresleny v jednotlivých elipsách činnosti (případy užití). (12)

## Sekvenční diagram

Sekvenční diagram (sequence diagram) jako jeden z nejpoužívanějších interakčních diagramů slouží k modelování interakcí z pohledu zpráv, které si objekty v systému vyměňují. Sekvenční diagram graficky zobrazuje jeden scénář jednoho případu užití. Základní prvky těchto diagramů jsou tzv. čáry života (originálně Lifelines). Čáry život mají nahoře obvykle obdélník s vepsanou identifikací a symbolizují účastníka interakce. Celá interakce je pak ohraničená obdélníkem a pojmenovaná. Zprávy, které si objekty v rámci interakce vyměňují, jsou v diagramu znázorněny pomocí různých šipek. Diagram může být použit opět v různé míře abstrakce, tedy například k modelování složitějších komunikací,

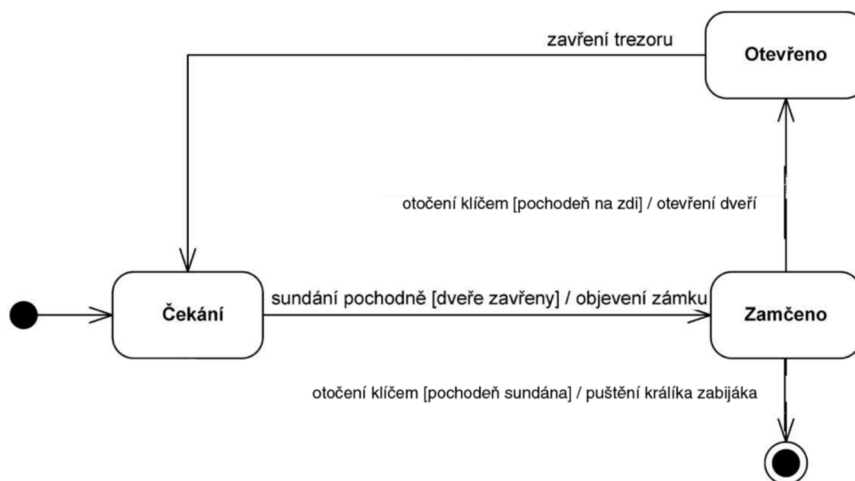
kde zprávy představují přímo volání metod vč. parametrů atd., nebo k více abstraktnímu modelování interakcí podle případů užití. Důležitá je v diagramu také posloupnost posílaných zpráv. (12)



Obrázek 14: příklad sekvenčního diagramu s popisem; zdroj: (12)

### Stavový diagram

Stavové diagramy jsou v UML používány opět k modelování dynamického chování objektů v systému. Stavový diagram se obvykle vztahuje k určité třídě, jejíž objekty se mohou nacházet v průběhu času v různých stavech. Na základě různých událostí se mohou stavy objektů měnit, tzn. objekt přechází z jednoho stavu do jiného. (14)

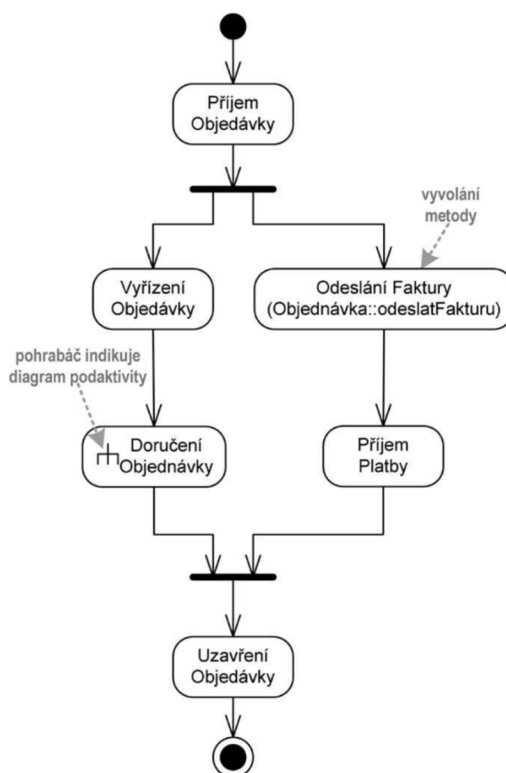


Obrázek 15: příklad stavového diagramu; zdroj: (3)

Hlavní prvky stavových diagramů jsou stavy objektu, přechody mezi stavy, události, podmínky a aktivity. Diagram může začínat v pseudostavu. Je to fiktivní stav, ze kterého je veden přechod k počátečnímu stavu. Přechody mezi procesními stavy zobrazují pravidla, podle kterých dochází ke změně stavů objektu. Pravidla přechodů jsou popsána tak, že na prvním místě každého přechodu může být uvedena spouštěcí událost, dále v hranatých

závorkách podmínka, která musí být splněna, aby k přechodu došlo, a na posledním místě aktivita vykonávaná během přechodu. Ani jedna z uvedených tří částí není v diagramu povinná. Přechod nemusí obsahovat ani aktivitu, ani podmínku. Vzácně se vyskytují přechody i bez události. Průběh stavového diagramu může být ukončen koncovým stavem, ze kterého již není možné přejít do jiného stavu. (3)

## Diagram aktivit



Obrázek 16: příklad diagramu aktivit; zdroj: (3)

Diagramy aktivit začínají počátečním uzlem a končí uzlem koncovým. Chování (akce) jsou zakreslovány pomocí příslušného symbolu, obdélníku se zaoblenými rohy, s vepsaným názvem chování uvnitř. Procesy mohou být v diagramu větveny pomocí rozhodovacích symbolů do paralelních větví a následně opět slučovány. Zatímco sekvenční diagramy zachycují interakci objektů v systému v rámci jednoho scénáře, v diagramu aktivit mohou být všechny scénáře zachyceny teoreticky do jednoho diagramu díky rozhodování, větvení a slučování. Mimo to je v diagramech aktivit užitečná také možnost dekompozice jednotlivých akcí. Akce může být rozkreslena dalším diagramem aktivit do většího detailu. (3)

## **3.2 Problematika společenství vlastníků jednotek**

### **Společenství vlastníků jednotek**

Společenství vlastníků jednotek je podle § 1189 - § 1216 zákona č. 89/2012 Sb. občanský zákoník v platném znění (2) nezisková organizace vznikající s cílem hospodárně spravovat nemovitosti, převážně bytové domy, v zájmu všech majitelů jednotlivých bytových nebo nebytových jednotek. Společenství se primárně zabývá udržováním a zlepšováním společných prostor a zařízení, které nejsou výhradním vlastnictvím jednoho vlastníka, ale slouží ke společnému užívání (15). Do jeho působnosti spadá i plánování a realizace oprav, údržba, modernizace či případné stavební změny pro zvýšení komfortu či hodnoty nemovitosti. (16)

### **Vznik**

Založení společenství vlastníků vyžaduje schválení stanov na ustavující schůzi, kde je zapotřebí souhlasu většiny všech vlastníků jednotek. Toto společenství pak oficiálně vzniká zápisem do veřejného rejstříku. Stanovy definují základní pravidla pro fungování společenství, včetně názvu, sídla, práv a povinností členů, stejně jako struktury a pravomocí jeho orgánů. (2)

### **Orgány společenství**

Nejvyšším rozhodovacím orgánem společenství je shromáždění vlastníků, které sestává ze všech majitelů jednotek a koná se podle zákona minimálně jednou ročně. Shromáždění rozhoduje o významných záležitostech týkajících se správy a rozvoje nemovitosti, schvaluje rozpočet a uzávěrku, volí nebo odvolává členy výboru, nebo předsedu, případně přijímá změny stanov. Rozhodnutí jsou přijímána na základě většinového hlasování přítomných členů. Statutárním orgánem společenství může být výbor nebo předseda společenství. Je zodpovědný za běžnou správu a zastupování společenství vůči třetím stranám. Tento orgán zajišťuje, že rozhodnutí shromáždění jsou provedena a že správa nemovitosti probíhá v souladu s právními předpisy a interními pravidly. (16)

### **Zánik**

Společenství vlastníků může zaniknout několika způsoby. Jedná se o situace, kdy je nemovitost celá zlikvidována, nebo pokud se vlastníci rozhodnou společenství dobrovolně rozpustit, což je možné v případě, že počet jednotek, nebo vlastníků klesne pod stanovenou hranici. V takovém případě dojde k převodu práv a povinností společenství na jednotlivé vlastníky v poměru jejich podílů na společných částech bez nutnosti likvidace. (2)

## 4 Vlastní práce

### 4.1 Sběr a vyhodnocení požadavků

Sběr požadavků byl proveden formou nestrukturovaných rozhovorů s osobami z řad potenciálních uživatelů, tedy běžných řadových členů SVJ i členů výborů či předsedů SVJ, která jsou spravována vlastními členy. Pozvání k rozhovorům přijalo i několik odborníků z oboru správy SVJ. Potenciální uživatelé, se kterými byl sběr požadavků proveden, byli vybráni z několika SVJ z obce Kařez a města Plzeň. Jako odborníci z oblasti správy společenství vlastníků jednotek byli vybráni profesionální předsedové a pracovníci specializované správcovské firmy PRODOMIA, s.r.o. z Plzně. Níže je uvedeno shrnutí a vyhodnocení sběru požadavků.

Osoby z řad běžných členů SVJ nejčastěji zmiňovali, že se potýkají s problematou dostupností informací o hospodaření společenství. O stavu hospodaření společenství během roku obvykle nemají přehled a dozvídají se o něm pouze na dotaz, nebo jednou ročně, když je předkládána uzávěrka.

Pokud jde o komunikaci v rámci společenství, uváděli dotazovaní, že ostatní členy SVJ znají většinou jen povrchně a komunikace probíhá zpravidla velmi omezeně. Většina osob ve společenství zná dobře jen mizivé procento ostatních členů. Dotazovaní často uváděli jako frekventovaně používaný nástroj komunikace nástěnku v domě. Pouze zanedbatelné procento dotazovaných osob uvedlo, že jsou s tímto způsobem komunikace spokojeni. Naopak drtivá většina dotazovaných uvedla, že by uvítala efektivnější nástroje pro vhodnou komunikaci v různých situacích. Někdy se hodí komunikace soukromá a jindy spíše veřejná. Individuálně to často řeší pomocí výměny kontaktů, nebo sociálních sítí, ale pro širší skupinovou komunikaci takové prostředky používány nejsou, protože ne všichni chtějí podobnou komunikaci vést přes sociální sítě, nebo telefon.

V oblasti komunikace se statutárními orgány SVJ uváděli dotazovaní nejčastěji situace, kdy se v domě vyskytne nějaká záležitost k řešení (závada, nebo jiný problém) a nikdo z členů neví, jestli už ji někdo výboru nahlásil, a pokud ano, jakým způsobem ji výbor řeší a v jaké fázi řešení se věc nachází atd. Často pak záležitost nenahlásí k řešení nikdo, protože si všichni myslí, že ji nahlásili ostatní. Někdy naopak bývají hlášení duplicitní a o řešení není přehled.

Podle dotazovaných odborníků začíná být zajímavým tématem také společné vybavení bytových domů. V moderních rezidenčních domech se začínají častěji vyskytovat prvky společného vybavení, jako zvláštní prostory, sauny, bazény, sportoviště, salonky, zahradní záhony, venkovní kuchyně atp. Ty mohou sloužit ke každodennímu použití všemi vlastníky, ale s tím může vlastníkům vznikat potřeba se na jejich používání nějakým způsobem domlouvat. Například může být za určitých situací vhodné si takové společné vybavení rezervovat, nebo dát vědět, že je mimo provoz, například kvůli údržbě, závadě, ročnímu období atp. Z hlediska harmonického komunitního soužití všech vlastníků by bylo vhodné mít k takovým účelům vhodné nástroje.

Dotazovaní z řad členů statutárních orgánů SVJ uváděli nejčastěji, že jsou zatěžováni velmi častou komunikací ze strany některých vlastníků, a požadavky dostávají duplicitně, nebo vůbec. Jako další bod byla zmiňována administrativa související se zasedáním shromáždění společenství. Členové výborů SVJ (zvláště těch početnějších) zmiňovali, že zasedání shromáždění pro ně bývá komplikované administrativně kvůli vyhodnocování účasti a hlídání schopnosti usnášet se, někdy i kvůli nízké účasti, protože pak shromáždění svolávají opakovaně, což je opět neefektivní, nebo výjimečně přistupují také k hlasování mimo shromáždění. Mohlo by se zdát, že administrativní zátěž členů výboru v souvislosti se zasedáním shromáždění je neopodstatněný argument, ale je potřeba uvést, že členové výboru často nemají odměny za svoji činnost pro SVJ v takové výši, aby jim řešení podobných věcí adekvátně vynahradila a opakované svolávání shromáždění kvůli nízké účasti dopadá i na vlastníky. Skoro nikdo nechce na zasedání chodit zbytečně, ani opakovaně. Zároveň například v případě nově vznikajících SVJ mohou být zasedání frekventovanější, než předepisuje legislativa, protože nějakou dobu může trvat, než se vztahy a záležitosti ve společenství ustálí.

Nejčastěji a nejobsáhleji byly opakovaně zmiňovány problémy v následujících oblastech:

- komunikace v SVJ, absence nástrojů pro specifické druhy komunikace v různých situacích (veřejná vs. soukromá), řešení záležitosti maximálně přes nástěnku,
- administrativní zátěž způsobená neefektivní komunikací,
- dostupnost informací o řešení záležitostí v SVJ a o hospodaření SVJ,
- administrativní zátěž související se zasedáním shromáždění SVJ,

- v případě větších SVJ problémy s účastí na zasedání shromáždění, vyhodnocování účasti, hlasování atd.

## 4.2 Koncepce systému

Následující analýza systému vychází ze získaných požadavků a poznatků (podkapitola 4.1) a nahlíží na systém z několika různých pohledů – z pohledu statické struktury objektů v systému a souvislostí mezi nimi prostřednictvím objektového modelu, z pohledu chování systému v průběhu času prostřednictvím stavového modelu a z pohledu vzájemné spolupráce objektů systému prostřednictvím modelu interakcí. Výsledkem je podklad pro mnohem podrobnější návrh pro realizaci se širokými možnostmi dalšího rozvoje.

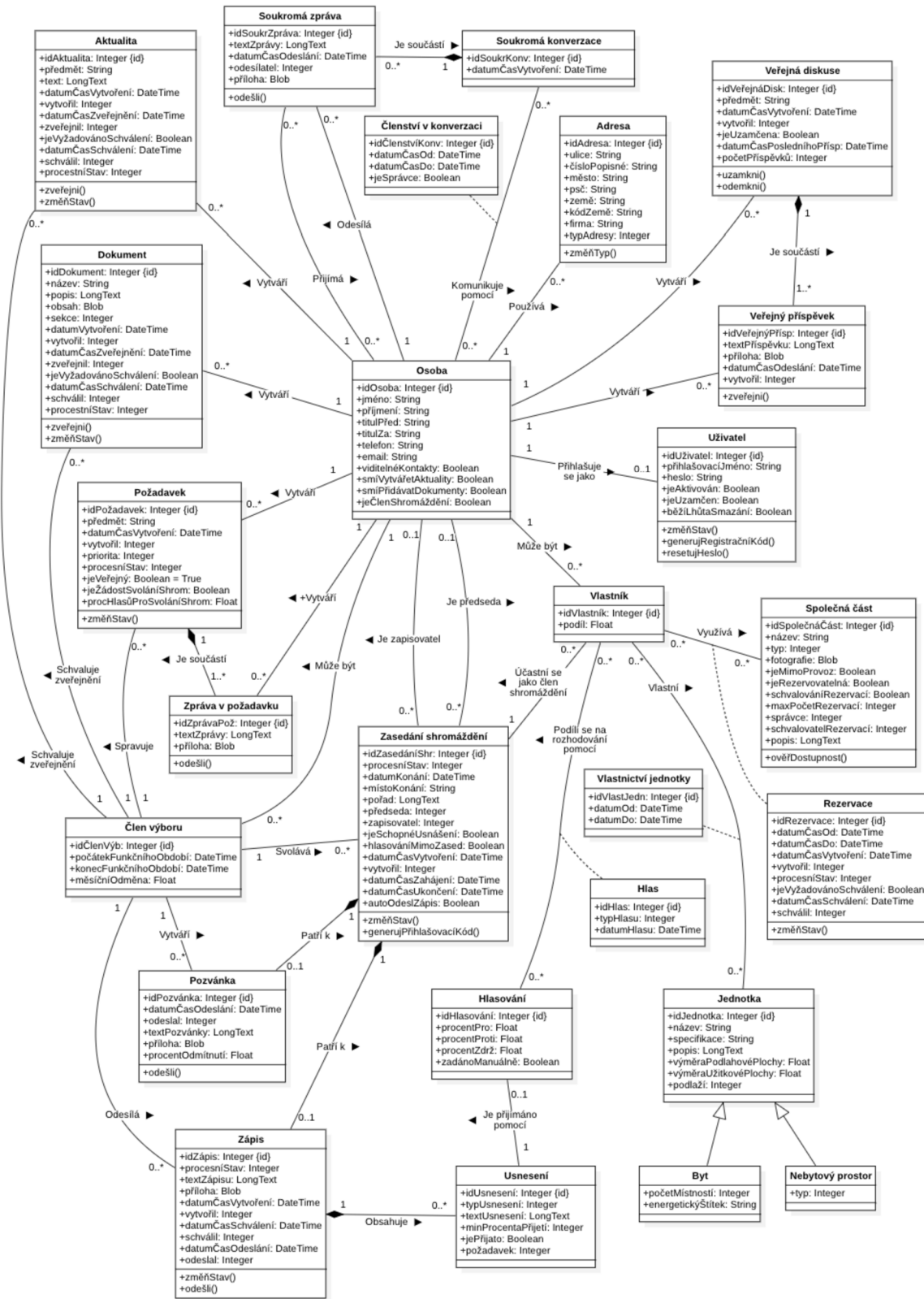
Systém, jehož návrhu se tato práce věnuje, by měl zefektivnit a usnadnit vzájemnou komunikaci členů společenství vlastníků jednotek a jeho orgánů, zvýšit transparentnost správy společenství a podpořit komunitní bydlení. Neměl by to být například systém pro vedení účetnictví, ale nástroj pro lidi, kteří spolu žijí na společném místě a tvoří tak společenství nejen po stránce legislativy. Systém by měl nabídnout i mechanismy k obsluze významných procesů opět s důrazem na vzájemnou komunikaci všech osob ve společenství. Informace související se společenstvím by pro jeho členy měly být díky systému dostupné snadno na jednom místě.

## 4.3 Objektový model

Diagram tříd (Obrázek 17) je konceptuálním modelem statické struktury objektů systému a vazeb mezi nimi. Je zpracován do takové míry detailu, aby umožnil co nejlepší pochopení analyzovaného systému. Finální návrh bude zahrnovat více objektových tříd a vazeb, jejichž zakreslení do tohoto modelu mohlo způsobit významné snížení jeho přehlednosti pro účely této analýzy.

Analýza nepracuje s konkrétní platformou či databázovým systémem. Datové typy jsou u jednotlivých atributů uvedeny spíše pro pochopení jejich významu a ve finálním návrhu pro realizaci se pravděpodobně budou mírně lišit. Pokud jde o operace, diagram tříd (Obrázek 17) zahrnuje operace specifické vždy pro danou třídu a podstatné pro účely analýzy. V diagramu nejsou zahrnuty běžně se vyskytující operace pro vytváření, úpravu, mazání objektů atp.

### 4.3.1 Diagram tříd



Obrázek 17: diagram tříd systému Domovnice; zdroj: vlastní zpracování



### **4.3.2 Datový slovník**

#### **Osoba**

Reprezentace skutečné osoby z reálného světa. Budou to primárně všechny osoby se vztahem k SVJ, tedy každý člověk, který bude systém používat, členové, osoby vykonávající nějakou funkci v SVJ, osoby zastupující správce SVJ, ale může to být také dodavatel atd.

#### **Vlastník**

Osoba vlastníci bytové či nebytové jednotky v SVJ (jednu či více), nebo osoba zastupující právnickou osobu vlastníci bytové či nebytové jednotky v SVJ. Vlastník je členem shromáždění, nejvyššího orgánu SVJ.

#### **Člen výboru**

Osoba vykonávající funkci statutárního orgánu SVJ, tedy člen výboru SVJ, nebo předseda SVJ jako jednočlenný výbor, pokud je to tak zakotveno ve stanovách SVJ. Člen výboru i předseda může (ale nemusí) být zároveň vlastníkem.

#### **Uživatel**

Reprezentuje uživatelský účet osoby v systému Domovnice. Musí existovat pro všechny osoby používající systém, protože nese informace o přihlašovacích údajích. Ty jsou od osoby odděleny, protože ne všechny osoby související se SVJ musí mít v systému uživatelský účet a zároveň je nutné zachovávat určitou historii evidence osoby.

#### **Adresa**

Reprezentuje adresu osoby. Jedna osoba může mít v systému evidováno více adres a ty mohou být různého typu (trvalá, kontaktní, firemní).

#### **Jednotka**

Třída reprezentující jednotky, což jsou právně samostatné části jedné budovy a jejich přesná definice je stanovena platnou legislativou. Zobecnění tříd Byt a Nebytový prostor.

#### **Byt**

Objekty této třídy jsou podtypem třídy Jednotka. Reprezentují samostatné bytové jednotky podle platné legislativy.

#### **Nebytový prostor**

Objekty této třídy jsou podtypem třídy Jednotka. Reprezentují samostatné nebytové prostory podle platné legislativy.

## **Vlastnictví jednotky**

Záznam, kterým SVJ eviduje vlastnictví jednotky vlastníkem. Představuje vazbu mezi Vlastníkem a Jednotkou.

## **Zasedání shromáždění**

Objekty této třídy reprezentují jednotlivé schůze nejvyššího orgánu SVJ, tedy schůzi vlastníků jednotek. Ve výjimečných případech slouží objekty této třídy také jako reprezentace hlasování mimo zasedání shromáždění SVJ.

## **Požadavek**

Objekt reprezentující speciální konverzační vlákno vytvořené osobou obsahující zprávy. Osoba podává vytvořením požadavku žádost členům výboru a očekává její řešení. Může jít o požadavky, žádosti, podněty nejrůznějšího charakteru. Reprezentuje také speciální požadavky jako žádost o svolání zasedání shromáždění v mechanismu svolání zasedání shromáždění SVJ z podnětu vlastníků.

## **Zpráva v požadavku**

Reprezentuje jeden příspěvek v požadavku. Je součástí požadavku. Může být odpovědí na jinou zprávu v rámci požadavku, doplnění informací relevantních k řešení požadavku, zprávou o řešení požadavku atd.

## **Pozvánka**

Reprezentuje dokument oznamující blížící se konání schůze shromáždění SVJ rozesílaný jeho členům. Alternativně také oznamuje blížící se hlasování mimo zasedání shromáždění SVJ. Obsahuje datum a čas konání zasedání, místo konání zasedání a pořad zasedání.

## **Usnesení**

Třída reprezentující jednotlivá usnesení, o jejichž přijetí rozhodují členové shromáždění pomocí hlasování. Objekty této třídy nesou podstatné informace popisující zasedání, jako text usnesení, informace o tom, zda bylo nebo nebylo přijato atd. Typ usnesení je důležitý z hlediska poměru hlasů nutných pro přijetí usnesení. Pro některé typy usnesení může být nutné například 75 %, nebo dokonce 100 % všech hlasů.

## **Hlasování**

Reprezentuje hlasování jako celek, tedy objekt nesoucí informace o počtu hlasů pro přijetí usnesení, počtu hlasů proti přijetí usnesení a počtu hlasů, které se zdržely hlasování.

## **Hlas**

Jednotlivý hlas jednoho vlastníka k jednomu usnesení. Představuje vazbu mezi Vlastníkem a Hlasováním. Může sloužit pro zachování historie po nějakou dobu. Po určité době podle nastavení parametrů systému mohou ze systému vymazány.

## **Zápis**

Dokument, který je výstupem ze zasedání shromáždění SVJ, nebo výstupem z hlasování mimo shromáždění SVJ. Obsahuje všechny údaje o proběhlém zasedání shromáždění SVJ jako datum a čas zahájení, datum a čas ukončení, usnesení, o kterých bylo rozhodováno, výsledky hlasování atd.

## **Soukromá konverzace**

Objekt reprezentující soukromé konverzační vlákno vytvořené osobou. Soukromá konverzace obsahuje soukromé zprávy. Jde o prostředek soukromé konverzace mezi dvěma a více osobami v systému. Spojení osob a konverzací je realizováno pomocí vazební třídy Členství v konverzaci.

## **Soukromá zpráva**

Každý objekt této třídy reprezentuje jednu soukromou zprávu, která je součástí soukromé konverzace. Soukromá zpráva může obsahovat přílohu.

## **Členství v konverzaci**

Vazební třída spojující osobu s konverzačním vláknem soukromé konverzace. Do konverzace mohou být osoby přidávány a mohou z ní být správcem konverzačního vlákna odebírány. Jedna soukromá konverzace může mít i více než jednoho správce. Automaticky je správcem zakladatel vlákna a poslední uživatel zbývajících v konverzačním vlákně.

## **Veřejná diskuse**

Třída objektů reprezentujících veřejná konverzační vlákna vytvořená osobami. Veřejné diskuse obsahují veřejné příspěvky. Veřejné diskuse tvoří dohromady diskusní fórum, které slouží jako nástroj otevřené komunikace mezi osobami používajícími systém.

## **Veřejný příspěvek**

Veřejné příspěvky jsou součástí Veřejné diskuse. Veřejný příspěvek může obsahovat přílohu.

## **Společná část**

Společné části nemovitosti, podle platné legislativy, které mají sloužit vlastníkům společně. Z hlediska použití v systému má smysl evidovat všechny společné části, zejména ty, které jsou rezervovatelné, což mohou být například zahradní pec, gril, bazén, sauna, sportoviště, konferenční (školící) místnost, společenská místnost, salonek atp.

## **Rezervace**

Rezervace objektů třídy Společná část. Jde o prvky, které se v moderních bytových domech začínají více vyskytovat a může být vhodné (nebo vyžadované) si jejich využití rezervovat.

## **Dokument**

Třída objektů reprezentujících dokumenty určené ke čtení osobám používajícím systém. Může se jednat o dokumenty základní dokumentace SVJ, technické dokumentace, smluvní dokumentace, dokumenty související se zasedáním shromáždění SVJ, evidenční listy a nejrůznější další dokumenty.

## **Aktualita**

Třída objektů sloužící jako další nástroj pro komunikaci všem uživatelům systému. Aktualita je jednoduchá zpráva obsahující předmět a text.

## **Další třídy v modelu**

Diagram tříd (Obrázek 17) obsahuje zejména takové objektové třídy a vazby mezi nimi, které jsou významné pro pochopení podstaty analyzovaného systému pro účely této analýzy. Finální návrh bude zahrnovat více objektových tříd a vazeb, jejichž zakreslení do diagramu tříd by mohlo způsobit významné snížení jeho přehlednosti pro účely této analýzy.

## **Atributy tříd**

Diagram tříd (Obrázek 17) obsahuje většinu atributů podstatných pro pochopení analyzovaného systému a zachycení jeho podstaty.

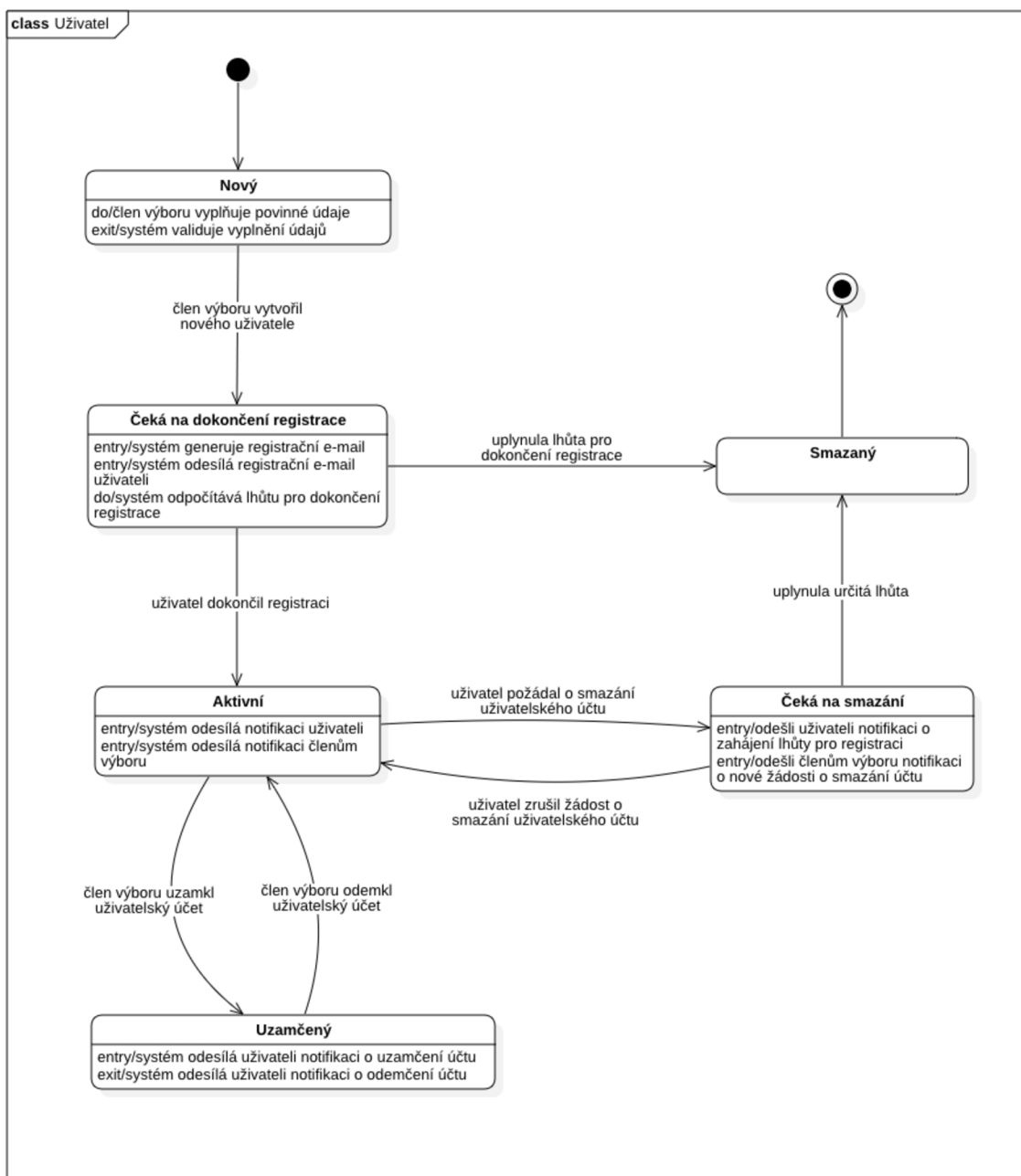
## **Operace tříd**

Diagram tříd (Obrázek 17) zahrnuje operace specifické vždy pro danou třídu a podstatné pro pochopení podstaty analyzovaného systému. V diagramu nejsou zahrnuty běžně se vyskytující operace pro vytváření, úpravu, mazání objektů atp. Finální návrhový diagram tříd bude obsahovat mnohem větší množství operací.

## 4.4 Stavový model

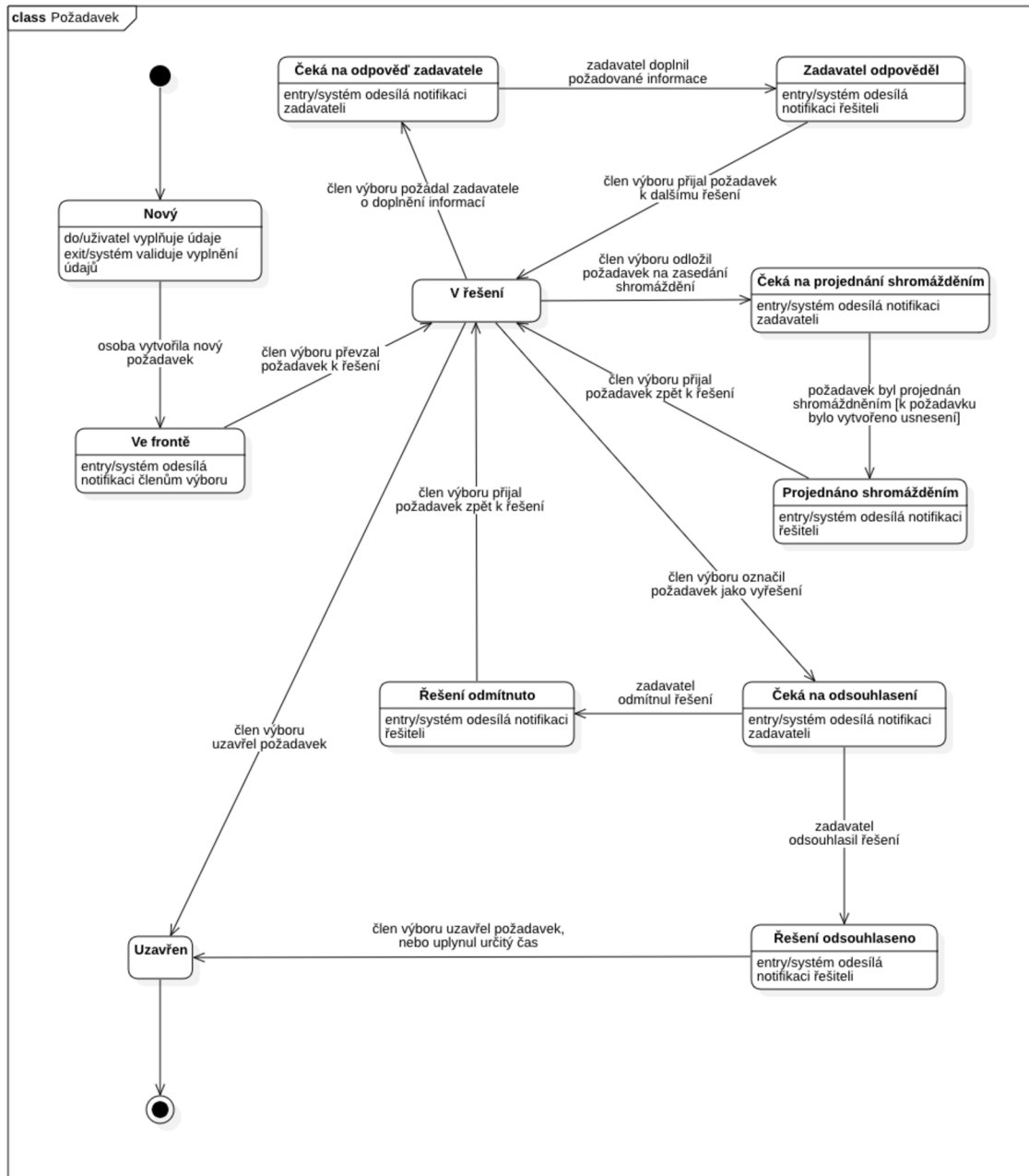
Stavový model nahlíží na analyzovaný systém z hlediska změny chování jeho objektů v průběhu času. Souvislosti událostí a stavů jsou zachyceny pomocí grafů – stavových diagramů.

### Třída Uživatel



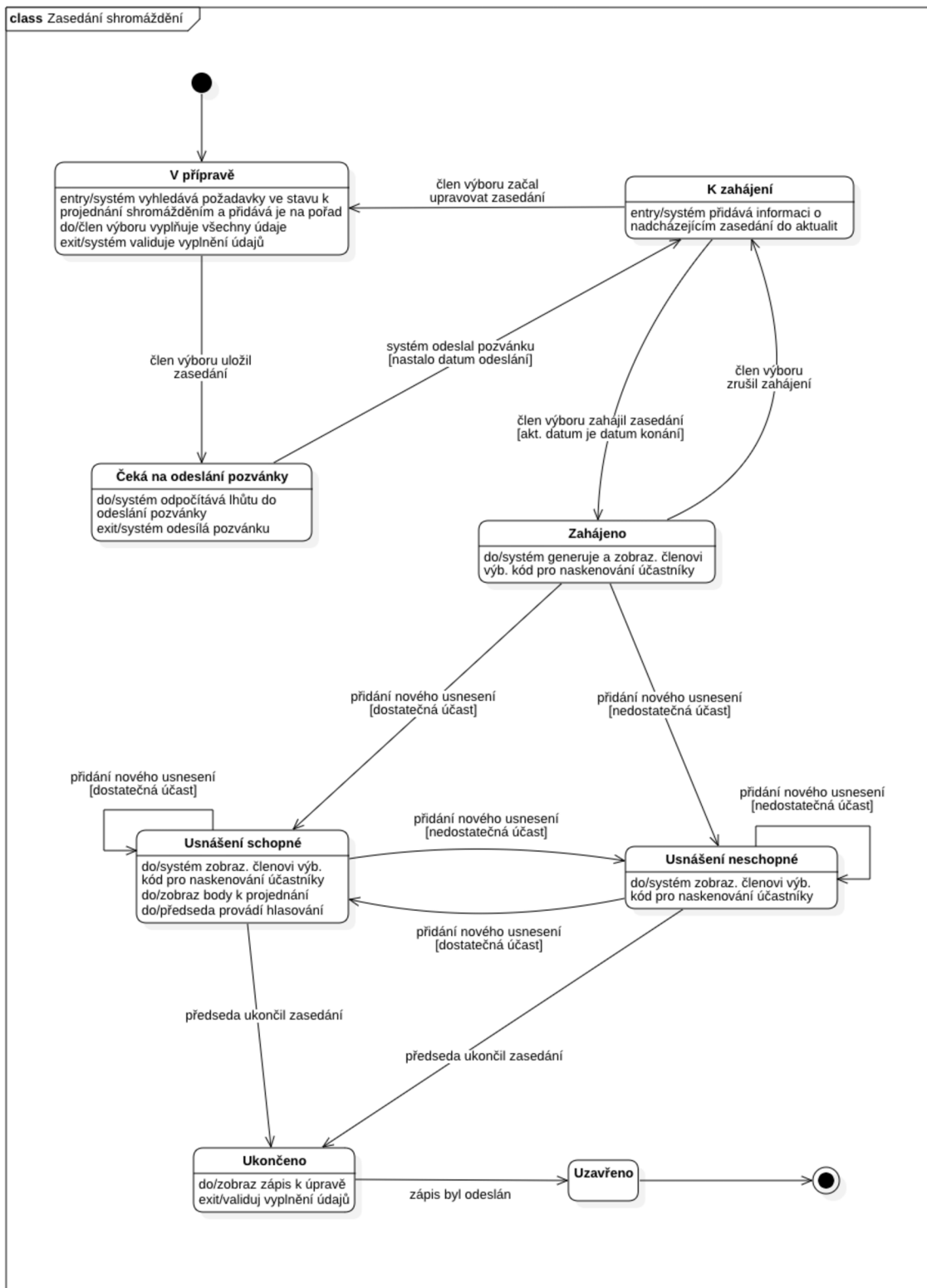
Obrázek 18: stavový diagram třídy Uživatel; zdroj: vlastní zpracování

## Třída Požadavek



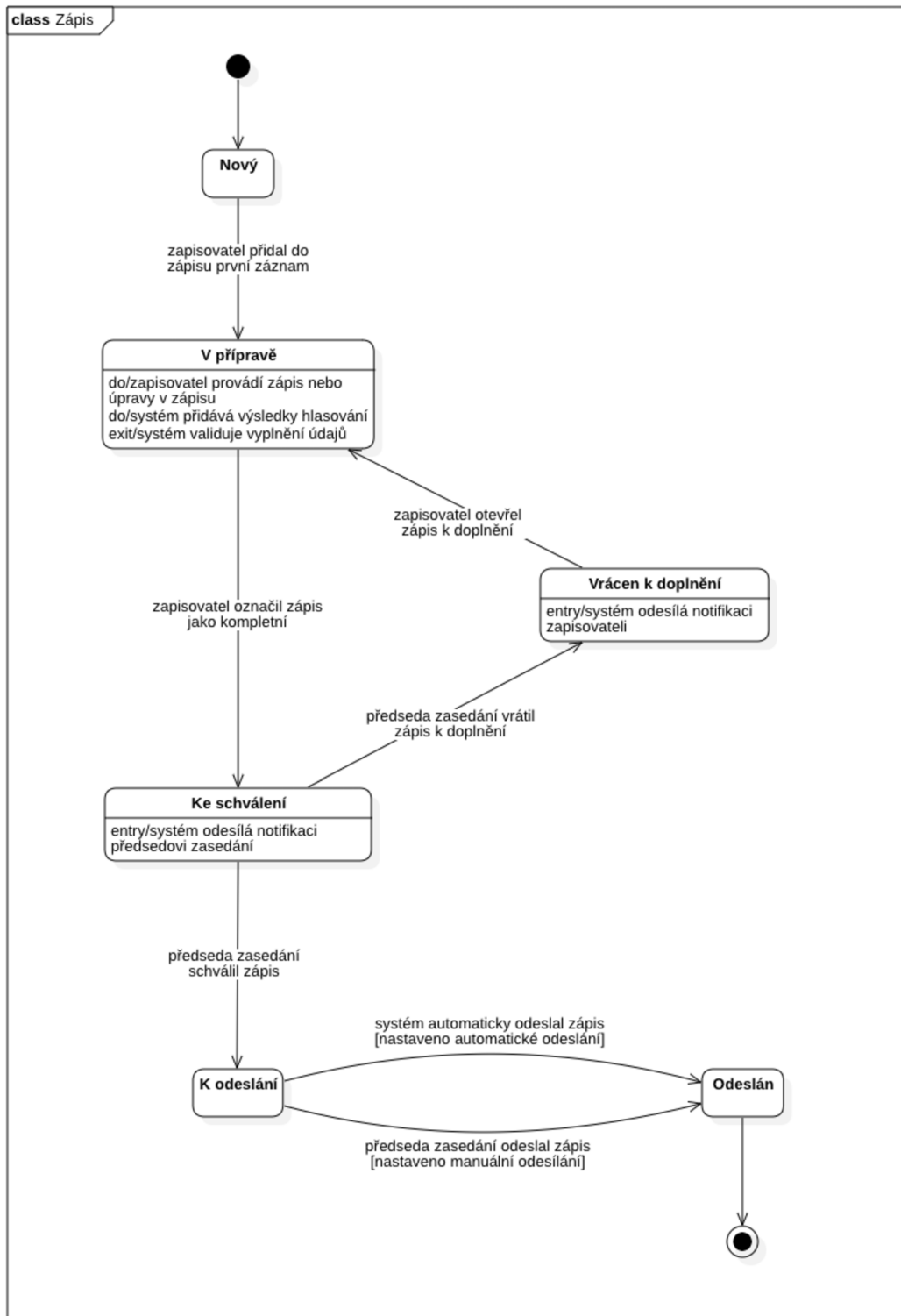
Obrázek 19: stavový diagram třídy Požadavek; zdroj: vlastní zpracování

## Třída Zasedání



Obrázek 20: stavový diagram třídy Zasedání; zdroj: vlastní zpracování

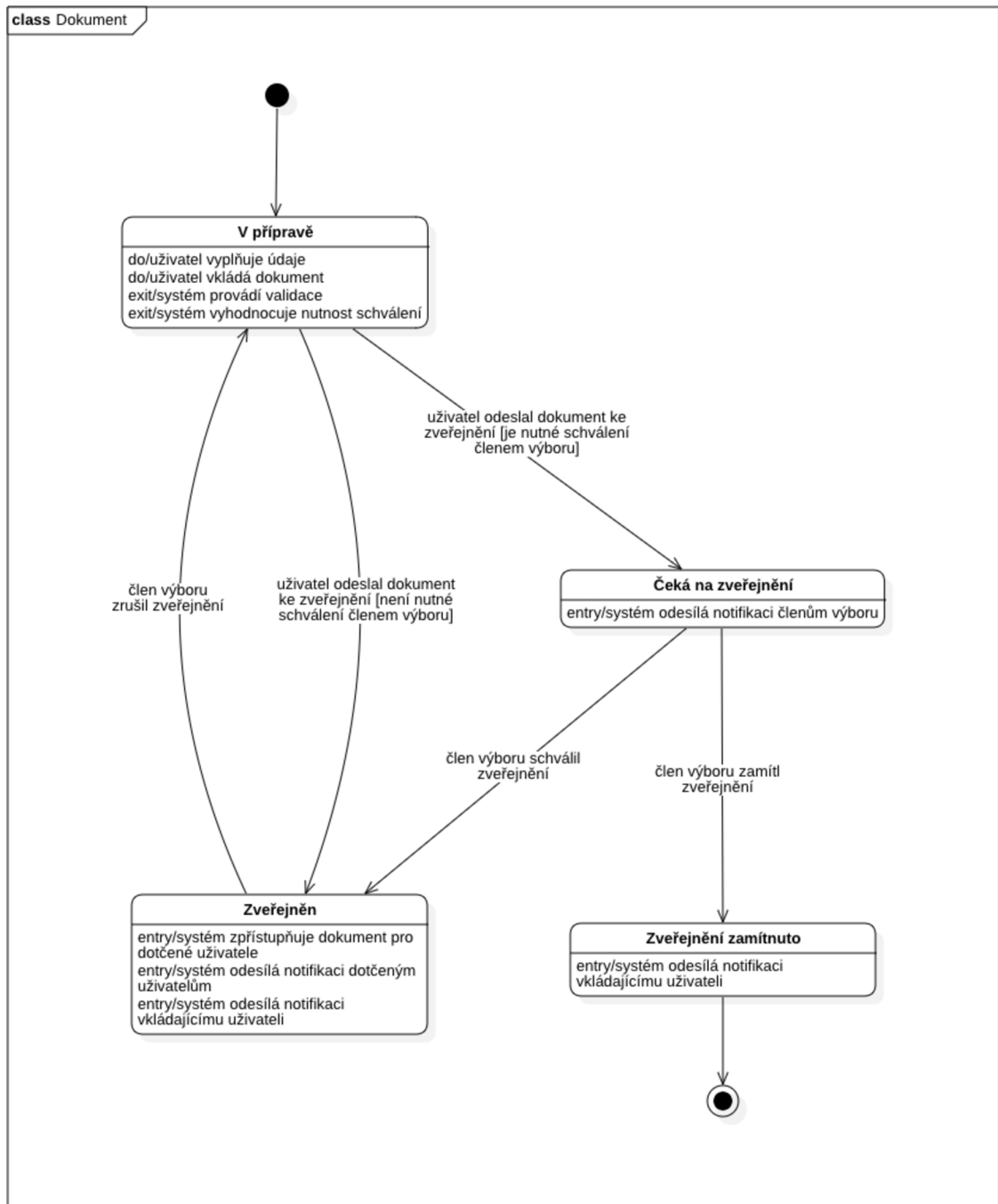
## Třída Zápis



Obrázek 21: stavový diagram třídy Zápis; zdroj: vlastní zpracování

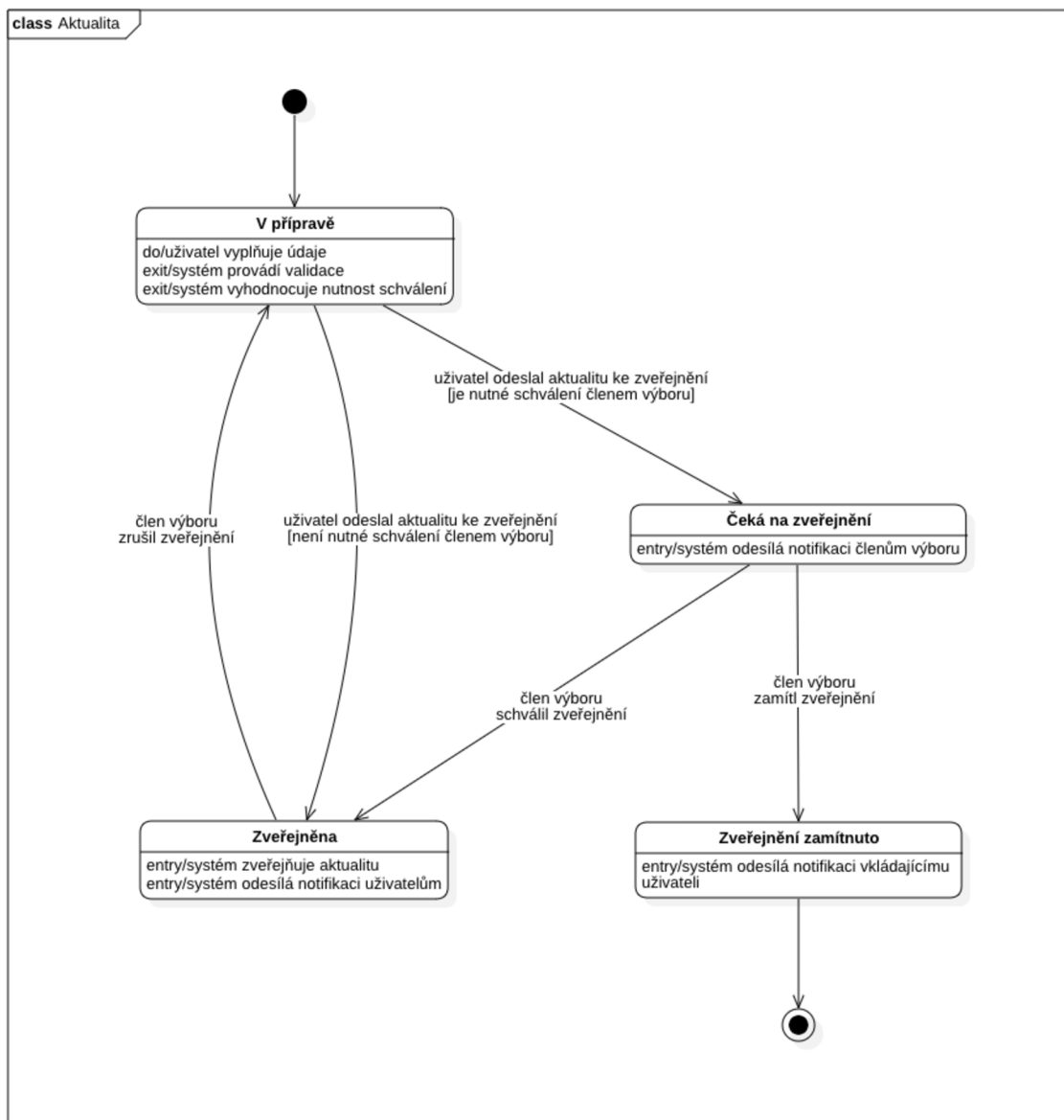


## Třída Dokument



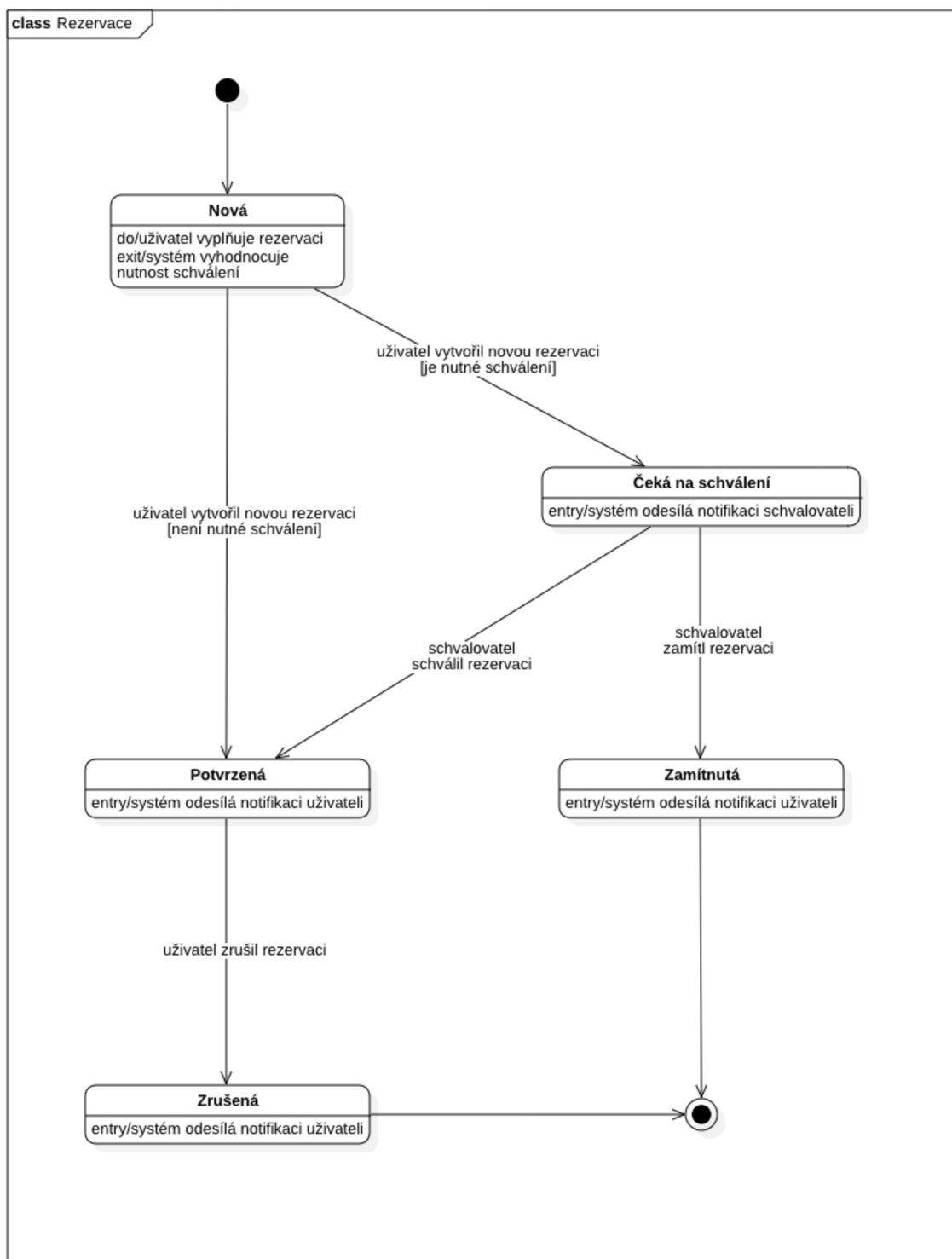
Obrázek 22: stavový diagram třídy Dokument; zdroj: vlastní zpracování

## Třída Aktualita



Obrázek 23: stavový diagram třídy Aktualita; zdroj: vlastní zpracování

## Třída rezervace



Obrázek 24: stavový diagram třídy Rezervace; zdroj: vlastní zpracování

## 4.5 Model interakcí

Objektový a stavový model jsou v této analýze doplněny modelem interakcí, který zachycuje vzájemné působení objektů v systému. Model obsahuje popis případů užití

(4.5.1), které jsou zachyceny do diagramu případů užití a rozepsány do scénářů. Z těchto scénářů pak vychází sekvenční diagramy (4.5.2) a diagramy aktivit (4.5.3).

### 4.5.1 Případy užití



Obrázek 25: diagram případů užití; zdroj: vlastní zpracování

#### 4.5.1.1 Administrace

##### **Aktéři**

- Člen výboru SVJ, nebo předseda SVJ (dále jen „člen výboru“)

##### **Popis případu užití**

Člen výboru v prostředí administrace vytváří uživatelské účty, vytváří a nastavuje bytové a nebytové jednotky a společné části, nastavuje parametry systému. Člen výboru přiřazuje k účtům vlastníků bytové a nebytové jednotky. Člen výboru provádí parametrizaci systému.

#### 4.5.1.2 Vytvoření a správa uživatelského účtu

##### **Aktéři**

- Vlastník
- Člen výboru SVJ, nebo předseda SVJ (dále jen „člen výboru“)
- Správce

##### **Popis případu užití**

Typičtí uživatelé systému, tedy vlastníci, členové výboru, nebo osoby zastupující správce si mohou samostatně spravovat uživatelský účet a jeho vybraná nastavení.

##### **Scénář vytvoření nového uživatelského účtu**

- Člen výboru v prostředí administrace vytváří nového uživatele, kde vyplňuje základní údaje jako jméno, příjmení, e-mail a telefonní číslo.
- Člen výboru ukládá nového uživatele. Spolu s ním vzniká v systému automaticky ze zadaných údajů také záznam v adresáři osob a je s nově vytvořeným uživatelským účtem automaticky provázán. Před vytvořením záznamu v adresáři osob je systémem kontrolováno, že osoba ještě v adresáři neexistuje. Pokud by existovala, systém nabídne do vazby nalezenou existující osobu.
- Systém uživateli automaticky odesílá na zadaný e-mail odkaz pro dokončení registrace. Na provedení registrace má uživatel určitou lhůtu. Po jejím uplynutí je odkaz pro dokončení registrace neplatný a uživatelský účet je automaticky smazán.
- Uživatel použije odkaz a dokončuje registraci zadáním všech potřebných údajů a zadáním nového hesla.

- Uživatel potvrzuje dokončení registrace.
- Uživatelský účet je tím v systému aktivován a uživatel se může do systému přihlásit.
- Systém odesílá členům výboru notifikaci, že uživatel dokončil registraci.
- Systém odesílá uživateli notifikaci o úspěšné registraci.

### **Scénář uzamčení uživatelského účtu**

Scénář je využíván, pokud nastala situace, která vyžaduje uzamknutí uživatelského účtu, což může být například:

- osoba opustila SVJ,
- došlo ke ztrátě přihlašovacích údajů,
- objevilo se podezření na zneužití přihlašovacích údajů,
- došlo k bezpečnostnímu incidentu atd.

Tento scénář je obdobným způsobem použitelný také k odemknutí uživatelského účtu, pokud potřeba uzamknutí profilu pominula a uživatel se bude do systému opět přihlašovat.

- Člen výboru v prostředí administrace vyhledává uživatelský účet.
- Zamyká účet. Tím znemožňuje použití přihlašovacích údajů, ale zachovává všechna data v systému s možností opětovného odemčení.
- Člen výboru opouští prostředí administrace.

### **Scénář smazání uživatelského účtu na žádost uživatele**

- Uživatel vstupuje do prostředí správy svého uživatelského účtu.
- Použije funkci k vytvoření žádosti o smazání uživatelského profilu.
- Během podávání žádosti může nepovinně vyplnit text.
- Uživatel odesílá žádost.
- Systém odesílá uživateli notifikaci o zahájení lhůty, během které může žádost o smazání uživatelského účtu odvolat.
- Systém odesílá členům výboru notifikaci o žádosti uživatele na smazání uživatelského účtu.
- Po uplynutí lhůty systém automaticky maže uživatelský účet.

### **Scénář přerušení smazání uživatelského účtu na žádost uživatele**

- Uživatel vstupuje do prostředí správy svého uživatelského účtu.

- Použije funkci k vytvoření žádosti o smazání uživatelského profilu.
- Během podávání žádosti může nepovinně vyplnit text.
- Uživatel odesílá žádost.
- Systém odesílá uživateli notifikaci o zahájení lhůty, během které může žádost o smazání uživatelského účtu odvolat.
- Systém odesílá členům výboru notifikaci o žádosti uživatele na smazání uživatelského účtu.
- Před uplynutím lhůty pro smazání se uživatel opětovně přihlašuje do svého účtu.
- Systém ukončuje odpočet lhůty do smazání účtu a uživatelský účet je opět aktivní.
- Systém odesílá notifikaci uživateli a členům výboru o zrušení žádosti o smazání účtu.

#### 4.5.1.3 Požadavky k řešení

##### **Aktéři**

- Vlastník
- Člen výboru SVJ, nebo předseda SVJ (dále jen „člen výboru“)
- Správce

##### **Popis případu užití**

Všichni uživatelé, ať už vlastníci, správce, nebo členové výboru, vytvářejí požadavky k řešení. Uživatelé mohou požadavky vytvářet jako neveřejné. To je možné využít k podávání nejrůznějších žádostí. Neveřejné požadavky vidí pouze členové výboru.

Požadavky je možné využít také k podání žádosti o svolání zasedání shromáždění. To je zvláštní typ požadavku, na jehož vytvoření budou všichni členové shromáždění a členové výboru upozorněni a mohou se k žádosti přidat. Statutární orgán SVJ je pak při splnění určitých podmínek povinen zasedání shromáždění svolat. Scénář použití těchto speciálních požadavků je popsán ve scénáři v podkapitole 4.5.1.4.

Uživatelé se mohou podívat na všechny zadané veřejné požadavky a jejich obsah. Vzhledem k tomu by je neměli zadávat duplicitně. Veřejné požadavky zadané duplicitně členové výboru slučují pomocí vazby (související požadavky).

Členové výboru stanovují prioritu řešení každého požadavku a nastavují stavy požadavků. Vlastníci vidí stavy všech požadavků. Požadavky, které nemusí být projednávány shromážděním, řeší členové výboru průběžně mimo shromáždění.

Požadavkům, které podle stanov či svou významností spadají do kompetence shromáždění, členové výboru nastavují stav, který znamená, že bude požadavek projednán na nejbližším zasedání shromáždění.

Členové výboru uzavírají vyřízené požadavky. Uživatelé nemohou otevřít uzavřené požadavky, ale mohou vytvořit nový a na uzavřený se odkázat.

### **Scénář vytvoření a řešení požadavku**

- Osoba disponující uživatelským účtem, ať už vlastník, člen výboru, nebo osoba zastupující správce, se přihlašuje do systému a vstupuje do sekce požadavků.
- Vytváří nový požadavek, vyplňuje předmět, text, přidává přílohy. Ukládá požadavek.
- Systém odesílá členům výboru notifikaci o vytvoření nového požadavku.
- Člen výboru změnou stavu přebírá požadavek do řešení a stává se řešitelem požadavku.
- Potřebuje-li od zadavatele požadavku doplnit nějaké informace, napíše mu do požadavku zprávu, kterou ho o tom informuje a změní stav požadavku.
- Systém odesílá zadavateli notifikaci, že jeho požadavek čeká na doplnění informací.
- Zadavatel do požadavku opět formou zprávy doplní požadované informace.
- Systém automaticky změní stav požadavku a odesílá notifikaci řešiteli požadavku.
- Řešitel požadavku si přebírá změnou stavu požadavek zpět k řešení.
- Po vyřešení požadavku, napíše do požadavku novou zprávu o vyřešení a změní stav požadavku na stav, ve kterém požadavek čeká na potvrzení zadavatele, že s řešením souhlasí. Alternativně může řešitel požadavek rovnou uzavřít a následující bod přeskočit.
- Zadavatel požadavku řešení potvrzuje.
- Řešitel požadavku požadavek uzavírá. Pokud by ho řešitel neuzavřel v určité lhůtě podle nastavení systému řešitel, dojde k jeho uzavření automaticky.



### **Scénář vytvoření a řešení požadavku spadajícího do kompetence shromáždění**

- Osoba disponující uživatelským účtem, ať už vlastník, člen výboru, nebo osoba zastupující správce, se přihlašuje do systému a vstupuje do sekce požadavků.
- Vytváří nový požadavek, vyplňuje předmět, text, přidává přílohy. Ukládá požadavek.
- Systém automaticky odesílá členům výboru zprávu o vytvoření nového požadavku.
- Člen výboru změnou stavu přebírá požadavek do řešení a stává se řešitelem požadavku.
- Zjišťuje, že o řešení požadavku musí rozhodnout shromáždění. Změní tedy stav požadavku na stav, ve kterém požadavek čeká na projednání shromážděním.
- Pro projednání shromážděním systém do požadavku vkládá usnesení a výsledek hlasování.
- Systém řešiteli požadavku notifikaci, že byl projednán.
- Řešitel požadavku si přebírá změnou stavu požadavek zpět k řešení.
- Po vyřešení požadavku, napíše do požadavku novou zprávu o vyřešení a změni stav, ve kterém požadavek čeká na potvrzení zadavatelem, že s řešením souhlasí. Alternativně může řešitel požadavek rovnou uzavřít a následující bod přeskočit.
- Zadavatel požadavku řešení potvrzuje.
- Řešitel požadavku požadavek uzavírá. Pokud by ho řešitel neuzavřel v určité lhůtě podle nastavení systému řešitel, dojde k jeho uzavření automaticky.

#### 4.5.1.4 Zasedání shromáždění

##### **Aktéři**

- Vlastník
- Člen výboru SVJ, nebo předseda SVJ (dále jen „člen výboru“)
- Předseda zasedání shromáždění
- Zapisovatel

## **Svolání zasedání**

Člen výboru vytváří v systému nové zasedání shromáždění. Zadává termín, místo a pořad zasedání. Při vytváření nového zasedání shromáždění člen výboru vidí požadavky, které jsou ve stavu, který znamená, že je potřeba požadavky projednat shromážděním.

Člen výboru vidí systémem navržené datum, kdy bude vlastníkům odeslána pozvánka. Odeslání pozvánky musí proběhnout s určitým předstihem podle stanov nebo platné legislativy. Navržené datum může člen výboru upravit. Ve zvoleném termínu dostávají vlastníci pozvánku.

Nemůže-li se vlastník zasedání shromáždění zúčastnit, dává toto prostřednictvím systému členům výboru na vědomí. Pokud by nastala situace, že by účast na zasedání shromáždění odmítlo příliš mnoho členů shromáždění, mohou na to členové výboru reagovat a datum konání zasedání shromáždění přeplánovat, nebo ho nechat přesto uskutečnit.

## **Scénář svolání zasedání**

- Člen výboru vytváří v systému nové zasedání shromáždění.
- Systém vyhledává požadavky ve stavu k projednání shromážděním a přidává je na pořad.
- Člen výboru zadává termín (datum a čas) konání, místo konání, datum odeslání pozvánky, upravuje pořad zasedání shromáždění a seznam pozvaných, zadává datum odeslání pozvánky a ukládá nové zasedání.
- Systém před uložením validuje vyplnění všech údajů a validuje, že datum odeslání pozvánky je nastaveno s dostatečným časovým odstupem vzhledem k datu konání zasedání shromáždění.
- Ve stanovené datum systém automaticky odesílá pozvánku.

## **Scénář svolání zasedání s přeplánováním termínu**

- Člen výboru vytváří v systému nové zasedání shromáždění.
- Systém vyhledává požadavky ve stavu k projednání shromážděním a přidává je na pořad.
- Člen výboru zadává termín (datum a čas) konání, místo konání, datum odeslání pozvánky, upravuje pořad zasedání shromáždění a seznam pozvaných, zadává datum odeslání pozvánky a ukládá nové zasedání.

- Systém validuje vyplnění všech údajů a validuje, že datum odeslání pozvánky je nastaveno s dostatečným časovým odstupem vzhledem k datu konání zasedání shromáždění.
- Ve stanovené datum systém automaticky odesílá pozvánku.
- Vlastník dostává pozvánku.
- Pokud se nemůže zasedání shromáždění zúčastnit, dává toto prostřednictvím systému členům výboru na vědomí.
- Systém s každou odmítnutou pozvánkou přepočítá poměr hlasů, který vyjadřuje, kolik vlastníků účast na zasedání shromáždění odmítlo. Pokud by nastala situace, že účast na zasedání shromáždění odmítne takové množství vlastníků, že by shromáždění nebylo schopné usnášení, dostává o tom svolavatel a všichni členové výboru informaci.
- Člen výboru může na takovou skutečnost reagovat změnou termínu konání zasedání shromáždění, což povede k tomu, že systém odešle novou pozvánku. Alternativně mohou členové výboru tuto skutečnost ignorovat a nechat zasedání přesto uskutečnit s vědomím toho, že pravděpodobně nebude usnášení schopné.

#### **Scénář svolání zasedání z podnětu vlastníků**

- Osoba, která je členem shromáždění se přihlašuje do systému a vstupuje do sekce požadavků.
- Vytváří nový požadavek, vyplňuje předmět, text, přidává přílohy a na požadavku nastavuje, že jde o žádost o svolání zasedání shromáždění SVJ. Ukládá požadavek.
- Systém odesílá členům výboru a všem členům shromáždění notifikaci o vytvoření nového požadavku, který je žádostí o svolání zasedání shromáždění SVJ.
- Člen shromáždění požadavek otvírá a má možnost se připojit k žádosti.
- Systém na požadavku automaticky vypočítává a zobrazuje poměr hlasů, který o svolání zasedání shromáždění žádá.
- Ve chvíli, kdy jsou splněny podmínky, za kterých je statutární orgán povinen svolat zasedání shromáždění, odesílá systém notifikaci členům výboru a všem členům shromáždění.

- Člen výboru změnou stavu přebírá požadavek do řešení a stává se řešitelem požadavku.
- Člen výboru přímo z požadavku vytváří v systému nové shromáždění.
- Systém vytvořením nového zasedání shromáždění automaticky změni stav požadavku na stav, ve kterém požadavek čeká na projednání shromážděním.
- Pro projednání shromážděním, odešle systém řešiteli požadavku notifikaci, že byl projednán.
- Řešitel požadavku si přebírá změnou stavu požadavek zpět k řešení.
- Po vyřešení požadavku, napíše do požadavku novou zprávu o vyřešení a změni stav, ve kterém požadavek čeká na potvrzení zadavatelem, že s řešením souhlasí. Alternativně může řešitel požadavek rovnou uzavřít a následující bod přeskočit.
- Zadavatel požadavku řešení potvrzuje.
- Řešitel požadavku požadavek uzavírá. Pokud by ho řešitel neuzavřel v určité lhůtě podle nastavení systému řešitel, dojde k jeho uzavření automaticky.

### **Zahájení zasedání**

Člen výboru zahajuje v systému zasedání shromáždění. Členové shromáždění svoji přítomnost stvrzují pomocí systému načtením kódu z obrazovky člena výboru, nebo předsedy zasedání. Člen výboru a později předseda zasedání mají možnost manuálně v systému přidat mezi přítomné členy shromáždění ty osoby, které z nějakého důvodu nemohou svoji prezenci zaznamenat prostřednictvím systému a musí také ze zasedání odebírat členy shromáždění, kteří zasedání v průběhu opustili. Kód pro zaznamenání prezence je dostupný po celou dobu, dokud není zasedání ukončeno.

Člen výboru v systému podle prezence rovnou vidí procento přítomných hlasů a zda může být hlasování usnášení schopné.

### **Scénář zahájení zasedání**

- Člen výboru zahajuje shromáždění.
- Systém generuje QR kód a zobrazuje ho členovi výboru.
- Členové shromáždění postupně pomocí svých zařízení načítají generovaný kód a stvrzují tím svoji přítomnost na zasedání.
- Člen výboru v systému navíc potvrzuje přítomnost každého člena na zasedání.

- Manuálně také přidává členy shromáždění, kteří z nějakého důvodu nemohou svoji přítomnost potvrdit pomocí svého zařízení.
- Systém vyhodnocuje a zobrazuje podíl přítomných hlasů a schopnost shromáždění se přijímat usnesení.

### **Hlasování během zasedání**

Pokud je shromáždění usnášení schopné, probíhá pomocí systému hlasování o volbě předsedy zasedání, nebo dalších osob, je-li to stanovami vyžadováno.

Zapisovatel do systému zapisuje návrh formulace usnesení a procentní hranici nutnou pro přijetí usnesení podle stanov. Pak zahajuje hlasování. Vlastníci v systému hlasují. Zapisovatel manuálně zadává hlasy těch, kteří nemohou hlasovat na svém zařízení, nebo za které hlasuje zprostředkovaně jiný vlastník (skrže plnou moc). Hlasování je automaticky ukončeno, pokud hlasovali všichni přítomní vlastníci. Předseda manuálně ukončuje hlasování například pokud nehlasovali všichni. Účastníci po skončení hlasování vidí výsledek hlasování.

### **Scénář hlasování o předsedovi zasedání**

- Člen výboru v systému k zahájenému zasedání shromáždění přidává nové usnesení. Nastavuje typ usnesení, kde vybírá, že jde o volbu předsedy zasedání a nastavuje osobu navrženou na roli předsedy zasedání. Typem usnesení je dán také minimální procento hlasů pro přijetí.
- Systém vyhodnocuje, zda je pro nastavený typ usnesení přítomný dostatečný počet hlasů. Pro některé typy usnesení může být nutné například 75 %, nebo dokonce 100 % všech hlasů. Pokud by na zasedání nebyl přítomný dostatek hlasů, systém to oznámí.
- Člen výboru zahajuje hlasování.
- Členové shromáždění v systému hlasují.
- Člen výboru manuálně zadává hlasy těch, kteří nemohou hlasovat na svém zařízení, nebo za které hlasuje zprostředkovaně jiný vlastník (skrže plnou moc).
- Systém automaticky ukončuje hlasování, pokud hlasovali všichni přítomní členové shromáždění. Člen výboru manuálně ukončuje hlasování například pokud nehlasovali všichni.
- Systém po skončení hlasování všem zobrazuje výsledek.

- Systém automaticky přidává usnesení a výsledek hlasování do zápisu ze zasedání shromáždění.
- Pokud bylo usnesení o osobě předsedy zasedání shromáždění přijato, člen výboru v systému potvrzuje, že zvolenému předsedovi zasedání předává další řízení zasedání.

Podobným způsobem, jaký je popsán v předchozím scénáři, je možné hlasovat také o zapisovateli, nebo jiných rolích, pokud to stanovy SVJ vyžadují.

Pro následující scénář je předpokládáno, že byl zvolen předseda zasedání a zapisovatel.

### **Scénář hlasování v průběhu zasedání shromáždění**

- Zapisovatel v systému do zápisu k zahájenému zasedání přidává nové usnesení. Nastavuje odpovídající typ usnesení. Typem usnesení je dán minimální procento hlasů pro přijetí. Zapisuje formulaci usnesení.
- Systém vyhodnocuje, zda je pro nastavený typ usnesení přítomný dostatečný počet hlasů. Pro některé typy usnesení může být nutné například 75 %, nebo dokonce 100 % všech hlasů. Pokud by na zasedání nebyl přítomný dostatek hlasů, systém to oznámí.
- Zapisovatel na pokyn předsedy zasedání zahajuje hlasování.
- Členové shromáždění v systému hlasují.
- Zapisovatel manuálně zadává hlasy těch, kteří nemůžou hlasovat na svém zařízení, nebo za které hlasuje zprostředkovaně jiný vlastník (skrže plnou moc).
- Systém automaticky ukončuje hlasování, pokud hlasovali všichni přítomní členové shromáždění. Zapisovatel na pokyn předsedy zasedání manuálně ukončuje hlasování například pokud nehlasovali všichni.
- Systém po skončení hlasování všem zobrazuje výsledek hlasování.
- Systém automaticky přidává výsledek hlasování do zápisu ze zasedání shromáždění.

### **Ukončení zasedání shromáždění**

Z průběhu zasedání shromáždění automaticky vzniká kompletní zápis tím, jak jsou do systému během zasedání přidávány návrhy usnesení a výsledky hlasování. Předseda

zasedání v systému ukončuje zasedání. Zapisovatel zápis dokončuje a označuje jako kompletní. Předseda zasedání zápis sám doplňuje, pokud je potřeba, nebo jej vrací k doplnění zapisovateli, nebo jej schvaluje a odesílá účastníkům. Vlastníci dostávají zápis ze zasedání shromáždění.

Následující scénáře počítají s tím, že na zasedání je nastaveno automatické odeslání zápisu. Člen výboru, který v systému zasedání vytvářel, má možnost nastavit, že zápis má být odeslán ručně. V takovém případě by bylo ještě odeslání zápisu ze systému ručně potvrdit.

#### **Scénář ukončení zasedání shromáždění**

- Předseda zasedání v systému ukončuje zasedání shromáždění.
- Zapisovatel má k dispozici zápis ze zasedání, který během zasedání průběžně vzniká. Tento zápis upravuje a označuje jako dokončený.
- Systém odesílá notifikaci předsedovi zasedání, kterou ho upozorňuje na skutečnost, že zápis byl zapisovatelem dokončen a čeká na schválení.
- Předseda schvaluje zápis.
- Systém odesílá zápis všem členům SVJ. Odesláním zápisu je zasedání shromáždění uzavřeno.

#### **Scénář ukončení zasedání s vrácením zápisu k doplnění**

- Předseda zasedání v systému ukončuje zasedání shromáždění.
- Zapisovatel má k dispozici zápis ze zasedání, který během zasedání průběžně vzniká. Tento zápis upravuje a označuje jako dokončený.
- Systém odesílá notifikaci předsedovi zasedání, kterou ho upozorňuje na skutečnost, že zápis byl zapisovatelem dokončen a čeká na schválení.
- Předseda vrací zápis zapisovateli k doplnění.
- Zapisovatel provádí požadované úpravy v zápisu a po dokončení úprav ho označuje jako dokončený
- Systém odesílá notifikaci předsedovi zasedání, kterou ho upozorňuje na skutečnost, že zápis byl zapisovatelem dokončen a čeká na schválení.
- Předseda schvaluje zápis.
- Systém odesílá zápis všem členům SVJ. Odesláním zápisu je zasedání shromáždění uzavřeno.

## **Hlasování mimo zasedání shromáždění**

K uskutečnění hlasování mimo zasedání shromáždění se dají využít, za splnění podmínek definovaných platnou legislativou a stanovami SVJ, podobné mechanismy, které byly popsány v předchozích scénářích, ale ve zjednodušené podobě. Níže je popsán scénář hlasování mimo zasedání shromáždění.

### **Scénář hlasování mimo zasedání shromáždění**

- Člen výboru vytváří v systému nové zasedání shromáždění.
- Člen výboru označí vytvářené zasedání jako hlasování mimo zasedání shromáždění.
- Zadává termíny (data a časy) zahájení a ukončení. Ty představují interval, v rámci kterého bude možné skrze systém vzdáleně hlasovat.
- Zapisuje návrhy usnesení, o kterých bude možné hlasovat.
- Nastavuje datum pozvánky, což v tomto případě představuje datum, kdy bude členům shromáždění odesláno oznámení o hlasování mimo zasedání shromáždění.
- Člen výboru ukládá nový záznam o zasedání.
- System před uložením validuje vyplnění všech údajů.
- V den nastavený jako datum pozvánky odesílá systém členům shromáždění oznámení o nadcházejícím hlasování mimo zasedání shromáždění. Oznámení obsahuje plné znění návrhů všech usnesení, o kterých bude hlasováno.
- V den a čas zahájení hlasování odesílá systém členům shromáždění notifikaci, že hlasování je přístupné.
- Členové shromáždění v systému vzdáleně hlasují o navržených usneseních.
- Po provedení hlasování odesílá systém členovi shromáždění dokument, který obsahuje sestavu hlasovacího lístku s hodnotami jeho hlasování.
- Člen shromáždění tento dokument tiskne, vlastnoručně podepisuje a odesílá na adresu výboru SVJ.
- Po obdržení všech hlasovacích lístků potvrzuje člen výboru v systému všechny platné hlasy.
- Člen výboru odesílá zápis obsahující všechny údaje vč. výsledků. Odesláním zápisu je záznam o zasedání shromáždění uzavřen.



#### 4.5.1.5 Komunikace mezi členy SVJ

##### **Aktéři**

- Vlastník
- Člen výboru SVJ, nebo předseda SVJ (dále jen „člen výboru“)
- Správce

##### **Kontakty**

Vlastník v administraci svého profilu povinně nastavuje své kontakty a jejich viditelnost pro ostatní. Vlastník může své kontakty nastavit jako viditelné pro ostatní vlastníky, nebo jen pro členy výboru. Členové výboru a správce mají kontakty automaticky nastavené jako viditelné pro všechny členy SVJ a nemají možnost je skrýt. Vlastníci, kteří budou mít své kontakty nastaveny jako skryté mít stále možnost přijímat soukromé zprávy, pouze nebude pro ostatní zveřejněna jejich telefonní čísla a e-maily atd.

##### **Aktuality**

Člen výboru, vlastník, nebo osoba zastupující správce, která potřebuje o něčem hromadně informovat všechny členy SVJ a všechny uživatele systému, vytváří novou aktualitu, což je příspěvek viditelný pro všechny. Člen SVJ, ať už je vlastník, nebo člen výboru může být na novou aktualitu upozorněn v závislosti na svém nastavení notifikací. Uživatelé vidí všechny aktuality hned po přihlášení.

##### **Scénář vytvoření aktuality členem výboru**

- Člen výboru vytváří a ukládá novou aktualitu.
- Systém vyhodnocuje nutnost schválení zveřejnění nové aktuality. V případě, že aktualitu vytváří člen výboru, není schválení nutné.
- Systém zveřejňuje aktualitu v sekci aktualit.
- Systém odesílá notifikaci o nové aktivitě uživatelům, kteří to mají v nastavení svého uživatelského účtu povolené.

##### **Scénář vytvoření aktuality jiným uživatelem**

- Vlastník, nebo osoba zastupující správce, případně jiný uživatel, který není členem výboru (dále v tomto scénáři jen „uživatel“) a má v administraci systému povoleno přidávat nové aktuality, vytváří a ukládá novou aktualitu.

- Systém vyhodnocuje nutnost schválení zveřejnění nové aktuality. V případě tohoto scénáře je systémem vyhodnoceno, že zveřejnění aktuality musí schválit člen výboru.
- Systém odesílá notifikace členům výboru o nové aktualitě, která čeká na zveřejnění.
- Člen výboru schvaluje zveřejnění aktuality.
- Systém zveřejňuje aktualitu v sekci aktualit.
- Systém odesílá notifikaci o nové aktualitě uživatelům, kteří to mají v nastavení svého uživatelského účtu povolené.

### **Soukromé zprávy**

Uživatel, ať už vlastník, správce, nebo člen výboru, který chce soukromě něco sdělit jinému uživateli, může poslat jednomu či více jiným uživatelům soukromou zprávu. Příjemce zprávy může být na příchozí zprávu upozorněn v závislosti na svém nastavení notifikací. Na soukromou zprávu může odpovědět, ignorovat ji, nebo ji smazat. Příchozí zprávy nalézá uživatel uspořádané chronologicky do konverzací na jednom místě. Soukromé zprávy budou mohou vlastníci posílat i těm členům, kteří mají své kontakty v profilu nastaveny jako skryté.

### **Veřejné diskuse**

Uživatel, ať už je vlastník, správce, nebo člen výboru, který potřebuje oslovit ostatní uživatele, vytváří příspěvek, který je viditelný pro všechny. Od soukromých zpráv se liší tím, že zprávy ostatních uživatelů, kteří na původní zprávu reagují, jsou opět viditelné pro všechny a jsou chronologicky členěny do vlákna. Jde v podstatě o fórum, kde je možné veřejně řešit vhodná témata. Fórum je možné využít například pro nabídku a poptávku sousedské pomoci. Někdo z vlastníků může být například lékař, inženýr, automechanik, zahradník, instalatér, profesor atp. a může nabídnout svoje služby v rámci sousedské výpomoci prostřednictvím fóra ostatním vlastníků. Naopak ostatní vlastníci mohou zase poptávat prostřednictvím fóra pomoc, nebo sdílet informace atd. Může jít i o domlouvání hromadných objednávek různých služeb či zboží atd.

#### 4.5.1.6 Bilance

### **Aktéři**

- Vlastník

- Člen výboru SVJ, nebo předseda SVJ (dále jen „člen výboru“)
- Správce

### **Popis**

Vlastník prohlíží údaje o platbách, které SVJ uhradil včetně historie, údaje o aktuální blížící se platbě včetně rozpadu na jednotlivé položky, veškerá vyúčtování, stavy účtů SVJ (fond oprav, běžný účet, úvěrový účet, spořicí účet).

Data jsou do systému importována automatizovaně, nebo správcem, nebo členem výboru.

#### 4.5.1.7 Ukládání a čtení dokumentů

### **Aktéři**

- Vlastník
- Člen výboru SVJ, nebo předseda SVJ (dále jen „člen výboru“)
- Správce
- Dodavatel

### **Popis případu užití**

Osoba s oprávněním přidávat dokumenty (členové výboru automaticky, ostatní osoby po schválení členem výboru, pokud to mají povolené v administraci) ukládá dokumenty do vybraných sekcí podle následujícího členění:

- Základní dokumentace SVJ
- Technická dokumentace
- Smluvní dokumentace
- Dokumenty ke shromáždění
- Evidenční listy
- Ostatní dokumenty

Uživatelé prohlíží v jednotlivých sekcích uložené dokumenty.

### **Scénář vložení nového dokumentu členem výboru**

- Člen výboru vstupuje do sekce dokumentů, vytváří nový dokument, vkládá potřebné údaje, vybírá sekci dokumentu. V závislosti na nastavení sekce dokumentů vybírá, nebo nevybírá vkládající člen výboru okruh uživatelů, kteří dokument v systému uvidí. V případě některých sekcí bude dokument

viditelný pro všechny uživatele. V případě některých sekcí je nutné vybrat okruh uživatelů, nebo konkrétní uživatele, pro které bude dokument viditelný.

- Člen výboru ukládá nový dokument.
- Systém provádí validace zadaných údajů.
- Systém vyhodnocuje nutnost schválení zveřejnění nového dokumentu. V případě, že dokument vkládá člen výboru, není schválení nutné.
- Člen výboru odesílá dokument ke zveřejnění.
- Systém zveřejňuje dokument v příslušné sekci dokumentů.
- Systém odesílá notifikaci o novém dokumentu uživatelům, pro které byl zveřejněn, pokud to mají v nastavení svého uživatelského účtu povolené.

### **Scénář vložení nového dokumentu jiným uživatelem**

- Vlastník, nebo osoba zastupující správce, případně jiný uživatel, který není členem výboru (dále v tomto scénáři jen „uživatel“) vstupuje do sekce dokumentů a pokud má v administraci systému povoleno přidávat nové dokumenty, použije funkci pro přidání nového dokumentu.
- Vkládá potřebné údaje, vybírá sekci dokumentu. V závislosti na nastavení sekce dokumentů vybírá, nebo nevybírá vkládající uživatel okruh uživatelů, kteří dokument v systému uvidí. V případě některých sekcí bude dokument viditelný pro všechny uživatele. V případě některých sekcí je nutné vybrat okruh uživatelů, nebo konkrétní uživatele, pro které bude dokument viditelný.
- Uživatel ukládá nový dokument.
- Systém provádí validace zadaných údajů.
- Systém vyhodnocuje nutnost schválení zveřejnění nového dokumentu. V případě tohoto scénáře je systémem vyhodnoceno, že zveřejnění dokumentu musí schválit člen výboru.
- Uživatel odesílá dokument ke zveřejnění.
- Systém odesílá notifikace členům výboru o novém dokumentu, který čeká na zveřejnění.
- Člen výboru schvaluje zveřejnění čekajícího dokumentu.
- Systém zveřejňuje dokument v příslušné sekci dokumentů.
- Systém odesílá notifikaci o schválení zveřejnění dokumentu uživateli, který dokument do systému vložil.

- Systém odesílá notifikaci o novém dokumentu uživatelům, pro které byl zveřejněn, pokud to mají v nastavení svého uživatelského účtu povolené.

#### 4.5.1.8 Rezervace společných částí

##### **Aktéři**

- Vlastník
- Člen výboru SVJ, nebo předseda SVJ (dále jen „člen výboru“)
- Schvalovatel rezervací společné části
- Správce společné části

##### **Popis**

Vlastník si může rezervovat společné části, které má právo užívat, které rezervace podporují, a jejichž rezervování je to vhodné, vyžadované, nebo kde systém rezervací může podpořit komunitní bydlení a harmonické soužití obyvatel.

Může jít například o zahradní pec, gril, bazén, saunu, sportoviště, konferenční (školící) místnost, společenskou místnost, salonek atd. Některé společné části je možné používat i více vlastníky zároveň a rezervace zde můžou sloužit jako nástroj pro organizaci společných událostí. Některé společné části hromadně použitelné být nemusí.

Člen výboru do rezervací vstupuje v roli administrátora, který v systému vytváří a nastavuje společné části. Nastavuje, jestli je u nich vyžadováno schválení, případně za jakých podmínek. U některých společných částí může být vyžadováno schvalování rezervací až od nějakého počtu rezervací. Některé části mohou být rezervovatelné jen po přechodné období. Člen výboru při nastavování každé společné části vybírá, kdo je jejím správcem, a kdo je jejím schvalovatelem rezervací.

Následující scénář popisuje vytvoření rezervace společné části vlastníkem s vyžadovaným schválením.

##### **Scénář rezervace společné části s vyžadovaným schválením**

- Vlastník v systému vstupuje do sekce společných částí.
- Vybírá konkrétní společnou část.
- Systém zobrazuje všechny základní informace o společné části vč. dostupnosti a existujících rezervacích a požadavcích v řešení.
- Vlastník vytváří novou rezervaci.

- Systém potvrzuje vlastníkovvi vytvoření nové rezervace.
- Systém odesílá notifikaci schvalovateli o nové rezervaci čekající na schválení.
- Schvalovatel schvaluje rezervaci.
- Systém odesílá notifikaci vlastníkovvi o schválení jeho rezervace.

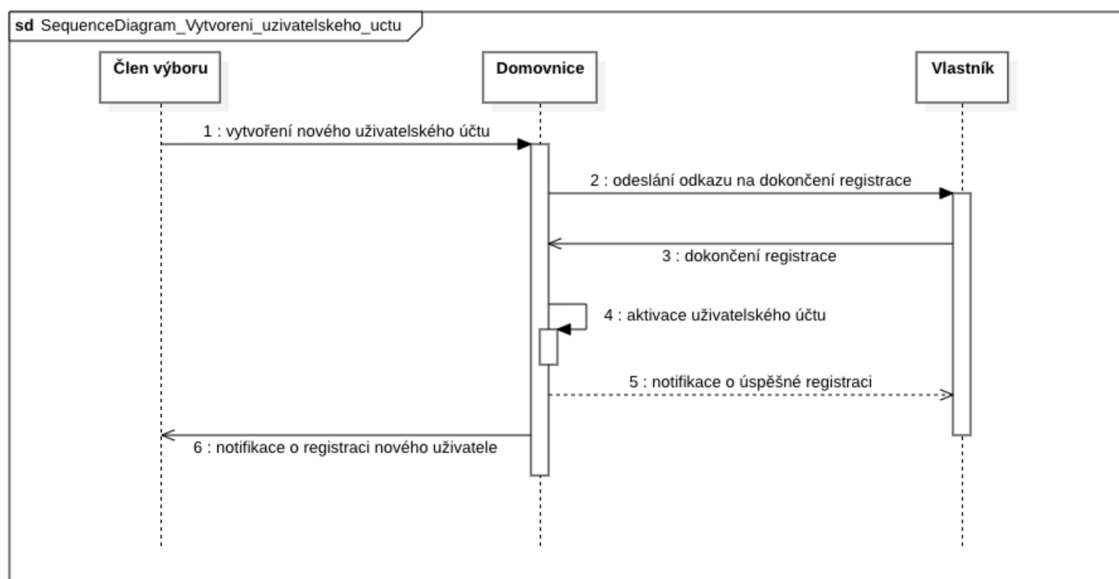
Následující scénář popisuje nahlášení závady správci vybrané společné části. Ve scénáři je jako zadavatel uvažován vlastník, ale zadavatelem může být i jiná osoba než vlastník. Scénář souvisí s případem užití Požadavky k řešení (4.5.1.3).

#### **Scénář nahlášení závady správci společné části**

- Vlastník v systému vstupuje do sekce společných částí.
- Vybírá konkrétní společnou část.
- Systém zobrazuje všechny základní informace o společné části vč. dostupnosti a existujících rezervacích a požadavcích v řešení.
- Vlastník použije funkci k vytvoření nového požadavku k vybrané společné části.
- Vytváří nový požadavek, vyplňuje předmět, text, přidává přílohy. Ukládá požadavek.
- Systém potvrzuje vytvoření nového požadavku.
- Systém odesílá správci společné části notifikaci o vytvoření nového požadavku k řešení.
- Další průběh řešení požadavku se může lišit. Některé možnosti jsou popsány v podkapitole 4.5.1.3 a znázorněny pomocí diagramů v podkapitolách 4.4 a 4.5.2.

## 4.5.2 Sekvenční diagramy

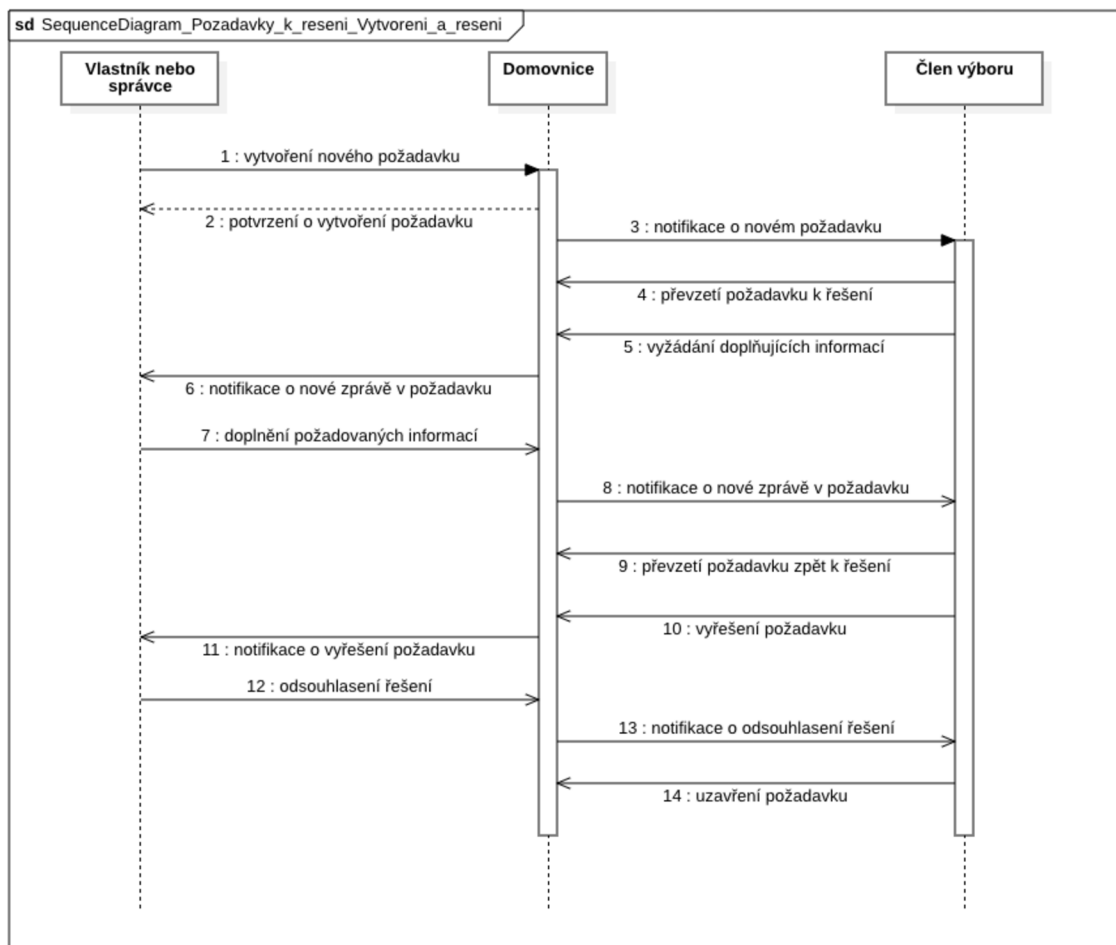
### Vytvoření a správa uživatelského účtu



Obrázek 26: sekv. d. vytvoření nového uživatelského účtu; zdroj: vlastní zpracování

Sekvenční diagram znázorňuje scénář případu užití vytvoření nového uživatelského účtu, který je popsán v 4.5.1.2.

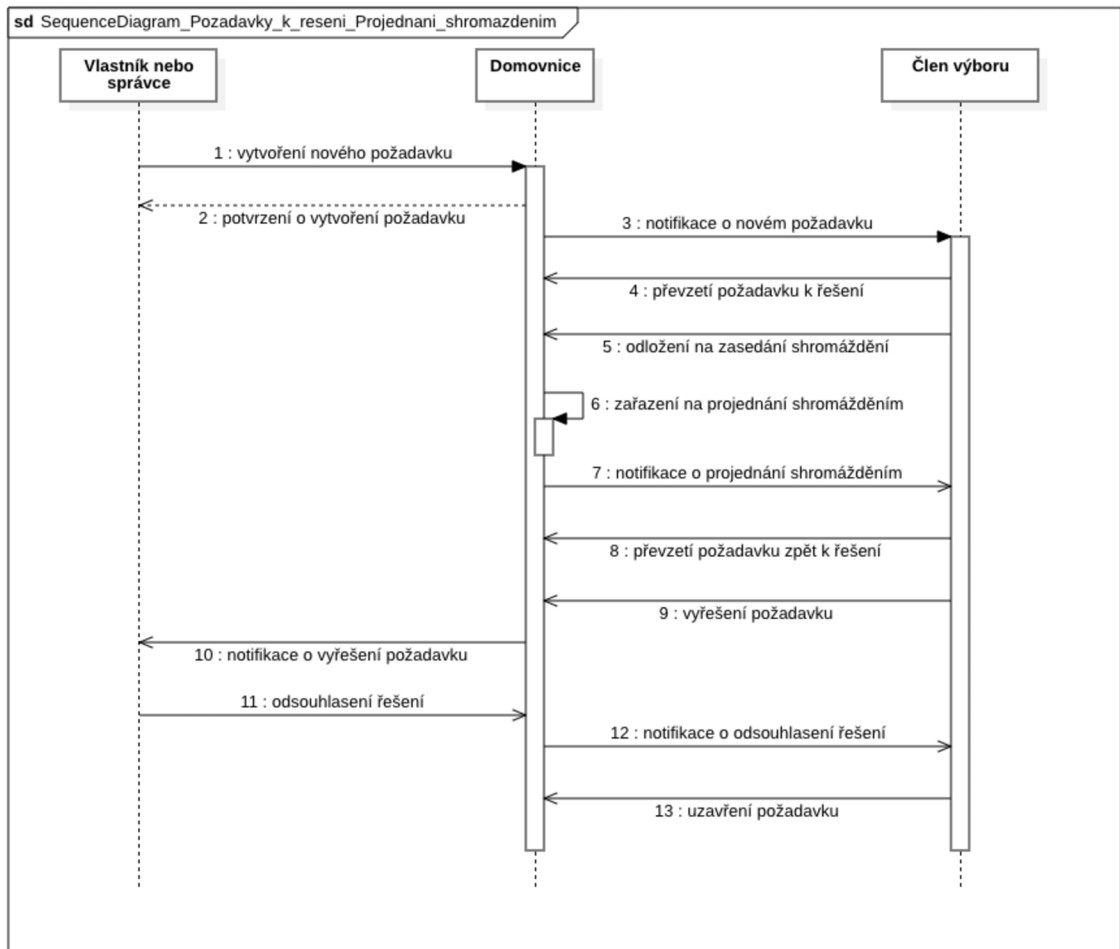
## Požadavky k řešení



Obrázek 27: sekv. d. vytv. a řešení požadavku; zdroj: vlastní zpracování

Sekvenční diagram znázorňuje scénář vytvoření a řešení požadavku, který je popsán v 4.5.1.3.

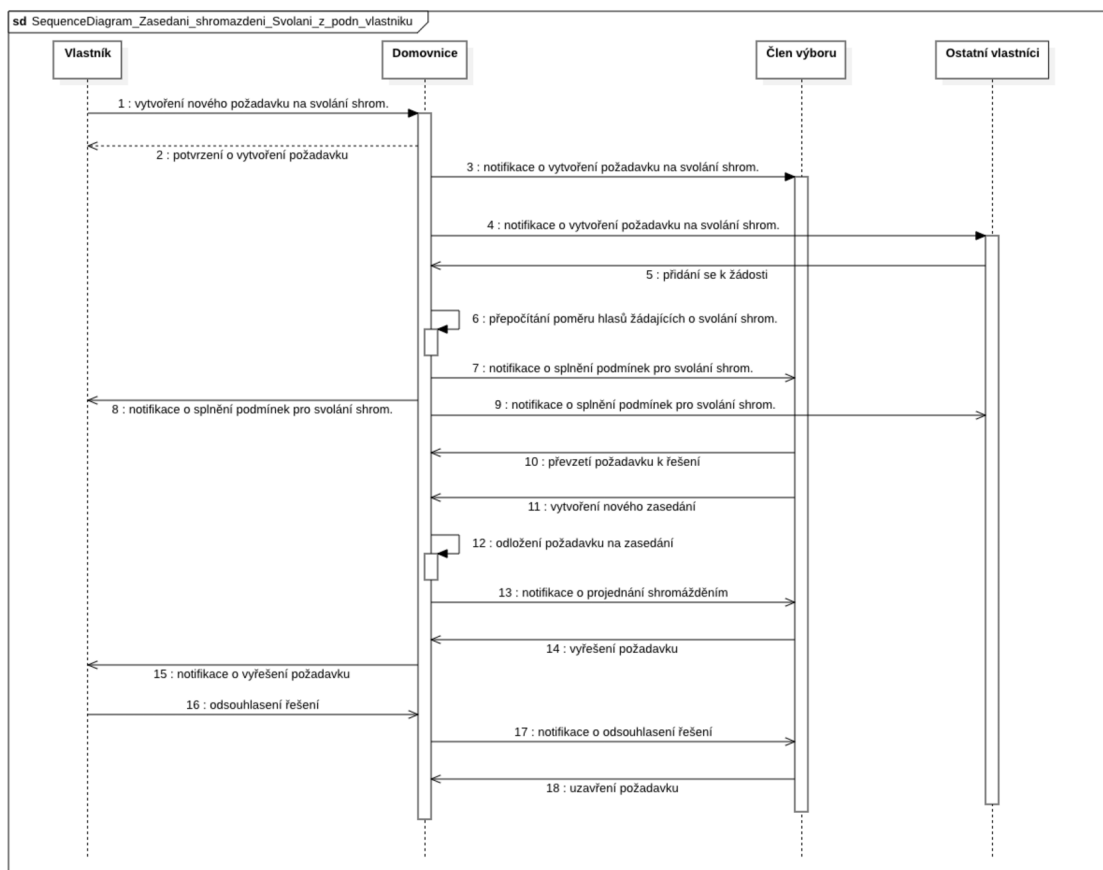




Obrázek 28: sekv. d. vytv. a řešení požad. shromážděním; zdroj: vlastní zpracování

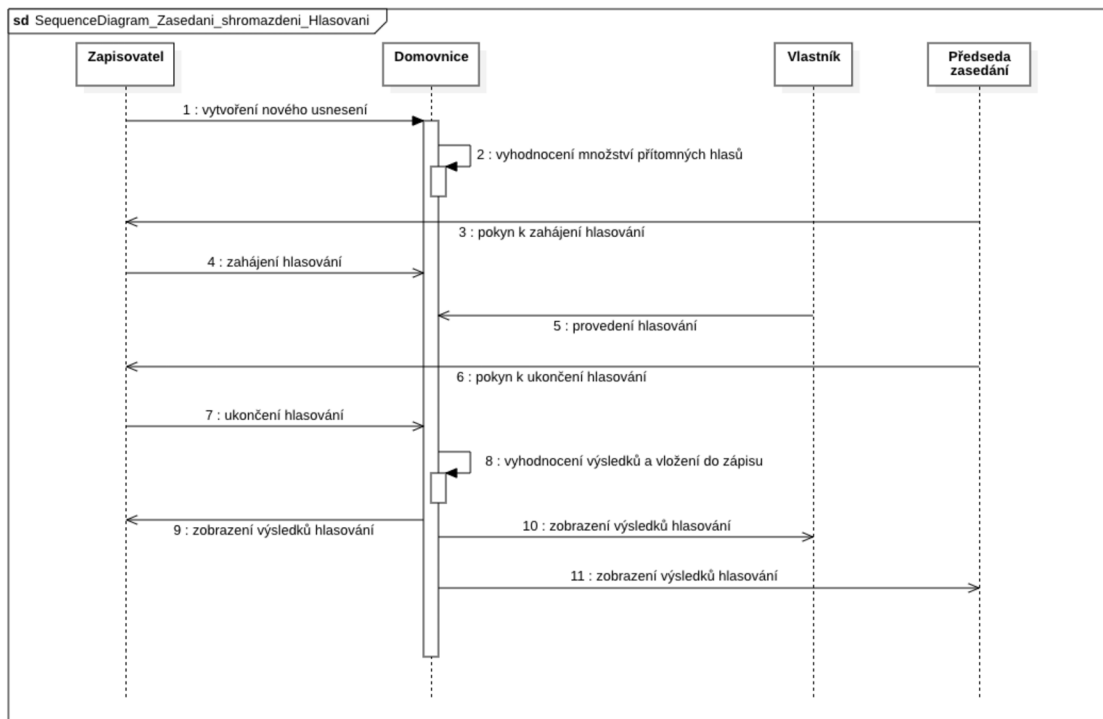
Sekvenční diagram znázorňuje scénář vytvoření a řešení požadavku spadajícího do kompetence shromáždění, který je popsán v 4.5.1.3.





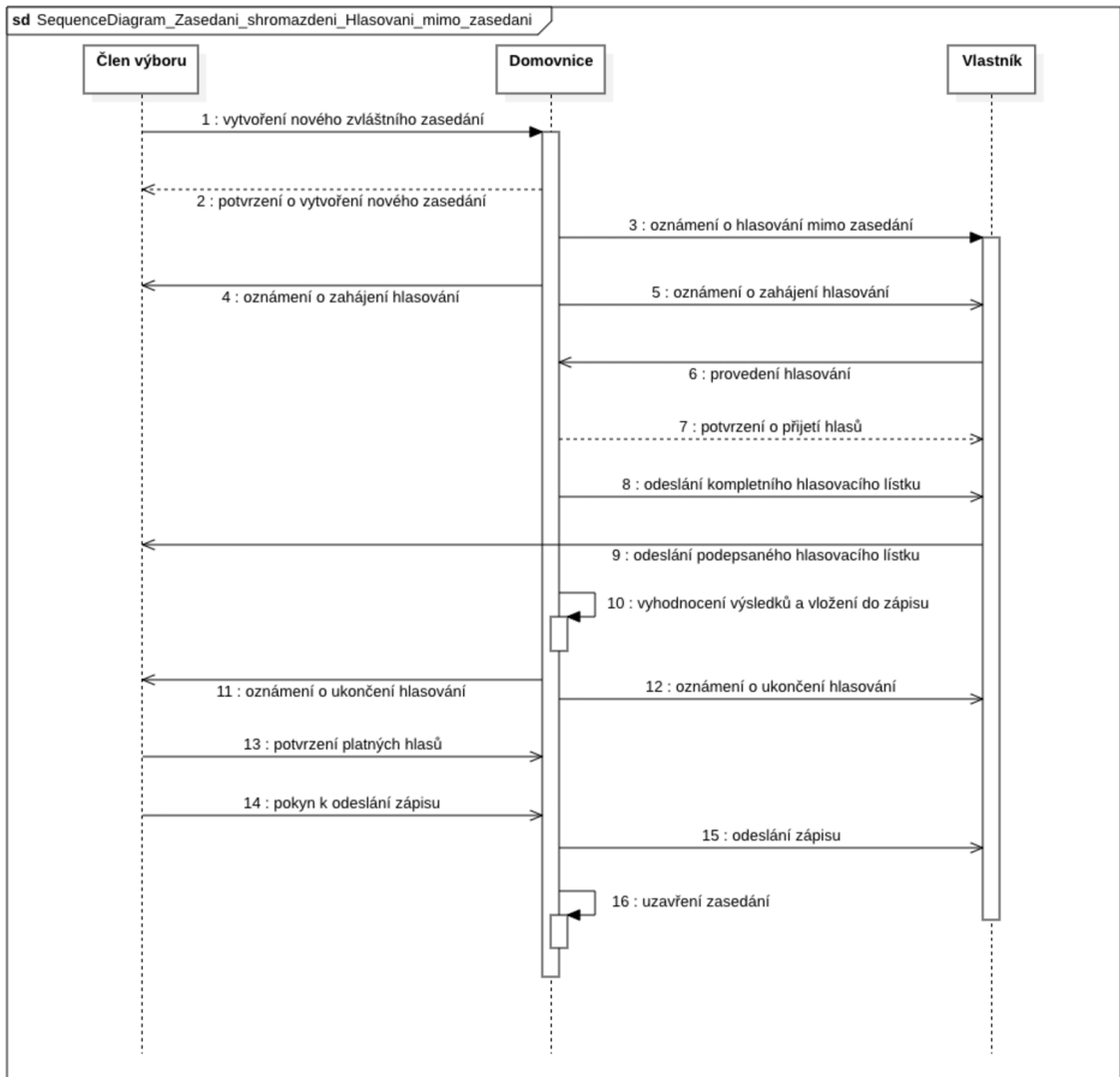
Obrázek 30: sekv. d. svolání zasedání z podnětu vlastníků; zdroj: vlastní zpracování

Sekvenční diagram znázorňuje scénář svolání zasedání z podnětu vlastníků, který je popsán v podkapitole 4.5.1.4. Čáry života Vlastník a Ostatní vlastníci znázorňují objekty stejné třídy a takto odděleně jsou v diagramu znázorněny z důvodu co nejlepšího pochopení jejich vzájemných interakcí v systému.



Obrázek 31: sekv. d. hlasování v průběhu zasedání; zdroj: vlastní zpracování

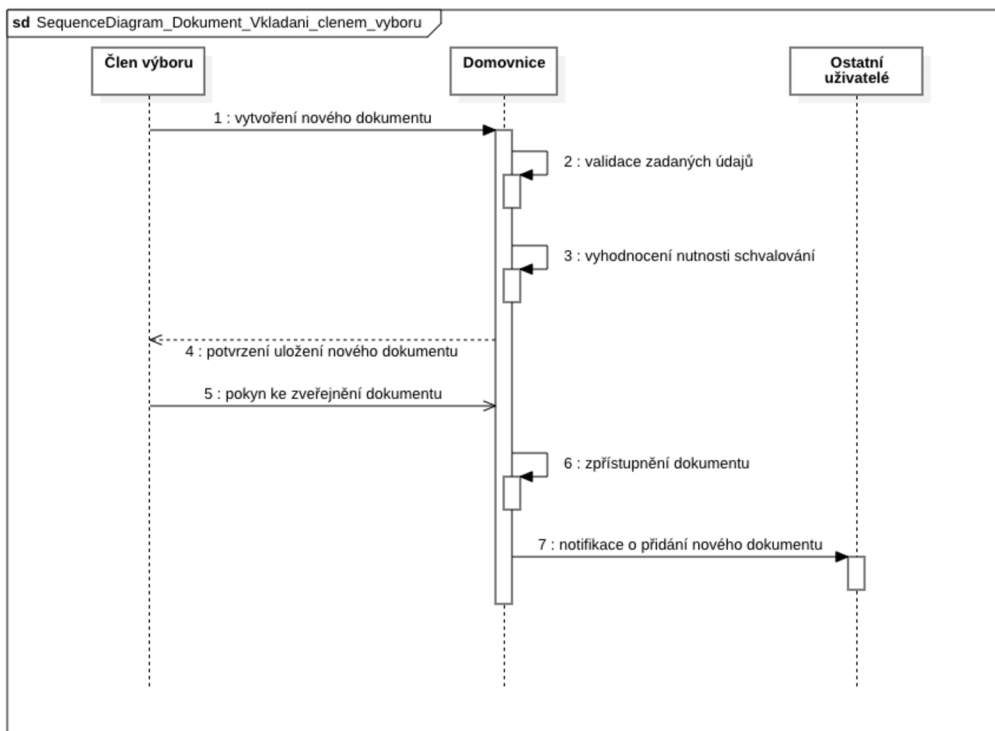
Sekvenční diagram znázorňuje scénář hlasování v průběhu zasedání shromáždění, který je popsán v 4.5.1.4. Tento diagram obsahuje i zprávy (zprávy č. 3 a č. 6), které mohou, ale nemusí, proběhnout přímo skrze systém. Jsou zde znázorněny pro lepší pochopení interakcí mezi samotnými aktéry.



Obrázek 32: sekv. d. hlasování mimo zasedání shromáždění; zdroj: vlastní zpracování

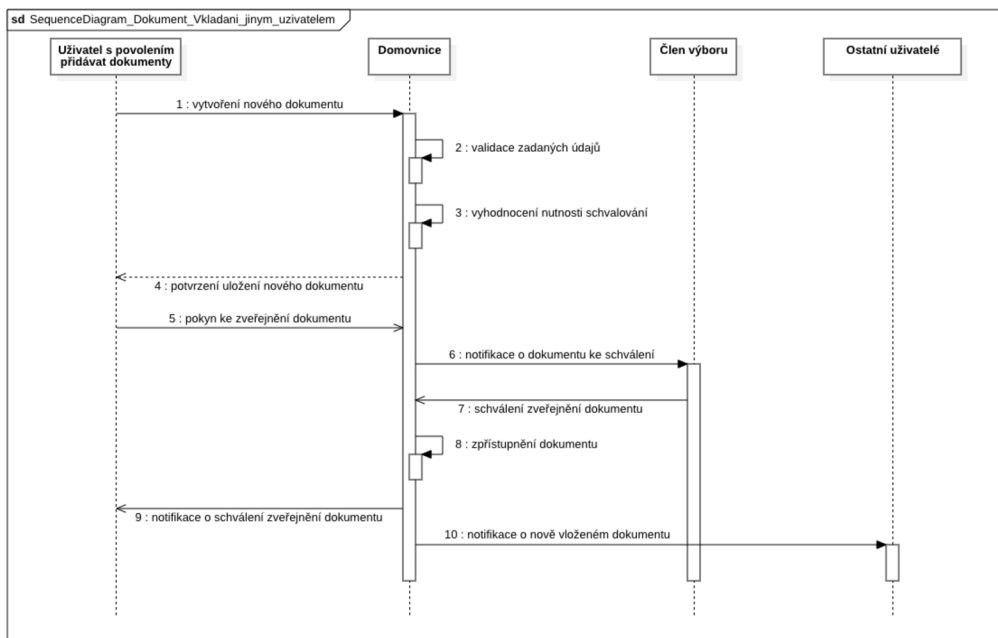
Sekvenční diagram znázorňuje scénář hlasování mimo zasedání shromáždění, který je popsán v 4.5.1.4. Tento diagram obsahuje i interakci (zpráva č. 9), která nebude probíhat přímo skrze systém. V diagramu je zahrnuta pro lepší pochopení interakcí mezi aktéry.

## Ukládání a čtení dokumentů



Obrázek 33: sekv. d. ukládání dokumentů členem výboru; zdroj: vlastní zpracování

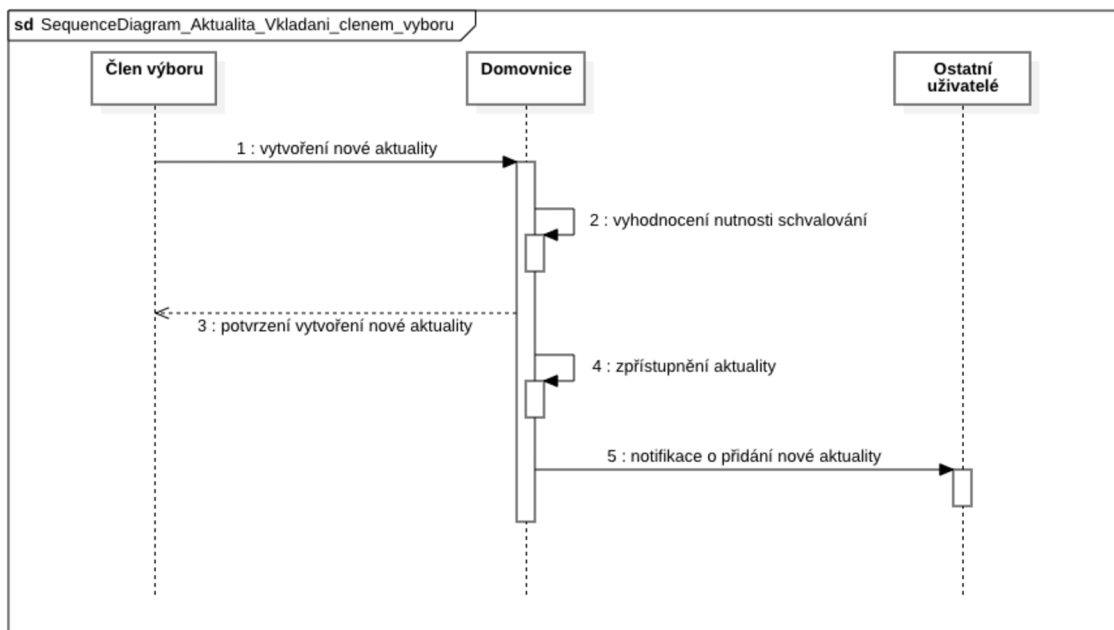
Sekvenční diagram (Obrázek 33) znázorňuje scénář vložení nového dokumentu členem výboru, který je popsán v 4.5.1.7.



Obrázek 34: sekv. d. ukládání dokumentu jiným uživatelem; zdroj: vlastní zpracování

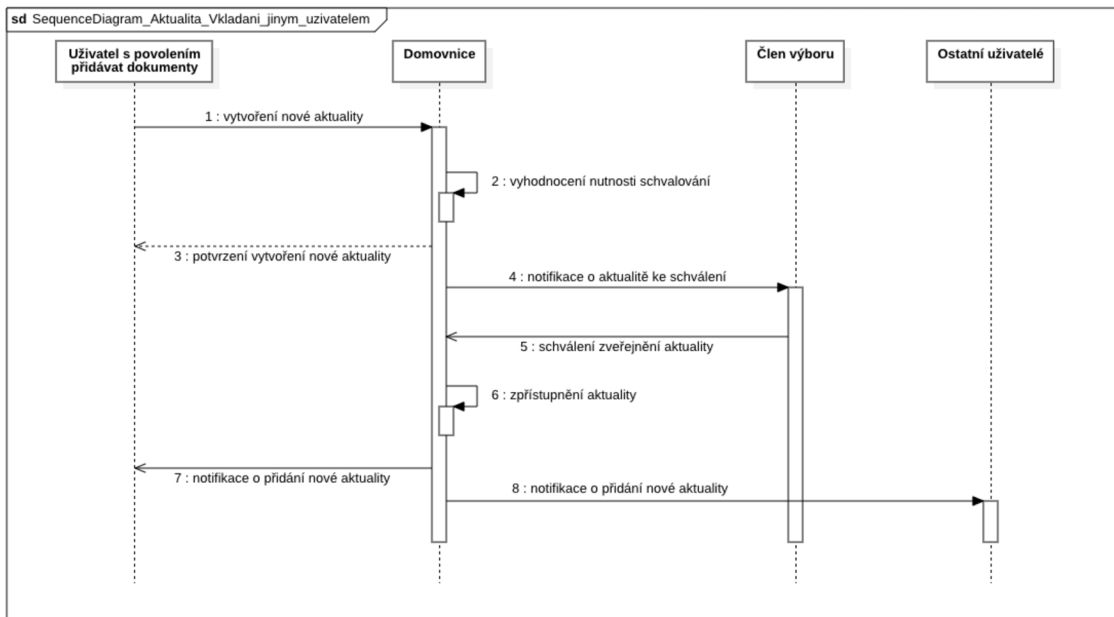
Sekvenční diagram (Obrázek 34) znázorňuje scénář vložení nového dokumentu jiným uživatelem, který je popsán v 4.5.1.7.

## Aktuality



Obrázek 35: sekv. d. vytvoření aktuality členem výboru; zdroj: vlastní zpracování

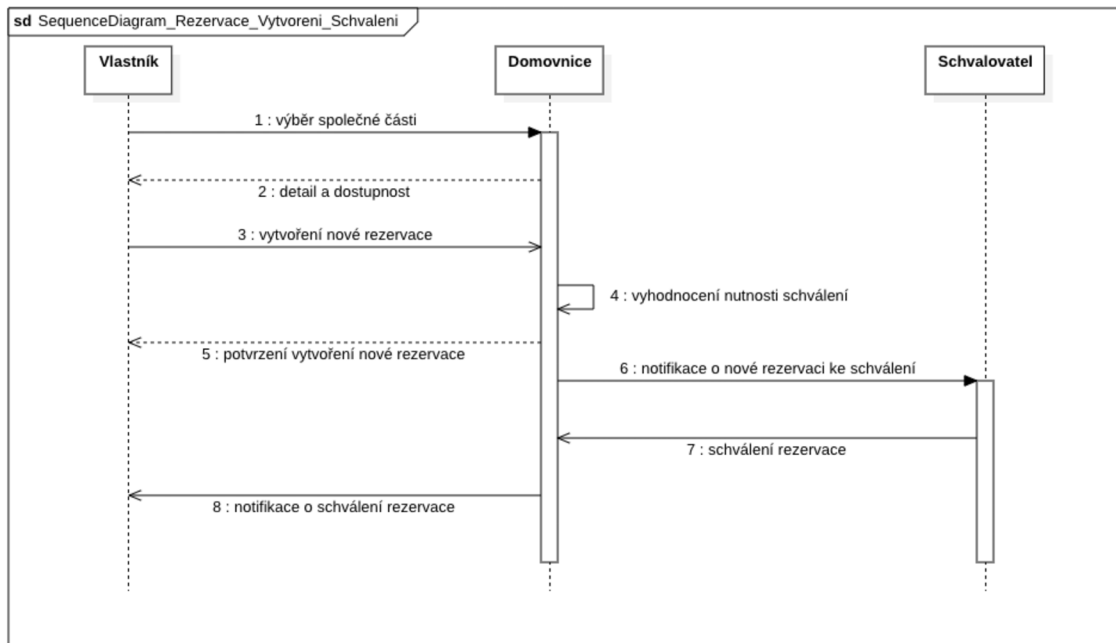
Sekvenční diagram (Obrázek 35) znázorňuje scénář vytvoření aktuality členem výboru, který je popsán v 4.5.1.5.



Obrázek 36: sekv. d. vytvoření aktuality jiným uživatelem; zdroj: vlastní zpracování

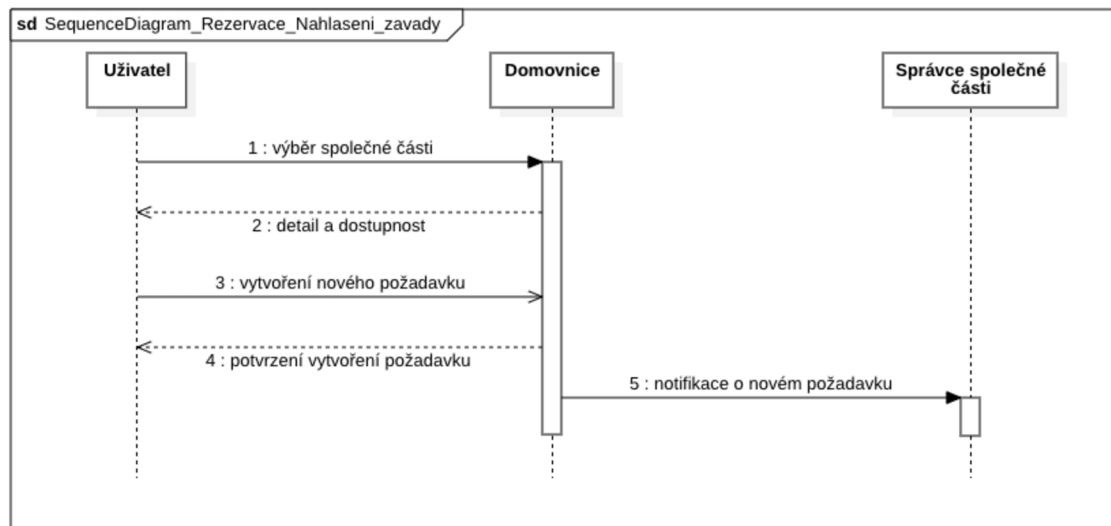
Sekvenční diagram (Obrázek 36) znázorňuje scénář vytvoření aktuality jiným uživatelem, který je popsán v 4.5.1.5.

## Rezervace společných částí



Obrázek 37: sekv. d. vytvoření rezervace se schválením; zdroj: vlastní zpracování

Sekvenční diagram (Obrázek 37) znázorňuje scénář rezervace společné části s vyžadovaným schválením, který je popsán v 4.5.1.8.



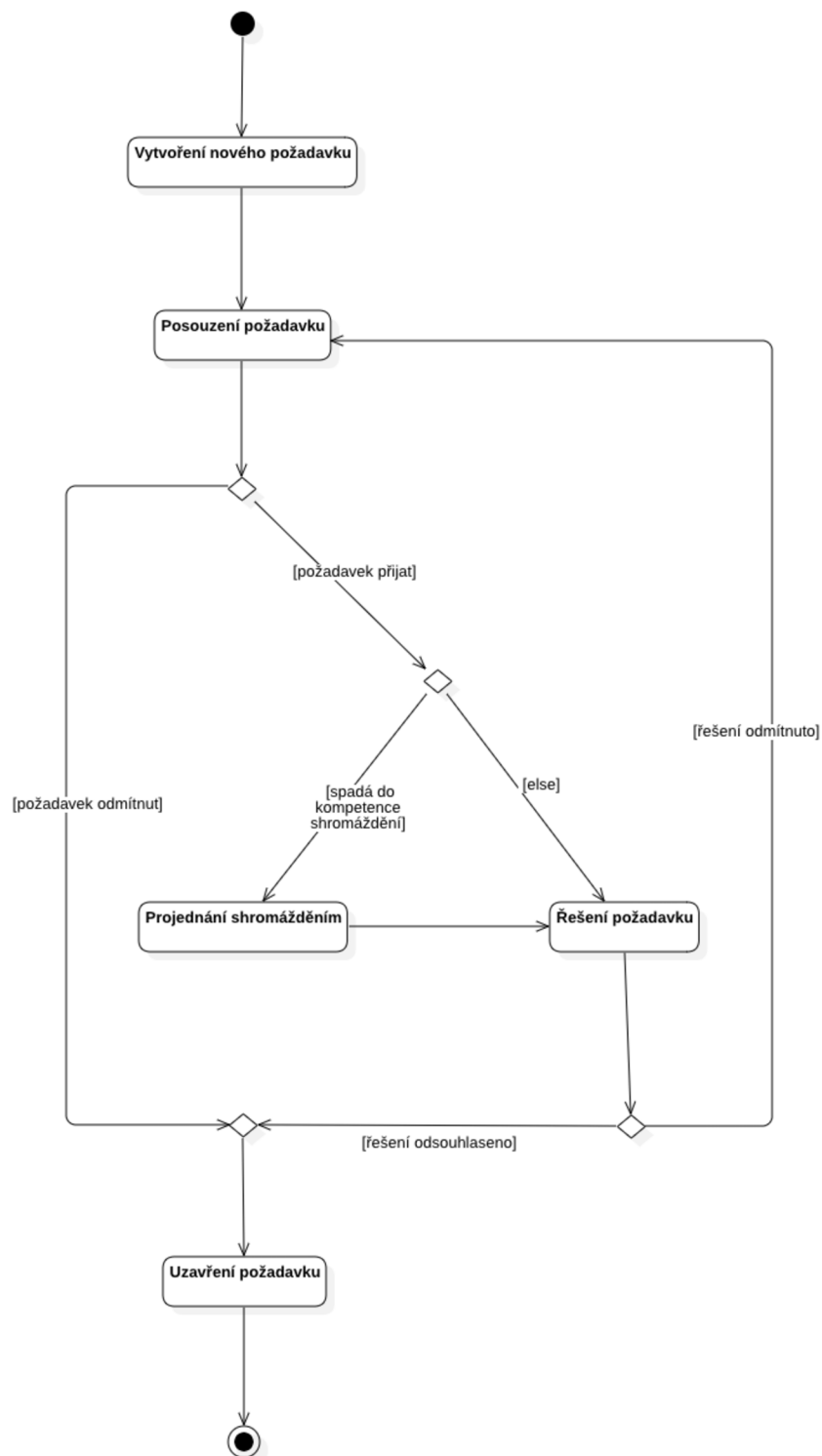
Obrázek 38: sekv. d. nahlášení závady správci společné části; zdroj: vlastní zpracování

Sekvenční diagram (Obrázek 38) znázorňuje scénář nahlášení závady správci společné části, který je popsán v 4.5.1.8. Další průběh řešení požadavku se může lišit. Některé možnosti jsou popsány v podkapitole 4.5.1.3 a znázorněny pomocí diagramů (Obrázek 27 a Obrázek 28).



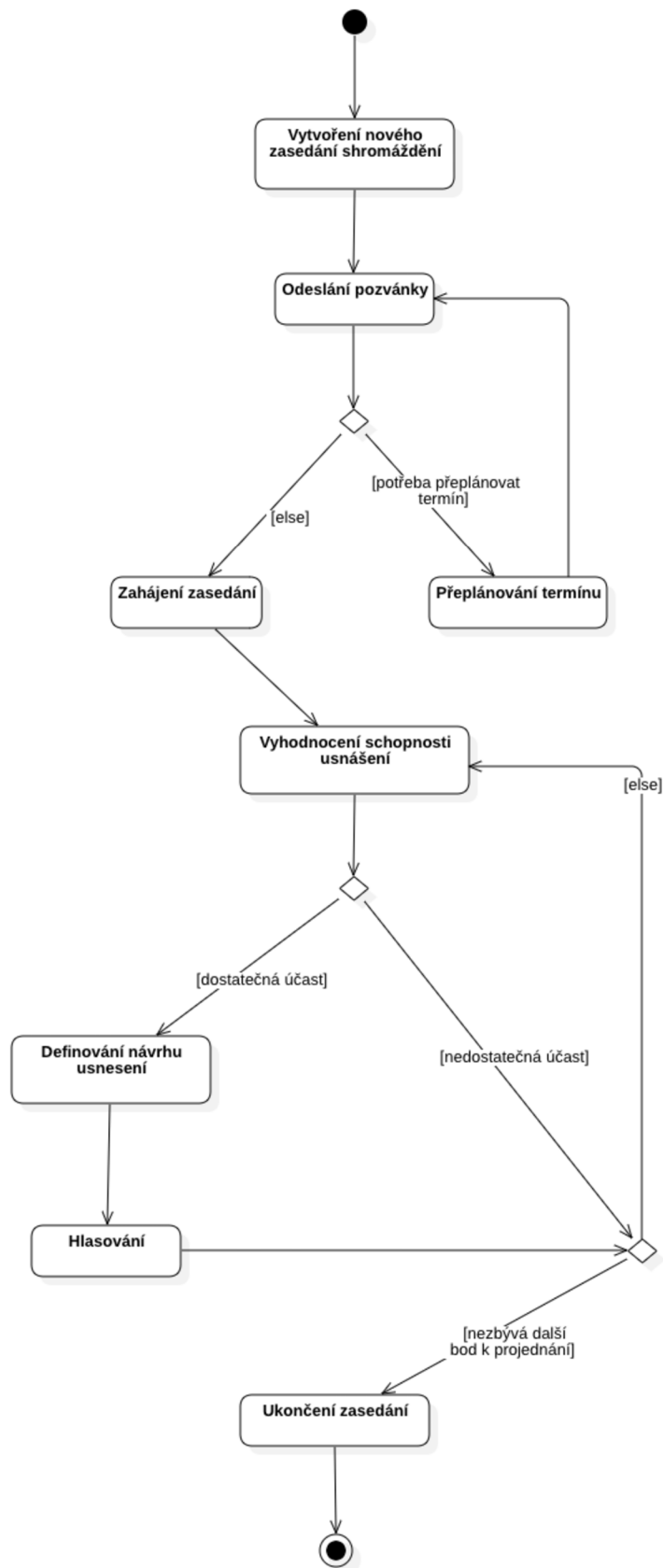
### 4.5.3 Diagramy aktivit

#### Vytvoření a řešení požadavku



Obrázek 39: diag. aktivit pro vytvoření a řešení požadavku; zdroj: vlastní zpracování

## Zasedání shromáždění



Obrázek 40: diag. aktivit zasedání shromáždění; zdroj: vlastní zpracování

## 5 Výsledky a diskuse

Cílem práce byl návrh informačního systému pro správu bytového domu pomocí modelovacího jazyka UML. Tento cíl byl dále specifikován na vytvoření podkladu pro budoucí mnohem podrobnější implementační návrh systému. Systém je určen pro společenství vlastníků jednotek spravovaná vlastními členy. Cíle bylo v této práci dosaženo vytvořením podkladu pro budoucí analýzu na základě sběru požadavků, ze kterých byla formulována koncepce systému shrnující významné procesy a oblasti, které by měl systém pokrýt. Analýza byla provedena s využitím objektově orientovaného přístupu. Na systém nahlíží z několika různých pohledů – z pohledu statické struktury objektů v systému a souvislostí mezi nimi, z pohledu chování systému v průběhu času a z pohledu vzájemné spolupráce objektů systému. K vytvoření diagramů byl použit CASE nástroj StarUML.

Analýza byla vytvořena tak, aby pokryla požadavky vzešlé ze sběru požadavků, který byl proveden formou nestrukturovaných rozhovorů s osobami z řad potenciálních uživatelů, tedy běžných řadových členů SVJ i členů výborů či předsedů SVJ, která jsou spravována vlastními členy. Pozvání k rozhovorům přijalo i několik odborníků z oboru správy SVJ. Potenciální uživatelé, se kterými byl sběr požadavků proveden, byli vybráni z několika SVJ z obce Kařez a města Plzeň.

Analýza systému byla zaměřena na:

- zefektivnění a usnadnění vzájemné komunikace mezi členy společenství vlastníků jednotek a jeho orgánů,
- na zvýšení transparentnosti správy společenství,
- snížení administrativní zátěže spojené se zasedáním shromáždění,
- na podporu komunitního bydlení.

Pro usnadnění komunikace všech osob ve společenství byl navržen systém požadavků, soukromých konverzací, veřejných příspěvků a aktualit. Transparentnost správy společenství by měla být zvýšena tlakem na dostupnost informací pro všechny členy společenství vlastníků. Na podporu komunitního bydlení byla v analýze navržena celá řada mechanismů, které se primárně týkají komunikace. Osoby ve společenství mohou jejich prostřednictvím nabídnout ostatním svoje služby či výpomoc, nebo naopak o pomoc požádat, rezervovat si společné části atd.

## Možnosti dalšího rozvoje

Vytvořená analýza poskytuje široké možnosti dalšího rozvoje.

- Návrh je možné snadno rozšířit o možnost vytváření vlastních procesních stavů nad některými objekty a vlastního workflow pomocí definovatelných přechodů mezi procesními stavy.
- Pro členy výboru by mohl být návrh rozšířen o možnost plánování a hlídání revizí.
- Členům výboru by mohla práci usnadnit také integrace na rejstříky. Existují služby, které nabízí pravidelné kontroly rejstříků Insolvencí, Exekucí, Likvidací, Dražebních vyhlášek a Zápisů do rejstříku. Je to důležité například z toho důvodu, že pokud se některý vlastník ocitne v exekuci, je zastaven i společný majetek, které má právo užívat. S majetkem pak není možné nijak nakládat, což je jistě nežádoucí stav.
- Systém by pomocí otevřeného aplikačního rozhraní mohl být integrován na IoT. Například nejrůznější čidla v domě a pozemku by mohla reportovat stav do systému, sloužit k zabezpečení, oznamování problémů atd.

Zvažováno bylo rozšíření využití i pro bytová družstva, ale takové rozšíření není doporučeno vzhledem k jejich mizivému počtu ve srovnání s počtem SVJ.

## Rizika

Použití systému by neslo také svá rizika.

- Ne všichni lidé jsou nakloněni použití elektronických systémů atd. Návrh obsahuje omezené prostředky k řešení některých situací offline bez použití systému. Návrh bude potřeba rozšířit o další mechanismy pro členy SVJ, kteří systém nebudou chtít, nebo nebudou moci, používat systém přímo, například zasílání různých důležitých informací prostřednictvím pošty, nebo způsoby zadávání požadavků atd.
- Rizikem pro systém mohou být také legislativní změny. Bude potřeba, aby měl systém zajištěnou dostatečnou podporu a byly do něj pravidelně zapracovávány legislativní požadavky.

## 6 Závěr

Cílem práce byl návrh informačního systému pro správu bytového domu pomocí modelovacího jazyka UML. Tento cíl byl dále specifikován na vytvoření podkladu pro budoucí mnohem podrobnější implementační návrh systému. Systém byl navržen pro společenství vlastníků jednotek, která jsou spravována vlastními členy. Cíle bylo v této práci dosaženo vytvořením podkladu pro budoucí analýzu na základě sběru požadavků, ze kterých byla formulována koncepce systému shrnující významné procesy a oblasti, které by měl systém pokrýt. K provedení analýzy byl použit objektově orientovaný přístup. Na systém je prostřednictvím analýzy nahlíženo z pohledu jeho statické struktury, z pohledu chování v průběhu času a z pohledu vzájemných interakcí. K vytvoření diagramů byl použit CASE nástroj StarUML.

Analýza byla vytvořena s ohledem na požadavky vzešlé ze sběru požadavků, který byl proveden formou nestrukturovaných rozhovorů s osobami z řad potenciálních uživatelů, tedy běžných řadových členů SVJ i členů výborů či předsedů SVJ, která jsou spravována vlastními členy. Pozvání k rozhovorům přijalo i několik odborníků z oboru správy SVJ. Potenciální uživatelé, se kterými byl sběr požadavků proveden, byli vybráni z několika SVJ z obce Kařez a města Plzeň. Jako odborníci z oblasti správy společenství vlastníků jednotek byli osloveni profesionální předsedové a pracovníci specializované správcovské firmy PRODOMIA, s.r.o. z Plzně. Analýza systému byla zaměřena na podporu komunitního bydlení, na zefektivnění a usnadnění vzájemné komunikace mezi členy společenství vlastníků jednotek a jeho orgánů, na zvýšení transparentnosti správy společenství a na snížení administrativní zátěže spojené se zasedáním shromáždění.

Pro usnadnění komunikace všech osob ve společenství byl navržen systém požadavků, soukromých konverzací, veřejných příspěvků a aktualit. Transparentnost správy společenství by měla být zvýšena tlakem na dostupnost informací pro všechny členy společenství vlastníků. Na podporu komunitního bydlení byla v analýze navržena celá řada mechanismů, které se primárně týkají komunikace. Osoby ve společenství mohou jejich prostřednictvím nabídnout ostatním svoje služby či výpomoc, nebo naopak o pomoc požádat, rezervovat si společné části atd.

V rámci dalšího vývoje by mohlo proběhnout vytvoření prototypu a jeho testování s potenciálními uživateli a stakeholdery. Následně by mohl být vytvořen podrobný

implementační návrh jako podklad pro realizaci. System má také široké možnosti dalšího rozvoje, které jsou popsány v kapitole 5.

## 7 Seznam použitých zdrojů

1. **BRUCKNER, Tomáš a kol.** *Tvorba informačních systémů*. [epub] Praha: Grada Publishing, a.s., 2012. ISBN 978-80-247-7903-4 [Citace: 1. březen 2024].
2. **ČESKO.** Zákon č. 89/2012 Sb., občanský zákoník - znění od 1. 1. 2024. *Zákony pro lidi*. [Online] <https://www.zakonyprolidi.cz/cs/2012-89>. [Citace: 10. březen 2024.]
3. **FOWLER, Martin.** *Destilované UML*. [epub] Praha: Grada Publishing, a. s., 2009. ISBN 978-80-271-2846-4 [Citace: 2. březen 2024].
4. **NILSSON, Andreas a WILSON, Timothy.** *Reflections on Barry W. Boehm's "A spiral model of software development and enhancement"*. [pdf] International Journal of Managing Projects in Business. 5. 737-756. 2012. Sv. 5. DOI:10.1108/17538371211269031 [Citace: 4. březen 2024].
5. **Principles behind the Agile Manifesto.** *Manifesto for Agile Software Development*. [Online] <http://agilemanifesto.org>. [Citace: 5. březen 2024.]
6. **Agility Academy.** What is DSDM Atern. [Online] <https://agility.ac/frequent-agile-questions/what-is-dsdm-atern>. [Citace: 5. březen 2024.]
7. **The Home of Scrum.** What is Scrum. [Online] <https://www.scrum.org/resources/what-scrum-module>. [Citace: 6. březen 2024.]
8. **Scrum Guides.** The 2020 Scrum Guide. [Online] <https://scrumguides.org/scrum-guide.html>. [Citace: 6. březen 2024.]
9. **ŘEPA, Václav.** *Podnikové procesy: procesní řízení a modelování*. [pdf] 2. aktualizované vydání. Praha: Grada Publishing, a.s., 2007. ISBN 978-80-247-6722-2 [Citace: 7. březen 2024].
10. **The Object Management Group Standards Development.** *Business Process Model and Notation*. [Online]. Version 2.0.2, 2013. <https://www.omg.org/spec/BPMN/2.0.2/PDF>. [Citace: 7. březen 2024]
11. **BOOCH, Grady a kol.** *Object-Oriented Analysis and Design with Applications*. Upper Saddle River: Addison-Wesley Professional, 2007. ISBN 978-02-018-9551-3.
12. **The Object Management Group Standards Development.** *OMG Unified Modeling Language*. [Online]. Version 2.5.1, 2017. <https://www.omg.org/spec/UML/2.5.1/PDF>. [Citace: 7. březen 2024]
13. **The Object Management Group Standards Development.** MISSION & VISION. *Object Management Group*. [Online] <https://www.omg.org/about/>. [Citace: 7. březen 2024.]
14. **VRANA, Ivan a RICHTA, Karel.** *Zásady a postupy zavádění podnikových informačních systémů*. [pdf] Praha: Grada Publishing, a.s., 2005. ISBN 978-80-247-6324-8 [Citace: 10. březen 2024].

15. **ELIÁŠ, Karel a kol.** *Občanské právo pro každého.* Praha: Wolters Kluwer ČR, a. s., 2013. ISBN 978-80-7478-013-4.

16. **ZÍTEK, Adam.** *Společenství vlastníků jednotek. Prakticky a bez obav.* Praha: C. H. Beck, 2022. ISBN 978-80-7400-872-6.



## 8 Seznam obrázků

Obrázek 1: schéma vodopádového životního cyklu; zdroj: (1) .....	14
Obrázek 2: schéma iterativního životního cyklu; zdroj: (1) .....	15
Obrázek 3: schéma inkrementálního životního cyklu; zdroj: (1) .....	16
Obrázek 4: schéma procesu metodiky OpenUP; zdroj: (1) .....	17
Obrázek 5: schéma etap metodiky DSDM; zdroj: (6) .....	17
Obrázek 6: schéma agilní metodiky Scrum; zdroj: (7) .....	18
Obrázek 7: příklad choreografie v BPMN; zdroj: (10) .....	19
Obrázek 8: příklad kolaborativního procesu v BPMN; zdroj: (10) .....	19
Obrázek 9: příklad Entity Relationship diagramu; zdroj: (1) .....	20
Obrázek 10: rozdělení UML diagramů podle významu; zdroj: (12) .....	22
Obrázek 11: základní koncept elementů a vztahů v UML; zdroj: (12) .....	23
Obrázek 12: příklad jednoduchého diagramu tříd; zdroj: (3) .....	24
Obrázek 13: příklad use case diagramu; zdroj: (12) .....	25
Obrázek 14: příklad sekvenčního diagramu s popisem; zdroj: (12) .....	26
Obrázek 15: příklad stavového diagramu; zdroj: (3) .....	26
Obrázek 16: příklad diagramu aktivit; zdroj: (3) .....	27
Obrázek 17: diagram tříd systému Domovnice; zdroj: vlastní zpracování .....	32
Obrázek 18: stavový diagram třídy Uživatel; zdroj: vlastní zpracování .....	37
Obrázek 19: stavový diagram třídy Požadavek; zdroj: vlastní zpracování .....	38
Obrázek 20: stavový diagram třídy Zasedání; zdroj: vlastní zpracování .....	39
Obrázek 21: stavový diagram třídy Zápis; zdroj: vlastní zpracování .....	40
Obrázek 22: stavový diagram třídy Dokument; zdroj: vlastní zpracování .....	41
Obrázek 23: stavový diagram třídy Aktualita; zdroj: vlastní zpracování .....	42
Obrázek 24: stavový diagram třídy Rezervace; zdroj: vlastní zpracování .....	43
Obrázek 25: diagram případů užití; zdroj: vlastní zpracování .....	44
Obrázek 26: sekv. d. vytvoření nového uživatelského účtu; zdroj: vlastní zpracování .....	63
Obrázek 27: sekv. d. vytv. a řešení požadavku; zdroj: vlastní zpracování .....	64
Obrázek 28: sekv. d. vytv. a řešení požad. shromážděním; zdroj: vlastní zpracování .....	65
Obrázek 29: sekv. d. svolání zased. s přeplánováním termínu; zdroj: vlastní zpracování ..	66
Obrázek 30: sekv. d. svolání zasedání z podnětu vlastníků; zdroj: vlastní zpracování .....	67
Obrázek 31: sekv. d. hlasování v průběhu zasedání; zdroj: vlastní zpracování .....	68
Obrázek 32: sekv. d. hlasování mimo zasedání shromážděním; zdroj: vlastní zpracování ....	69
Obrázek 33: sekv. d. vkládání dokumentů členem výboru; zdroj: vlastní zpracování .....	70
Obrázek 34: sekv. d. vkládání dokumentu jiným uživatelem; zdroj: vlastní zpracování ....	70
Obrázek 35: sekv. d. vytvoření aktuality členem výboru; zdroj: vlastní zpracování .....	71
Obrázek 36: sekv. d. vytvoření aktuality jiným uživatelem; zdroj: vlastní zpracování .....	71
Obrázek 37: sekv. d. vytvoření rezervace se schválením; zdroj: vlastní zpracování .....	72
Obrázek 38: sekv. d. nahlášení závady správci společné části; zdroj: vlastní zpracování ..	72
Obrázek 39: diag. aktivit pro vytvoření a řešení požadavku; zdroj: vlastní zpracování .....	73
Obrázek 40: diag. aktivit zasedání shromážděním; zdroj: vlastní zpracování .....	74

## **Přílohy**

1. Externí soubor - html-docs.zip – 2,7 MB (export ze StarUML)