



**FAKULTA  
INFORMAČNÍCH  
TECHNOLOGIÍ  
ČVUT V PRAZE**

## Zadání bakalářské práce

<b>Název:</b>	Systém pro správu nemovitosti
<b>Student:</b>	Jan Starůstka
<b>Vedoucí:</b>	Ing. Jiří Chludil
<b>Studijní program:</b>	Informatika
<b>Obor / specializace:</b>	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	do konce letního semestru 2022/2023

### Pokyny pro vypracování

Cílem práce je vytvořit prototyp webové aplikace pro správu (provozní příjmy a výdaje, majetek, atd.) nemovitosti, ve které jsou pronajímány byty.

1. Analyzujte požadavky potenciálních uživatelů.
2. Analyzujte dostupná softwarová řešení.
3. Pomocí metod softwarového inženýrství navrhnete modulární architekturu prototypu. Návrh přizpůsobte technologiím Express.js a React.js.
4. Implementujte prototyp serveru a webového klienta.
5. Prototyp podrobte vhodným testům (akceptační a uživatelské).

---

*Elektronicky schválil/a Ing. Michal Valenta, Ph.D. dne 30. března 2021 v Praze.*



Bakalářská práce

# SYSTÉM PRO SPRÁVU NEMOVITOSTI

**Jan Starůstka**

Fakulta informačních technologií ČVUT v Praze  
Katedra softwarového inženýrství  
Vedoucí: Ing. Jiří Chludil  
12. května 2021

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2021 Jan Starůstka. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bez uplatněných zákonných licencí nad rámec oprávnění uvedených v Prohlášení, je nezbytný souhlas autora.*

Odkaz na tuto práci: Jan Starůstka. *Systém pro správu nemovitosti*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

## Obsah

Poděkování	viii
Prohlášení	ix
Abstrakt	x
Shrnutí	xi
Seznam zkratk	xii
<b>1 Úvod</b>	<b>1</b>
<b>2 Cíle práce</b>	<b>3</b>
<b>3 Sběr požadavků</b>	<b>5</b>
3.1 První kolo . . . . .	5
3.2 Druhé kolo . . . . .	6
3.2.1 Výsledky prvního kola . . . . .	7
3.2.2 Priority požadavků . . . . .	7
3.2.3 Funkční požadavky . . . . .	7
3.2.4 Nefunkční požadavky . . . . .	10
<b>4 Analýza konkurence</b>	<b>13</b>
4.1 Dostupné softwary . . . . .	13
4.1.1 Domsys . . . . .	13
4.1.2 Anasoft Domus . . . . .	13
4.1.3 Správa bytů a nemovitostí (SBN) . . . . .	13
4.1.4 Jirra inkasník/vlastník . . . . .	13
4.1.5 Qi správa nemovitosti . . . . .	14
4.1.6 CSH Domovník . . . . .	14
4.2 Způsob hodnocení . . . . .	14
4.2.1 Hodnocení funkčních požadavků . . . . .	14
4.2.2 Hodnocení nefunkčních požadavků . . . . .	16
4.3 Závěr hodnocení konkurence . . . . .	18
<b>5 Návrh</b>	<b>19</b>
5.1 Doména . . . . .	19
5.1.1 Popis domény . . . . .	19
5.1.2 Doménový model . . . . .	20
5.2 Návrh obrazovek . . . . .	22
5.2.1 Hlavní menu . . . . .	22
5.2.2 Byt . . . . .	22
5.2.3 Přehled financí . . . . .	23
5.2.4 Přehled jednotek . . . . .	24

5.2.5	Přidání nákladu . . . . .	24
5.2.6	Příchozí platby . . . . .	25
5.2.7	Nájemník . . . . .	25
5.2.8	Stažení vyúčtování . . . . .	26
5.2.9	Rozpočítání služeb . . . . .	26
5.2.10	Přechodová mapa . . . . .	27
5.3	Procesy . . . . .	28
5.3.1	Přidání nákladu . . . . .	28
5.3.2	Přijetí plateb za nájem . . . . .	29
5.3.3	Změna nájemníka . . . . .	29
5.4	Databáze . . . . .	31
<b>6</b>	<b>Implementace</b>	<b>33</b>
6.1	Architektura a technologie . . . . .	33
6.1.1	React . . . . .	34
6.1.2	Node.js a Express . . . . .	34
6.1.3	MongoDB . . . . .	34
6.1.4	Mongoose . . . . .	34
6.2	Uživatelské rozhraní prototypu . . . . .	34
6.2.1	Přihlašování . . . . .	34
6.2.2	Navigační lišta . . . . .	35
6.2.3	Hlavní menu . . . . .	35
6.2.4	Finance . . . . .	36
6.2.5	Přehled jednotek . . . . .	36
6.2.6	Přehled nájemníků . . . . .	37
6.2.7	Přidání nákladu . . . . .	38
6.2.8	Příchozí platby . . . . .	39
6.2.9	Nájemník . . . . .	40
6.2.10	Jednotka (část) . . . . .	40
6.2.11	Přechodová mapa . . . . .	42
6.3	Server . . . . .	42
6.4	Databáze . . . . .	44
6.4.1	Mongo Atlas . . . . .	44
6.4.2	Část domu . . . . .	45
6.4.3	Nájemník . . . . .	45
6.4.4	Náklady . . . . .	45
6.4.5	Příjmy . . . . .	45
6.4.6	Uživatel . . . . .	45
6.5	Přidaná hodnota prototypu . . . . .	45
<b>7</b>	<b>Testování</b>	<b>47</b>
7.1	Uživatelské testy . . . . .	47
7.1.1	Přihlášení a přidání nákladu . . . . .	47
7.1.2	Zobrazení nájemníků a práce s nimi . . . . .	48
7.1.3	Zobrazení jednotek a práce s bytem . . . . .	50
7.1.4	Přijetí plateb . . . . .	52
7.1.5	Průběh a vyhodnocení uživatelského testování . . . . .	53
7.2	Akceptační testy . . . . .	54
7.2.1	Splnění funkčních požadavků . . . . .	54
7.2.2	Splnění nefunkčních požadavků . . . . .	56
7.2.3	Celkové hodnocení prototypu . . . . .	57
<b>8</b>	<b>Závěr</b>	<b>59</b>

<b>A</b>	<b>Instalační příručka</b>	<b>61</b>
A.1	Databáze . . . . .	61
A.2	Instalace git . . . . .	61
A.3	Instalace serveru . . . . .	61
A.4	Instalace frontend . . . . .	62
A.5	Vytvoření prvního uživatele . . . . .	63
<b>B</b>	<b>Uživatelská příručka</b>	<b>65</b>
	<b>Obsah přiloženého média</b>	<b>69</b>

## Seznam obrázků

3.1	Use case diagram . . . . .	11
5.1	Doménový model domu . . . . .	21
5.2	Návrh brazovky hlavního menu . . . . .	22
5.3	Návrh obrazovky bytu . . . . .	23
5.4	Návrh obrazovky přehledu financí . . . . .	23
5.5	Návrh obrazovky přehledu jednotek . . . . .	24
5.6	Návrh brazovky přidání nákladu . . . . .	25
5.7	Návrh brazovky příchozích plateb . . . . .	25
5.8	Návrh obrazovky nájmeníka . . . . .	26
5.9	Návrh brazovky stažení vyúčtování . . . . .	26
5.10	Návrh obrazovky rozpočítání služeb . . . . .	27
5.11	Návrh přechodové mapy . . . . .	28
5.12	Proces přidání nákladu . . . . .	29
5.13	Proces přijetí plateb . . . . .	29
5.14	Proces vystěhování nájmeníka . . . . .	30
5.15	Proces přiřazení nájmeníka k bytu . . . . .	30
5.16	Návrh databáze . . . . .	32
6.1	Architektura prototypu . . . . .	33
6.2	Přihlašovací obrazovka . . . . .	35
6.3	Navigační lišta . . . . .	35
6.4	Hlavní menu . . . . .	36
6.5	Finance . . . . .	36
6.6	Přehled jednotek . . . . .	37
6.7	Přehled nájmeníků . . . . .	38
6.8	Přidání nákladu . . . . .	38
6.9	Příchozí platby . . . . .	39
6.10	Obrazovka nájmeníka . . . . .	40
6.11	Editace nájmeníka . . . . .	40
6.12	Obrazovka jednotky . . . . .	41
6.13	Editace jednotky . . . . .	42
6.14	Přechodová mapa prototypu . . . . .	42
6.15	API Jednotky (byty) . . . . .	43
6.16	API Nájmeníci . . . . .	43
6.17	API Náklady . . . . .	44
6.18	API Příjmy . . . . .	44
6.19	API Uživatel . . . . .	44



## Seznam tabulek

4.1	Hodnocení funkčních požadavků u dostupných softwarů . . . . .	17
4.2	Hodnocení nefunkčních požadavků u dostupných softwarů . . . . .	17
7.1	Hodnocení funkčních požadavků prototypu . . . . .	57
7.2	Hodnocení nefunkčních požadavků prototypu . . . . .	57

*Chtěl bych především poděkovat vedoucímu práce Ing. Jiřímu Chludilovi za jeho cenné rady ke tvorbě práce.*

*Také bych chtěl poděkovat svému otci za pomoc při sběru požadavků. Poděkování patří také testerům z řad rodiny a přátel za to, že si na mě udělali čas a dali mi zpětnou vazbu na aplikaci.*

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 12. května 2021

.....

## Abstrakt

Ve své práci se věnuji analýze, návrhu, implementaci a testování aplikace, která má za cíl usnadnit správu nemovitosti, ve které jsou pronajímány byty, případně jiné prostory.

V rámci analýzy jsem sesbíral funkční a nefunkční požadavky na takovou aplikaci. Následně jsem zkoumal a hodnotil již dostupné aplikace pro správu nemovitosti. Zjistil jsem, že dostupné softwary jsou dosti složité a drahé. Nejsou příliš vhodné pro správu malých nemovitostí.

Navrhl jsem vzhled webového rozhraní, strukturu aplikace a databáze. Pak jsem implementoval prototyp serveru (Node.js, Express) a webového klienta (React). Server komunikuje s klientem přes REST API. Pro uchování dat jsem využil MongoDB umístěnou v cloudu.

Prototyp jsem následně podrobil uživatelským a akceptačním testům. V uživatelských testech procházeli testěři prototypem podle předem definovaných scénářů. Poté mi dali zpětnou vazbu na uživatelské rozhraní aplikace. V akceptačních testech jsem hodnotil, jak se mi podařilo splnit jednotlivé funkční a nefunkční požadavky definované v analýze.

**Klíčová slova** Správa nemovitosti, Analýza softwarů pro správu domu, Návrh softwaru, Implementace jednoduchého prototypu, Testování prototypu, Node.js, Express, React, MongoDB, Mongoose, REST API

## Abstract

In my thesis I do analysis, design, implementation and testing of application that should help with management of property, where flats or other parts are being rented.

In analysis I found out, what should such an application do. Then I researched already existing softwares for property management. I discovered that they are too complicated and not good enough for little properties with only a few flats to rent.

I designed the application and implemented a prototype of server (Node.js, Express) and web client (React). Server communicates with the client using the REST API. To store data I used MongoDB hosted in cloud.

Afterwards I performed user and acceptance tests. In user testing, testers went through scenarios I prepared. Then they gave me feedback on usability of the application. In acceptance tests I evaluated how well I managed to implement requirements collected in analysis.

**Keywords** Property management, Property management programs analysis, Designing software, Implementation of an intuitive prototype, Testing prototype, Node.js, Express, React, MongoDB, Mongoose, REST API

# Shrnutí

## Motivace

Můj otec vlastní dům, ve kterém pronajímá byty. Udržet přehled v příjmech a nákladech v takovém domě není triviální úkol. Myslím si, že by část úkonů šla zautomatizovat a zjednodušit. Na to mysleli už i jiní a softwary za tímto účelem už vznikly. Žádný ale nevyhovuje požadavkům člověka, který pronajímá jednotky bytů a nestará se o ně na plný úvazek.

## Cíl práce

Vytvořit aplikaci, která by pomohla se správou domu majitelům, kteří chtějí mít přehled o výsledcích svého hospodaření, ale nemají čas se domu věnovat naplno. Aplikace by měla být jednoduchá a intuitivní, aby se v ní vyznali nejen počítačový experti. Měla by umožnit to stejné, co dělá majitel nyní „na papíře“, ale poskytnout mu navíc přehlednější výstupy.

## Postup

Od otce jsem si zjistil, co by se při správě domu dalo zautomatizovat nebo zjednodušit. Rozhodl jsem se aplikaci udělat webovou, protože pak nebude závislá na operačním systému.

Analyzoval jsem aplikace, které k tomuto tématu již vznikly. Z mého výzkumu vyplývá, že nesplňují všechny požadavky, které by taková aplikace měla umět a které vyplývaly z mé analýzy. Tyto softwary jsou zbytečně složité, komplexní a drahé na to, aby s nimi pracoval někdo, kdo pronajímá pár bytů a stará se o ně při plném úvazku v jiném zaměstnání.

Pak jsem navrhl, jak by mohla aplikace vypadat a jak by měla fungovat. To jsem pak validoval u zadavatele. Následně jsem implementoval prototyp, který sice nesplňuje všechny požadavky, ale i tak může majiteli pomoci se správou. Prototyp jsem podrobil uživatelským a akceptačním testům.

## Výsledky práce

Implementovaný prototyp může pomoci majitelům se správou jejich nemovitosti. Nesplňuje zatím všechny požadavky, které jsem v analýze definoval, ale již umí zobrazit majiteli přehledně, za co, kde a kolik utratil a jak jsou které části domu ziskové či ztrátové.

Uživatelské testování prototypu odhalilo menší nedostatky v uživatelské přívětivosti prototypu. Všichni testeři ale dokázali splnit všechny úkony, které jsem od nich žádal bez větších problémů.

## Závěr

Zadání mi udávalo sběr požadavků, analýzu konkurence, návrh, implementaci a testování prototypu. Vše se mi podařilo splnit. Mým osobním cílem bylo implementovat takovou aplikaci, která bude majitelům užitečná při správě jejich nemovitosti. Prototyp sice nesplňuje všechny funkcionality, zjištěné v analýze, ale i přesto má co nabídnout. Mám v plánu ho dál upravovat a doplňovat funkcionality, které jsem nestihl implementovat.

## Seznam zkratk

API	application programming interface
CD	Continuous deployment
CI	Continuous integration
CRUD	create, read, update, delete
FP	funkční požadavek
JS	javascript
JSON	javascript object notation
JWT	JSON web token
NA	not available
NP	nefunkční požadavek
REST	representational state transfer

# Kapitola 1

## Úvod

Správa nemovitosti je, pokud nechceme aby nemovitost zchátrala, nezbytná. Obecně se jedná o činnost, která je nudná a nezáživná. Na druhou stranu ve správě je a vždy bude prostor pro zjednodušení, zefektivnění nebo pro automatizaci činností.

Jedna možnost, jak zefektivnit správu nemovitosti, je použití vhodné platformy pro práci s daty. Pokud si bude majitel důležité věci o budově pouze pamatovat, pravděpodobně nebude vědět tak dobře, co se mu doma děje, jako kdyby si důležité věci někam psal.

Ve chvíli, kdy máme z nemovitosti nějaký příjem, je nutné data spravovat pečlivě. Už jen pro každoroční podání daňového přiznání je potřeba mít archivovaná data o příjmech a výdajích. U nemovitostí to jsou zpravidla příjmy z pronájmu a náklady na správu, provoz a údržbu.

Vedení účetnictví je časově náročné. Ale dalo by se s ním pomoci. Majitel je ze zákona nucen uchovávat data o svých příjmech. Pokud to dělá nejjednodušeji, jak může, vše „hází“ do jednoho šanonu a na konci roku sečte, nedostane tak z nemovitosti spoustu užitečných dat. Pokud by třeba majitel místo založení faktury do šanonu pouze vložil fakturu do aplikace, mohla by za něj následně spočítat, jak je která část jeho nemovitosti výdělečná. Už to je posun oproti tomu, kdy majitel vše sečte jednou za rok a zjistí, kolik za celý rok vydělal.

Aplikace by majiteli také mohla ušetřit čas, který správou tráví. Například tím, že příjmy a náklady sečte za něj. Majitel pak nebude muset před podáním daňového přiznání strávit víkend s kalkulačkou, utopený ve fakturách, poznámkách a účtenkách. Aplikace by také mohla pomoci s rozpočítáním nákladů za energie, hlídat platnost nájemních smluv a upozorňovat na nutné revize.

Softwary, které slouží ke správě nemovitosti s nájemními prostory, samozřejmě již existují a majitelé je využívají. Tyto aplikace však nejsou vhodné pro všechny. Většinou jsou zaměřeny na majitele, kteří pronajímají desítky či stovky bytů. Pro někoho, kdo pronajímá pouze pár jednotek, jsou tyto aplikace zbytečně drahé a složité. Ale i jednoduchá aplikace by mohla majiteli ušetřit čas a přinést mu výstupy, kterých se sešitem těžko dosáhne.

Aplikace, která by menším majitelům pomohla, by měla být jednoduchá, intuitivní a levná. Jednoduchá tak, aby ji člověk, který dosud řeší správu nemovitosti na papíře s kalkulačkou, zvládl používat a dělal podobné úkony, na které byl zvyklý. Aplikace by také měla vést majitele k tomu, aby věci dělal správně (ukládal si faktury, evidoval příjmy, ...).

Aplikace by pak majitele odměnila úsporou času nutného věnovat správě a přinesla by mu lepší data o jeho nemovitosti. Například by se majitel dozvěděl, který byt je jak výdělečný a kde jak často musí jednotky opravovat či rekonstruovat.

Nejdřív zjistím, co by taková aplikace měla dělat proto, aby opravdu byla majitelům nemovitostí užitečná. Spolupracoval jsem na tom se svým otcem, který je přesně ten typ majitele, kterého jsem popsal výše. Pokud se mi podaří vytvořit dobrou aplikaci, která mu pomůže, tak by mohla pomoci i ostatním, podobným majitelům. Na základě zjištěných požadavků na aplikaci,

prozkoumám již dostupné softwary řešící správu domu.

Následně navrhnu aplikaci, u které budu klást důraz na to, aby byla co nejjednodušší na používání, ale zároveň splňovala požadavky zjištěné v analýze. Ty nejdůležitější požadavky implementuji v prototypu aplikace. Následně aplikaci podrobím uživatelským a akceptačním testům.





## Kapitola 2

# Cíle práce

Hlavním cílem práce je vymyslet a navrhnout, jak usnadnit správu nemovitosti lidem, kteří nemovitost pronajímají. Abych toho dosáhl, potřebuji zjistit, co by taková aplikace měla umět. Potřebuji na ni sesbírat funkční a nefunkční požadavky. Užitečným aspektem bude zjistit kvalitu již dostupných softwarů, abych se při návrhu poučil z jejich chyb.

Dalším cílem je navrhnout a implementovat prototyp takovéto aplikace. Technologie, ve kterých budu prototyp implementovat mám dané v zadání, proto jejich výběr cílem této práce není. Na prototyp by se v budoucnu mělo dát navázat dalším vývojem, není tedy cílem vytvořit prototyp v produkční kvalitě, ani ho nasadit do ostrého provozu.

Mým osobním cílem je primárně pomoci se správou našeho domu. Pokud se to podaří, budu rád aplikaci dál rozšiřovat, aby mohla pomoci s evidencí spojenou se správou i jiným majitelům nemovitostí. Přeji bych si, aby aplikaci někdo ke správě nemovitosti opravdu používal byla mu užitečná.



## Kapitola 3

# Sběr požadavků

V této kapitole popisuji, jak probíhal sběr a zadávání požadavků na aplikaci. Požadavky sbírám proto, že v tuto chvíli neznám doménu a nerozumím ji. Doméně ale rozumí můj otec, majitel domu. Aby jemu, případně dalším majitelům nemovitostí byla užitečná, potřebuji od něj získat požadavky.

Sběr požadavků jsem provedl dvoukolově. V prvním kole jsem se snažil vymyslet co nejvíce různých funkcionalit, aby pak měl otec (zadavatel) z čeho vybírat. Ve druhém kole jsme vybrali (pro něj) ty důležité funkcionality a upřesnili je.

### 3.1 První kolo

Následující požadavky vyplynuly z první debaty mezi mnou a otcem [1]. Na základě jeho potřeb a nabídek konkurenčních softwarů jsem rámcově zformuloval funkční a nefunkční požadavky a poslal mu je k náhledu.

Cílem tohoto kola bylo předložit zákazníkovi vědět co nejvíce různých funkcionalit o které by potenciálně mohl mít zájem.

#### Automatická kontrola plateb

Po přihlášení si majitel může zobrazit platby od nájemníků. Případná neprovedená platba bude zvýrazněna a majitel dostane možnost zaslat nájemníkovi upozornění (e-mailem). Notifikace dlužníkům může majitel nechat odesílat automaticky.

#### Rozpočítání plateb za energie a odpady

Aplikace za majitele rozpočítá platby pro jednotlivé byty za energie a vodu. Majitel pak může buď hodnoty odsouhlasit, nebo upravit. Výši těchto poplatků si pak nájemník může zobrazit v aplikaci.

#### Dostupnost všech kontaktů na jednom místě

Důležité kontakty (majitel, správce, zámečnick, oprava výtahu, ...) budou jednoduše dostupné na jednom místě a každý nájemník se k nim může jednoduše dostat, aniž by si o ně musel žádat u majitele/správce.

Majitel bude mít přístup ke kontaktům nájemníků. V případě změny kontaktu (telefonní číslo, e-mail) si nájemník pouze změní příslušný kontakt v aplikaci.

## Evidování nákladů a jejich ukládání k bytům

Majitel bude mít možnost uložit fakturu nebo účtenku do účetnictví k jednotlivým bytům, případně jako náklad do celého domu.

## Monitorování příjmů a nákladů

Majiteli bude umožněno zobrazit si (v grafech) výnosy a výdaje za celý dům, případně za jednotlivé byty. V přehledu budou zahrnuty nájmy a na druhé straně provedené investice.

## Generování podkladů pro daňové přiznání

Možnost sečíst příjmy a náklady a uspořádat do excelovské tabulky tak, aby bylo pro majitele jednodušší vyplnit daňové přiznání.

## Virtuální nástěnka

V případě, že je potřeba hromadně informovat nájemníky o nějaké události, může majitel napsat zprávu na virtuální nástěnku v systému. Zpráva se nemusí týkat všech nájemníků a majitel pak může vybrat pouze ty, kterým se to zobrazí (například zpráva o změně poplatku za psa zajímá jen ty, kteří psa mají).

## Evidence vybavení

Možnost sledování stáří a stavu spotřebičů a vybavení v jednotlivých prostorách domu. V případě zájmu lze nastavit notifikace pro kontrolu, revizi nebo výměnu. U jednotlivých prostor si pak majitel může zobrazit, jaké vybavení se v nich nachází.

## Hlášení závad

Pokud něco v domě nefunguje, jak má (nesvítí žárovka, nejde výtah, . . . ), může každý nájemník tuto chybu nahlásit. Čím dříve si majitel (nebo správce) závady všimne, tím dříve se dá začít řešit.

## Přehled plateb pro nájemníka

Nájemník uvidí v systému přehled poplatků za nájem, energie, . . . Aplikace mu taky vygeneruje QR kód pro provedení platby za měsíční nájem včetně všech poplatků.

## Generování nájemních smluv

Místo přepisování informací o novém nájemníkovi ve wordové šabloně, zadá majitel informace o nájemníkovi do systému a ten vyplní šablonu za něj. Odpadá rutina a minimalizuje se riziko chyby.

### 3.2 Druhé kolo

Ve druhém kole sběru požadavků již přesně specifikují jednotlivé funkční i nefunkční požadavky na aplikaci.

### 3.2.1 Výsledky prvního kola

Mezi první a druhou verzí požadavků jsem se zadavatelem probral návrh funkcionalit. Požadavky z prvního kola počítají s rolí majitele/správce a s rolí nájemníka. Rozhodli jsme se vypustit účet nájemníka ze systému. Stále pro něj bude možné se dívat na důležité kontakty, ale ztratí možnost se podívat na přehled plateb za nájem. Možná by tato funkcionalita mohla být zajímavá pro nájemní bydlení na kratší dobu (ubytovny, koleje). Pro dlouhodobé nájmy nemá funkcionalita takový smysl, protože většina nájemníků má v bance nastavený trvalý příkaz, a proto tuto možnost nepotřebuje a nevyužívali by ji. Funkcionality určené pro majitele jsme upřesnili.

### 3.2.2 Priority požadavků

Požadavkům přiděluji priority „vysoká“, „střední“, „nízká“. Požadavky s vysokou prioritou jsou pro fungování aplikace esenciální. Bez nich nebude mít smysl aplikaci používat. Požadavky se střední prioritou jsou víceméně podpůrné funkcionality pro pohodlnější práci s aplikací. Nízkou prioritu jsem volil u požadavků, u kterých nebude jejich absence zásadně chybět. Bud' se jedná o „nice to have“ vychytávky, nebo možné budoucí rozšiřování aplikace.

### 3.2.3 Funkční požadavky

#### FP1 Rozdělení domu na části

Majitel může svůj dům pro přehlednost rozdělit na části. Toto členění je především pro lepší přehled nákladů a příjmů domu. Nájmy a náklady se budou v celém domě sčítat, aby bylo usnadněno podání daňového přiznání majitele.

Aplikace bude mít několik předdefinovaných typů prostor. U všech si bude možné vést základní informace (plocha, nájem) a sledovat u nich příjmy a náklady.

#### ■ Byt

Bude potřeba byty očíslovat tak, aby byly jednoznačně identifikovatelné. Podle čísla bytu bude možné kontrolovat, jak který byt vydělává nebo prodělává. Byty budou nejdůležitější částí příjmů, proto je u nich evidováno více informací, než u zbylých částí.

- Číslo bytu
- Aktuální nájem
- Aktuální nájemník
- Cena služeb
- Počet lidí nahlášených v bytě
- Počet klíčů zapůjčených nájemníkovi
- Předpokládaný nájem (za kolik by si majitel přál byt pronajímat)
- Cena bytu (kolik by byt aktuálně stál)
- Sklep a půda (mohou patřit k bytu a být pronajímány jako celek)

#### ■ Chodba

#### ■ Sklep

#### ■ Půda

#### ■ Fasáda

#### ■ Kotelna

- Výtah
- Zahrada

Priorita požadavku: vysoká

## FP2 Evidování nákladů a investic

### Náklady

Majitel bude mít možnost přidat do účetnictví fakturu nebo účet za jakýkoliv náklad v domě (nákup vybavení, údržba, služby, ...). U nákladu zapíše částku a datum, případně přidá popis. Tento výdaj může přidat k jednotlivé části domu, aby později mohl kontrolovat, kolik kde vynaložil prostředků. Majitel bude moci vybrat, jestli se tato věc má ukládat a zobrazovat i jako vybavení, v takovém případě může doplnit další informace.

Priorita požadavku: vysoká

### Investice

Investice lze odepisovat několik let, dokonce to je většinou žádoucí pro daňovou optimalizaci. Další rozdíl oproti nákladům je, že investice se dá odečíst od základu daně ještě dříve, než je provedena. Aplikace umožní majiteli investice a plán investic sledovat.

Podobně jako u nákladů investici přidá mezi investice v aplikaci. Uvede přitom, jestli je už realizovaná, nebo jestli teprve na realizaci čeká a kolik investice stála (bude stát). Aplikace pak spočítá a doporučí majiteli, kdy si má jakou částku odečíst ze základu daně tak, aby platil na daních co nejméně a zároveň, aby nebyl ve zbytečně velké daňové ztrátě.

Priorita požadavku: střední

## FP3 Monitorování financí

### Příchozí platby

Majitel bude mít možnost vybrat, kteří nájemníci již za daný měsíc zaplatili nájem. Udělá to zaškrtnutím políčka (checkboxu) v přehledu plateb. Když takto platbu označí, přidají se peníze za nájem k výnosům a majitel je pak uvidí v přehledu toku peněz.

Zatím se nepočítá s verzí, která by uměla kontrolovat platby automaticky napojením na bankovní aplikaci. Zákazník si to spíše nepřeje s tím, že úspora času by nebyla velká. Navíc někteří nájemníci stále platí raději nájem hotově. To se automaticky kontrolovat nedá, takže by stejně bylo na majiteli, aby alespoň část příchozích plateb zpracovával ručně.

Pro zajímavost si bude majitel moci zobrazit i částku, kterou by vydělával za předpokladu, že by byl dům plně obsazen. Tato částka se vypočítá z předpokládaného nájmu u bytů a nebytových prostor spolu s nájmy všech ostatních částí domu.

Priorita požadavku: vysoká

### Odchozí platby

Majitel bude moci zadat do systému i odchozí platbu z účtu domu nesouvisející přímo s domem. Tato funkcionality slouží pro lepší přehled, aby bylo vidět, za co se utráčí zisky domu.

Priorita požadavku: střední

## FP4 Rozpočítání plateb za energie a odpady

### Rozpočítání záloh

Majitel může nechat rozpočítat zálohy za jednotlivé služby. Rozpočítání proběhne podle jím vybraného klíče (počet lidí v bytě, plocha bytu, ...). Aplikace mu zobrazí návrh cen pro jednotlivé nájemníky, který může majitel případně změnit.

Priorita požadavku: střední

### Rozpočítání nedoplatků

V případě nedoplatku za služby se majitel může rozhodnout tento nedoplatek vymáhat po nájemnících. Zadá do systému výši nedoplatku a ten rozpočítá nedoplatek mezi nájemníky.

Priorita požadavku: střední

V obou případech dostane majitel možnost odeslat všem upozornění. Každému nájemníkovi pak přijde zpráva s výší zálohy nebo nedoplatku.

Priorita požadavku: střední

## FP5 Generování podkladů pro daňové přiznání

Na základě dat vkládaných do systému bude možné vygenerovat souhrnnou excelovou tabulku na vybraný kalendářní rok. Přímo v systému si majitel může zobrazit důležité položky pro své daňové přiznání (příjem, náklady, investice) bez stahování tabulky.

Priorita požadavku: vysoká

## FP6 Evidence vybavení

Aplikace umožní majiteli sledovat stáří, cenu a stav vybavení v domě. Bude možné vložit jednotlivé kusy vybavení buď při vkládání nějaké položky jako náklad, nebo i bez nákladu. Vybavení bude možné rozdělovat do kategorií (nábytek, spotřebiče, kuchyňská linka, kotel, bojler, ...) a přiřazovat ho k jednotlivým prostorám. Majitel si pak bude moci zobrazit, v jakém stavu jsou jednotlivé části domu a kde se jak často kupuje nové vybavení. U jednotlivých vybavení bude možné automaticky poslat notifikaci po stanovené době fungování (třeba u kotlů nebo bojlerů je každých pár let potřeba pořádná kontrola).

Priorita požadavku: nízká

## FP7 Hromadná notifikace nájemníků

Pomocí systému bude možné notifikovat nájemníky, nebo nějakou podskupinu o události, která se jich týká (změna poplatku za služby, revize kotlů, odečet elektroměru, ...). Notifikace bude nájemníkům zaslána pomocí sms nebo e-mailu.

Změna oproti předchozí verzi požadavků je v tom, že nyní už se nepočítá s účty pro nájemníky, proto se posílá sms nebo e-mail. Nájemník tak není nucen sledovat nástěnku na webu.

Priorita požadavku: střední

## FP8 Dostupnost kontaktů

Důležité kontakty (majitel, správce, zámečnick, oprava výtahu, ...) budou veřejně dostupné na jednom místě a každý nájemník se k nim může jednoduše dostat, aniž by si o ně musel žádat u majitele/správce. Kromě nájemníků si takto může kontakt na majitele najít i případný zájemce o bydlení.

Priorita požadavku: nízká

## FP9 Ukládání smluv

K jednotlivým částem domu bude možné přidávat nájemní smlouvy ve formátu pdf. Ke smlouvě bude potřeba přidat dobu platnosti. Aplikace bude umět majitele upozornit na končící smlouvy.

Priorita požadavku: nízká

## 3.2.4 Nefunkční požadavky

### NP1 Webová aplikace

Aplikace bude přístupná ve webovém prohlížeči Chrome (verze 90.0.4430.66), Safari (verze 14.0.3) a Firefox (verze 87.0).

Priorita požadavku: vysoká

### NP2 Responzivita

Webová aplikace bude responzivní.

Priorita požadavku: vysoká

### NP3 Mobilní aplikace

Aplikace pro mobilní telefony s operačními systémy IOS a Android.

Priorita požadavku: nízká

### NP4 Autentizace uživatele

Aby mohl uživatel (majitel) s nemovitostí v aplikaci pracovat, bude se muset přihlásit na svůj účet. V pozdějších verzích přibude možnost jednu nemovitost spravovat z více účtů.

Priorita požadavku: vysoká

### NP5 Autorizace uživatele

V první verzi počítám pouze s jednou rolí (administrátor). Časem ale bude moct existovat více účtů k jedné nemovitosti s různými právy. Například pro účetní může být účet „smí pouze nahlížet na finance a stahovat o nich data“.

Priorita požadavku: střední

### NP6 Bezpečné ukládání hesla uživatele

Přihlašovací údaje nebudou ukládány v plain textu, ale budou soleny a hešované podle SHA-2 a ukládat budou pouze heše. V pozdějších verzích by šlo využít autorizace například pomocí Google nebo Facebook účtu.

Priorita požadavku: vysoká

### NP7 Podpora více jazyků

Aplikace bude v češtině a v angličtině. Mezi jazyky bude možnost přepínat.

Priorita požadavku: nízká (v prototypu pouze čeština)

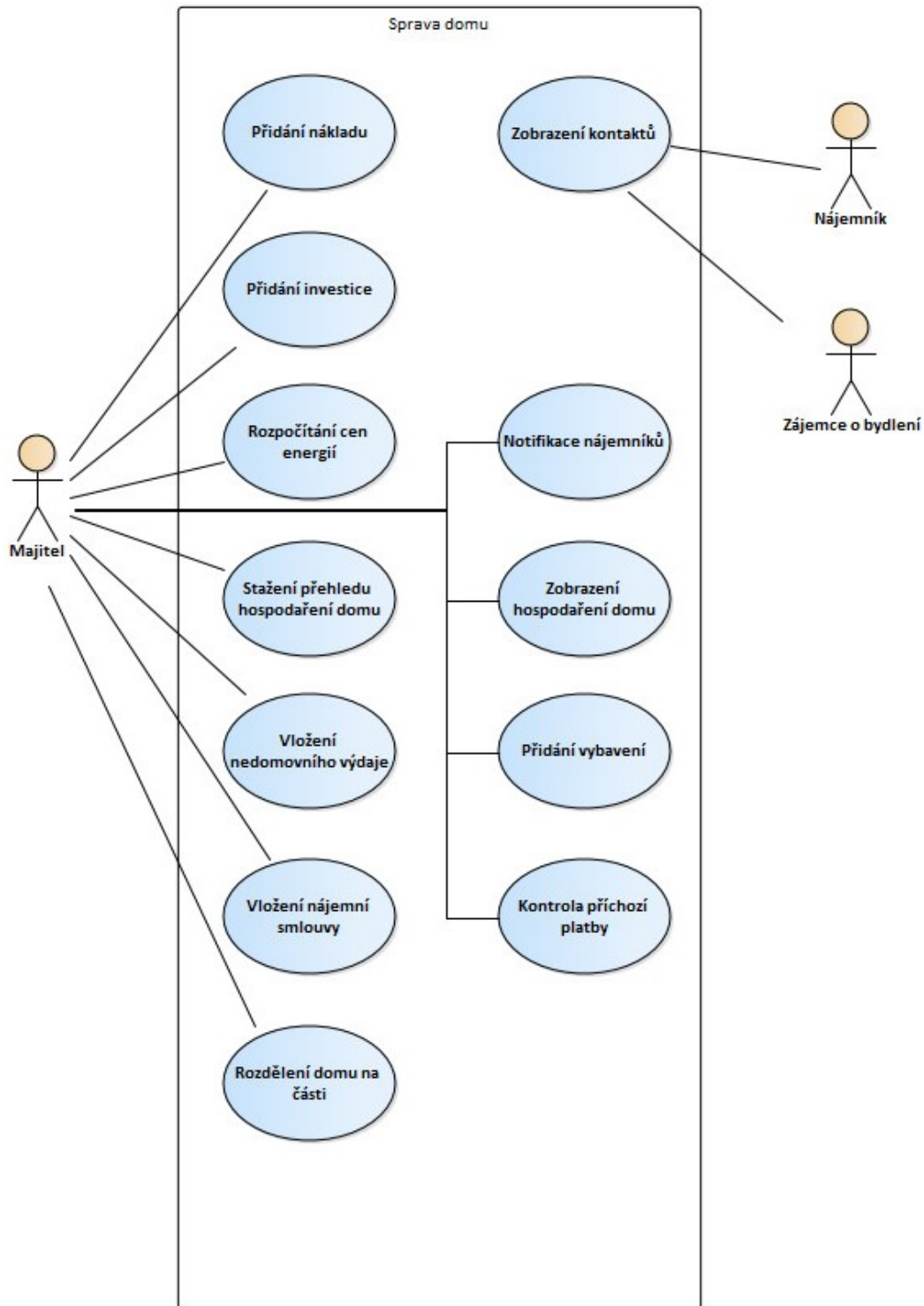
### NP8 CI/CD

V projektu bude k sestavování a nasazování aplikace využívána průběžná integrace (CI - continuous integration) a automatické nasazení (CD - continuous deployment).

Priorita požadavku: střední



■ Obrázek 3.1 Use case diagram





# Analýza konkurence

V této kapitole se zaměřuji na již dostupná softwarová řešení. Zkoumám jejich funkcionality a hodnotím je na základě mnou sesbíraných požadavků.

## 4.1 Dostupné softwary

### 4.1.1 Domsys

Webová i mobilní aplikace, která nabízí majiteli dobrou kontrolu a přehled o jeho nemovitostech. Aplikace počítá s více účty k jednomu domu a umí přiřazovat různá práva ke správě i zadávat úkoly mezi správci. Umožňuje napojení na účetní software, ukládání smluv do databáze i sledování příjmů a výdajů. [2]

### 4.1.2 Anasoft Domus

Anasoft nabízí několik softwarů a umožňuje dosti komplexní správu nejen domu. Domus majitelům umožní kontrolu měřičů spotřeby pomocí mobilní aplikace a spolupráci s účetními softwary. Software se stará primárně o správné rozpočítání a vyúčtování služeb. Až na druhém místě jsou finance okolo nájmů. Majitel se k datům může dostat přes webový portál, který slouží také nájemníkům. Nájemník si na webu může kontrolovat vyúčtování a platby. [3]

### 4.1.3 Správa bytů a nemovitostí (SBN)

SBN eviduje bytové a nebytové jednotky, vytváří předpisy měsíčních plateb, automaticky zpracovává příchozí platby a provádí vyúčtování záloh na služby. Lze napojit na účetní software Pohoda, sledovat stav technických zařízení a ukládat faktury. Vše běží lokálně na počítači majitele/správce. U vyšších verzí lze připojit počítače až dva. [4]

### 4.1.4 Jirra inkasník/vlastník

Aplikace pro majitele a správce bytových i nebytových prostor. Umožňuje spravovat nájemní domy, prodané domy i domy vedené SVJ. Nemovitost lze rozčlenit na menší části (byty i nebytové prostory). U jednotlivých částí i celku může majitel sledovat příjmy, výdaje, nebo opravy. Software dále nabízí automatickou kontrolu plateb, vyúčtování za nájem a služby. Napojení na účetnictví je pouze ve verzi vlastník. Jirra má i další podpůrné softwary pro správu dalších aspektů domu nebo jiného podnikání. [5]

### 4.1.5 Qi správa nemovitosti

Software pro usnadnění správy nemovitostí pro družstva, podniky, developery a správce komerčních nebo bytových prostor. Poskytne jim přehled o nákladech a příjmech z jednotlivých prostor a rozpočítá služby mezi nájemníky. V aplikaci lze plánovat údržbu a servis v nemovitostech. Spolupracuje s dalšími moduly od Qi a lze ho propojit s účetnictvím. [6]

### 4.1.6 CSH Domovník

Domovník umožňuje pasportizaci nemovitosti, plánovat revizí a oprav v nemovitost a evidovat osoby i vybavení v domě. Kontroluje příchozí platby a rozúčtuje služby podle dat z měřičů, počtu lidí v jednotce, nebo plochy jednotky. Je určen pro bytová družstva, majitele domů i pro správcovské firmy. Program se umí spojit s účetnictvím SYSEL. [7]

## 4.2 Způsob hodnocení

Hodnotící metriku jsem stanovil na základě nasbíraných funkčních a nefunkčních požadavků. Jednotlivým požadavkům jsem poté přiřazoval váhu podle důležitosti v mé analýze. Pokud software splňuje podmínku k udělení bodů, dostane všechny. Pokud podmínku nesplňuje, body nedostane žádné. Body se poté sečtou a já získám povědomí o tom, jak daný software požadavky mého zadavatele splňuje.

### 4.2.1 Hodnocení funkčních požadavků

#### 4.2.1.1 Automatická kontrola plateb

**1 bod:** V aplikaci lze označit, kdo již nájem za dané období platil a kdo ještě ne.

**1 bod:** Aplikace umí kontrolovat příchozí platby na zvolený účet automaticky.

#### 4.2.1.2 Rozpočítání plateb za energie a odpady

**2 body:** Aplikace umí spočítat cenu za energie a odpady pro jednotlivé byty podle počtu lidí v bytě.

**1 bod:** Aplikace umí jako vstup pro rozpočítání vzít data z měřičů a rozpočítat cenu za služby pro jednotlivé části domu podle nich.

**2 body:** V aplikaci lze vygenerované částky upravovat.

**1 bod:** Lze pomocí aplikace upozornit nájemníky (e-mail, sms) o nové výši poplatku za služby.

#### 4.2.1.3 Databáze kontaktů

**0,5 bodu:** Aplikace umožňuje majiteli uložit si kontakty na nájemníky a další důležité kontakty (údržbáři, řemeslníci). Aplikace mu musí umožnit kontakty aktualizovat.

**0,5 bodu:** Nájemník má díky aplikaci přístup k důležitým kontaktům (majitel, údržbáři).

#### 4.2.1.4 Monitorování cash flow

**3 body:** V aplikaci lze přidávat k jednotlivým bytům částky jako výdaje.

**1 bod:** Pokud kromě bytů umí aplikace evidovat náklady a příjmy i u jiných prostor domu (chodby, sklepy, fasáda).

**1 bod:** V aplikaci lze ke konkrétnímu nákladu uložit fotku účtenky nebo fakturu.

**1 bod:** Pokud lze účtenku nebo fakturu vyfotit a vložit pomocí mobilní aplikace.

**2 body:** Aplikace umí zobrazit alespoň nějaký přehled o nákladech a příjmech za celý dům.

**1 bod:** Aplikace umí vykreslit na obrazovku graf nákladů a příjmů za vybrané období.

**2 body:** Pokud si navíc majitel může zobrazit informace o prosperitě v jednotlivých částech domu.

#### 4.2.1.5 Generování podkladů pro daňové přiznání

**1 bod:** Aplikace umí sečíst evidované příjmy a náklady za vybraný kalendářní rok. Tyto dvě čísla pak umí předat majiteli.

**1 bod:** Aplikace umí zobrazit všechny příjmy a náklady za vybrané období. Rozdíl příjmů a nákladů pak zobrazí majiteli.

**1 bod:** Vložené příjmy a náklady lze z aplikace stáhnout (například excelová tabulka).

**1 bod:** Aplikace umí předat data účetnímu systému (SAP, POHODA, ...).

#### 4.2.1.6 Upozornění nájemníků

**0,5 bodu:** Pomocí aplikace lze upozornit nájemníky domu na nějakou událost.

**2 body:** Pokud se tak děje posláním sms nebo e-mailu).

#### 4.2.1.7 Evidence vybavení

**1 bod:** V aplikaci lze přidat vybavení a evidovat u něj alespoň název, cenu a datum pořízení.

**1 bod:** V aplikaci lze přidat a přiřadit jednotlivé kusy vybavení ke konkrétní části domu (byt, chodba).

#### 4.2.1.8 Přehled plateb pro nájemníky

**0,5 bodu:** Nájemník může díky aplikaci přímo zjistit, kolik má za určité období platit.

**0,5 bodu:** Aplikace umí nájemníkovi vygenerovat QR kód pro platbu za měsíční nájemné.

#### 4.2.1.9 Nájemní smlouvy

**1 bod:** Aplikace umí na základě vložených dat (z formuláře) upravit šablonu nájemní smlouvy pro konkrétního nájemníka (vložit jméno, nájem, platnost, ...).

**1 bod:** Aplikace umí nájemní smlouvy ukládat a zachovávat.

Tabulka 4.1 obsahuje hodnocení funkčních požadavků pro jednotlivé funkční požadavky, které jsem zdefinoval v první kapitole. Bodové ohodnocení každého požadavku je dáno prioritou požadavku z mé analýzy.

Ne u všech softwarů se mi podařilo najít jednoznačnou odpověď na to, zda splňuje některé požadavky. Ty jsou pak ohodnoceny „NA“ (not available) a do výsledných bodů se počítají jako nula. Nejsem ale schopen na sto procent říct, že toto by daný software neuměl, proto kromě součtu bodů, které softwary získaly uvádím i potenciální rozsah. Ten říká, kolik by mohl software dostat bodů, kdyby splnil všechny požadavky, u kterých získal „NA“. Tato nepřesnost je způsobena tím, že ne všechny softwary byly volně přístupné.

## 4.2.2 Hodnocení nefunkčních požadavků

### 4.2.2.1 Dostupnost

**6 bodů:** Webová aplikace

**4 body:** Mobilní aplikace IOS

**4 body:** Mobilní aplikace Android

**2 body:** Desktopová aplikace

Tabulka 4.2 hodnocení nefunkčních požadavků obsahuje získané body za dostupnost podle platformy. Zbylé nefunkční požadavky pro mě nemá cenu hodnotit, protože jsou buď příliš specifické, nebo bych je nebyl schopen ověřit.

Požadavek	Popis	Body	Domsys	Domus	SBN	Inkasník	Qi správa	Domovník
1a	Kontrola příchozích plateb	1	1	1	1	1	1	1
1b	Automatická kontrola příchozích plateb	1	1	1	1	1	NA	1
2a	Rozpočítání služeb podle lidí	2	2	2	2	2	2	1
2b	Rozpočítání služeb z dat měřičů	1	1	1	1	1	1	1
2c	Možnost částku upravit	2	0	0	0	NA	NA	NA
2d	Upozornění nájemníků o změně služeb	1	1	1	0	NA	NA	0
3a	Kontakty pro majitele	0,5	0,5	0,5	0	NA	NA	0,5
3b	Kontakty pro nájemníky	0,5	0	0,5	0	NA	NA	NA
4a	Náklady k bytům	3	3	0	3	3	3	1
4b	Náklady nejen k bytům	1	1	0	1	1	1	1
4c	Přiřazení účtenky (faktury)	1	1	0	0	1	NA	NA
4d	Zobrazení přehledu cash flow za celý dům	2	2	0	2	2	2	NA
4e	Vykreslení grafu příjmů a nákladů	1	1	0	NA	NA	NA	NA
4f	Přehled o jednotlivých bytech	2	2	0	2	2	2	NA
5a	Součet nákladů a příjmů za rok	1	1	1	1	1	1	1
5b	Zobrazení nákladů a příjmů za rok	1	1	1	1	1	1	1
5c	Stažení dat	1	1	NA	1	1	1	1
5d	Napojení na účetní systém	1	1	1	1	1	1	1
6a	Hromadná zpráva nájemníkům	0,5	0,5	0,5	0	0,5	NA	0,5
6b	Upozornění nájemníků e-mailem nebo sms	2	2	0	0	NA	NA	0
7a	Přidání vybavení	1	0	0	NA	1	NA	1
7b	Data o vybavení	1	0	0	NA	1	NA	1
8a	Zjištění nájemného nájemníkem	0,5	0	0	0	NA	0,5	0
8b	Generování QR kódu pro platbu	0,5	0	0	0	NA	NA	0
9a	Generování nájemních smluv	0,5	0	0	NA	NA	NA	0,5
9b	Uchovávání nájemních smluv	2	1	0	NA	NA	NA	0
	Součet bodů	31	24	11,5	16	20,5	16,5	13
	Potenciál bodů	31	24	11,5	16-20,5	20,5-31	16,5-31	13-16,5

■ **Tabulka 4.1** Hodnocení funkčních požadavků u dostupných softwarů

Požadavek	Popis	Body	Domsys	Domus	SBN	Inkasník	Qi správa	Domovník
NP1a	Webová aplikace	6	6	6	0	0	6	0
NP1b	IOS aplikace	4	4	4	0	0	4	0
NP1c	Android aplikace	4	4	4	0	0	4	0
NP1d	Offline desktopová aplikace	2	2	0	2	2	2	2
	Součet bodů	16	16	14	2	2	16	2

■ **Tabulka 4.2** Hodnocení nefunkčních požadavků u dostupných softwarů

### 4.3 Závěr hodnocení konkurence

Celkem šlo podle mého hodnocení získat 47 bodů. Nejlépe podle průzkumu dopadl Domsys se 40 body. Jeho velkou nevýhodou je, že si ho nepořídíte za méně než 897 Kč měsíčně [2]. Už i to, že za tuto cenu dostanete možnost aplikací spravovat až 30 jednotek naznačuje, že software je vhodný pro jiného zákazníka. Já se snažím udělat software pro správu jednotek bytů. Pro někoho, kdo chce mít přehled ve svých pár bytech je Domsys pravděpodobně zbytečně drahé a robustní řešení.

Většina ze zkoumaných softwarů se zaměřuje na větší majitele, družstva nebo společnosti, zatímco já se pokouším zjednodušit správu majitelům, kteří vlastní pouze pár bytových jednotek. Zkoumané softwary jsou pro tyto účely zbytečně složité a mým cílem je dosáhnout podobných výstupů s nižšími nároky na majitele. Některé softwary jsou navíc (minimálně vzhledem) zastaralé a uživatelsky nepřívětivé.



V této kapitole se budu zabývat návrhem aplikace. Nejdříve navrhnu strukturu domény. Model domény mi později poslouží jako strukturální předloha pro server a pro databázi. Zabývat se budu i návrhem vzhledu obrazovek webového klienta.

### 5.1 Doména

Zkoumanou doménou je nemovitost, ve kterém se pronajímají byty a jiné prostory. Zde navrhuji, jaká je struktura domény na základě toho, co jsem o doméně zjistil při sběru požadavků.

#### 5.1.1 Popis domény

Ze zájemce o byt se stává nájemník, když si vybere byt, podepíše nájemní smlouvu a složí kauci. Kauce je součástí nájemní smlouvy a pokud se nájemník chová k bytu tak jak má a nic v něm nepoškodí, bude mu při odstěhování vrácena. Pokud se kauce vrací nájemníkovi celá, nespadá do účetnictví a nepočítá se do základu daně. Pouze když ji majitel nevrací celou, pak musí tu část kauce, kterou nevrátil, zahrnout do účetnictví. Součástí smlouvy je také výše nájmu.

Nájemníci platí každý měsíc pravidelně nájem. Nájem se platí za konkrétní jednotku (byt, garáž, ...) a k tomu se zvlášť za některé jednotky platí služby. Služby se platí paušálně a patří mezi ně voda, odpady, úklid společných prostor, údržba výtahu, ... Služby a nájem se platí zpravidla v jedné platbě jednou měsíčně. Po zaplacení se nájem eviduje v účetnictví mezi zisky domu. Nájemník může platit nájem buď hotově, nebo bankovním převodem.

Nákladem může být buď vybavení, nebo služba. Vybavení je cokoliv, co je hmotné a zůstane to v bytě k používání (křeslo, postel, kuchyňská linka, barvy, ...). Služba je cokoliv nehmotného, za co majitel zaplatil a souvisí to s domem. Mezi služby může spadat například plyn, voda, elektřina, oprava výtahu nebo údržba domu. Náklady se evidují u konkrétní části domu. To slouží pro přehlednost ohledně utracených peněz v domě (která část je jak výdělečná).

Ke každému nákladu musí existovat nějaký daňový doklad. Tím je buď faktura, nebo účtenka. Daňový doklad je možné evidovat v účetnictví. O takové náklady si majitelé snižují zisk z domu a tím i základ daně.

Příjmy z nájmu a výdaje za dům se evidují v účetnictví. Účetnictví slouží pro přehled financí domu a lze z něj vygenerovat souhrn (vyúčtování). Vyúčtování pomůže majiteli při podání daňového přiznání, protože nebude muset manuálně přepočítávat příjmy a výdaje.

### 5.1.2 Doménový model

OntoUML model domény 5.1 odráží předchozí popis domény. Model odpovídá popisu domény a zároveň představuje strukturu budoucí implementace. Jelikož budu prototyp implementovat v JS (zadání), tak jednotlivé třídy v diagramu neodpovídají nutně třídám v implementaci. JS sice třídy už nabízí, ale ne vždy jsou vhodné použít.



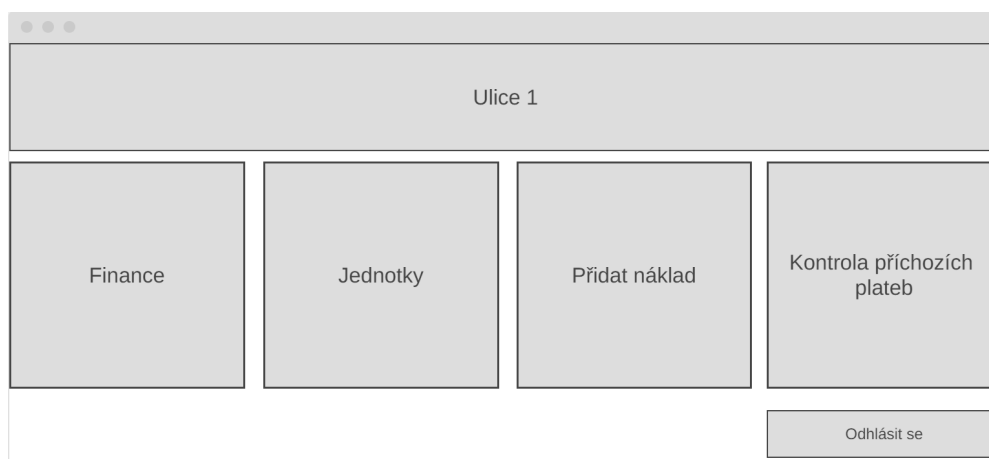
## 5.2 Návrh obrazovek

Obrazovky vznikají na základě sesbíraných požadavků. Cílem této sekce je navrhnout vzhled jednotlivých obrazovek webového klienta. Hlavním cílem je, aby bylo používání aplikace pro uživatele co nejjednodušší. Snažím se proto dát každé obrazovce pouze jeden účel, aby byla pro uživatele čitelnější a srozumitelnější.

### 5.2.1 Hlavní menu

Do hlavního menu je uživatel přesměrován hned po přihlášení. Slouží jako rozcestník pro práci s aplikací. Tlačítko „Finance“ přesměruje uživatele na stránku umožňující pohled na příjmy a náklady za celý dům. Za tlačítkem „Jednotky“ se schovává přehled částí domu jejich nájemníků. Z hlavního menu se také může majitel přímo dostat k přidání nákladu nebo kontrole příchozích plateb. Pro odhlášení se z aplikace slouží tlačítko „Odhlásit se“.

■ **Obrázek 5.2** Návrh obrazovky hlavního menu



### 5.2.2 Byt

Obrazovka bytu poskytuje na nejdůležitější informace o bytě (nájemné, nájemník, ...). Na pravé straně obrazovky se majiteli zobrazí graf ukazující, jak byl byt ve kterých měsících v zisku nebo ztrátě. Níže na obrazovce se může podívat, jaké útraty se v bytě v poslední době děly. Při potřebě bližšího prozkoumání prvku si může majitel vybraný prvek „rozbalit“ a zobrazit si tak další informace (datum, cena, ...).

Při editaci bytu může majitel měnit jakékoliv údaje. Pokud se mění nájemník, může majitel v obrazovce editace bytu kliknout na „Vystěhovat byt“ a aplikace v jednom kroku odstraní přiřazeného nájemníka a vynuluje kauci i zapůjčené klíče.

Kauce zde slouží pouze jako informace pro majitele, kolik na kauci od nájemníka vybral. Kauce, pokud se nájemníkovi vrací při odstěhování celá, se vůbec nedostane mezi příjmy. Pouze pokud se nájemníkovi celá nevrací, částka, o kterou je vrácený obnos menší, se dostává do účetnictví. A to tak, jak byly peníze z kaucí využity.

Po výběru „Vybavení v bytě“ bude majitel moci vybrat náklady, které jsou vybavením (lednička, pračka, gauč, ...). Tyto vybrané náklady se pak budou majiteli zobrazovat v tomto menu, jako vybavení bytu. Toto slouží majiteli pro lepší přehled o stáří a stavu vybavení v jednotlivých bytech. Majitel zde uvidí pouze vybavení, ne například náklady na údržbu.

Pro zobrazení informací o jiných částech domu (garáž, sklep) bude sloužit podobná obrazovka jako 5.3.

■ **Obrázek 5.3** Návrh obrazovky bytu

Byt 1

Patro: 1.  
 Nájemník: Karel Novák  
 Počet lidí v bytě: 3  
 Plocha: 80 m<sup>2</sup>  
 Pokoje: 3+1  
 Měsíční nájem: 10 000 Korun  
 Kauce: 50 000 Korun  
 Zapůjčené klíče: 3  
 Služby: 3 000 Korun

Vybavení a investice:  
 Lednička  
 Pračka  
 Revize bojleru  
 Myčka  
 Sušička  
 Kuchyňská linka

Přidat náklad    Editovat byt    Vybavení v Bytě

graf výdělku za poslední rok

### 5.2.3 Přehled financí

V přehledu financí uvidí majitel jednotlivé položky (příjmy a náklady) za jím zvolené období. Pod Formulářem kterým majitel vybere období se mu zobrazí celkové příjmy a náklady a jejich rozdíl. Níže majitel uvidí jednotlivé položky, které si může „rozbalit“ a zobrazit si tak o nich podrobnější informace. Na pravé straně majitel uvidí graf za vybrané období a vpravo nahoře se tlačítkem „Stáhnout vyúčtování“ dostane k formuláři na stažení vyúčtování v excelové tabulce.

■ **Obrázek 5.4** Návrh obrazovky přehledu financí

Přehled financí

Hospodaření za:  
 od   
 do

Příjmy celkem: 100 000  
 Výdaje celkem: 50 000  
 Zisk: 50 000

Výdaje ve vybraném období  
 Služby dům  
 Lednice byt 4  
 Pračka byt 1

Příjmy ve vybraném období  
 Nájem byt 1  
 Nájem byt 2  
 Nájem byt 3  
 Nájem byt 4  
 Nájem byt 5  
 Nájem byt 6

Stáhnout vyúčtování

Graf příjmů a výdajů za dům

## 5.2.4 Přehled jednotek

V přehledu jednotek uvidí majitel všechny části domu, které má vložené do systému. U každé části se mu zobrazuje, kde se část nachází, kolik se v platí nájem, kdo je aktuálním nájemníkem a do kdy je platná nájemní smlouva. U bytů se navíc zobrazuje, kolik lidí v bytě bydlí.

V editačním módu, do kterého se majitel přepne tlačítkem „Editovat“, bude možné jednotky přidat nebo smazat, samotná editace jednotek bude možná pouze v detailní obrazovce konkrétní jednotky. Editační mód zde existuje pouze proto, aby majitel nepřidal nebo neodstranil nějakou jednotku omylem (majitel se musí do editačního módu vědomě přepnout).

Tlačítko „Rozpočítat služby“ zobrazí formulář, který vypočítá zálohu za služby pro jednotlivé jednotky.

■ **Obrázek 5.5** Návrh obrazovky přehledu jednotek

Garáže	nájem	jméno	počet lidí	konec smlouvy
Garáž 1	5 000	Josef Dvořák		31. 12. 2022
Garáž 2	5 000	....		31. 12. 2023
1. patro				
Byt 1	15 000	Karel Novák	3	31. 12. 2024
Byt 2	12 000	....	2	31. 12. 2023
2. patro				
Byt 3	16 000	....	4	31. 8. 2022
Byt 4	14 000	....	2	31. 12. 2021

Rozpočítat služby      Editovat

## 5.2.5 Přidání nákladu

Majitel může do aplikace pomocí formuláře přidávat náklady. Musí vybrat, do které jednotky nebo části domu náklad patří, musí ho pojmenovat a zadat cenu a datum nákupu. Volitelné u nákladu jsou popis a poznámka.

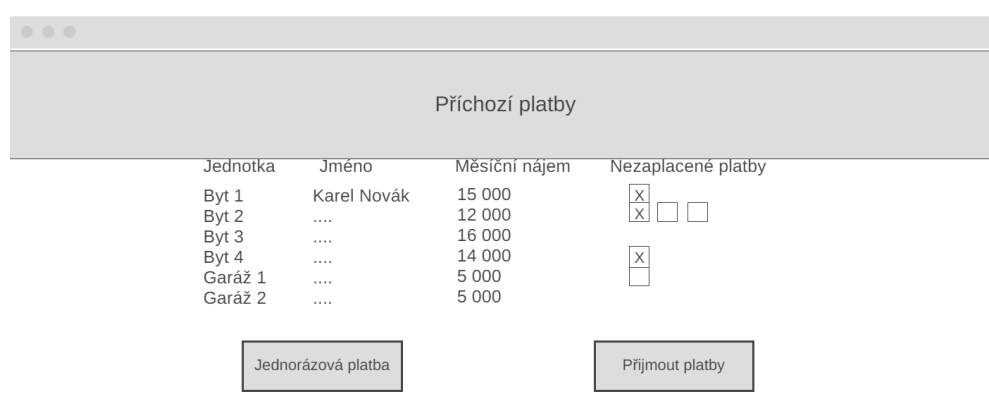
Po stisknutí tlačítka „Přidat“ se vytvoří nová položka, kterou majitel následně uvidí mezi výdaji buď u jednotlivých jednotek, nebo v přehledu financí za celý dům. Po přidání nákladu se také formulář resetuje a majitel může přidat další položku. Tlačítko „Zpět“ slouží pro návrat do hlavního menu.

■ **Obrázek 5.6** Návrh brazovky přidání nákladu


### 5.2.6 Příchozí platby

Tato obrazovka doplňuje napojení aplikace na banku. Majitel vidí všechny části domu, kdo je má v současnosti pronajaté a kolik činí měsíční nájem. V posledním sloupečku je přehled dosud nezaplacených plateb za každou jednotku. Každý měsíc přibude u každé obsazené jednotky checkbox, jehož zaškrtnutím majitel označí, že tato platba proběhla. Aby se ale platby přidaly do aplikace, musí ještě dojít k potvrzení tlačítkem „Přijmout platby“.

Tlačítkem jednorázová platba se majiteli otevře dotazník podobný přidávání nákladu, ve kterém může přidat jednorázovou příchozí platbu. Jednorázovou platbou může být například nevrácení celé kauce nájemníkovi nebo třeba pronajmutí zahrady na oslavu.

■ **Obrázek 5.7** Návrh brazovky příchozích plateb


Jednotka	Jméno	Měsíční nájem	Nezaplacené platby
Byt 1	Karel Novák	15 000	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Byt 2	....	12 000	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Byt 3	....	16 000	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Byt 4	....	14 000	<input checked="" type="checkbox"/>
Garáž 1	....	5 000	<input type="checkbox"/>
Garáž 2	....	5 000	

### 5.2.7 Nájemník

Obrazovka nájemníka umožní majiteli zobrazit si bližší informace o konkrétním nájemníkovi v domě. Majitel uvidí jméno, prostory které má daný člověk pronajaté a kontaktní údaje. Tlačítkem

„Editovat“ se majitel přepne do editačního režimu a bude moct upravovat údaje o nájemníkovi nebo ho vystěhovat z pronajatých prostor.

Údaj o nájemníkovi, zda platí spolehlivě slouží ke zjednodušení kontroly příchozích plateb. U těch nájemníků, kteří budou označeni že spolehlivě platí, bude na obrazovce „Příchozí platby“ každý nový checkbox automaticky zaškrtnutý. Toto slouží jako podpůrná informace pro majitele, že tento nájemník pravděpodobně zaplatil a majitel tomu tak nemusí věnovat takovou pozornost. Samozřejmě, že pokud se nájemník s platbou zpozdí, může majitel při kontrole platebu zrušit zaškrtnutí příslušného checkboxu.

■ **Obrázek 5.8** Návrh obrazovky nájemníka

## 5.2.8 Stažení vyúčtování

Ve formuláři vyúčtování, do kterého se majitel dostane ze stránky „Finance“, se specifikují parametry toho, co se dostane do excelové tabulky, kterou si majitel bude moct vygenerovat a stáhnout. Vyúčtování se dá stáhnout buď za celý dům, nebo pouze za vybranou část. Majitel musí specifikovat dobu, za kterou chce vyúčtování vygenerovat (typicky kalendářní rok). Může také specifikovat, zda chce v tabulce mít pouze příjmy, pouze náklady, nebo obojí. Stažená tabulka by měla majiteli usnadnit podání daňového přiznání.

■ **Obrázek 5.9** Návrh obrazovky stažení vyúčtování

## 5.2.9 Rozpočítání služeb

Aplikace může majiteli pomoci s vypočítáním a nastavením zálohy za služby. Služby se (zatím pouze) rozpočítávají podle počtu lidí bydlících v bytě. Majitel napíše cenu služeb, kterou zaplatil



za celý dům a zvolí, za jaké období tato cena je. Tlačítkem „Vypočítat návrh“ mu systém vyplní textové pole „Návrh měsíční ceny na osobu“ návrhem na cenu za služby pro jednoho člověka.

Pokud s číslem není majitel spokojen (například ho chce zaokrouhlit), může toto číslo měnit. Když bude s cenou spokojen, může říct aplikaci ať změni cenu služeb u všech bytů v domě. Udělá to stisknutím tlačítka „Použít ke změně služeb“. Pro lepší představu o vývoji ceny za služby uvidí majitel poslední platnou cenu za osobu.

Při změně zálohy za služby budou nájemníci upozorněni e-mailem, který majitel může upravovat pod tlačítkem „Upravit notifikační e-mail“. Tlačítko „Zpět“ vrátí majitele do přehledu jednotek.

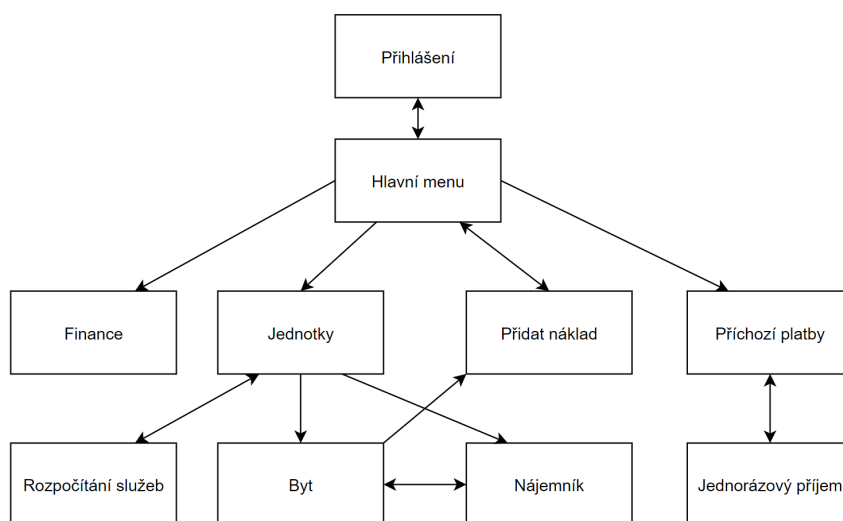
■ **Obrázek 5.10** Návrh obrazovky rozpočítání služeb

### 5.2.10 Přechodová mapa

Pro lepší přehled, jak jsou na sebe navzájem obrazovky navázány, přikládám přechodovou mapu. Šipka značí, že na obrazovce se nachází tlačítko, kterým se lze dostat na obrazovku další. Ne vždy je možná cesta zpět vymyšlena tlačítkem, ale často se spoléhá na prohlížečové zpět, protože by se někde význam tlačítka „Zpět“ mohl měnit v závislosti na tom, ze které obrazovky uživatel přišel.

Některé přechody nejsou realizovány tlačítkem. Pak se odkazem na další stránku stává například jméno nájemníka, nebo číslo bytu (přechod mezi bytem a nájemníkem).

■ **Obrázek 5.11** Návrh přechodové mapy



## 5.3 Procesy

V této části práce se snažím přiblížit a přesněji popsat procesy v domě. Také se snažím ukázat, jak se daný proces realizuje pomocí nové aplikace, nebo alespoň podle toho, jak jsem navrhl její uživatelské rozhraní.

### 5.3.1 Přidání nákladu

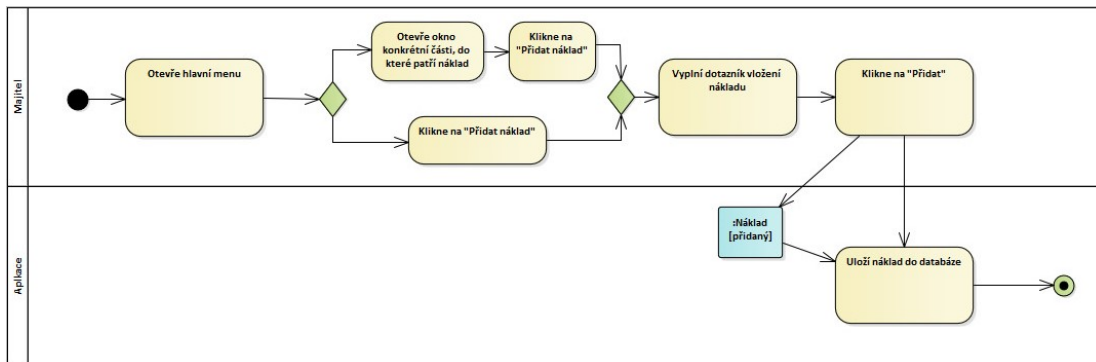
Položka, která je přidána do aplikace jako náklad se dostane na konci roku do vyúčtování a bude majiteli snižovat základ daně. Takové položky mohou být například vybavení nebo rekonstrukce bytů, které slouží k pronájmu.

Proces přidání začíná v hlavním menu aplikace. Zde může majitel přímo zvolit „Přidání nákladu“ a dostane se tak k dotazníku, který vyplní údaji o nákladu a potvrdí tlačítkem „Přidat náklad“.

Jiná cesta k dotazníku vede z předem vybrané části domu. Pokud si majitel nejříve prohlíží danou část domu, nemusí se vracet do hlavního menu a náklad může přidat přímo z vybrané stránky, znovu přes tlačítko „Přidat náklad“.

Po přidání nákladu ho aplikace uloží do databáze a zahrne ho při zobrazování přehledů a ve vyúčtování.

■ **Obrázek 5.12** Proces přidání nákladu

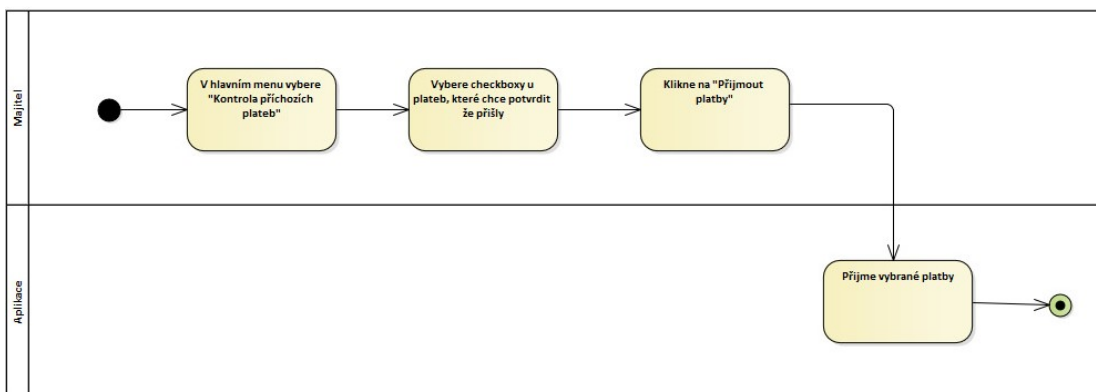


## 5.3.2 Přijetí plateb za nájem

Protože zatím nebude aplikace připojená k bance, bude nutné zadávat příchozí platby ručně. Jediný možný příjem je z nájmu (byty, sklepy, garáže, fasády na reklamu, ...). Tyto platby se dějí pravidelně a je tedy možné pouze požádat majitele o potvrzení, zda byla tato platba provedena.

Tento proces je v aplikaci řešený tak, že si majitel nejdřív otevře obrazovku „Příchozí platby“. V ní vidí, které platby aplikaci chybí, aby byly přijaty. U plateb, které už se provedly a majitel je chce aplikaci potvrdit, zaškrtně příslušné checkboxy a potom klikne na „Přijmout platby“, aplikace pak vybrané platby zpracuje jako provedené. V případě, že bude někdo dlužit více nájmu, bude u příslušné jednotky více checkboxů.

■ **Obrázek 5.13** Proces přijetí plateb



## 5.3.3 Změna nájemníka

### 5.3.3.1 Vystěhování nájemníka

Když se nájemník stěhuje pryč, je potřeba zkontrolovat stav bytu a přepočítat klíče. Je-li vše v pořádku, s nájemníkem se podepíše předávací protokol a on dostane zpět svou kauci a majitel klíče.

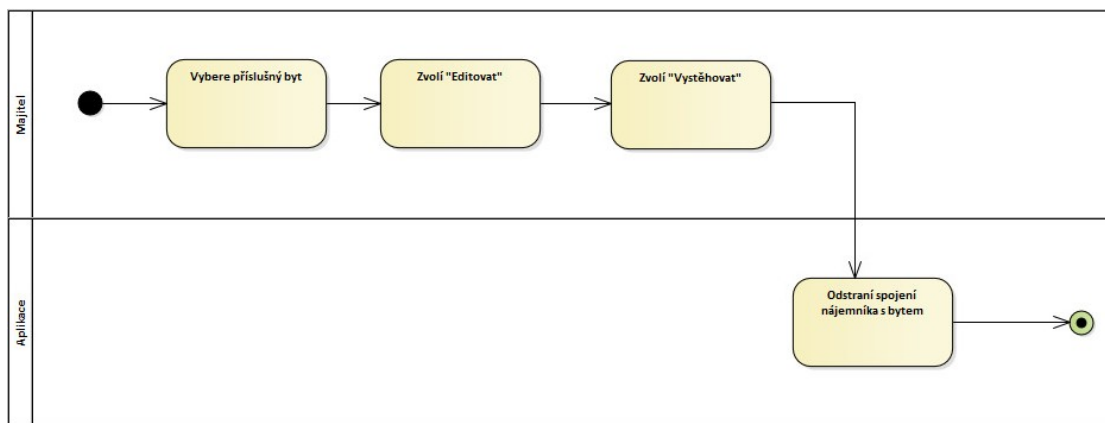
V situaci, kdy jeden nájemník střídá druhého se plyn a elektřina přehlásí přímo z nájemníka na nájemníka. Pokud se byt na nějakou dobu uvolní, elektřina a plyn se přehlásí na majitele domu. V aplikaci se toto děje tak, že se majitel prokliká ke konkrétnímu bytu a klikne na „Editovat“.

U jména nájemníka potom klikne na tlačítko „Vystěhovat“, byt přijde o přiřazeného nájemníka a nebude generovat zisk (nebude se dát přidat příchozí platba). Na koho je nahlášena elektřina a plyn zatím aplikace neřeší.

Pokud se nájemníci v bytě střídají, místo „Vystěhovat“ zvolí majitel „Změna nájemníka“ a vybere nového nájemníka.

Aplikace neřeší kauce hlouběji, než jako informaci pro majitele. Kauce se váže ke konkrétní jednotce, proto tato informace u bytu zůstane i při vystěhování nebo nastěhování nájemníka. S kaucí může majitel pracovat pouze tak, že ji v editačním módu bytu nastaví.

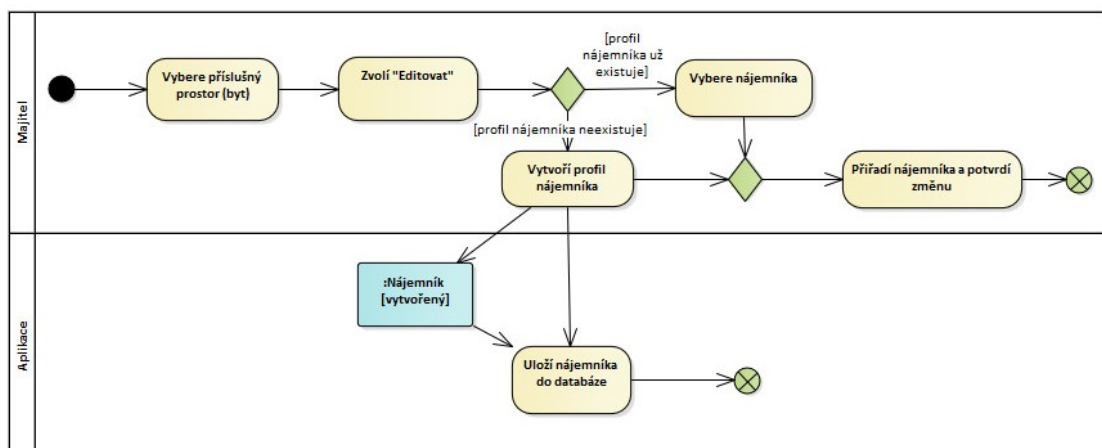
**Obrázek 5.14** Proces vystěhování nájemníka



### 5.3.3.2 Přiřazení nájemníka k bytu

Při přiřazování nového nájemníka k bytu musí majitel tento byt najít v systému a kliknout na editovat. Pokud má již nájemníka v systému vedeného (například už bydlel v jiném bytě), tak majitel klikne na „Změna nájemníka“ a vybere ze seznamu toho nájemníka, kterého chce přiřadit. Jestli je nájemník nový, musí mu majitel nejdříve vytvořit profil a ten pak k bytu přiřadit.

**Obrázek 5.15** Proces přiřazení nájemníka k bytu



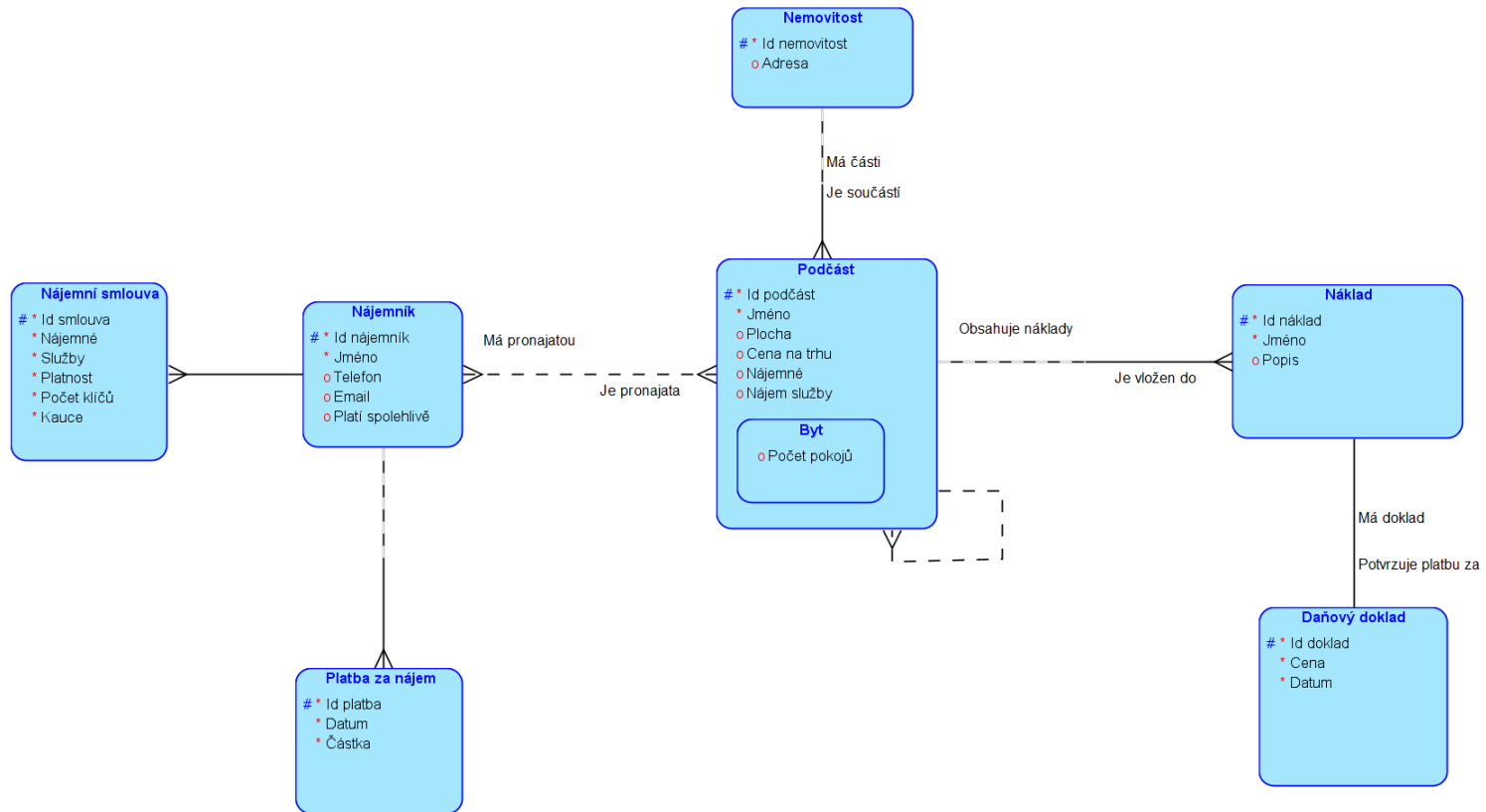
## 5.4 Databáze

Model databáze 5.16 vznikl z doménového modelu 5.1. V databázovém modelu jsou ty třídy z doménového modelu, které uchovávají data.

Dům bude třeba členit na části, dokonce struktura členění by mohla být víceúrovňová (část má další části). Snahou je části pronajímat nájemníkům, kteří budou za nájem platit. Nájemníci jsou k pronajaté části vázáni nájemní smlouvou, kterou bude moct majitel v pozdějších verzích do aplikace vložit (scan, pdf). K podčástí se dá přidat náklad, ke kterému se bude dát přidat daňový doklad (účtenka, faktura).

Nájemné se váže k nájemníkovi, nikoliv k podčásti proto, že pro majitele je snazší kontrolovat platbu od člověka než za část. Například pokud má nájemník pronajatý byt a garáž, tak majitel může kontrolovat platbu za obě tyto části od jednoho nájemníka jednodušeji, než kdyby to byly dvě různé položky.

■ Obrázek 5.16 Návrh databáze



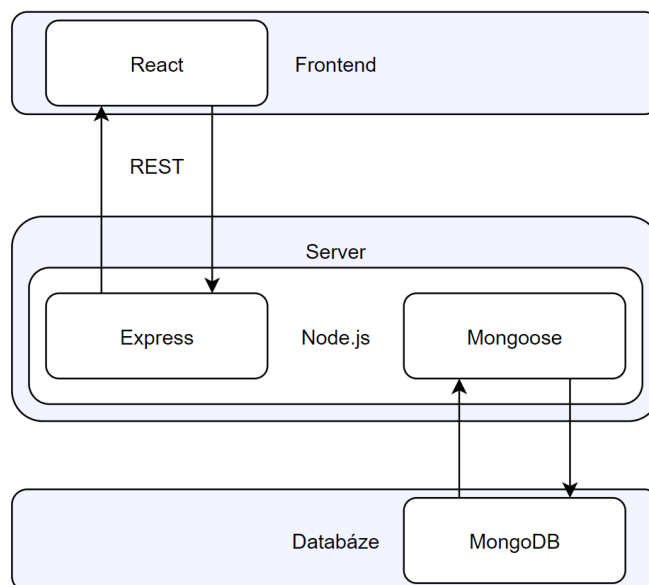
# Implementace

V této kapitole se budu věnovat implementaci prototypu. Nejdřív přestavím jeho architekturu a technologie, které jsem použil. Pak detailně popíši webové rozhraní aplikace a pokážu na změny oproti návrhu. Dále představím RESTové API, které používají pro komunikaci server a frontend a krátce se zmíním o serveru a databázi.

## 6.1 Architektura a technologie

Prototyp je postaven na třívrstvé MVP (model-view-presenter) architektuře [8]. Klient je webový a je naprogramovaný v Reactu, což je knihovna pro javascript. Server běží na Node.js s frameworky Express a Mongoose. Klient komunikuje se serverem přes REST API. Databázi jsem zvolil MongoDB.

■ **Obrázek 6.1** Architektura prototypu



### 6.1.1 React

React je knihovna pro javascript na vytváření uživatelského rozhraní. Umožňuje jednoduše vytvářet interaktivní pohledy pomocí programátorem definovaných komponent. Komponenty jsou definovány pomocí HTML tagů a mohou se dynamicky měnit díky JS. [9]

Kromě webové aplikace lze pomocí Reactu vytvářet i aplikace mobilní, děje se tak pomocí knihovny React Native. Pro mobilní aplikaci není nutné přepisovat celý kód do jiného jazyka, stačí změnit vykreslování na obrazovku. Logika může pro mobilní aplikaci zůstat stejná (stejný zdrojový kód) jako pro webovou.

### 6.1.2 Node.js a Express

Node.js je open source multiplatformní prostředí pro vývoj v jazyce java script (JS) na straně serveru. Node.js, stejně jako JS, nemá vlákna ale je event-driven, což zjednodušuje vývoj a zabraňuje deadlockům. [10]

Express je nejpoblárnější framework pro Node.js. Zjednodušuje především vývoj REST API. To je v Expressu mnohonásobně jednodušší než v základním Node.js. [11]

### 6.1.3 MongoDB

MongoDB je NoSQL databáze. Místo tabulek, jako relační databáze, ukládá data do dokumentů a ty skládá do kolekcí. Dokumenty jsou ve formátu JSON, který je založen na JS objektech. MongoDB je hojně používaná při Node.js. [12]

### 6.1.4 Mongoose

Mongoose je knihovna pro node.js zjednodušující práci s MongoDB.

K definování struktury dokumentů slouží schémata. V nich se pomocí notace podobné JSON definují jednotlivé atributy dokumentu a jejich datové typy. Ze schémat se lze vytvořit modely. Modely jsou konstruktory objektů definovaných pomocí schémat. Instancováním modelu vznikne dokument, který lze uložit do MongoDB.

Každému dokumentu přidá Mongoose atribut „\_id“ sloužící jako unikátní identifikátor. [13]

## 6.2 Uživatelské rozhraní prototypu

V této části práce představím a popíšu obrazovky webového klienta prototypu. Tato část zároveň může sloužit uživateli jako příručka pro pochopení jednotlivých obrazvek v prototypu.

### 6.2.1 Přihlašování

Na obrazovce přihlašování se vyskytuje pouze formulář pro přihlášení uživatele. Uživatel zadá uživatelské jméno a heslo a stiskne tlačítko „Přihlásit se“. Pokud zadá uživatel platné údaje, server pošle zpět webovému klientovi JSON web token (JWT), kterým se potom klient prokazuje při dalších dotazech na server. Pokud uživatel zadá špatné uživatelské jméno nebo heslo, zobrazí se mu informace, že zadal neplatnou kombinaci a že se má zkusit přihlásit znovu.

JWT se po přihlášení uloží do cookie ve webovém prohlížeči a je platný 24 hodin. Po úspěšném přihlášení je uživatel přesměrován do hlavního menu.



■ **Obrázek 6.2** Přihlašovací obrazovka

Uživatelské jméno

Heslo

Přihlásit

## 6.2.2 Navigační lišta

Navigační lišta se zobrazuje na všech obrazovkách kromě přihlašovací. Nápis vlevo „Správa domu“ slouží jako odkaz do hlavního menu odkudkoliv z aplikace. Na pravé straně má uživatel možnost se odhlásit kliknutím na „Odhlásit se“.

V dalších verzích na navigační lištu ještě přibude tlačítko pro správu účtu uživatele. Prototyp v tuto chvíli neumožňuje uživateli jeho účet jakkoliv měnit.

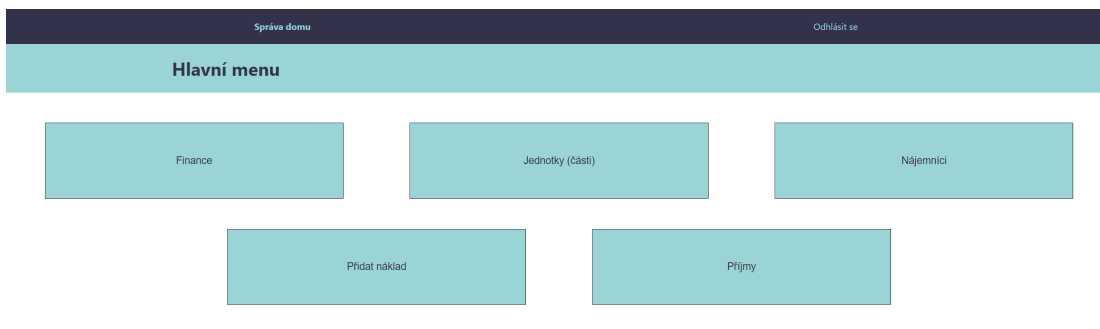
Pod navigační lištou se na každé obrazovce vždy nachází nadpis, na které obrazovce se majitel právě nachází. V případě většiny obrazovek je to přímo jméno dané obrazovky. Pouze v případě, že si majitel prohlíží stránku nájemníka nebo jednotky, je v hlavičce napsáno jméno nájemníka, nebo typ a číslo jednotky (například „Byt 1“).

■ **Obrázek 6.3** Navigační lišta

## 6.2.3 Hlavní menu

Hlavní menu slouží jako rozcestník pro práci s aplikací. Oproti návrhu zde přibylo tlačítko „Nájemníci“ a za ním se skrývající nová obrazovka zobrazující přehled nájemníků s možností nájemníky přidávat.

■ **Obrázek 6.4** Hlavní menu



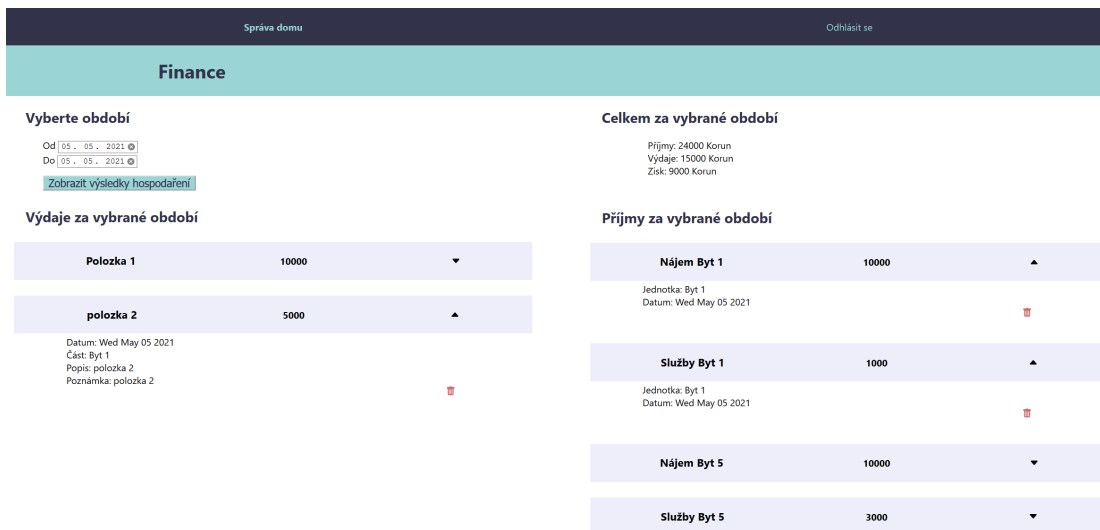
## 6.2.4 Finance

Stránka financí zobrazí majiteli přehled o příjmech a nákladech za vybrané období. Období majitel specifikuje ve formuláři vlevo nahoře, kde vybere datum „od, do“. Tlačítkem „Zobrazit výsledky hospodaření“ potom volbu data potvrdí. Vpravo nahoře se mu pak vyplní částky za součet všech příjmů, nákladů a jejich rozdíl za zvolené období.

Po potvrzení období se majiteli také zobrazí všechny příjmy a náklady seřazené od nejstarších po nejnovější. V levém sloupečku uvidí majitel náklady, v pravém příjmy. V základní přehledu je vidět pouze jméno nákladu nebo příjmu a částka v korunách. Při kliknutí na šipku vpravo u každé položky si může majitel zobrazit další detaily o ní (jednotku do které patří, datum). Kliknutím na ikonku koše vpravo dole u každé „rozbalené“ položky může položku odstranit.

Oproti návrhu neumí prototyp zatím zobrazit graf nákladů a příjmů. Umí je pouze sečíst a zobrazit majiteli. Zatím se také nedá stáhnout vyúčtování. Prototyp umí data pouze zobrazit na webu.

■ **Obrázek 6.5** Finance



## 6.2.5 Přehled jednotek

V přehledu jednotek uvidí majitel abecedně seřazené všechny jednotky a části, které v domě má. Jednotkou se rozumí například byty, garáže a sklepy. Zpravidla to, co se dá pronajímat. Pro lepší

přehled ale může majitel přidat i nějakou jinou část, například výtah a ukládat pak náklady k této konkrétní části a zjistit tak, kolik provoz této konkrétní části stojí.

Při načtení stránky je u všech jednotek vidět pouze typ jednotky (části), číslo a aktuální nájemník. Kliknutím na typ nebo číslo jednotky se majitel dostane na detailní stránku jednotky, kliknutím na jméno nájemníka se dostane na stránku nájemníka.

Pokud si majitel chce zobrazit bližší informace o jednotce, ale nechce jít na její stránku, může si detailní přehled zobrazit kliknutím na šipku vpravo. Pak se mu „rozbalí“ bližší informace o jednotce: nájemné, služby a kolik lidí v ní bydlí. Pokud chce detail jednotky schovat, klikne znovu na šipku vpravo. Detailnější informace i s možností modifikace jednotky získá majitel až na stránce jednotky.

Pokud chce majitel nějakou jednotku přidat, pak si dole na stránce rozbalí formulář „Přidat část“. Musí pak vyplnit typ jednotky (byt, garáž, ...) a její číslo. Číslo je nutné vyplnit vždy, i když bude jednotka jediná svého typu. Jinak by totiž mohlo snadno dojít k duplikaci nějaké jednotky nebo části.

■ **Obrázek 6.6** Přehled jednotek

Byt 1	Nájemník jedna	▲
Nájemné: 10000 Služby: 1000 Počet lidí v bytě: 1		
Byt 2	Nájemník dva	▲
Nájemné: 12000 Služby: 2000 Počet lidí v bytě: 2		
Byt 3	Nájemník tři	▼
Byt 4	Nájemník čtyři	▼
Byt 5	nájemník pět	▼
Garaz 1	Nájemník jedna	▼

**Přidat část** ▲

Typ jednotky  Číslo jednotky

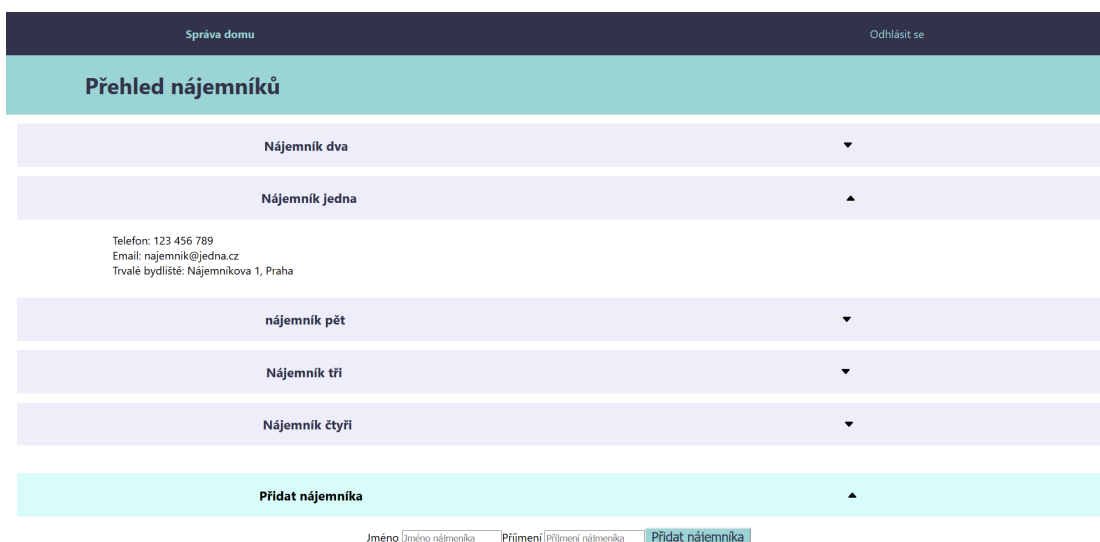
## 6.2.6 Přehled nájemníků

Přehled nájemníků je nová obrazovka oproti návrhu. V době návrhu to nevypadalo, že by měla být potřeba. Dokonce i při implementaci jsem s ní nepočítal až do poslední chvíle, než jsem začal pracovat s nájemníky detailněji. Až v pozdější fázi se ukázalo, že nájemník lze pouze svázat s jednotkou a nemůže existovat zvlášť, není si ho kde zobrazit, pokud zrovna nikde nebydlí. Nebyla by to asi zásadní chyba, kdyby byl nájemník pouze součástí jednotky, ale někdy se může stát, že se nájemník přestěhuje z jednoho bytu do druhého. Proto jsem vytvořil obrazovku přehledu nájemníků, aby bylo možné pracovat i s nájemníky, kteří zrovna nemají nic pronajatého.

Na obrazovce jsou pod sebou v abecedním pořadí vidět všichni nájemníci, které má majitel v systému. Kliknutím na šipku vpravo si může majitel zobrazit další informace o nájemníkovi (telefon, e-mail, trvalé bydliště). Pokud by chtěl majitel data o nájemníkovi editovat, musí přejít na obrazovku nájemníka. To udělá kliknutím na jméno Nájemníka v přehledu nájemníků.

Na této obrazovce je také možné nájemníka vytvořit. Podobně jako při vytváření nové části, nejdříve majitel rozbálí formulář dole na obrazovce kliknutím na „Přidat nájemníka“. Majitel musí poté vyplnit formulář jménem a příjmením nájemníka a kliknutím na tlačítko „Přidat nájemníka“ se vytvoří nový nájemník s daným jménem a příjmením. Jeho další editace je možná až na stránce tohoto nájemníka.

■ **Obrázek 6.7** Přehled nájemníků



## 6.2.7 Přidání nákladu

Obrazovka přidání nákladu slouží k ukládání výdajů do domu. Oproti návrhu se na ní nic, kromě detailů na vzhledu, nezměnilo.

Pokud chce majitel přidat náklad musí vybrat část, do které náklad patří. Pak musí zadat jméno položky, cenu a datum nákupu. Může přidat popis nebo poznámku. Tlačítkem „Přidat“ pak majitel položku vloží do aplikace, zobrazí se mu informace o tom, že položka byla přidána a data o položce zmizí z formuláře. Tlačítkem „Zpět“ se majitel může vrátit do hlavního menu.

■ **Obrázek 6.8** Přidání nákladu



## 6.2.8 Příchozí platby

Tato obrazovka slouží k přidávání příjmů do aplikace. Příjmy jsou většinou pravidelné za nájem jednotek. Mohou se občas vyskytnout i nepravidelné příjmy, například za pronájem zahrady na oslavu.

Jednorázový příjem lze přidat v dolní části obrazovky. Majitel musí vybrat část, do které příjem spadá, jméno příjmu, částku a a datum. Stisknutím tlačítka “Přidat příjem” přidá aplikace příjem do databáze a resetuje formulář.

Pro pravidelné příjmy slouží tabulka v horní části obrazovky. V tabulce je pro každou část domu připravený jeden řádek, ve kterém jsou nejdříve typ a číslo jednoty a jméno aktuálního nájemníka.

Potom následují dvě textová pole, nájem a služby. Ta jsou předvyplněná čísla, která lze konfigurovat u každé jednotky. Následuje datum a tlačítko „Přijmout platbu“, kterým se přidá příjem zvlášť za nájem a za služby k vybranému datu ke konkrétní jednotce. Po úspěšném přidání platby se vedle tlačítka objeví zelená fajfka, která značí že přidání proběhlo úspěšně.

Předvyplněná pole za nájem a služby lze v tabulce upravovat. Lze tedy například přijmout pouze část platby. Stává se totiž, že jednotku má pronajato více lidí a každý platí nájemné zvlášť. Takto je možné kontrolovat platbu od každého plátce.

Prototyp sice umí platby evidovat, ale nehlídá za majitele, které nájemy jsou již za daný měsíc zaplacený. Zelená fajfka po přidání platby je pouze informativní v tom smyslu, že se právě teď podařila platba přidat. Při dalším načtení obrazovky tam již fajfky nebudou. Je zatím tedy na majiteli, aby správně přidával platby ke správným měsícům.

### Změna oproti návrhu

V návrhu jsem vymyslel variantu s checkboxy za každý nezaplacený měsíc. Tato varianta ale neumožňovala částečné platby. Další problém s touto variantou byl, že nebylo jednoduché říct, ke kterému datu se má checkbox vztahovat.

Mohlo by to být datum kdy majitel tuto platbu přijme. V tomto případě by mohl vzniknout problém, pokud by majitel přijal platby v jiném měsíci, případně roce, než ji nájemník poslal. Platba by se pak dostala jinam než kam měla přijít a to by mohlo způsobit problémy (příjem je ve špatném účetním období).

Druhá možnost data, ke kterému by checkbox mohl patřit, je nějaké vybrané datum v měsíci, ve kterém se platba měla uskutečnit. Zde upozornila na problém současná situace, kdy bylo pro nájemníky možné odložit si nájem. Potom by se ale znovu mohlo stát, že platba se dostane do jiného účetního období. Proto jsem zvolil variantu, kdy musí majitel vybrat datum platby.

■ **Obrázek 6.9** Příchozí platby

**Příjmy**

Pravidelné příjmy z nájmu

Část	Nájemník	Nájem	Služby	Datum	Potvrzení platby
Byt 5	nájemník pět	10000	3000	05. 05. 2021	Přijmout platbu ✓
Garaz 1	Nájemník jedna	15000	100	05. 05. 2021	Přijmout platbu
Byt 1	Nájemník jedna	10000	1000	05. 05. 2021	Přijmout platbu ✓
Byt 2	Nájemník dva	12000	2000	05. 05. 2021	Přijmout platbu
Byt 3	Nájemník tři	10000	5000	05. 05. 2021	Přijmout platbu
Byt 4	Nájemník čtyři	18000	5000	05. 05. 2021	Přijmout platbu ✓

Jednorázový příjem

Místo/Vyberte jednotku | Jméno příjmu | Částka | Datum | Přidat příjem

## 6.2.9 Nájemník

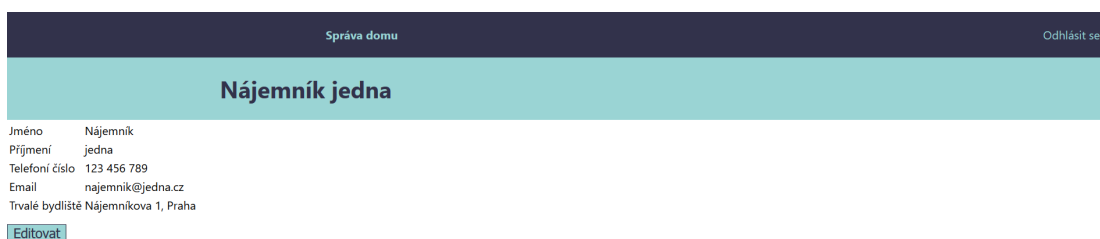
Při otevření obrazovky nájemníka jsou vidět údaje o něm (jméno, příjmení, telefon, e-mail, trvalé bydliště). Pokud by chtěl majitel něco o nájemníkovi změnit, musí se přepnout do editačního módu. To udělá tlačítkem „Editovat“.

V editačním módu se informace o nájemníkovi zobrazí v textových polích, které je možné editovat. Majitel tak může napsat nebo smazat cokoliv z kteréhokoliv pole. Pokud má nájemníka upraveného jak chtěl, může tlačítkem „Potvrdit změny“ potvrdit nové informace o nájemníkovi. Pokud by něco upravil, ale rozmyslel si to a nechtěl změny uložit, stiskne opět tlačítko „Editovat“ a z editačního módu se dostane zpět do náhledu na nájemníka bez toho, aby se uložily změny v textových polích.

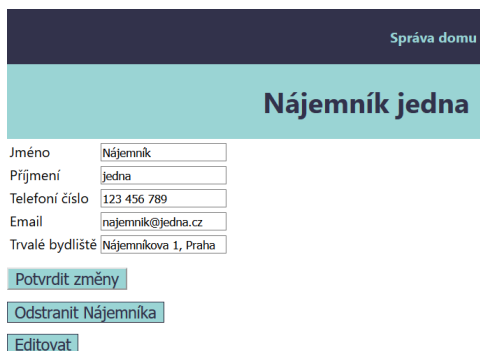
Nájemníka je možné smazat pouze v editačním módu, a to tlačítkem „Odstranit nájemníka“. Mazání je pouze zde, aby se minimalizovala šance, že majitel smaže nájemníka omylem.

Na obrazovce nájemníka by do budoucna mohly být například také informace o jím pronajatých jednotkách, nájemních smlouvách nebo platbách. To jsem ale bohužel v rámci prototypu nestihl implementovat.

■ **Obrázek 6.10** Obrazovka nájemníka



■ **Obrázek 6.11** Editace nájemníka



## 6.2.10 Jednotka (část)

Obrazovka jednotky zobrazí majiteli bližší informace a data o konkrétní jednotce (části).

### Informace a editování

Vlevo nahoře na obrazovce jednotky jsou vidět základní informace o ní (patro, rozloha, nájemník, nájemné, služby, ...). Důležitá informace pro fungování programu je, kolik se v ní platí nájem a kolik stojí služby. Podle ceny nájmu a služeb se totiž automaticky vyplňuje formulář pro příchozí platby.

Další číslo, které využije jiný formulář, bude v některé z dalších verzí počet lidí žijících v bytě. Prototyp zatím neumí mezi jednotky rozpočítat služby, ale až se tato funkcionalita bude realizovat, tak se budou služby rozpočítávat právě na základě počtu lidí žijících v bytě.

Pokud chce majitel měnit údaje o jednotce, musí se přepnout do editačního režimu tlačítkem „Editovat“. V editačním režimu lze data o jednotce měnit stejně, jako se mění data o nájemníkovi v obrazovce nájemníka. Jiný přístup je ale k nájemníkovi obývajícím byt. Toho majitel vybírá ze seznamu nájemníků, které má zadané do systému. Může zde vyměnit nájemníka za nájemníka, nebo nájemníka z jednotky odebrat (právě nemá jednotku nikdo pronajatou). Odebrání se provede kliknutím na červený křížek v kolečku vedle výběru nájemníka.

Pokud chce majitel údaje o jednotce aktualizovat, musí kliknout na „Potvrdit změny“. Pokud by nějaké změny v textových polích udělal, ale chtěl je zahodit, může znovu kliknout na „Editovat“ a vrátí se tak zpět z editačního režimu bez uložení změn. Pro smazání části je nutné být v editačním režimu a kliknout na „Odstranit část“.

Odstraněním části budou, kromě části, zároveň smazány všechny příjmy a náklady k této části přidáné. Nepřeji si totiž, aby zůstaly v systému příjmy a náklady, které k žádné části nepatří. Takto sice existuje riziko, že si majitel smaže náklad nebo příjem, který nechce. Na druhou stranu, k mazání části by nemělo docházet často a pokud to majitel chce udělat, tak k tomu má pravděpodobně dobrý důvod.

Bezpečnější varianta mazání jednotek by byla, pokud by jednotku nebylo možné smazat, dokud k ní existují příjmy nebo náklady. Pak by ale bylo velmi náročné (časově) smazat jednotku, kterou majitel již nějakou dobu provozuje (příliš mnoho příjmů a nákladů). Proto jsem zvolil pro uživatele jednodušší, ale trochu méně bezpečný, způsob.

## Příjmy a náklady

Ve formuláři vpravo nahoře může majitel specifikovat, od kdy do kdy si chce nechat zobrazit příjmy a náklady. Výběr datumu potvrdí tlačítkem „Zobrazit výsledky hospodaření“.

Po potvrzení formuláře se majiteli zobrazí, podobně jako v přehledu financí, všechny příjmy a náklady za vybrané období. Ovšem pouze související s touto jednotkou nebo částí. Zároveň se sečtou všechny příjmy a všechny náklady a zobrazí se mu pod formulářem pro výběr období spolu s rozdílem všech příjmů a nákladů této jednotky.

Jednotlivé položky zobrazené ve spodní části obrazovky jsou ve stejném formátu, jako na stránce financí, lze je rozbalit pro detailnější informace, sbalit zpět, nebo smazat. Jediný rozdíl je, že jsou zde vyfiltrovány pouze pro konkrétní jednotku.

**Obrázek 6.12** Obrazovka jednotky

The screenshot shows a web interface for managing a unit ('Byt 1'). At the top, there are navigation links 'Správa domu' and 'Odhlásit se'. The main header is 'Byt 1'. Below this, there are two columns of information.

**Left Column (Unit Details):**

- Patro: 1
- Nájemník: Nájemník jedna
- Počet lidí v bytě: 3
- Plocha: 80 m<sup>2</sup>
- Pokoje: 3+1
- Měsíční nájem: 10000
- Měsíční služby: 1000
- Kauce: 60000 Kč
- Zapůjčené klíče: 3
- Buttons: Editovat

**Right Column (Financials):**

- Vyberte období:**
  - Od: 05. 05. 2021
  - Do: 05. 05. 2021
  - Button: Zobrazit výsledky hospodaření
- Celkem za vybrané období:**
  - Příjmy: 11000 Korun
  - Výdaje: 15000 Korun
  - Zisk: -4000 Korun
- Příjmy za vybrané období:**
  - Nájem Byt 1: 10000
  - Buttons: Jednotka: Byt 1, Datum: Wed May 05 2021, Delete icon
- Výdaje za vybrané období (Table):**

Polozka 1	10000	▼
polozka 2	5000	▲

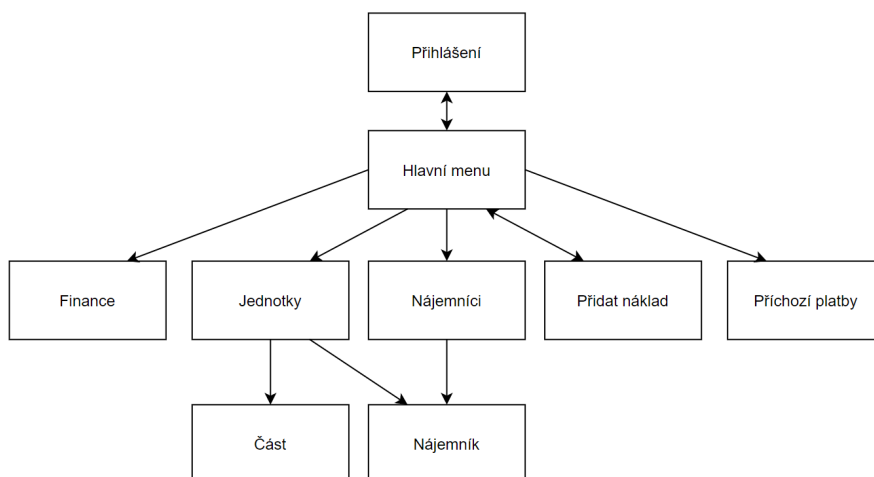
  - Datum: Wed May 05 2021
  - Část: Byt 1
  - Popis: polozka 2
  - Poznámka: polozka 2
  - Buttons: Delete icon
- Služby Byt 1:** 1000 ▼

■ **Obrázek 6.13** Editace jednotky

### 6.2.11 Přechodová mapa

Oproti návrhu doznaly přechody mezi obrazovkami několika změn. Hlavní menu se rozšířilo o tlačítko odkazující na přehled nájemníků. Obrazovka rozpočítání splužeb nebyla v prototypu implementována, proto není ani v přechodové mapě. Další obrazovka, která v přechodové mapě oproti návrhu chybí, je obrazovka přidání jednorázového příjmu. Přidání jednorázového příjmu je nyní jedné obrazovce se všemi příjmy.

V prototypu se uživatel odhláší z navigační lišty, která je na všech obrazovkách kromě přihlašovací. Odhlásit se a tím se dostat na obrazovku přihlášení se tedy lze dostat z každé obrazovky. Pro přehlednost ale tyto přechody v přechodové mapě chybí (ze všech obrazovek vede přechod na přihlašovací obrazovku).

■ **Obrázek 6.14** Přechodová mapa prototypu

## 6.3 Server

Privátní API serveru umožňuje klientovi provádět CRUD (create, read, update, delete) operace. Pro autentizaci klienta je využíván Json Web Token (JWT). Toho server zašle klientovi po



úspěšném přihlášení na jediném public endpointu (`/api/users/login`).

Server nabízí endpointy pro práci s uživateli, jednotkami (byty), nájemníky, příjmy a náklady. Ne vždy jsou k dispozici všechny CRUD operace. Především proto, že na ně při implementaci prototypu nezbyl čas (změna hesla uživatele, editace příjmů a nákladů, ...).

Pro zdokumentování všech endpointů jsem využil aplikaci *Swagger*. [14]

■ **Obrázek 6.15** API Jednotky (byty)

GET	<code>/api/flats</code>	Returns all flats	🔒 ↶
POST	<code>/api/flats</code>	Adds flat	🔒 ↶
GET	<code>/api/flats{id}</code>	Returns flat by its ID	🔒 ↶
PUT	<code>/api/flats{id}</code>	Updates flat by ID	🔒 ↶
DELETE	<code>/api/flats{id}</code>	Removes flat by ID	🔒 ↶

■ **Obrázek 6.16** API Nájemníci

GET	<code>/api/renterers</code>	Lists all renterers	🔒 ↶
POST	<code>/api/renterers</code>	Adds renterer	🔒 ↶
GET	<code>/api/renterers/{id}</code>	Returns renterer by his ID	🔒 ↶
PUT	<code>/api/renterers/{id}</code>	Updates renterer by ID	🔒 ↶
DELETE	<code>/api/renterers/{id}</code>	Removes renterer by ID	🔒 ↶

■ **Obrázek 6.17** API Náklady

<b>GET</b>	<code>/api/expenses</code>	Returns all expenses	🔒 ↩
<b>POST</b>	<code>/api/expenses</code>	Creates expense	🔒 ↩
<b>DELETE</b>	<code>/api/expenses/{id}</code>	Deletes expense by its ID	🔒 ↩

■ **Obrázek 6.18** API Příjmy

<b>GET</b>	<code>/api/revenues/</code>	Returns all revenues	🔒 ↩
<b>POST</b>	<code>/api/revenues/</code>	Creates revenue	🔒 ↩
<b>DELETE</b>	<code>/api/revenues/{id}</code>	Deletes revenue by its ID	🔒 ↩

■ **Obrázek 6.19** API Uživatel

<b>GET</b>	<code>/api/users</code>	Returns all users	🔒 ↩
<b>POST</b>	<code>/api/users/create</code>	Creates user	🔒 ↩
<b>POST</b>	<code>/api/users/login</code>	Login	🔒 ↩

## 6.4 Databáze

V této části popisují modely a schémata, na základě kterých Mongoose vytváří dokumenty v kolekcích v databázi MongoDB. Model je konstruktor objektu definového schématem. Mongoose pak s takto vytvořeným objektem umí pracovat (ukládat, vyhledávat, editovat, mazat).

### 6.4.1 Mongo Atlas

Mongo atlas je cloudové řešení databáze MongoDB. Atlas umožňuje využívat databázi jako službu. Využíváním atlasu se zjednoduší instalace, protože není potřeba instalovat na server databázi, tu stačí pouze připojit z cloudu. [15]

## 6.4.2 Část domu

Část domu má povinné atributy číslo (String) a typ části (String). Typ části slouží k rozlišení, zda se jedná o byt, garáž, chodbu, ... V aplikaci je potřeba nemovitost rozdělit na jednotlivé části, aby šlo přesně přiřazovat příjmy a náklady. Číslo části je schválně String, protože ten dává majiteli větší možnosti v označování. Například lze identifikovat byt pomocí čísla patra a na patře písmenem (byt 2A, 3C, ...).

Nepovinné atributy části jsou patro, plocha, počet pokojů, měsíční nájem, měsíční služby, počet lidí aktuálně bydlících v bytě, kauce a počet zapůjčených klíčů od části. Poslední atribut je reference na aktuálního nájemníka části.

Většina částí budou v aplikaci byty. Proto jsou nepovinné atributy vymyšleny tak, aby seděly co nejlépe k části, která bude pronajímána jako byt. U některých jiných částí nebude mít smysl tyto atributy používat (počet lidí bydlících v garáži).

## 6.4.3 Nájemník

U nájemníka ukládám jeho jméno (String) a příjmení (String), telefon (String), email (String) a adresu trvalého bydliště (String). Telefon je ve formátu String schválně, umožňuje to lepší formátování telefonního čísla při zobrazování (mezery mezi trojčíslím) a navíc je jednoduché přidat předvolbu státu.

## 6.4.4 Náklady

Náklady mají povinné atributy: jméno nákladu (String), pořizovací cenu (Number), datum pořízení (Date). Majitel ještě může přidat popis nebo poznámku.

## 6.4.5 Příjmy

U příjmů do databáze ukládám jméno příjmu (String), částku (Number), datum (Date) a referenci na část domu, ke které příjem patří.

## 6.4.6 Uživatel

V tuto chvíli jsou jediné dva atributy uživatele username (String) a hashované heslo (String). Všechnu účty uživatelů tak mají stejná práva pro práci s nemovitostí.

Do budoucna by mohlo dávat smysl jednotlivým účtům omezovat co mohou a co nemohou dělat. Například by dávalo smysl omezit přístup ke kontaktním údajům pro toho, kdo má za úkol pouze vkládat do systému výdaje. Na druhou stranu, tím že se snažím vytvořit aplikaci podporující "rodinnou správu" nemovitosti, nejsou role uživatelů prioritou.

## 6.5 Přidaná hodnota prototypu

Prototyp v tuto chvíli přinese majiteli lepší přehled o financích v domě. Majitel si může zjistit, kolik mu dům vydělal či prodělal za nějaké období. Toto majitel stejně musel jednou ročně dělat pro podání daňového přiznání, ale zpravidla čísla sečetl pouze za rok a nemohl tak vědět, jak se mu daří po jiných časových úsecích.

Podobně neměl majitel jednoduchý přístup k datům za jednotlivé části domu. V prototypu si může jednoduše zobrazit jednotlivé části a u nich náklady, příjmy a jejich rozdíl. Dozví se tak více informací o své nemovitosti a dostane lepší podklady pro svá rozhodnutí ohledně správy.



## Kapitola 7

# Testování

V této kapitole se věnuji testování prototypu. Podrobuji ho uživatelským testům a hodnotím splnění funkčních a nefunkčních požadavků z kapitoly 3 (Sběr požadavků).

### 7.1 Uživatelské testy

V této části se zabývám testování uživatelského rozhraní aplikace. Testuji různé průchody aplikací a její funkcionality podle scénářů. V této kapitole popsané testy mohou zároveň sloužit jako příklady pro uživatele, jak s aplikací pracovat, která funkcionalita se jak používá a kde je na co umístěné tlačítko.

Testy na sebe navazují a při započetí dalšího testu se předpokládá, že předchozí test je zdárně dokončen. Z předchozího testu se totiž mohou využívat výstupy (něco bylo vytvořeno).

#### 7.1.1 Přihlášení a přidání nákladu

##### Odhadovaný čas

3 minuty

##### Účel testování

Cílem je otestovat přihlášení uživatele do aplikace, vytvoření nové jednotky a přidání nákladu k ní.

##### Počáteční bod

- Otevřená přihlašovací obrazovka v novém okně prohlížeče.

##### Koncový bod

- Uživatel je úspěšně přihlášen do aplikace.
- Vytvořený nový byt.
- Nový náklad je úspěšně přidán k novému bytu.

### Instrukce pro testera

- Pokuste se přihlásit uživatelem „admin“ a neplatným heslem (jiným než „Admin123“).
- Přihlaste se do aplikace uživatelem „admin“ a heslem „Admin123“.
- Vytvořte nový byt „Byt 42“.
- Přidejte náklad k tomuto bytu.
  - Jméno „Náklad 1“
  - Částka 10000
  - Dnešní datum
  - Popis a poznámka libovolné
- Zkontrolujte v detailu bytu zda byl náklad vložen.

### Očekávané kroky

1. Vyplnění formuláře přihlášení a stisknutí „Přihlásit“.
2. Znovu vyplnění formuláře přihlášení a stisknutí „Přihlásit“.
3. Kliknutí na „Jednotky (části)“.
4. Kliknutí na „Přidat část“ dole na obrazovce.
5. Vyplnění formuláře a kliknutí na „Přidat část“.
6. Kliknutí na správa domu v navigační liště.
7. Kliknout na „Přidat náklad“.
8. Vybrat „Byt 42“ v nabídce částí a vyplnit formulář.
  - Jméno „Náklad 1“
  - Částka 10000
  - Dnešní datum
  - Popis a poznámka libovolné
9. Přejít do detailu bytu.
10. Zobrazit si náklady a příjmy za dnešek (tlačítko „Zobrazit výsledky hospodaření“).

## 7.1.2 Zobrazení nájemníků a práce s nimi

### Odhadovaný čas

5 minut

### Účel testování

Tento test má za úkol ověřit funkcionality týkající se zobrazování nájemníků v přehledu a jejich přidávání. Zároveň ověřuje i editaci a mazání nájemníků na obrazovce nájemníka.

## Počáteční bod

- Přihlášený uživatel.
- Otevřené okno prohlížeče s hlavním menu aplikace.

## Koncový bod

- Noví nájemníci jsou přidáni.
- U nájemníků jsou nastavené údaje.

## Instrukce pro testera

- Zobrazte si přehled nájemníků.
- Vytvořte nájemníka se jménem a příjmením „Testovaný jedna“.
- Vytvořte nájemníka se jménem a příjmením „Druhý testovaný“.
- Vytvořte nájemníka se jménem a příjmením „Další testovaný“.
- Přepněte se na nájemníka jedna a nastavte údaje o něm.
  - Jméno: „Testovaný nájemník“
  - Telefon: „123456789“
  - E-mail: „test@najemnik.cz“
  - Trvalé bydliště: „tady“
- Jděte zpět do přehledu nájemníků a zobrazte si detaily o tomto nájemníkovi.
- Zobrazte si Detaily o „Druhém Testovaném“ nájemníkovi.
- Smažte „Dalšího Testovaného“ nájemníka.

## Očekávané kroky

1. Kliknutí na tlačítko „Nájemníci“.
2. Kliknutí na „Přidat nájemníka“ v dolní části obrazovky.
3. Vyplnění formuláře prvním nájemníkem a kliknutí na „Přidat nájemníka“.
4. Vyplnění formuláře druhým nájemníkem a kliknutí na „Přidat nájemníka“.
5. Vyplnění formuláře třetím nájemníkem a kliknutí na „Přidat nájemníka“.
6. Kliknutí na jméno prvního přidaného nájemníka.
7. Kliknutí na tlačítko „Editovat“.
8. Vyplnění kolonek údaje o nájemníkovi.
  - Jméno: „Testovaný nájemník“
  - Telefon: „123456789“
  - E-mail: „test@najemnik.cz“
  - Trvalé bydliště: „tady“

9. Kliknutí na tlačítko „Potvrdit změny“.
10. Vrácení se zpět na přehled nájemníků.
11. Kliknutí na šipku vpravo u prvního nájemníka.
12. Kliknutí na šipku vpravo u druhého nájemníka.
13. Kliknutí na jméno třetího nájemníka.
14. Kliknutí na „Editovat“ a na „Odstranit nájemníka“.

### 7.1.3 Zobrazení jednotek a práce s bytem

#### Odhadovaný čas

5 minut

#### Účel testování

Cílem tohoto testu je ověřit funkcionalitu okolo jednotek. Testovat se bude obrazovka přehledu jednotek i funkcionalita na obrazovce konkrétní jednotky (editace, mazání).

#### Počáteční bod

- Přihlášený uživatel.
- Otevřené hlavní menu aplikace.
- Vytvořený prázdný Byt 42 z prvního testu.
- Dva vytvoření nájemníci ze druhého testu.

#### Koncový bod

- Vytvořené nové jednotky.
- U jednotek jsou nastavené údaje.

#### Instrukce pro testera

- Otevřete si přehled jednotek.
- Vytvořte „Byt 666“.
- Vytvořte „Garáž 42“.
- Upravte atributy bytu 42.
  - Patro: 1
  - Nájemník: Testovný nájemník
  - Počet lidí v bytě: 2
  - Plocha: 50
  - Pokoje: 2
  - Měsíční nájemné: 10000
  - Měsíční služby: 5000



- Kauce: 30000
- Zapůjčené klíče: 3
- Přejděte zpět do přehledu jednotek.
- Zobrazte si detaily o bytu 42.
- Upravte atributy bytu 666.
  - Patro: 2
  - Nájemník: Druhý testovaný
  - Počet lidí v bytě: 3
  - Plocha: 80
  - Pokoje: 3
  - Měsíční nájemné: 15000
  - Měsíční služby: 6000
  - Kauce: 40000
  - Zapůjčené klíče: 5
- Přepněte se na obrazovku přehledu jednotek.
- Zobrazte si detaily bytu 42 a bytu 666.
- Odstraňte garáž 42.

### Očekávané kroky

1. Kliknutí na „Jednotky (části)“.
2. Kliknutí na „Přidat byt“ v dolní části obrazovky.
3. Vyplnění formuláře bytem 666 a kliknutí na „Přidat část“.
4. Vyplnění formuláře garáží 42 a kliknutí na „Přidat část“.
5. Kliknutí na Byt 42.
6. Kliknutí na tlačítko „Editovat“.
7. Vyplnění kolonek údajů o bytě.
  - Patro: 1
  - Nájemník: Testovný nájemník
  - Počet lidí v bytě: 2
  - Plocha: 50
  - Pokoje: 2
  - Měsíční nájemné: 10000
  - Měsíční služby: 5000
  - Kauce: 30000
  - Zapůjčené klíče: 3
8. Kliknutí na „Potvrdit změny“.

9. Vrácení se zpět do přehledu jednotek.
10. Kliknutí na šipku vpravo u bytu 42.
11. Kliknutí na Byt 666.
12. Kliknutí na tlačítko „Editovat“.
13. Vyplnění kolonek údaji o bytě.
  - Patro: 2
  - Nájemník: Druhý testovaný
  - Počet lidí v bytě: 3
  - Plocha: 80
  - Pokoje: 3
  - Měsíční nájemné: 15000
  - Měsíční služby: 6000
  - Kauce: 40000
  - Zapůjčené klíče: 5
14. Kliknutí na „Potvrdit změny“.
15. Vrácení se zpět do přehledu jednotek.
16. Kliknutí na šipku vpravo u bytu 666.
17. Kliknutí na garáž 42.
18. Kliknutí na tlačítko „Editovat“.
19. Kliknutí na tlačítko „Smazat část“.

### 7.1.4 Přijetí plateb

#### Odhadovaný čas

3 minuty

#### Účel testování

Tento test má za úkol ověřit, že k vytvořeným jednotkám lze přidat příjmy a správně si je zobrazit v přehledu.

#### Počáteční bod

- Přihlášený uživatel.
- Otevřené hlavní menu aplikace.
- Vytvořené byty 42 a 666 z předchozího testu.

#### Koncový bod

- K bytům jsou přidány příjmy.
- Příjmy a výdaje bytů jsou zobrazeny na obrazovce „Finance“.

## Instrukce pro testera

- Otevřete si obrazovku Příjmy.
- Potvrďte platby za nájem k dnešnímu datu u bytů 42 a 666.
- Upravte ve formuláři částku nájmu na 2000 a služeb na 0 a u bytu 42 a přijměte platbu ke včerejšímu datu.
- Přidejte jednorázový příjem k bytu 666.
  - Jméno příjmu: „Příjem 666“
  - Částka: 4000
  - Datum: dnes
- Zkontrolujete na stránce financí detaily přidaných příjmů.

## Očekávané kroky

1. Kliknutí na „Příjmy“.
2. Kliknutí na „Přijmout platbu“ u bytu 42.
3. Kliknutí na „Přijmout platbu“ u bytu 666.
4. Upravení částky nájmu (2000) a služeb (0) ve formuláři u bytu 42.
5. Kliknutí na „Přijmout platbu“ u bytu 42.
6. Vyplnění formuláře jednorázového příjmu k bytu 666.
  - Jméno příjmu: „Příjem 666“
  - Částka: 4000
  - Datum: dnes
7. Kliknutí na „Přidat příjem“
8. Kliknutí na „Správa domu“ v navigační liště.
9. Kliknutí na „Finance“ v hlavním menu.
10. Zadání rozsahu od včerejšího data do dnešního data ve formuláři.
11. Kliknutí na „Zobrazit výsledky hospodaření“.

### 7.1.5 Průběh a vyhodnocení uživatelského testování

Předchozí scénáře prošlo dne 8. 5. 2021 pět testerů ve věku od 15 do 57 let. Všechny testy měly podle plánu být dohromady na 16 minut, testerům zabraly od 15 do 27 minut s průměrným časem zhruba 20 minut. Pravděpodobně jsem tedy přecenil rychlost uživatelů, kteří aplikaci neznají. Ne všichni testeři se dokázali přesně držet stanovených scénářů, ale všichni nakonec prošli všechny části testů, které jsem od nich potřeboval otestovat.

Nejčastější stížnosti od testerů na uživatelské rozhraní aplikace byly na obrazovku příjmů. Testeři zpravidla neměli problém s přidáním standardního pravidelného příjmu k jednotce. Po jejich přidání se vedle tlačítka zobrazí zelená fajfka, která potvrzuje, že byl příjem přidán. Problém testeři měli s druhým příjmem, u kterého museli měnit čísla ve formuláři. Vyplnit správně formulář nebyl problém, po vyplnění všichni zvládli příjem přidat, ale chybělo jim potvrzení, že

platba byla přidána. V tu chvíli totiž už byla zelená fajfka za přidání platby zobrazená po předchozím příjmu. Po stisknutí tlačítka „Přijmout platbu“ se tedy pouze resetoval formulář u příslušné jednotky a testeři si nebyli jistí, zda byl příjem opravdu přidán.

Možná nebyl vhodně zvolený test, protože testeři by neměli problém s přidáním nestandardního příjmu, kdyby viděli jednoznačné potvrzení (stačilo by stránku obnovit). Potvrdili mi, to co jsem si myslel, že kontrola pravidelných příjmů je nejméně přehledné místo prototypu a při jeho dalším rozvoji bude potřeba se mu věnovat více než zbytku prototypu.

Další věc, která se testerům nelíbila bylo vytváření nových jednotek a částí. Tři z pěti testerů by raději místo textového pole, do kterého se píše typ jednotky, uvítali výběr z předem definovaných částí.

Zde je ale problém s univerzalitou. Kdybych, jakožto programátor, definoval nějaké části a dal je k dispozici uživatelům, tak se dříve či později stane, že uživatel bude chtít vytvořit jinou část, než kterou má k dispozici. Lepší variantou by bylo, kdyby si mohl majitel typy jednotek sám zvolit. V rámci prototypu jsem ale neměl čas na vytvoření „přizpůsobovacího menu“, proto je variabilita vyřešena textovým vstupem od uživatele.

Další stížnosti a návrhy testerů už byly ojedinělé. Navrhovali například některá tlačítka pojmenovat trochu jinak, nebo chtěli, aby odkazem na jednotku z přehledu jednotek byl celý šedý obdélník, ne pouze typ a číslo jednotky.

I přes výše popsané výhrady si myslím, že uživatelské testy dopadly úspěšně. Všichni testeři zvládli všechny úkony, které jsem po nich požadoval a řekli mi, že by si na používání aplikace dokázali bez větších problémů zvyknout. Uznávám ale, že je v uživatelské přívětivosti určitě prostor pro zlepšení.

## 7.2 Akceptační testy

Akceptační testy klasicky slouží k tomu, aby si zadavatel ověřil, zda doručенý produkt odpovídá jeho požadavkům. V této práci budu dělat akceptační testy já, podle požadavků, které jsme se zadavatelem definovali v kapitole 3. Budu hodnotit jejich splnění či nesplnění podobně jako jsem hodnotil konkurenci. Testy budu psát já z více důvodů. Zadavatelem byl můj otec, který nerozumí softwarovému inženýrství natolik, aby mi pomohl akceptační testy udělat. Druhý problém je, že se v průběhu psaní této práce přestěhoval za prací do Švýcarska a proto je spolupráce na tomto projektu složitější.

### 7.2.1 Splnění funkčních požadavků

Pro každý funkční požadavek definovaný z kapitoly 3.2.3 (Funkční požadavky) jsem rozepsal, jak se mi ho povedlo splnit v implementovaném prototypu. Výsledek hodnotím procenty. Procentuální splnění požadavku závisí na jeho prioritě a na jeho implementovaných i neimplementovaných částí.

#### FP1 Rozdělení domu na podčásti

- V prototypu lze dům rozdělit na libovolné části.
- Části může majitel pojmenovat a označit jakkoliv (byt, sklep, ...).
- Zatím nelze spojovat jednotky do komplexů (nelze přiřadit sklep k bytu).

Splnění požadavku: 90 %.

### FP2 Evidování nákladů a investic

- Náklady lze do prototypu přidávat.
- Zatím nelze přidat do aplikace fakturu nebo účtenku.
- Investice v prototypu chybí.

Splnění požadavku: 55 % (náklady mají vysokou prioritu).

### FP3 Monitorování financí

- Do aplikace lze přidávat příchozí platby.
- Do aplikace zatím nelze přidávat odchozí platby, které nejsou náklady.

Splnění Požadavku: 65 % (příchozí platby mají vysokou prioritu).

### FP4 Rozpočítání plateb za energie a odpady

- Tento požadavek není v prototypu implementován.

Splnění požadavku: 0 %.

### FP5 Generování podkladů pro daňové přiznání

- Tento požadavek není v prototypu implementován.

Splnění požadavku: 0 %.

### FP6 Evidence vybavení

- Tento požadavek není v prototypu implementován.

Splnění požadavku: 0 %.

### FP7 Hromadná notifikace nájemníků

- Tento požadavek není v prototypu implementován.

Splnění požadavku: 0 %.

### FP8 Dostupnost kontaktů

- Majitel má dostupné kontakty na nájemníky.
- Pro nájemníky žádné dostupné kontakty nejsou.

Splnění požadavku: 50 %.

### FP9 ukládání smluv

- Tento požadavek není v prototypu implementován.

Splnění požadavku: 0 %.

## 7.2.2 Splnění nefunkčních požadavků

Podobně jako v předchozí části hodnotím splnění funkčních požadavků, tak v této části hodnotím splnění nefunkčních požadavků. Jsou to požadavky z kapitoly 3.2.4 (Nefunkční požadavky). Pro každý jsem procentuálně ohodnotil, jak se mi ho podařilo splnit.

### NP1 Webová aplikace

- Klient prototypu je webová aplikace.

Splnění požadavku: 100 %.

### NP2 Responzivita

- Klient je responzivní.

Splnění požadavku 100 %.

### NP3 Mobilní aplikace

- Mobilní aplikace není implementována.

Splnění požadavku 0 %.

### NP4 Autentizace uživatele

- K práci s prototypem je nutné se přihlásit uživatelským jménem a heslem.

Splnění požadavku 100 %.

### NP5 Autorizace uživatele

- Prototyp má pouzejednu úroveň práv uživatele.

Splnění požadavku 50 %.

### NP6 Bezpečné ukládání hesla uživatelů

- Hesla uživatelů jsou před uložením soleny a hešovány.

Splnění požadavku 100 %.

### NP7 Podpora více jazyků

- Prototyp je pouze v češtině.

Splnění požadavku 50 %.

### NP8 CI/CD

- Prototyp nemá nastavené automatické sestavování ani nasazování.

Splnění požadavku 0 %.

### 7.2.3 Celkové hodnocení prototypu

Prototyp jsem také podrobil stejnému hodnocení, kterým jsem hodnotil konkurenční softwary v kapitole 4 (Analýza konkurence). V součtu za hodnocení funkčních i nefunkčních požadavků získal můj prototyp 17,5 bodů. To je víc, než v hodnocení získal Domovník (15 bodů) a jen půl bodu méně než získalo SBN (18 bodů). V hodnocení funkčních požadavků zaostává můj prototyp výrazněji, bodovou ztrátu z hodnocení funkčních požadavků dohání tím, že se jedná o webovou, ne desktopovou aplikaci.

Výsledek hodnocení prototypu považuji za úspěch. Sice se zatím nedokáže vyrovnat těm nejlepším z dostupných softwarů, ale již teď je funkční. Na prototypu plánuji dál pracovat a zlepšovat ho.

Požadavek	Popis	Body	Prototyp
1a	Kontrola příchozích plateb	1	1
1b	Automatická kontrola příchozích plateb	1	0
2a	Rozpočítání služeb podle lidí	2	0
2b	Rozpočítání služeb z dat měřičů	1	0
2c	Možnost částku upravit	2	0
2d	Upozornění nájemníků o změně služeb	1	0
3a	Kontakty pro majitele	0,5	0,5
3b	Kontakty pro nájemníky	0,5	0
4a	Náklady k bytům	3	3
4b	Náklady nejen k bytům	1	1
4c	Přiřazení účtenky (faktury)	1	0
4d	Zobrazení přehledu cash flow za celý dům	2	2
4e	Vykreslení grafu příjmů a nákladů	1	0
4f	Přehled o jednotlivých bytech	2	2
5a	Součet nákladů a příjmů za rok	1	1
5b	Zobrazení nákladů a příjmů za rok	1	1
5c	Stážení dat	1	0
5d	Napojení na účetní systém	1	0
6a	Hromadná zpráva nájemníkům	0,5	0
6b	Upozornění nájemníků e-mailem nebo sms	2	0
7a	Přidání vybavení	1	0
7b	Data o vybavení	1	0
8a	Zjištění nájemného nájemníkem	0,5	0
8b	Generování QR kódu pro platbu	0,5	0
9a	Generování nájemních smluv	0,5	0
9b	Uchovávání nájemních smluv	2	0
Součet bodů		31	11,5

■ **Tabulka 7.1** Hodnocení funkčních požadavků prototypu

Požadavek	Popis	Body	Prototyp
NP1a	Webová aplikace	6	6
NP1b	IOS aplikace	4	0
NP1c	Android aplikace	4	0
NP1d	Offline desktopová aplikace	2	0
Součet bodů		16	6

■ **Tabulka 7.2** Hodnocení nefunkčních požadavků prototypu





## Kapitola 8

# Závěr

Cílem bylo vymyslet, navrhnout, implementovat a otestovat aplikaci pro usnadnění správy nemovitosti, ve které jsou pronajímány byty. Požadavky na takovou aplikaci jsem sesbíral a rozepsal. Požadavků je mnohem více, než jsem v prototypu stihl implementovat, je tedy aplikaci kam dál rozvíjet.

Prozkoumal jsem již dostupná softwarová řešení, ohodnotil je a odhalil jejich nedostatky. Zároveň jsem se z nich poučil, jak může aplikace vypadat a naopak, kterým směrem při vývoji vlastní aplikace nejít.

V návrhu jsem zakomponoval většinu požadavků z analýzy, nebo alespoň ty nejdůležitější. Navrhl jsem uživatelské rozhraní aplikace vypadat a popsal jak by ji měl majitel pro správu používat.

Implementoval jsem prototyp, který sice zatím nemá produkční kvalitu, přesto jsem přesvědčen, že by přinesl majiteli hodnotu navíc oproti poznámkám v sešitu nebo například v excelu. Prototyp je kam dál rozšiřovat, protože jsem neimplementoval všechny požadavky vyplývající jak z analýzy, tak z návrhu.

Prototyp jsem následně podrobil uživatelským testům. Testeři procházeli prototyp podle mnou definovaných scénářů. Následně mi dali zpětnou vazbu na uživatelské rozhraní aplikace. Hlavní problém viděli v obrazovce příjmů, kterou bude potřeba zrevidovat. V rámci akceptačních testů jsem zkoumal, jak se mi podařilo splnit funkční a nefunkční požadavky definované v analýze. Prototyp jsem také ohodnotil stejnou metrikou, jakou jsem hodnotil konkurenční aplikace. Za nejlepšími z hodnocené konkurence stle velmi zaostává, ale nedopadl nejhůře.

Prototyp není v produkční kvalitě, což také znamená, že jsem ho ještě neotestoval v ostrém provozu. Není tedy jasné, zda bude aplikace opravdu užitečná a zda majiteli ušetří čas strávený správou nemovitosti. Doufám, že se mi podaří ho v dohledné době nasadit alespoň pro náš dům, abych na něm mohl dále pracovat, přidávat funkcionality a odstraňovat nedostatky, aby se z prototypu někdy stala funkční a užitečná aplikace.



# Příloha A

## Instalační příručka

Instalační příručka obsahuje návod, jak nainstalovat a spustit aplikaci na počítači s operačním systémem Debian 10.

### A.1 Databáze

Server je připraven na používání MongoDB pomocí mongoose. Databáze k mé práci běží v cloudu Mongo Atlas [15]. Při instalování nové instance serveru je ale třeba v kódu změnit konstantu `DBURI` v souboru `index.js`, aby url odkazovala na vaši databázi.

Instalování MongoDB na Debian 10 je popsáno v manuálových stránkách MongoDB. [16]

### A.2 Instalace git

Zdrojové kódy budu v příručce stahovat z git repozitáře. Je tedy potřeba instalovat nejdříve git. Instalaci provedete následujícími příkazy:

```
sudo apt update;
sudo apt install git;
```

### A.3 Instalace serveru

#### Klonování repozitáře

Repozitář je umístěn na <https://gitlab.fit.cvut.cz/xchludil-bp-dp/starujan>. Klonovat lze buď přes HTTPS příkazem

```
git clone https://gitlab.fit.cvut.cz/xchludil-bp-dp/starujan.git;
```

nebo přes ssh příkazem

```
git clone git@gitlab.fit.cvut.cz:xchludil-bp-dp/starujan.git;
```

Přejděte do stažené složky.

## Instalace Node.js a použitých knihoven

Následující příkazy nainstalují Node.js a knihovny potřebné pro spuštění projektu.

```
sudo apt install nodejs npm;
npm install;
```

## Vytvoření tajného klíče

Aby mohl server komunikovat s klientem, je potřeba vytvořit soubor `.env` v kořenové složce projektu a vložit do něj vygenerovaný klíč. Příkazy pro vytvoření klíče jsou následující:

```
node
require('crypto').randomBytes(64).toString('hex')
.exit
```

Tajným klíčem je výstup `require('crypto').randomBytes(64).toString('hex')`. Je potřeba ho zkopírovat (bez uvozovek) a uložit do souboru `.env`. Obsah souboru `.env` by měl vypadat následovně:

```
ACCESS_TOKEN_SECRET=klíč
```

## Spuštění serveru

Server následně spustíte příkazem:

```
npm start;
```

Aplikace pak vypíše na obrazovku port, na kterém je dostupná.

### A.4 Instalace frontend

Před instalováním frontendu odejděte ze složky, kam jste instalovali server (`cd ..`).

## Klonování repozitáře

Repozitář je umístěn na:

```
https://gitlab.fit.cvut.cz/xcludil-bp-dp/starujan---react-frontend
```

Klonovat lze buď přes HTTPS příkazem

```
git clone https://gitlab.fit.cvut.cz/xcludil-bp-dp/
starujan---react-frontend.git;
```

nebo přes ssh příkazem

```
git clone git@gitlab.fit.cvut.cz:xcludil-bp-dp/
starujan---react-frontend.git;
```

Přejděte do stažené složky.

## Instalace použitých knihoven

Knihovny používané aplikací nainstalujete příkazem:

```
npm install;
```

## Změna IP serveru

V souboru `./src/SERVER.js` je potřeba změnit IP adresu (případně port) serveru, se kterým bude klient komunikovat.

## Sestavení a nasazení

K sestavení a spuštění dojde příkazy:

```
npm run build;
npm install -g serve;
serve -s build;
```

Aplikace se spustí a vypíše, na které IP adrese a portu je dostupná.

## A.5 Vytvoření prvního uživatele

Rozhraní webové aplikace zatím nenabízí možnost vytvořit nového uživatele. Pro vytvoření uživatele je nutné poslat POST request s tělem

```
{
  "username": "uživatelské jméno",
  "password": "heslo"
}
```

na endpoint `/api/users/create`. Pro vytvoření uživatele je ale nutné se prokázat platným JWT vydaným serverem. Pro přidání prvního uživatele je proto nutné dočasně zrušit kontrolu tokenů v hlavičce této funkce (odebrat `authenticateToken`).



## Uživatelská příručka

Uživatelskou příručku nahrazují kapitoly 6.2 (Obrazovky webové aplikace) a 7.1 (Uživatelské testy). V kapitole 6.2 je vysvětleno, k čemu slouží která obrazovka, co zobrazuje a jaký je její smysl, a co dělá které tlačítko na ní. V kapitole 7.1 jsou popsány scénáře pro testery. Ty lze také využít jako „návod“, které kroky udělat pro dosažení nějakého cíle.





# Bibliografie

1. *Rozhovor s otcem* [Osobní rozhovor]. 2020.
2. DATA EVOLUTION S.R.O. *Online software pro správu nemovitostí Domsys* [online] [cit. 2021-04-17]. Dostupné z: <https://domsys.cz>.
3. ANASOFT. *Systém pro správu domu Domus* [online] [cit. 2021-04-17]. Dostupné z: <https://www.anasoft.com/domus/cz/>.
4. VYDAS SOFTWARE. *Správa bytů a nemovitostí SBN* [online] [cit. 2021-04-17]. Dostupné z: <http://spravabytu.cz/sprava-bytu/sprava-bytu-informace-o-programu/>.
5. JIRRA. *Jirra vlastník* [online] [cit. 2021-04-17]. Dostupné z: <https://jirra.cz>.
6. IQ. *Správa nemovitostí* [online] [cit. 2021-04-17]. Dostupné z: <https://www.qi.cz/moduly/sprava-nemovitosti/>.
7. CSH. *Domovník* [online] [cit. 2021-04-17]. Dostupné z: <https://www.qi.cz/moduly/sprava-nemovitosti/>.
8. MLEJNEK, Jiří. *Architektonické vzory* [online] [cit. 2021-04-17]. Dostupné z: [https://moodle-vyuka.cvut.cz/pluginfile.php/388212/mod\\_resource/content/1/06.prednaska.pdf](https://moodle-vyuka.cvut.cz/pluginfile.php/388212/mod_resource/content/1/06.prednaska.pdf).
9. *React.js* [online] [cit. 2021-04-17]. Dostupné z: <https://reactjs.org>.
10. *Node.js* [online] [cit. 2021-04-17]. Dostupné z: <https://nodejs.org/en/about/>.
11. *Express* [online] [cit. 2021-04-17]. Dostupné z: <https://expressjs.com>.
12. *MongoDB* [online] [cit. 2021-04-17]. Dostupné z: <https://docs.mongodb.com/manual/introduction/>.
13. *Mongoose* [online] [cit. 2021-04-17]. Dostupné z: <https://mongoosejs.com/docs/>.
14. *Swagger* [online] [cit. 2021-05-06]. Dostupné z: <https://swagger.io>.
15. *Mongo Atlas* [online] [cit. 2021-04-17]. Dostupné z: <https://www.mongodb.com/cloud/atlas>.
16. *MongoDB Debian installation* [online] [cit. 2021-04-17]. Dostupné z: <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-debian/>.



# Obsah přiloženého média

readme.txt.....	stručný popis obsahu média
src	
├ server.....	zdrojové kódy server
├ frontend.....	zdrojové kódy frontend
└ thesis.....	zdrojová forma práce ve formátu $\text{\LaTeX}$
text	
└ thesis.pdf.....	text práce ve formátu PDF