

Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra počítačů

## Aplikace pro podporu externího správce malých počítačových sítí

**Mirza Hasanbegović**

Vedoucí: Ing. Pavel Náplava, Ph.D.  
Obor: Softwarové inženýrství a technologie  
Studijní program: Enterprise systémy  
Květen 2024



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Hasanbegović** Jméno: **Mirza** Osobní číslo: **491870**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra počítačů**  
Studijní program: **Softwarové inženýrství a technologie**  
Specializace: **Enterprise systémy**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Aplikace pro podporu externího správce malých počítačových sítí**

Název bakalářské práce anglicky:

**Application to support the external administrator of small computer networks**

Pokyny pro vypracování:

Navrhněte a vytvořte aplikaci, která bude pomáhat externím správcům v malých společnostech, s malou počítačovou sítí (řádově jednotky až desítky počítačů), udržovat přehled o aktuálním stavu vybavení. Postupujte následovně:

- 1) Definujte pojmy – počítačová síť, malá společnost, správce počítačové sítě, IT vybavení atd.
- 2) Popište základní činnosti, které externí správce obvykle provádí.
- 3) Analyzujte existující nástroje, které lze použít pro evidenci IT vybavení a vyhodnoťte je s ohledem na potřeby externího správce malé sítě, včetně použití u více zákazníků.
- 4) Navrhněte vlastní aplikaci v podobě „deníčku“, umožňující vedení přehledné evidence IT vybavení dle zákazníků, plánování pravidelných akcí, upozorňování na důležité termíny (např. blížící se konec platnosti licence), vyhledání vybavení, získávání informací o aktuálním stavu (např. pro účely rozhodování o náhradě dle stáří) atd.
- 5) Aplikaci implementujte.
- 6) Vytvořenou aplikaci otestujte pomocí uživatelských testů.

Seznam doporučené literatury:

MySQL Documentation [<https://dev.mysql.com/doc/>]. [on-line]  
Spring Boot Reference Documentation  
[<https://docs.spring.io/spring-boot/docs/current/reference/pdf/spring-boot-reference.pdf>]. [on-line]  
Java SE Documentation, [<https://docs.oracle.com/en/java/javase/>]. [on-line]  
AuditPro 10 [<https://www.auditpro.cz/download/RC/AuditPro10.pdf>]. [on-line]

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Pavel Náplava, Ph.D. Centrum znalostního managementu FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **15.02.2024**

Termín odevzdání bakalářské práce: **24.05.2024**

Platnost zadání bakalářské práce: **21.09.2025**

Ing. Pavel Náplava, Ph.D.  
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

### III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

## Poděkování

Na tomto místě bych rád poděkoval zejména svému vedoucímu Ing. Pavlu Náplavovi, Ph.D. za jeho odborné vedení, ochotu a celkovou vstřícnost při konzultacích a vypracování této bakalářské práce. Dále bych také rád poděkoval své rodině a přátelům za podporu a pomoc během studia. Také bych chtěl poděkovat testerům, kteří otestovali mou 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.

V Praze dne 17. 05. 2024

.....

Mirza Hasanbegović

## Abstrakt

Tato bakalářská práce se zaměřuje na návrh, vývoj a testování desktopové aplikace určené pro externí správce počítačových sítí v malých podnicích. Cílem je vytvořit nástroj, který umožní externím správcům efektivně evidovat a spravovat IT infrastrukturu malých firem bez potřeby interního IT oddělení.

Práce se zabývá analýzou specifických výzev malých podniků v oblasti IT a zkoumá možnosti využití externích IT služeb. V rámci práce byly definovány klíčové požadavky na aplikaci, provedeny návrhy doménových a databázových modelů a vyvinut prototyp s implementací v prostředí JavaFX a SQLite.

Testování aplikace zahrnovalo uživatelské i integrační testy, které potvrdily její funkčnost a použitelnost. Výsledkem je aplikace, která efektivně podporuje externí správce sítí v malých firmách, umožňuje evidovat zařízení a spravovat informace o nich, čímž zvyšuje efektivitu správy IT bez potřeby stálého interního IT personálu.

**Klíčová slova:** Správa počítačových sítí, Externí správce, Deníček, SQL Lite, Java desktopová aplikace, Java FX, Licence

**Vedoucí:** Ing. Pavel Náplava, Ph.D.

## Abstract

This bachelor's thesis focuses on the design, development, and testing of a desktop application intended for external network administrators in small businesses. The aim is to create a tool that allows external administrators to efficiently record and manage the IT infrastructure of small companies without the need for an internal IT department.

The work addresses the specific challenges faced by small businesses in the IT domain and explores the possibilities of utilizing external IT services. Key requirements for the application were defined, domain and database models were designed, and a prototype was developed using JavaFX and SQLite.

The application underwent both user and integration testing, confirming its functionality and usability. The result is an application that effectively supports external network administrators in small businesses, allowing them to record devices and manage information about them, thereby increasing the efficiency of IT management without the need for a permanent internal IT staff.

**Keywords:** Network management, External administrator, Notebook, SQL Lite, Java desktop application, Java FX, License

# Obsah

<b>1 Úvod</b>	<b>1</b>	6.1.1 Vývojové prostředí: IntelliJ IDEA	29
1.1 Cíle práce	1	6.1.2 SDK: Java 8	29
1.2 Struktura práce	2	6.1.3 Verzovací systém: GitHub	30
<b>2 Definování základních pojmů</b>	<b>3</b>	6.1.4 Nástroj pro správu databáze: DB Browser for SQLite	30
2.1 Počítačová síť	3	6.2 Obecné informace o implementaci	30
2.2 Správce počítačových sítí	3	6.2.1 Model	30
2.3 Evidenční systém	4	6.2.2 View	31
2.4 Správa licencí	5	6.2.3 Controller	31
2.5 Malá firma	5	6.2.4 DAO (Data Access Objects)	32
2.6 IT vybavení	5	6.2.5 Service	33
2.7 Závěr	6	6.2.6 Validace	33
<b>3 Ze života správce</b>	<b>7</b>	6.3 Uživatelské rozhraní	34
3.1 User Story: Řešení hardwarových problémů a rozšiřování IT infrastruktury	7	6.3.1 Hlavní stránka aplikace	35
3.2 User Story: Správa verzí softwaru	8	6.3.2 Detail firmy	36
3.3 User Story: Správa licencí	8	6.3.3 Detail zařízení	38
3.4 User Story: Pravidelná údržba	9	6.4 Zajímavé funkce systému	39
3.5 Závěr kapitoly	10	6.4.1 Správa adresářů firem	39
<b>4 Analýza programů pro podporu správy počítačových sítí</b>	<b>11</b>	6.4.2 Pojmenování zařízení	39
4.1 Metodika vyhledávání	11	6.4.3 Bezpečnostní prvky	39
4.2 Kritéria hodnocení softwarových nástrojů	11	6.4.4 Enumerace a stabilní pojmenování zařízení	39
4.3 Dostupné nástroje	12	6.5 Instalace a spuštění	40
4.3.1 Audit Pro	12	6.6 Závěr	41
4.3.2 Spiceworks	13	<b>7 Testování aplikace</b>	<b>43</b>
4.3.3 Lansweeper	14	7.1 DAO testování	43
4.3.4 Závěr analýzy dostupných nástrojů	14	7.1.1 Mockování	43
4.4 Určení požadavků na systém	15	7.1.2 Integrace s databází	43
4.4.1 Funkční požadavky (FP)	15	7.2 Uživatelské testování	44
4.4.2 Nefunkční požadavky (NP)	16	7.2.1 Proces testování	44
4.5 Shrnutí kapitoly	17	7.2.2 Testovací scénáře	44
<b>5 Návrh aplikace</b>	<b>19</b>	7.3 Závěrečné shrnutí	46
5.1 Doménový model	19	<b>8 Aktuální stav a budoucí rozvoj</b>	<b>47</b>
5.2 Databázový model	21	8.1 Implementace návrhů na zlepšení	47
5.3 Případy užití - Use Case(UC)	22	8.1.1 Vysvětlení funkce tlačítka Zobrazit licence	47
5.4 Výběr technologií	25	8.1.2 Návrat k zobrazení všech zařízení	48
5.5 Návrh prototypu	25	8.1.3 Zvětšení pole pro poznámky při přidávání zařízení	48
5.5.1 Závěr a diskuze	27	8.1.4 Oprava chování ChoiceBoxu při editaci zařízení	48
<b>6 Implementace aplikace</b>	<b>29</b>	8.1.5 Změna správy licencí	48
6.1 Použité nástroje	29	8.2 Shrnutí a uzavření aktuálního stavu	49

8.3 Budoucí rozvoj .....	49
8.3.1 Funkční rozšíření a přechod na klient-server architekturu .....	49
<b>9 Závěr</b>	<b>51</b>
<b>Bibliografie</b>	<b>53</b>
<b>A Přílohy</b>	<b>55</b>
A.1 Obrazovky aplikace .....	55



## Obrázky

## Tabulky

2.1 Ilustrace správce sítě .....	4
4.1 Audit Pro .....	12
4.2 Spiceworks .....	13
4.3 Lansweeper .....	14
5.1 UML Doménový model popisující entity systému a jejich vazby .....	20
5.2 Databázový model .....	22
5.3 Externí správce jako actor .....	23
5.4 Use case externího správce .....	23
5.5 Hlavní stránka .....	26
5.6 Detail Firmy .....	26
5.7 Detail zařízení .....	27
6.1 Hlavní rozhraní aplikace zobrazující správu firem a akcí .....	35
6.2 Detail firmy .....	36
6.3 Detail zařízení .....	38
8.1 Vysvětlení funkce tlačítka "Zobrazit licence" .....	47
8.2 Zobrazení aktivních a neaktivních licencí .....	48
A.1 Obrazovka detailu akce .....	55
A.2 Obrazovka detailu licence .....	56
A.3 Okno přidání firmy .....	57
A.4 Okno přidání zařízení .....	58



# Kapitola 1

## Úvod

Tato bakalářská práce se zaměřuje na návrh, vývoj a testování desktopové aplikace určené pro externí správce počítačových sítí v malých firmách. Práce se zabývá specifickými výzvami, kterým čelí malé podniky v oblasti IT infrastruktury a správy sítí, a zkoumá možnosti využití externích IT služeb pro efektivní správu těchto sítí bez nutnosti investovat do interního IT oddělení.

Na téma této práce jsme přišli na základě diskuzí s vedoucím práce, kdy jsme po několika setkáních a rozpravách o námětech dospěli k závěru, že by bylo přínosné zkoumat právě tuto oblast. Nebyli jsme inspirováni žádným konkrétním zdrojem či předchozí prací.

Při vývoji této aplikace jsem použil následující nástroje:

- ChatGPT<sup>1</sup> - pro korekci textu, řešení komplikací s kódem a pomoc s psaním v LaTeXu.
- GitHub Copilot<sup>2</sup> - pro zjednodušení práce s psaním kódu.
- Perplexity<sup>3</sup> - pro hledání zdrojů na internetu.

## 1.1 Cíle práce

1. Vyvinout aplikaci, která umožní externím správcům počítačových sítí efektivně a jednoduše evidovat IT infrastrukturu malých firem.
2. Prozkoumat a definovat klíčové pojmy a rozdíly mezi interními a externími správci sítí.
3. Provést analýzu softwarových nástrojů určených pro evidenci IT nástrojů ve firmě, které jsou využívány externími správci.
4. Navrhnout a vyvinout aplikaci, zahrnující výběr technologií, modelování případů užití a vytvoření první verze prototypu demonstrujícího plánované funkcionality systému.

---

<sup>1</sup><https://www.openai.com/chatgpt>

<sup>2</sup><https://copilot.github.com/>

<sup>3</sup><https://www.perplexity.ai/>

## 1.2 Struktura práce

1. **Úvod:** Definování klíčových pojmů týkajících se počítačových sítí a správy IT, včetně rozdílů mezi interními a externími správci sítí.
2. **Analýza technik a nástrojů:** Zaměření na softwarové nástroje používané externími správci pro evidenci IT nástrojů ve firmách.
3. **Návrh aplikace:** Výběr technologií, modelování případů užití a vývoj prototypů demonstrujících plánované funkcionality systému.
4. **Implementace aplikace:** Popis klíčových funkcí a nástrojů použitých během vývoje aplikace, včetně architektury aplikace rozdělené do vrstev modelu, pohledu a kontroleru, a použití technologií jako JavaFX pro uživatelské rozhraní a SQLite pro správu databáze.
5. **Testování aplikace:** Metody a postupy použité k ověření funkcionality a uživatelské přívětivosti systému, včetně analýzy výsledků uživatelského testování a zpětné vazby za účelem identifikace oblastí pro další zlepšení.
6. **Aktuální stav a budoucí rozvoj:** Popis implementovaných vylepšení na základě zpětné vazby z testování a návrhy na budoucí rozvoj aplikace.
7. **Závěr:** Shrnutí dosažených výsledků a návrh možných směrů pro další vývoj a rozšíření aplikace, která může pomoci externím správčům sítí efektivněji spravovat IT infrastrukturu malých firem.

## Kapitola 2

### Definování základních pojmů

Na začátku je vhodné si definovat, co je správce počítačových sítí a jaký má význam. Pro lepší pochopení práce je nezbytné nejprve vysvětlit některé základní pojmy, jako jsou správce počítačové sítě, evidenční systém, licence, malá firma a IT vybavení. Tyto pojmy jsou klíčové pro celou práci a pomohou nám lépe porozumět souvislostem.

#### 2.1 Počítačová síť

Počítačová síť je systém, který spojuje dvě nebo více výpočetních zařízení za účelem přenosu a sdílení dat, aplikací a dalších zdrojů. Těmito zařízeními může být vše od notebooků přes stolní počítače, po mobilní telefony až po servery.[1]

Existuje několik druhů počítačových sítí ale v bakalářce se zaměřím na typ sítě, které většinou využívají menší firmy: LAN, WLAN a WAN.

Místní síť (LAN) je typ sítě, která propojuje zařízení v malé geografické oblasti, jako je domov, kancelář nebo škola. LAN obvykle používají kabelová připojení, jako jsou ethernetové kabely, k připojení zařízení k centrálnímu rozbočovači nebo přepínači. To umožňuje zařízením sdílet data, zdroje a zařízení, jako jsou tiskárny a úložná zařízení. [2]

Bezdrátová místní síť (WLAN) je typ sítě, která používá bezdrátovou technologii, jako je Wi-Fi, k připojení zařízení ve stejné oblasti. WLAN využívají bezdrátové přístupové body k přenosu dat mezi zařízeními, což umožňuje větší mobilitu a flexibilitu. [2]

Rozsáhlá síť (WAN) je typ sítě, která umožňuje připojení k internetu a propojení zařízení na velké vzdálenosti. [3]

#### 2.2 Správce počítačových sítí

Správce počítačové sítě je osoba odpovědná za instalaci, údržbu a aktualizaci jakéhokoli softwaru nebo hardwaru potřebného pro efektivní provoz počítačové sítě.[4] Přebírá, ověřuje, uvádí do provozu a nastavuje parametry operačních systémů a počítačových sítí. Jak je vidět na obrázku 2.1, můžeme si správce sítě ilustrovaně představit jako osobu odpovědnou za tyto úkoly.



**Obrázek 2.1:** Ilustrace správce sítě<sup>1</sup>

Existují různé typy správců, včetně interních a externích. Interní správci jsou zaměstnanci organizace, pro kterou pracují, fyzicky jsou přítomni na pracovišti organizace a mají hlubší znalost o interním prostředí a specifických potřebách organizace. Zatímco externí správci jsou dodavatelé, kteří poskytují služby více organizacím, nejsou zaměstnanci jedné konkrétní organizace a často pracují vzdáleně.[5]

Správci počítačových sítí musí mít široké znalosti o různých prostředích a technologiích, aby mohli efektivně spravovat sítě různých klientů. Je nezbytné, aby měli pevné základy v síťové administraci a rozuměli klíčovým konceptům, nástrojům a metodám používaným při správě sítí, jak je popsáno v literatuře o síťové administraci, například v díle Douglase Chicka.[6]

V této práci se budeme věnovat pouze externím správcům.

## 2.3 Evidenční systém

Evidenční systém, na který se zaměřím, úzce souvisí s IT Asset Managementem. Proto ho vysvětlím a budu ho definovat na základě principů Asset.

IT Asset Management (ITAM) je proces, který integruje různé obchodní praktiky, včetně finančního řízení, správy skladů a smluv, s cílem efektivně spravovat IT aktiva v organizaci. Tento proces zahrnuje řízení životního cyklu IT aktiv od pořízení až po vyřazení a podporuje strategické rozhodování v IT prostředí.[7]

V této bakalářské práci bude evidenční systém fungovat jako elektronický deník, který umožní uživatelům přidávat, mazat, nebo upravovat údaje o IT aktivách, jako jsou počítače, servery, licence a smlouvy. Tento systém nebude sloužit pouze k evidenci těchto aktiv, ale také umožní uživatelům spravovat související dokumentaci a smlouvy. Důležitou funkcí bude také schopnost systému včas upozorňovat na blížící se termíny ukončení licencí nebo potřebu

<sup>1</sup><https://chat.openai.com/> Vygenerovaný podle ChatGPT

obnovy smluv, což přispěje k lepšímu řízení IT vybavení v rámci organizace. Systém však nebude řešit finanční otázky spojené s těmito aktivy, což pomůže při rozhodování, proč nepoužít stávající řešení.

## 2.4 Správa licencí

Externí správce se setkává s různými typy licencí, které jsou relevantní pro provoz počítačových sítí a zařízení. Tyto licence mohou zahrnovat jak softwarové, tak hardwarové komponenty a další související technologie. Mezi typické příklady patří:

1. **Licencování Windows pro jednotlivé počítače:** Každý počítač bude vyžadovat vlastní licenci pro operační systém Windows, což je základní softwarová licence potřebná pro funkčnost zařízení.
2. **Licencování Office:** Zahrnujeme jak jednorázové licencí Office 2021, tak předplatné Microsoft 365, která umožňují instalaci na více zařízeních a nabízejí flexibilní řešení pro podnikové uživatele.[8]
3. **Hardwarové licence:** Některé specifické hardwarové komponenty mohou také vyžadovat licencování, například speciální síťové prvky nebo serverové technologie, které jsou chráněné patenty nebo jsou poskytovány na základě licenčního ujednání.

Tyto licence jsou zmíněny s cílem ukázat, s jakými se bude v této práci pracovat. Existuje mnoho různých typů licencí, ale v této práci se zaměřujeme na konkrétní druhy, se kterými pracujeme.

## 2.5 Malá firma

Abychom správně porozuměli pojmu malá firma, domluvili jsme se s vedoucím, že budeme brát v úvahu firmy, které se spíše nezabývají IT jako hlavní náplní své činnosti a zároveň nemají velký počet IT vybavení. Takové firmy mají obvykle méně než 50 zaměstnanců a předpokládáme, že vlastní do 50 až 100 zařízení.[9]

## 2.6 IT vybavení

V bakalářské práci jsem se zaměřil především na počítače, servery a tiskárny, které tvoří základní IT infrastrukturu malých firem, jak bylo definováno v předchozí podkapitole. Ostatní zařízení, která nejsou explicitně zahrnuta v této kategorii, budou v mé práci zahrnuta do kategorie "Ostatní". Tímto způsobem si udržíme přehledný přístup a budeme věnovat pozornost zejména těm prvkům IT vybavení, které jsou pro malé firmy klíčové.

## ■ 2.7 Závěr

V této kapitole jsme získali přehled o základních pojmech, které jsou klíčové pro pochopení této práce. Zaměřili jsme se na malé firmy využívající standardní IT vybavení a definovali jsme roli externího správce sítí, který zajišťuje jejich správu.

V následující kapitole se podíváme na nejčastější činnosti externího správce sítí a praktické aspekty správy IT infrastruktury v malých firmách.



## Kapitola 3

### Ze života správce

Tato kapitola představuje integraci teoretických základů definovaných v předchozí kapitole s praktickými scénáři a user stories, které ilustrují každodenní práci externího správce IT. Zde spojíme teoretické znalosti o správcích, počítačových sítích, evidenčních systémech a licencích s praktickými příklady z každodenního života správce, které ukazují, jak tyto pojmy ovlivňují jejich pracovní realitu. Tyto příběhy nám umožňují lépe porozumět, jak může navrhovaný evidenční systém usnadnit a zefektivnit práci správců. [10][11]

#### Struktura popisu user stories:

Struktura user stories je následující:

1. **Pozadí:** Krátký popis kontextu a úkolů správce IT infrastruktury.
2. **Problém:** Specifikace problému, který správce řeší, a proč je tento problém důležitý.
3. **Řešení:** Popis navrhovaného řešení, které by měl navrhovaný evidenční systém poskytnout.
4. **Informatická podpora:** Popis, jak konkrétně evidenční systém podporuje řešení problému a zjednodušuje správu IT infrastruktury.

### 3.1 User Story: Řešení hardwarových problémů a rozšiřování IT infrastruktury

**Pozadí:** Správce IT je zodpovědný za udržování hardwaru ve funkčním stavu napříč všemi klienty. To zahrnuje identifikaci a rychlou reakci na hardwarové problémy, které mohou ovlivnit provoz organizace, a také plánování a realizaci rozšíření nebo obnovy stávajícího hardwaru.

**Problém:** Hardwarové problémy mohou způsobit významné přerušení práce, a je důležité je rychle identifikovat a opravit. Bez přehledného systému pro evidenci a sledování historie zařízení může být diagnostika a oprava zpóžděna kvůli dohledávání informací, což vede k větším provozním ztrátám.



šení licenčních podmínek nebo selhání správného fungování systémů kvůli nedostatečnému sledování termínů expirace a přehledu o aktuálním stavu licencí.

**Řešení:** Navrhovaný evidenční systém poskytuje jednotnou platformu pro správu a sledování všech typů licencí. Tento systém umožňuje snadné sledování a obnovování licencí, a správci mohou kdykoliv zobrazit aktuální stav licencí a zjistit, které licence se blíží ke konci platnosti. To jim umožňuje obnovovat licence a zajistit jejich dodržování, což zvyšuje právní a operační bezpečnost organizace.

**Informatická podpora:** Systém nabízí přehledné uživatelské rozhraní, které zobrazuje aktuální stav všech licencí, včetně softwarových a hardwarových. Správci mohou jednoduše přidávat, upravovat nebo obnovovat licence přímo v systému, což zajišťuje, že všechny informace jsou aktuální a správně evidovány. Toto rozhraní zefektivňuje správu licencí a snižuje administrativní zátěž tím, že centralizuje veškeré licenční informace na jednom místě.

## 3.4 User Story: Pravidelná údržba

**Pozadí:** Správce IT infrastruktury je zodpovědný za pravidelnou údržbu a podporu IT systémů pro mnoho firemních klientů. To zahrnuje plánování návštěv u klientů za účelem řešení různých problémů a úkolů, jako je konfigurace sítě, aktualizace softwaru nebo výměna hardwaru.

**Problém:** Před návštěvou klienta potřebuje správce přesně vědět, jaké úkoly musí na místě vykonat, a zajistit, aby byly všechny potřebné informace dostupné včas. Bez centralizovaného nástroje na správu těchto informací a plánování návštěv je obtížné efektivně organizovat práci. Správci jsou často nuceni udržovat si poznámky o potřebných úkolech na různých místech, což zvyšuje riziko zapomenutí klíčových úkolů nebo opomenutí důležitých informací.

**Řešení:** Navrhovaný evidenční systém nabízí správcům možnost efektivně si zorganizovat všechny návštěvy u klientů. Mohou si do systému zapsat jednotlivé akce, poznámky k plánovaným úkolům a jejich termíny. Díky tomu se správcům zjednoduší práce, udrží si lepší přehled o svém rozvrhu a zvýší svou efektivitu bez nutnosti vést poznámky na více místech nebo si všechno pamatovat.

**Informatická podpora:** Systém poskytuje integraci kalendáře a plánovacích nástrojů, které usnadňují správci zapisování, úpravy a sledování všech nadcházejících akcí a návštěv u klientů. Centralizovaný přístup umožňuje snadný přístup ke všem informacím o plánovaných událostech na jednom místě, což eliminuje potřebu různých platforem nebo papírové formy. Správci mohou přidávat detailní poznámky a specifikace k jednotlivým úkolům, což zlepšuje přípravu a efektivitu jejich práce. Tato funkcionality zjednodušuje organizaci návštěv, pomáhá předcházet přehlédnutí klíčových úkolů a zajišťuje, že všechny potřebné akce jsou provedeny včas a podle plánu.

## ■ 3.5 Závěr kapitoly

V této kapitole jsem prostřednictvím user stories ilustroval, jak může dobře navržený evidenční systém usnadnit a zefektivnit práci externího správce IT. Příběhy pokrývají široké spektrum činností - od správy softwarových aktualizací, přes řešení hardwarových problémů, správu licencí, až po plánování akcí u firem. Každá z těchto oblastí představuje specifické výzvy, které evidenční systém pomáhá řešit efektivněji, a ukazuje význam integrace moderních IT nástrojů do každodenního provozu.

Další kapitola obsahuje analýzu existujících řešení, která by mohla být použita, s cílem pochopit jejich funkčnost a přínos k dosažení cílů této práce.

## Kapitola 4

### Analýza programů pro podporu správy počítačových sítí

Tato kapitola se zaměřuje na analýzu existujících softwarových nástrojů určených pro podporu správy počítačových sítí malých firem. V předchozí kapitole jsme prostřednictvím různých user stories ilustrovali, jak může dobře navržený evidenční systém usnadnit a zefektivnit práci externího správce IT. V této kapitole se nejprve zaměříme na vyhodnocení použitelnosti existujících programů a posouzení, zda splňují potřeby a očekávání správců sítí, aniž by obsahovaly zbytečně složité nebo nadbytečné funkce.

Druhým hlavním cílem kapitoly je definice specifikací funkčních a nefunkčních požadavků, které by software měl splňovat, aby efektivně sloužil cílové skupině. Tato analýza požadavků je klíčová pro návržení ideálního softwarového řešení, odpovídajícího specifickým potřebám a očekáváním malých firem v oblasti IT správy.

#### 4.1 Metodika vyhledávání

Pro identifikaci relevantních softwarových nástrojů pro správu počítačových sítí jsem využil vyhledávač Google.<sup>1</sup> Klíčová slova jako „software pro správce počítačových sítí“ a „IT asset management software“ byla použita k nalezení programů vhodných pro tuto analýzu. Na základě výsledků vyhledávání jsem se zaměřil na tři softwarové nástroje: Audit Pro, Spiceworks a Lansweeper, které se jeví jako nejčastěji používané v oblasti správy počítačových sítí.<sup>2</sup> Tyto nástroje byly vybrány kvůli jejich širokému rozšíření a popularitě v IT sektoru, stejně jako na základě jejich dostupnosti a funkcí, které se zdály být přínosné pro účely této analýzy.

#### 4.2 Kritéria hodnocení softwarových nástrojů

Při hodnocení vybraných softwarových nástrojů pro správu počítačových sítí jsem stanovil následující kritéria:

<sup>1</sup><https://www.google.com>

<sup>2</sup><https://www.capterra.com/it-asset-management-software/>

- **Relevance funkcí:** Toto kritérium hodnotí, zda jsou funkce softwaru přímo relevantní a užitečné pro běžné úkoly našeho správce sítí. Zjišťujeme, zda program obsahuje nadbytečné nebo málo využívané funkce, které mohou zbytečně zvyšovat složitost práce. Například v podkapitole 3.3 jsme viděli, jak je důležité mít funkce pro sledování a obnovu licencí.
- **Složitost rozhraní:** Hodnocení se zaměřuje na uživatelské rozhraní a jeho složitost. Kritérium posuzuje, zda je rozhraní intuitivní a snadno pochopitelné pro zkušené správce sítí, nebo zda obsahuje zbytečně složité prvky, které mohou práci ztěžovat. Například v podkapitole 3.1 je klíčové, aby správce mohl rychle najít a opravit chyby bez zbytečného zdržování složitým rozhraním.

## 4.3 Dostupné nástroje

Následující hodnocení softwarových nástrojů pro správu počítačových sítí vychází z veřejně dostupných informací na webu, včetně obrázků a popisů produktů. Protože nebyly provedeny praktické testy těchto nástrojů, je možné, že některé zjištěné vlastnosti a funkce nemusí úplně odpovídat reálnému užívání. V rámci této analýzy se zaměřujeme na placené i neplacené verze těchto nástrojů, abychom zjistili, zda dokážou pokrýt potřeby malých firem.

### 4.3.1 Audit Pro



Obrázek 4.1: Přehled zařízení v Audit Pro.<sup>3</sup>

**Krátký popis:** Audit Pro<sup>4</sup> je nástroj určený pro správu IT aktiv a inventarizaci. Tento software je dostupný jako placené řešení.

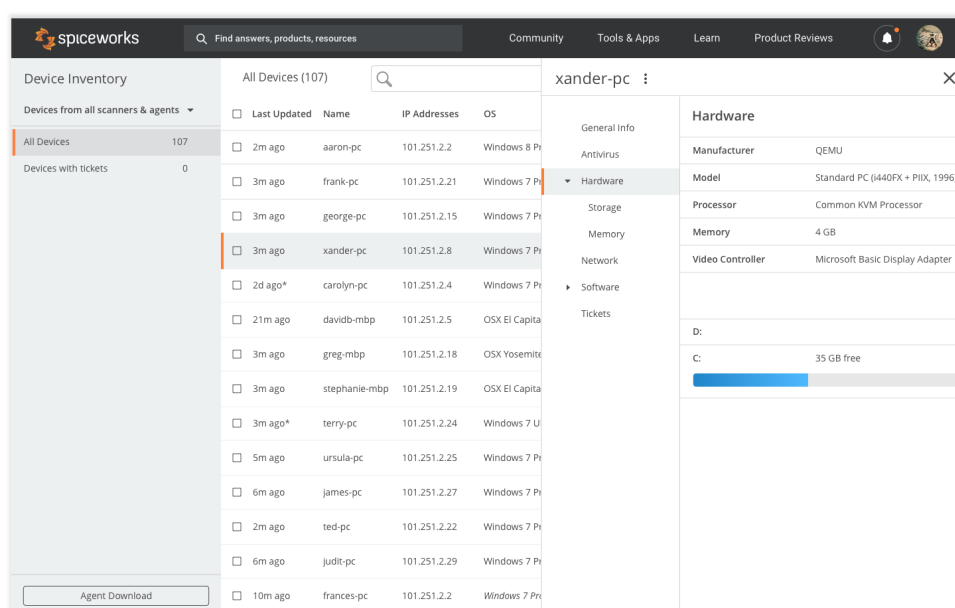
<sup>3</sup>Vystřižený obrázek z pdfka. Kapitola 5.5 Webové rozhraní Audit Pro

<sup>4</sup><https://www.auditpro.cz/cs/>

**Složitost rozhraní:** Rozhraní působí moderně a je intuitivně uspořádané, nicméně rozsáhlá nabídka funkcí a grafů může být pro externí správce malých firem až příliš komplexní.

**Relevance funkcí:** Ačkoliv funkce pokrývají široký rozsah potřeb správy IT, zdá se, že některé grafy a statistiky by mohly být pro malé firmy nadbytečné, což by mohlo vést k zbytečné složitosti a méně efektivnímu využití. Zdá se také, že program nenabízí správu více firem, což může omezovat jeho použitelnost pro externího správce.

### 4.3.2 Spiceworks



Obrázek 4.2: Přehled zařízení v Spiceworks<sup>5</sup>

**Krátký popis:** Spiceworks<sup>6</sup> je nástroj pro inventarizaci a správu IT aktiv, který umožňuje sledování hardware a software na síti. Nabízí užitečné funkce, jako je automatické objevování zařízení a jejich sledování. Tento software je dostupný ve verzi zdarma s možností placeného rozšíření.

**Složitost rozhraní:** Rozhraní je jasné a intuitivní, což se blíží ideálu, který bych si představoval pro vlastní řešení v mé práci. Nabízí efektivní přístup k informacím o zařízeních.

**Relevance funkcí:** Spiceworks nabízí komplexní funkce pro správu IT, ale nezahrnuje zdánlivě možnost přidávat a spravovat licence, ani efektivně rozlišovat zařízení a servery pro různé klienty. Toto omezení může být významnou překážkou pro správce, kteří potřebují spravovat IT prostředky pro více firem a preferují oddělený přehled pro každou z nich.

<sup>5</sup>Obrázek převzat z <https://www.spiceworks.com/free-pc-network-inventory-software/>

<sup>6</sup><https://www.spiceworks.com/>

### 4.3.3 Lansweeper

The screenshot shows the Lansweeper interface for asset D003. Key details include:

- Asset type:** Windows
- Last user:** D003\Lan
- OS:** Windows 10 Pro N (64 bit)
- Build:** 10.0.17134.648
- Version:** 1803
- Domain:** LS
- Manufacturer:** Dell Inc.
- Model:** OptiPlex 9020M
- Memory:** 8 GB DDR3
- Processor:** Intel Core i7-4785T CPU @ 2.20GHz
- Motherboard:** Dell Inc. 0Y5DDC
- Graphics:** Intel(R) HD Graphics 4600 1 GB
- Audio:** Intel(R) Display Audio, Realtek High Definition Audio
- Antivirus:** Windows Defender Enabled & Up To Date
- Network:** Intel(R) Ethernet Connection I217-LM - 98.90.96.CE.64.7C
- Harddisk:** C: SSD 500 (212.8 GB free of 465.3 GB)
- Warranty Service:** Ship date: 10/05/2015, Purchase country: Belgium. Service history table shows dates and descriptions.
- Relations:** Table showing connections to other assets.

Obrázek 4.3: Detailní přehled zařízení v Lansweeper<sup>7</sup>

**Popis:** Lansweeper<sup>8</sup> je nástroj pro správu IT aktiv, který umožňuje detailní sledování a inventarizaci hardwaru a softwaru v síti. Nabízí komplexní přehled o zařízeních, včetně operačních systémů a historie použití. Tento software je dostupný jako placené řešení, avšak nabízí omezenou verzi zdarma.

**Složitost rozhraní:** Rozhraní je podrobné, poskytující množství informací, které mohou být pro zkušené uživatele užitečné, avšak pro někoho může být tato složitost zbytečná. Zjednodušení rozhraní odstraněním nadbytečných dat by mohlo zlepšit jeho přehlednost a snadnější orientaci uživatelů.

**Relevance funkcí:** Lansweeper je plný funkcí, ale pro správu menších sítí se může zdát složitý a obsahuje mnoho funkcí, které nejsou vždy potřeba. Pro správce malých sítí by bylo lepší používat jednodušší nástroj, který se soustředí jen na základní funkce.

### 4.3.4 Závěr analýzy dostupných nástrojů

Analýza odhalila, že i když je na trhu dostupných mnoho softwarových nástrojů pro správu IT, často jsou příliš komplexní a finančně nákladné, což může být problematické pro malé firmy s omezenými zdroji. Tyto nástroje bývají navrženy s funkcemi vhodnými pro velké firmy nebo komplexní IT

<sup>7</sup>Obrázek převzat z Wikipedie

<sup>8</sup><https://www.lansweeper.com/>



infrastruktury, což v kontextu malých firem přináší zbytečnou zátěž a komplikuje procesy správy. Navíc, mnohé z těchto řešení neumožňují snadné použití pro správu IT více zákazníků bez složitých konfigurací nebo vysokých dodatečných nákladů.

Na základě této analýzy a diskuzí s vedoucím práce jsme identifikovali potřebu vytvoření softwarových řešení, která jsou speciálně přizpůsobena potřebám malých firem. Tyto nástroje by měly být jednoduché, cenově dostupné a flexibilní, což by umožnilo správcům efektivně řídit IT prostředky bez nepotřebné složitosti a přitom podporovat správu IT prostředků pro více zákazníků. Toto zjištění je klíčové pro definování relevantních funkčních a nefunkčních požadavků pro budoucí vývoj softwaru, který lépe vyhovuje potřebám malých podniků.

## 4.4 Určení požadavků na systém

Na základě předchozí analýzy softwarových nástrojů pro správu počítačových sítí, diskuzí s vedoucím práce a dříve definovaných user stories, jsem definoval následující specifikace funkčních a nefunkčních požadavků na systém. Tyto požadavky reflektují potřeby a očekávání malých firem v kontextu efektivity a jednoduchosti použití.[12]

### 4.4.1 Funkční požadavky (FP)

#### FP-1: Přehled firem

- Systém umožňuje zobrazení seznamu všech firem v systému.
- Systém poskytuje detailní informace o každé firmě, včetně kontaktů, adresy a dalších relevantních údajů.

#### FP-2: Úpravy informací o firmách

- Systém umožňuje přidávání nových firem.
- Umožňuje úpravy existujících informací o firmách, jako jsou kontaktní údaje, adresa a další.
- Poskytuje možnost odstranění firmy ze systému.

#### FP-3: Plánování událostí

- Systém umožňuje zobrazení akcí spojených s firmou, včetně návštěv a údržb.
- Umožňuje přidání poznámek k jednotlivým návštěvám nebo akcím.

#### FP-4: Seznam zařízení

- Systém umožňuje zobrazení všech zařízení přidružených k firmě.
- Poskytuje filtraci zařízení podle typu a stavu.



## ■ 4.5 Shrnutí kapitoly

Tato kapitola zhodnotila vybrané dostupné softwarové nástroje pro správu počítačových sítí s ohledem na potřeby malých firem. Na základě této analýzy a diskuzí s vedoucím práce byly definovány specifické funkční a nefunkční požadavky, které by ideální softwarové řešení mělo splňovat. Tyto požadavky jsou zásadní pro vytvoření systému, který je jednoduchý, intuitivní a efektivní, což je klíčové pro malé firmy s omezenými zdroji. Definované požadavky slouží jako základ pro návrh a vývoj optimálního softwarového řešení pro správu IT.

Následující kapitola se zaměří na návrh tohoto systému, který bude odpovídat specifikovaným požadavkům a potřebám cílové skupiny.



## Kapitola 5

### Návrh aplikace

V této kapitole popisují návrh aplikace, který vychází z požadavků sesbíraných v podkapitole 4.4 a z analýzy provedené ve spolupráci s vedoucím práce.

Návrh aplikace obsahuje doménový model, databázový model, případy užití, návrh technologií a pak nakonec návrh prototypu.

#### 5.1 Doménový model

Doménový model představuje druh diagramu ve fázi počátečního návrhu softwaru, který vytváří základ pro budoucí vývoj aplikace. Jde o specifickou formu class diagramu, která je orientována na objektové programování. Tento model je zjednodušený a obsahuje pouze základní entity systému reprezentované třídami bez metod, které jsou znázorněny pouze svými klíčovými atributy. Třídy mohou mít názvy a atributy zapsané s diakritikou a jsou vzájemně propojeny bez uvedení datových typů, což činí model platformově nezávislým.[13]

Návrh doménového diagramu na obrázku 5.1 zobrazuje strukturu systému pro správu zařízení ve firmě.

##### Zařízení

Hlavní entitou je *Zařízení*, které obsahuje všechny klíčové informace jako název zařízení, jeho status podle výčtového typu `DeviceStatusEnum` (Aktivní, Poškozené, Neaktivní, Ve službě) a typ zařízení podle výčtového typu `DeviceTypeEnum` (Počítač, Server, Tiskárna, Ostatní). Třídy jako *Počítač*, *Server*, a *Tiskárna* jsou specializacemi *Zařízení* a dědí všechny jeho atributy, přičemž rozšiřují tuto základní třídu o další specifické atributy a funkce.

##### Počítač

Specializace zařízení s atributy jako sériové číslo, operační systém dle `OperatingSystemEnum` (Windows, MacOS, Linux, Ostatní), RAM, CPU, úložiště a komentáře, které specifikují další detaily o počítači.



## ■ Adresa

Třída **Adresa** slouží k uchování adresních údajů firmy. Atributy této třídy zahrnují ulici, město, PSČ a stát. Tyto údaje jsou nezbytné pro identifikaci a komunikaci s danou firmou.

## ■ Akce

Navržena pro správce systému a obsahuje název akce, datum a popis. Tyto akce nejsou přímo organizovány firmou, ale slouží správci k plánování a sledování úkolů a aktivit spojených s každou firmou.

## ■ Licence

Každé zařízení může mít také přiřazenou **Licenci**, která obsahuje název licence, data platnosti, soubory a poznámky týkající se licence.

Celkově tato struktura umožňuje efektivní správu zařízení v organizaci s možností sledovat specifické události, licence a detailní informace o každém typu zařízení.

## ■ 5.2 Databázový model

Na základě doménového modelu a detailní specifikace systémových požadavků jsem vytvořil databázové schéma, na kterém je aplikace postavena. Návrh databázového modelu je zobrazen na obrázku 5.2.

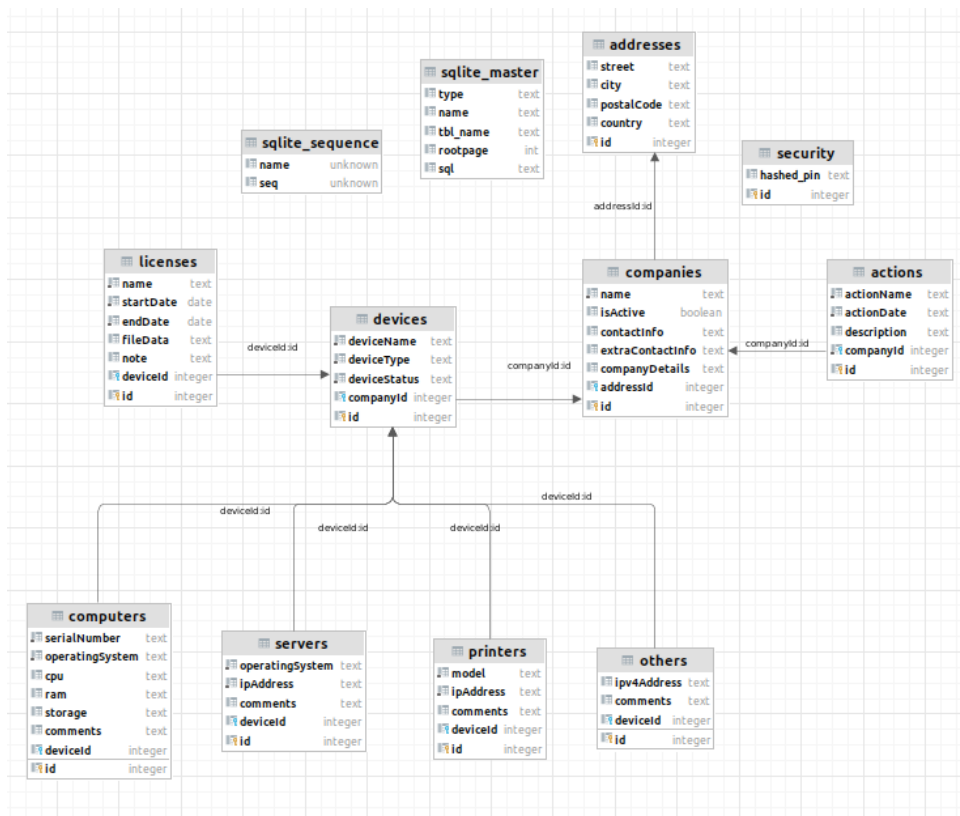
Databázový model zahrnuje systémovou tabulku `sqlite_master`, která je standardní součástí každé SQLite databáze<sup>1</sup>. Tato tabulka je používána pro správu a provoz databáze a obsahuje záznamy o všech databázových objektech, jako jsou tabulky, indexy, triggerů a pohledy. Je klíčová pro správné fungování celého systému, protože umožňuje SQLite engine procházet a modifikovat strukturu databáze.[14]

Tabulka `sqlite_master` obsahuje následující sloupce:

- **type** - Typ objektu v databázi (tabulka, index, trigger, view).
- **name** - Název objektu.
- **tbl\_name** - Název tabulky, k níž objekt patří.
- **rootpage** - Číslo kořenové stránky v databázovém souboru, kde začíná objekt.
- **sql** - SQL příkaz použitý k vytvoření objektu.

<sup>1</sup><https://www.sqlite.org/index.html>

Kromě `sqlite_master`, náš databázový model také využívá tabulku `sqlite_sequence` pro správu autoinkrementačních hodnot. Tato tabulka je automaticky vytvořena a aktualizována, kdykoli je použit atribut `AUTOINCREMENT`. Umožňuje bezpečnou sekvenční alokaci hodnot v primárních klíčích a zajišťuje, že každý nový záznam dostane jedinečné identifikační číslo.[14]



Obrázek 5.2: Databázový model

### 5.3 Případy užití - Use Case(UC)

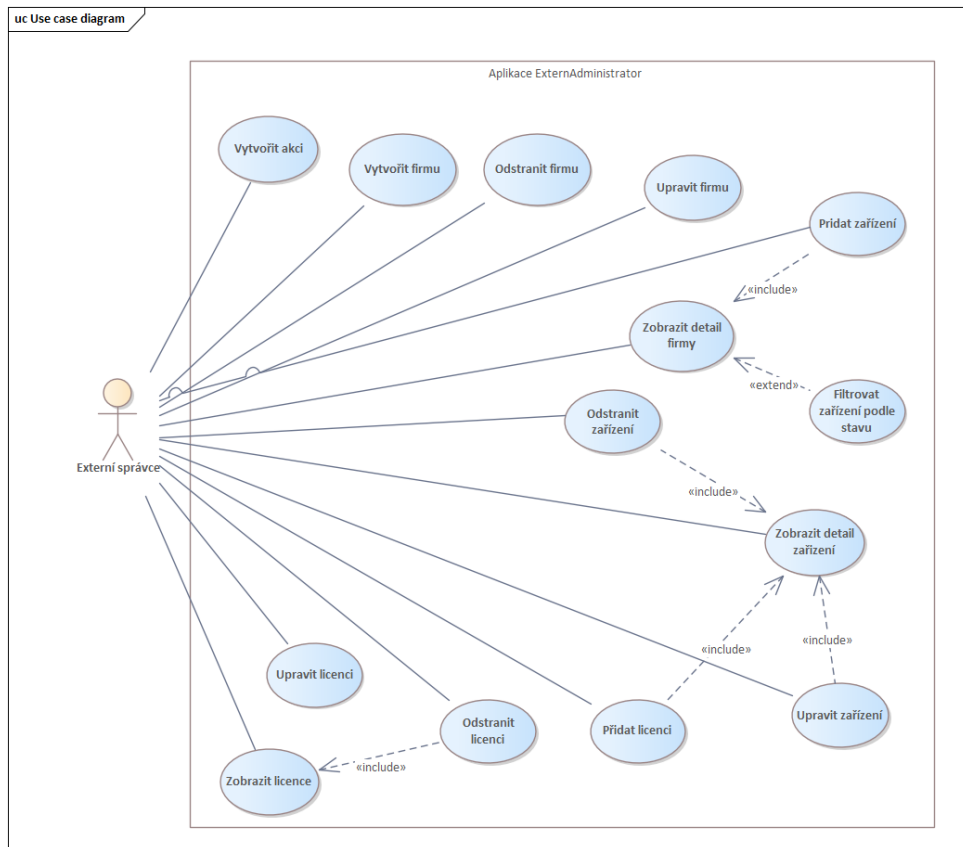
Případy užití popisují, jak a kým bude systém využíván. Hlavním cílem modelování případů užití je definovat rozsah systému. Diagram případů užití se skládá z jednotlivých případů užití, aktérů a vztahů mezi nimi. Aktéři jsou uživatelé nebo systémové entity, které se systémem interagují a ovlivňují jeho chování. Modelování případů užití představuje metodu pro zaznamenání funkcí, které systém musí splnit. [15]

**Externí správce:** Externí správce je hlavním uživatelem systému. Má na starosti správu firem, zařízení a licencí v rámci informačního systému. Všechny případy užití jsou zaměřeny na jeho interakce a operace v systému, jak je znázorněno na obrázku 5.3.





Obrázek 5.3: Externí správce jako actor



Obrázek 5.4: Use case externího správce

**UC1: Vytvořit akci**

Umožňuje externímu správci vytvořit novou akci. Detaily akce, jako jsou účel, datum, a čas jsou specifikovány.

**UC2: Vytvořit firmu**

Umožňuje externímu správci vytvořit nový záznam firmy. Detaily firmy, jako jméno, adresa, a kontaktní informace jsou vyžadovány.

**UC3: Odebrat firmu**

Umožňuje externímu správci odstranit existující záznam firmy z databáze.

**UC4: Upravit firmu**

Umožňuje externímu správci upravit detaily existující firmy, včetně názvu, adresy a kontaktních informací.

**UC5: Přidat zařízení**

Umožňuje externímu správci přidat nové zařízení do systému. Vyžaduje specifikace jako typ zařízení a přiřazení k firmě.

**UC6: Odebrat zařízení**

Umožňuje externímu správci odstranit existující zařízení z databáze.

**UC7: Upravit zařízení**

Umožňuje externímu správci upravit specifikace existujícího zařízení, jako jsou jeho parametry.

**UC8: Zobrazit detail firmy**

Umožňuje externímu správci zobrazit kompletní detaily firmy včetně souvisejících zařízení.

**UC9: Zobrazit detail zařízení**

Umožňuje externímu správci zobrazit kompletní detaily zařízení a včetně specifikací.

**UC10: Zobrazit licence**

Umožňuje externímu správci zobrazit seznam licencí přidružených k firmě nebo zařízení.

**UC11: Upravit licenci**

Umožňuje externímu správci upravit detaily existující licence, včetně její platnosti a parametrů.

**UC12: Odebrat licenci**

Umožňuje externímu správci odstranit existující licenci z databáze.

**UC13: Přidat licenci**

Umožňuje externímu správci přidat novou licenci do systému, specifikuje se typ a platnost.

**UC14: Filtr zobrazení zařízení podle stavu**

Umožňuje externímu správci filtrovat zařízení na základě jejich stavu (např. v provozu, mimo provoz).

## 5.4 Výběr technologií

Na počátku mého projektu jsem zvažoval použití frameworku Spring Boot<sup>2</sup>, což je často preferovaná volba pro webové aplikace. Po důkladném prozkoumání a praktických zkušenostech s tímto nástrojem jsem však přehodnotil svůj původní záměr. Zjistil jsem, že pro specifické požadavky mé aplikace považuji Spring Boot za příliš komplexní.[16]

V rámci diskuse s mým vedoucím práce jsme dospěli k závěru, že pro účely mé bakalářské práce bude vhodnější a efektivnější použít jednodušší technologii. Rozhodl jsem se tedy pro JavaFX<sup>3</sup>, která umožňuje efektivní vývoj desktopových aplikací bez potřeby komplexních frameworků. Toto rozhodnutí mi umožnilo příměji se zaměřit na specifické funkcionality a požadavky projektu. [17]

K další důležité změně došlo ve způsobu nasazení aplikace. Místo serverového řešení je aplikace navržena tak, aby fungovala lokálně na počítačích externího správce. Tento přístup zvyšuje kontrolu nad provozem a zabezpečením dat.

Pro správu dat jsem vybral databázový systém SQLite<sup>4</sup>, přístupovaný přes JDBC. SQLite je ideální pro menší informační systémy a nabízí optimální kombinaci funkcionality, jednoduchosti a efektivity. Tato databáze nevyžaduje složitou konfiguraci a poskytuje dostatečné možnosti pro správu dat v malých podnicích, což přesně odpovídá potřebám mé aplikace.[18]

Tato technologická rozhodnutí reflektují mé snahy o vytvoření efektivního a zároveň jednoduše udržovatelného řešení, které odpovídá specifickým potřebám správcům sítí malých firem.

## 5.5 Návrh prototypu

V souladu s rozhodnutím o vývoji desktopové aplikace, které bylo učiněno v předchozí kapitole 5.4, jsem se rozhodl vytvořit low-fidelity prototyp v programu Figma<sup>5</sup>. Cílem tohoto prototypu bylo vizualizovat základní layout a funkční prvky aplikace, jež odpovídají specifikacím projektu.[19]

Obrázky 5.5, 5.6, a 5.7 poskytují první náhledy s krátkým popisem jejich užití v kontextu aplikace.

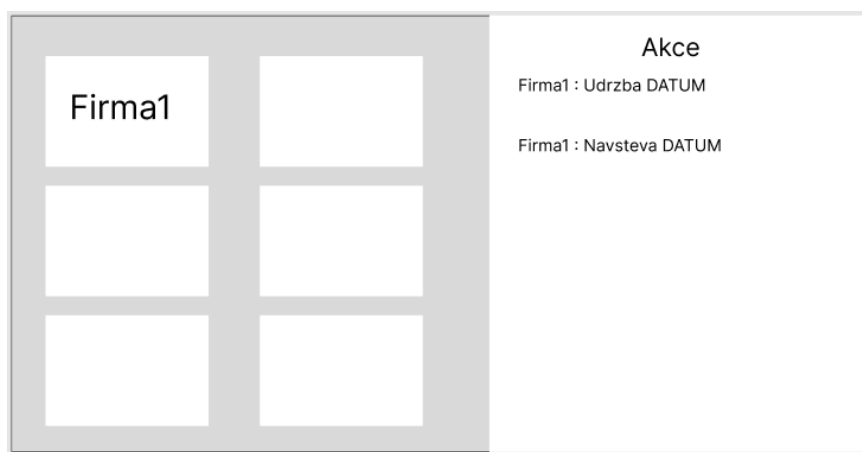
---

<sup>2</sup><https://spring.io/projects/spring-boot>

<sup>3</sup><https://openjfx.io/>

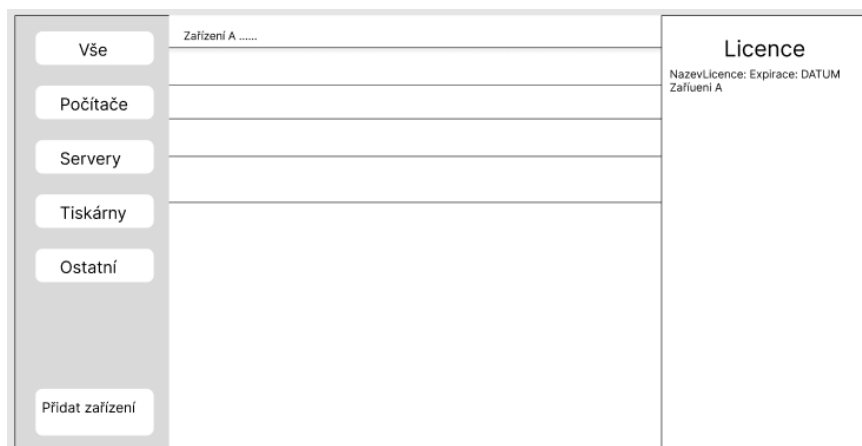
<sup>4</sup><https://www.sqlite.org/>

<sup>5</sup><https://www.figma.com/>



Obrázek 5.5: Hlavní stránka

**Hlavní obrazovka aplikace:** Na hlavní obrazovce aplikace jsou na levé straně umístěny firmy, které jsou klienty informačního systému. Napravo od seznamu firem se nachází sekce s akcemi externího správce, které se vztahují k jednotlivým firmám. Toto uspořádání umožňuje rychlý přehled o aktuálních úkonech a akcích, jak je znázorněno na obrázku 5.5.



Obrázek 5.6: Detail Firmy

**Detail firmy:** Po výběru konkrétní firmy z levé části se uživateli zobrazí detailní informace o firmě. V této části jsou zobrazeny seznamy zařízení spadajících pod danou firmu, a k nim přiřazené licence. Detailní pohled na zařízení obsahuje specifikace a další relevantní informace včetně přidružených licencí a jejich platnosti, jak je znázorněno na obrázku 5.6.

Obrázek Počítač	Seriové číslo: 3AD3492	Licence Windows podpora DATUM
	Operační systém: Windows	
	_____	
	_____	
Komentáře		

Obrázek 5.7: Detail zařízení

**Detail zařízení:** V detailu zařízení uživatel nalezne kompletní specifikace a technické informace týkající se konkrétního zařízení. Zahrnuty jsou informace jako model, sériové číslo, operační systém a status licencí. Tato sekce také umožňuje přístup k historii servisních zásahů a je možné z ní přímo spravovat nebo obnovovat licence, jak je znázorněno na obrázku 5.7.

### ■ 5.5.1 Závěr a diskuze

Tento prototyp slouží jako vizuální nástroj pro další diskuse s vedoucím práce a jako základ pro vývoj vysokoúrovňového prototypu, který bude dále rozvíjet uživatelské rozhraní. Po diskusích s vedoucím práce jsme se shodli na celkovém vzhledu aplikace a provedli několik drobných úprav, které se projeví ve finální verzi uživatelského rozhraní. Vzhledem k specifické povaze aplikace jsme se rozhodli, že tento prototyp dále netestujeme a testování jsme odložili až na finální verzi aplikace, kde jsme mohli lépe posoudit uživatelskou přívětivost.



## Kapitola 6

### Implementace aplikace

Na základě předchozího návrhu aplikace, který byl důkladně popsán v kapitole o návrhu, jsem přistoupil k implementační části bakalářské práce. Návrh aplikace poskytl základní kostru a směrnice pro vývoj, které byly nyní aplikovány do praktického řešení.

#### 6.1 Použité nástroje

Během implementace jsem využil řadu nástrojů, které podporovaly různé aspekty vývoje, od psaní kódu po správu databáze a verzí. Zde je přehled hlavních nástrojů a důvodů jejich výběru:

##### 6.1.1 Vývojové prostředí: IntelliJ IDEA

Pro psaní a správu kódu jsem použil komerční integrované vývojové prostředí (IDE) IntelliJ IDEA od společnosti JetBrains<sup>1</sup>. Toto prostředí je vysoce preferované pro jeho schopnost podporovat různé programovací jazyky a frameworky, což zahrnuje rozsáhlou podporu pro Java, která byla použita v tomto projektu. Dále IntelliJ IDEA nabízí pokročilé funkce pro refaktoring kódu, ladění a integraci s verzovacími systémy, což výrazně usnadňuje vývojový proces.[20]

##### 6.1.2 SDK: Java 8

Pro vývoj aplikace byl vybrán Java Development Kit (JDK) verze 8. Tato verze byla zvolena kvůli své stabilitě, široké podpoře a kompatibilitě s většinou knihoven, které jsem plánoval použít. Kromě toho, Java 8 je známá svou dobrou kompatibilitou s JavaFX, což bylo klíčové, jelikož některé novější verze Java mohou přinášet komplikace v integraci s JavaFX.[21]

Java 8 také přináší několik klíčových vylepšení, včetně lambda výrazů a stream API, které zjednodušují psaní čistého a efektivního kódu, což umožnilo efektivnější vývoj a udržitelnost aplikace.[22] I když je možné použít novější verze JDK, vycházel jsem ze svých zkušeností a zůstal jsem u verze 8, která se osvědčila jako spolehlivá a dobře podporovaná pro tento projekt.

---

<sup>1</sup><https://www.jetbrains.com/>





```

5     private String postalCode;
6     private String country;
7
8     public Address(int id, String street, String city,
9         String postalCode, String country) {
10        this.id = id;
11        this.street = street;
12        this.city = city;
13        this.postalCode = postalCode;
14        this.country = country;
15    }

```

Ukázka 6.1: Ukázka třídy Address

## 6.2.2 View

View, neboli zobrazovací vrstva, je zodpovědná za prezentaci dat uživateli. Ve své aplikaci jsem implementoval zobrazovací vrstvu pomocí JavaFX, což mi umožňuje dynamicky zobrazovat data a zároveň zachytávat uživatelské vstupy. Pomocí JavaFX Scene Builder jsem navrhl uživatelské rozhraní, které intuitivně prezentuje informace a umožňuje uživatelům snadno s aplikací interagovat. Vývoj uživatelského rozhraní je také podpořen externími CSS soubory, které definují vizuální styl komponent, jako jsou tlačítka a layouts, umožňující tak jednotný a přizpůsobitelný vzhled celé aplikace.

## 6.2.3 Controller

Controllery v mé aplikaci zajišťují komunikaci mezi modelem a zobrazovací vrstvou. Každý controller zpracovává logiku, která reaguje na uživatelské akce, a řídí aktualizace dat modelu, které jsou poté reflektovány v pohledu. Pro každou hlavní funkční oblast aplikace využívám samostatný controller, což mi umožňuje udržet kód organizovaný a snadno rozšiřitelný.

```

1 public class ApplicationController {
2     private Stage primaryStage;
3     private StackPane rootLayout;
4     private BorderPane mainBorderPane;
5
6     public ApplicationController(Stage primaryStage) {
7         this.primaryStage = primaryStage;
8         rootLayout = new StackPane();
9         mainBorderPane = new BorderPane();
10
11        rootLayout.getChildren().add(mainBorderPane);
12
13        Scene scene = new Scene(rootLayout, 1000, 800);
14        scene.getStylesheets().add(getClass().getResource
15            ("/style.css").toExternalForm());
16        primaryStage.setScene(scene);
17        primaryStage.setTitle("Prehled aplikace");

```

```

17     primaryStage.show();
18 }
19 }

```

**Ukázka 6.2:** Inicializace zobrazení a scény v ApplicationController

Výše uvedený kód (ukázka 6.2) demonstruje základní strukturu aplikace v JavaFX. *Stage* je hlavní okno aplikace, zatímco *StackPane* a *BorderPane* slouží jako layout manager pro organizaci obsahu uživatelského rozhraní. *Scene* pak obsahuje všechny UI komponenty. Toto uspořádání umožňuje efektivní oddělení logické struktury od grafické prezentace, což vede k modularitě a lepší udržitelnosti kódu.

## 6.2.4 DAO (Data Access Objects)

DAO třídy zajišťují přímou interakci s databází SQLite prostřednictvím JDBC. Tyto třídy poskytují metody pro CRUD operace – vytváření (Create), čtení (Read), úpravy (Update) a mazání (Delete) datových entit v databázi. Každá DAO třída zajišťuje:

- **Spojení s databází:** Inicializace spojení se serverem SQLite pomocí `DriverManager.getConnection()`.
- **Podpora cizích klíčů:** Aktivace podpory cizích klíčů v SQLite pro zachování referenční integrity dat.
- **CRUD operace:** Implementace metod jako `getActionById`, `insertAction`, `updateAction`, a `deleteAction`, které umožňují manipulaci s entitami v databázi.
- **Zpracování výjimek:** Zachycení a zpracování SQL výjimek pro informování o potenciálních chybách při databázových operacích.

Tato struktura a přístup umožňují efektivní správu a manipulaci s daty v rámci aplikace bez nutnosti použití komplexnějších ORM frameworků jako je Spring Data.

```

1 public class DeviceDao {
2     public Connection conn;
3
4     public DeviceDao(String dbUrl) {
5         try {
6             this.conn = DriverManager.getConnection(dbUrl
7             );
8         } catch (SQLException e) {
9             e.printStackTrace();
10        }
11    }
12
13    public int insertDevice(Device device) {
14        String sql = "INSERT INTO devices (deviceName,
15            deviceStatus, deviceType, companyId) VALUES
16            (?, ?, ?, ?)";

```

```

14         int id = 0;
15
16         try (PreparedStatement pstmt = conn.
17             prepareStatement(sql, Statement.
18                 RETURN_GENERATED_KEYS)) {
19             pstmt.setString(1, device.getDeviceName());
20             pstmt.setString(2, device.getDeviceStatus().
21                 name());
22             pstmt.setString(3, device.getDeviceType().
23                 name());
24             pstmt.setInt(4, device.getCompanyId());
25             int affectedRows = pstmt.executeUpdate();
26
27             if (affectedRows > 0) {
28                 try (ResultSet rs = pstmt.
29                     getGeneratedKeys()) {
30                     if (rs.next()) {
31                         id = rs.getInt(1);
32                     }
33                 }
34             } catch (SQLException e) {
35                 System.out.println(e.getMessage());
36             }
37
38             return id;
39         }
40     }

```

Ukázka 6.3: Ukázka Device DAO

Výše uvedený kód (ukázka 6.3) ukazuje implementaci třídy `DeviceDao`, která zajišťuje operace s databází pro zařízení. Metoda `insertDevice` vkládá nové zařízení do databáze a vrací vygenerované ID zařízení, což umožňuje další manipulaci s tímto zařízením v aplikaci.

### 6.2.5 Service

Vrstva služeb v aplikaci funguje jako most mezi DAO a controllery. Umožňuje to implementaci složitější logiky, která může zahrnovat transakční zpracování a operace rozšířené na více tabulek.

### 6.2.6 Validace

Validace vstupních údajů je důležitou součástí zajištění správného fungování aplikace. V této implementaci je validace zajištěna jak na frontendu (FE), tak na backendu (BE):

- **Omezení délky textu (FE):** Pro pole, jako jsou například název firmy a kontaktní informace, je použito omezení délky vstupu pomocí `TextFormatter`, aby uživatel nemohl zadat více než povolený počet

znaků. Například pro název firmy je maximální délka nastavena na 14 znaků a pro kontaktní informace na 9 znaků.

- **Formát čísel (FE):** Pro poštovní směrovací číslo je použit `TextFormatter`, který umožňuje pouze číslice a omezuje délku na 5 znaků.
- **Povinná pole (FE a BE):** Před uložením změn nebo přidáním nové firmy se kontroluje, zda jsou všechna povinná pole vyplněna. Pokud některé povinné pole chybí, zobrazí se uživateli chybová zpráva na frontendu a backend rovněž zajišťuje, že žádná prázdná pole nebudou zpracována.
- **Unikátní název firmy (BE):** Při úpravě názvu firmy se kontroluje, zda se nezměnil název firmy, pokud existují soubory asociované s firmou. Pokud ano, uživatel je upozorněn, že název firmy nelze změnit.
- **Neuložené změny (FE):** Při zavírání dialogového okna se kontroluje, zda došlo k neuloženým změnám. Pokud ano, zobrazí se uživateli upozornění s možností uložit změny před zavřením.

Backend zajišťuje další úroveň validace prostřednictvím `DeviceDao` a `CompanyDao` tříd, které kontrolují integritu dat před jejich uložením do databáze.

```

1 TextField companyNameField = new TextField();
2 companyNameField.setTextFormatter(new TextFormatter<>(
    change -> change.getControlNewText().length() <= 14 ?
    change : null));
3
4 TextField contactInfoField = new TextField();
5 contactInfoField.setTextFormatter(new TextFormatter<>(
    change -> change.getControlNewText().length() <= 9 ?
    change : null));
6
7 TextField postalCodeField = new TextField();
8 postalCodeField.setTextFormatter(new TextFormatter<>(
    change -> change.getControlNewText().matches("\\d*")
    && change.getControlNewText().length() <= 5 ? change :
    null));

```

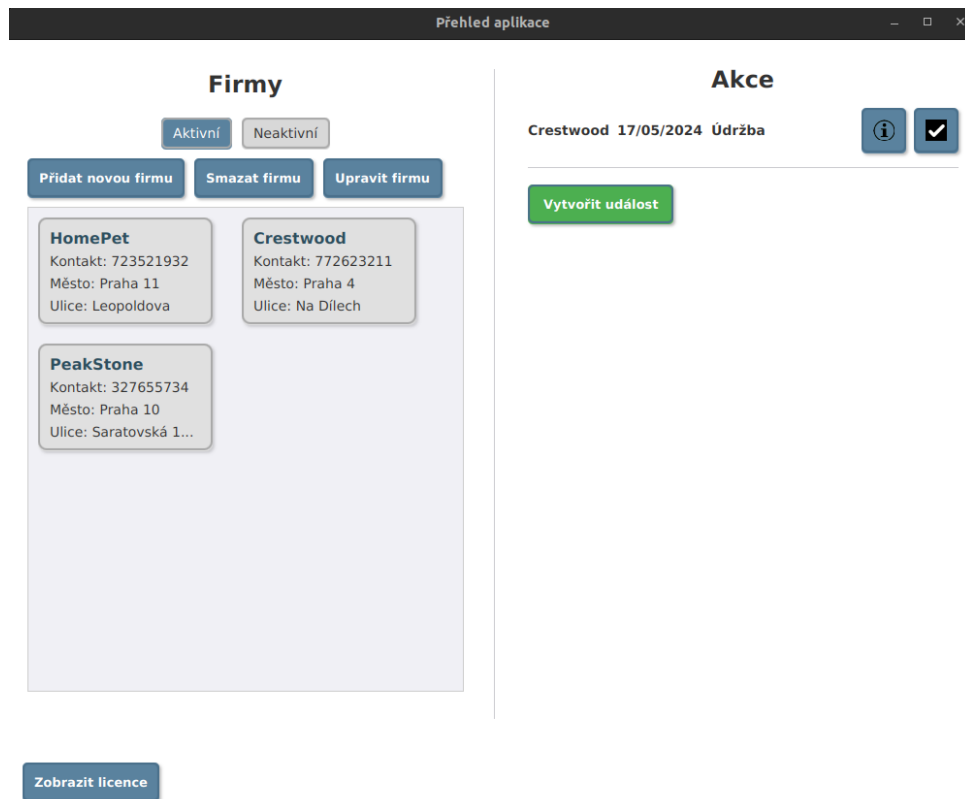
**Ukázka 6.4:** Validace vstupů v `CompanyController`

Výše uvedený kód ukazuje použití `TextFormatter` pro omezení délky vstupu a formátu čísel na frontendu. Na backendu je validace zajištěna například v metodách tříd `DeviceDao` a `CompanyDao`, které kontrolují správnost dat před jejich uložením do databáze.

## 6.3 Uživatelské rozhraní

Uživatelské rozhraní mé aplikace bylo navrženo s využitím JavaFX, frameworku, který umožňuje komplexní a flexibilní design GUI pro desktopové aplikace. Celkový design rozhraní vychází z návrhu prototypu vytvořeného v Figma, jak je detailně popsáno v sekci 5.5 *Návrh prototypu*.

### 6.3.1 Hlavní stránka aplikace



Obrázek 6.1: Hlavní rozhraní aplikace zobrazující správu firem a akcí

### Architektura rozhraní

Hlavní okno aplikace (ukázka na obrázku 6.1) je strukturováno pomocí *BorderPane*, což je layout manager poskytovaný JavaFX, který umožňuje efektivní rozdělení prostoru na centrální obsah a periferní obsahy, jako jsou hlavička, postranní menu a patička. Centrální část, kde jsou zobrazovány hlavní informace, je implementována pomocí *StackPane*, který umožňuje vrstvení komponent nad sebou a je ideální pro přechody mezi různými uživatelskými pohledy.

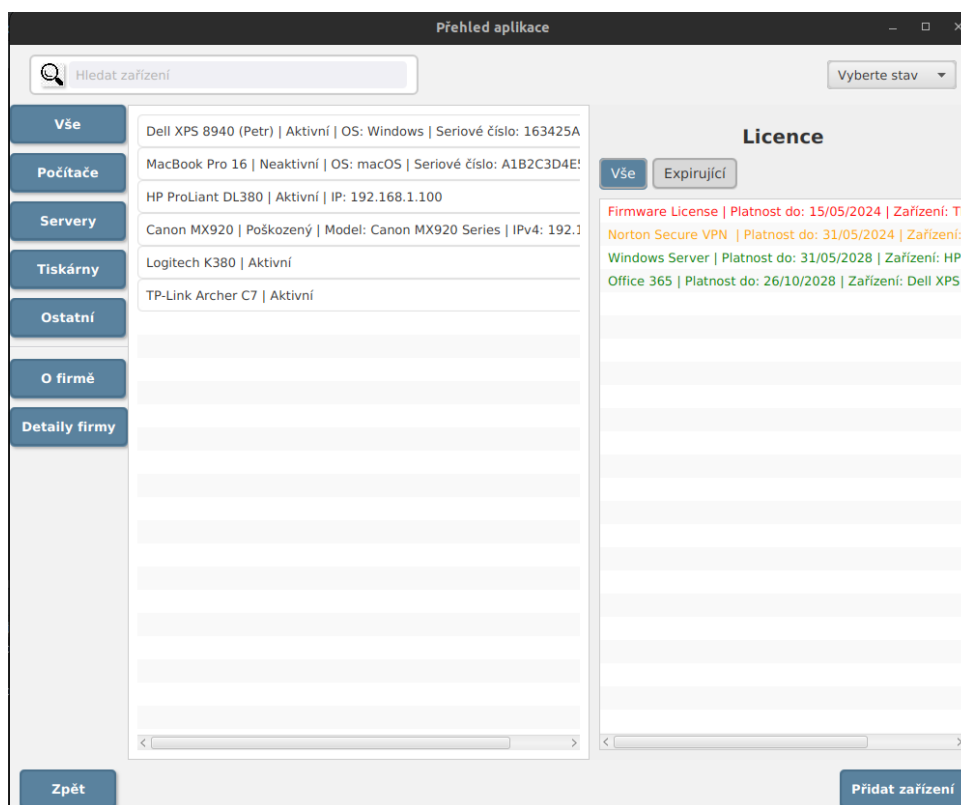
### Správa firem

V levé sekci *BorderPane* je umístěn *VBox*, obsahující dynamicky generované tlačítka, která reprezentují jednotlivé firmy. Tlačítka jsou implementována jako instance třídy *Button* s grafickými komponentami stylizovanými pomocí CSS, což umožňuje flexibilní a esteticky příjemné zobrazení. Tlačítka reflektují aktuální data a umožňují rychlý přístup k detailům o firmách.

## ■ Správa akcí

Pravá část *BorderPane* je vyhrazena pro správu akcí. Akce jsou zobrazeny v *TilePane*, což zajišťuje adaptivní rozvržení komponent v závislosti na velikosti okna. Pro každou akci je vytvořen *HBox*, který obsahuje popisky a interaktivní tlačítka s ikonami pro detailní zobrazení a změnu stavu akce. Toto uspořádání zlepšuje navigaci a usnadňuje uživatelské interakce.

### ■ 6.3.2 Detail firmy



Obrázek 6.2: Detail firmy

Pro detailní zobrazení informací o firmě (ukázka na obrázku 6.2) je v aplikaci implementován modální dialog, který se otevírá po výběru konkrétní firmy z hlavního seznamu firem. Tento dialog je realizován třídou *DeviceController*, která obsahuje logiku pro správu zařízení spojených s danou firmou. Zde je několik klíčových aspektů a funkcionality tohoto rozhraní:

## ■ Architektura a Layout

- **BorderPane:** Celkové rozložení dialogu je založeno na *BorderPane*, který rozděluje obsah na horní navigační pruh, centrální seznam zařízení, pravý panel pro detailní informace o licencích a dolní panel s tlačítky pro akce.

- **ListView:** Centrální část obsahuje `ListView` pro zobrazení zařízení, které jsou spojené s danou firmou. Každé zařízení je zobrazeno včetně základních informací, jako jsou typ zařízení, status a přiřazené licence.

### ■ Panel s licencemi

Pravý panel zobrazuje seznam licencí spojených se všemi zařízeními firmy. Licencím jsou přiřazeny barvy dle jejich aktuální platnosti:

- **Červená:** Licence, které již expirovaly nebo expirují do týdne.
- **Žlutá:** Licence, které expirují do tří měsíců.
- **Zelená:** Licence, které mají platnost delší než tři měsíce.

Licence jsou dynamicky filtrovány a řazeny v `ListView` na základě jejich expirace, což uživatelům umožňuje rychle identifikovat a spravovat ty, které vyžadují pozornost.

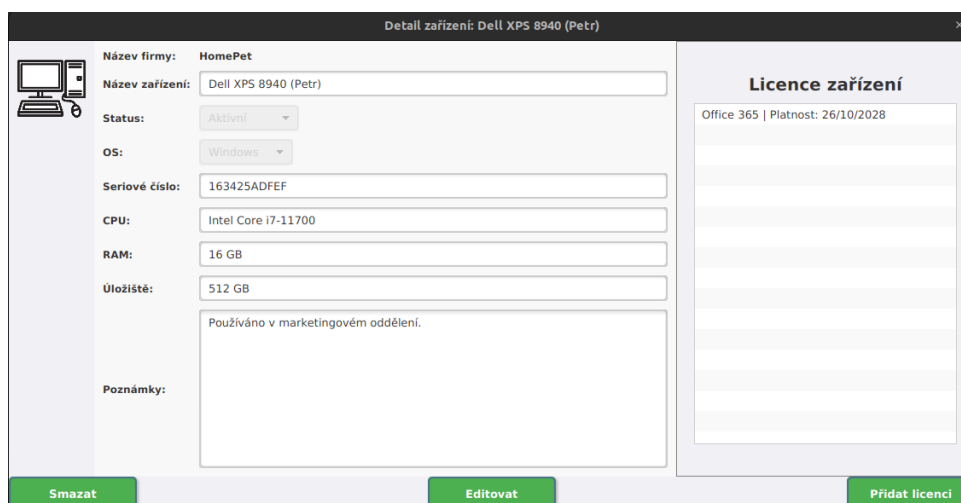
### ■ Funkcionalita

- **Dynamické vyhledávání:** Uživatelé mohou využívat vyhledávací pole pro filtraci zařízení podle jména. Toto pole je součástí horního navigačního pruhu a je integrováno s ikonou lupy pro vizuální identifikaci.
- **Filtrování zařízení:** Na levém panelu jsou tlačítka umožňující filtraci zařízení dle typu (počítače, servery, tiskárny, ostatní). Tímto způsobem může uživatel rychle najít potřebná zařízení bez nutnosti procházení celého seznamu.
- **Filtr stavu zařízení:** V pravém horním rohu dialogu je `ChoiceBox`, který umožňuje uživatelům filtraci zařízení podle jejich stavu, jako jsou 'Aktivní', 'Neaktivní', 'Poškozené' atd. Toto usnadňuje správu zařízení a umožňuje rychle identifikovat zařízení, která vyžadují pozornost.
- **Detailní informace a editace:** Po dvojkliku na zařízení v `ListView` se otevře další dialogové okno, kde jsou zobrazeny detailní informace o zařízení a kde je možné provádět editace. Toto zahrnuje změnu statusu zařízení, úpravu technických specifikací a přidávání nebo úpravu licencí. Více detailů viz Obrázek 6.3.

### ■ Interakce s uživatelem

- **Přidání nového zařízení:** Na dolním panelu je tlačítko pro přidání nového zařízení, které spustí dialog pro zadání parametrů nového zařízení.
- **Návrat do hlavního pohledu:** Tlačítko 'Zpět' umožňuje uživatelům návrat do hlavního pohledu aplikace.

### 6.3.3 Detail zařízení



Obrázek 6.3: Detail zařízení

Dialog pro zobrazení detailů zařízení v aplikaci (ukázka na obrázku 6.3) je implementován pomocí třídy `DeviceController`. Dialog využívá `BorderPane` pro rozdělení obsahu na několik segmentů: ikona a základní informace o zařízení na levém panelu, detailní informace a možnosti editace v centrálním panelu, seznam licencí na pravém panelu a panel s tlačítky pro akce v dolní části.

#### Funkcionality a rozvržení

- **Levý panel:** Obsahuje ikonu a základní identifikátory zařízení jako název a typ.
- **Centrální panel:** Zobrazuje detailní informace o zařízení, jako jsou operační systém, sériové číslo, specifikace hardwaru a poznámky. Uživatelé mohou informace editovat a ukládat změny.
- **Pravý panel:** Vypisuje seznam licencí přiřazených k zařízení, s možností přidání nové licence přímo k zařízení.
- **Dolní panel:** Poskytuje tlačítka pro akce jako je smazání zařízení, editace detailů a přidání nové licence.

#### Interaktivní prvky

- **Úpravy a správa detailů:** Dvojklik na zařízení v seznamu otevírá dialog pro detailní pohled a editaci, kde lze měnit stav zařízení, upravovat technické specifikace, a spravovat licence.



- **Přidání nové licence:** Tlačítko pro přidání nové licence umožňuje uživatelům rychle a efektivně rozšířit licenční portfolio zařízení bez opuštění detailu zařízení.

Další obrazovky aplikace, včetně detailů akcí, licencí, procesů přidávání firem a zařízení, jsou v příloze dokumentu pod sekci *Obrazovky aplikace (A.1)*.

## 6.4 Zajímavé funkce systému

### 6.4.1 Správa adresářů firem

Při přidání nové firmy do systému dochází automaticky k vytvoření dedikovaného adresáře pro tuto firmu v základním adresáři, který je definován v konfiguračním souboru `application.yml`. Tento adresář je určen pro uložení všech dokumentů a licencí, které se týkají dané firmy. Jakmile je k zařízení přiřazena nová licence, odpovídající soubor je zkopírován do adresáře příslušné firmy. Cesta k tomuto souboru je poté uložena v databázi systému jako textový řetězec. Tento systém umožňuje efektivně organizovat a spravovat licenční dokumenty bez nutnosti ukládání samotných souborů, jako jsou PDF, přímo do databáze, což zjednodušuje správu dat a snižuje nároky na databázový systém.

### 6.4.2 Pojmenování zařízení

Zařízení mohou být pojmenována dle konvence `[název zařízení]` (`[jméno uživatele]`), což umožňuje flexibilní identifikaci zařízení. Přestože jsou zařízení pojmenována uživatelskými jmény, vyhledávací funkce aplikace efektivně lokalizuje zařízení na základě jakéhokoli zadaného řetězce.

### 6.4.3 Bezpečnostní prvky

Aplikace pro zvýšení bezpečnosti vyžaduje při každém spuštění zadání čtyřmístného bezpečnostního PINu. Tento PIN je pro další zabezpečení uložen v databázi v zahashované formě. Pokud uživatel zadá PIN třikrát špatně, aplikace se automaticky vypne, aby se zabránilo neautorizovanému přístupu. V případě, že správce zapomene tento kód, má možnost jej resetovat pomocí *DB Browser for SQLite*<sup>4</sup> smazáním příslušné tabulky v databázi a nastavením nového bezpečnostního kódu.

### 6.4.4 Enumerace a stabilní pojmenování zařízení

Systém využívá statické enum typy pro definici stavů zařízení a typů operačních systémů. Tyto enumy jsou integrovány přímo do kódu aplikace, což vyžaduje zásah do kódu a jeho překompilaci pro jakékoli změny těchto typů.

<sup>4</sup><https://sqlitebrowser.org/>



- **DB Browser for SQLite:** Stáhněte a nainstalujte DB Browser for SQLite z oficiální stránky DB Browser for SQLite<sup>8</sup>.
- 2. **Vytvoření databáze:** Vytvořte prázdnou databázi SQLite podle konfigurace v `application.yml`.
- 3. **Nastavení konfiguračního souboru:** Upravte soubor `application.yml` na cestě, která je specifikována:

```
database:
  url: jdbc:sqlite:/path/to/your/databasefile

directory:
  path: /path/to/your/companyfiles
```

- 4. **Příprava prostředí:** Ujistěte se, že máte nainstalovaný Java Development Kit (JDK) a nastavené proměnné prostředí pro správnou funkčnost Java aplikací.
- 5. **Instalace závislostí:** V kořenovém adresáři projektu spusťte příkaz `mvn install` pro instalaci všech závislostí.

Spuštění aplikace je možné provést pomocí následujícího příkazu ve složce projektu:

```
mvn javafx:run
```

Ujistěte se, že konfigurační soubor `application.yml` je správně nastaven a že databázový soubor je přístupný před spuštěním aplikace.

## 6.6 Závěr

Implementační fáze projektu byla úspěšně dokončena s využitím nástrojů a technologií, které byly pečlivě vybrány v návrhové fázi. Použití JavaFX a MVC architektury umožnilo efektivní oddělení logiky aplikace od uživatelského rozhraní a zjednodušilo tak správu kódu a jeho rozšiřitelnost. Nástroje jako IntelliJ IDEA, Java Development Kit a DB Browser for SQLite poskytly stabilní prostředí pro vývoj, testování a správu databáze.

Abychom zajistili, že všechny implementované funkce fungují správně a splňují požadavky, bylo nutné provést testování. Následující kapitola se zaměřuje na metody a postupy testování, které byly použity k ověření správnosti systému a jeho komponent, včetně systematických testů a konkrétních scénářů uživatelského testování.

---

<sup>8</sup><https://sqlitebrowser.org/>



# Kapitola 7

## Testování aplikace

Testování je klíčovou fází vývoje softwaru, která umožňuje odhalení chyb a nedostatků, které mohou během implementace vzniknout. V této kapitole popíšu, jakým způsobem bylo testování prováděno na různých úrovních vývoje aplikace s ohledem na specifika implementace a struktury aplikace.

### 7.1 DAO testování

DAO (Data Access Object) testování je zaměřeno na ověření správnosti operací pro přístup k datům, což zahrnuje operace jako vkládání, aktualizace a mazání záznamů v databázi. Tyto testy jsou klíčové pro zajištění, že aplikace správně interaguje s databází a správně zpracovává data.

#### 7.1.1 Mockování

Pro simulaci databázových operací a ověření, že moje DAO metody správně volají SQL příkazy a zpracovávají výsledky, jsem použil knihovnu Mockito. Tento přístup mi umožnil izolovaně testovat funkcionality bez nutnosti připojení k reálné databázi, což mi efektivně pomohlo identifikovat a opravit chyby v kódu.

#### 7.1.2 Integrace s databází

Pro hlubší úroveň testování jsem implementoval integrační testy s použitím in-memory databáze. Tento krok mi umožnil testovat DAO vrstvu v podmínkách, které co nejvíce odpovídají reálnému provozu aplikace. Díky tomu jsem mohl ověřit správnou funkčnost všech databázových operací v kontextu celého systému.

Během testování jsem narazil na problém, kdy původní implementace vyžadovala samostatnou inicializaci databáze pro každé DAO. Toto řešení nebylo efektivní a vedlo k redundanci a složitější správě databázových spojení. Proto jsem přistoupil k úpravě systému tak, aby se databáze inicializovala centralizovaně pro celý systém najednou. Tato změna zjednodušila strukturu kódu, zlepšila rychlost provádění testů a zvýšila stabilitu databázových operací.

## 7.2 Uživatelské testování

Uživatelské testování, často nazývané také testování použitelnosti, je metoda užívaná k vyhodnocení, jak snadné je ovládnání desktopové aplikace. Tento typ testování se provádí se skutečnými uživateli, aby bylo možné zjistit, nakolik je aplikace intuitivní a jak snadno mohou uživatelé dosáhnout svých cílů.[25]

Před zahájením testování byla v aplikaci vytvořena řada ukázkových firm, akcí, zařízení a licencí. Tyto předpřipravené entity umožnily efektivněji demonstrovat a testovat funkcionality systému.

### 7.2.1 Proces testování

Testování bylo provedeno skupinou čtyř testerů, každý s různým zaměřením a pracovními návyky, aby se získala co nejširší zpětná vazba na funkčnost a intuitivnost aplikace. Veškeré testy byly prováděny na mém osobním počítači s operačním systémem Linux, což umožnilo přesně monitorovat a hodnotit chování aplikace.

### 7.2.2 Testovací scénáře

Testovací scénáře jsou navrženy tak, aby simulovaly reálné uživatelské interakce s aplikací a ověřily správnost jejích funkcí. Každý scénář zahrnuje konkrétní kroky, které uživatel provede, a sleduje, jak aplikace reaguje. Kromě těchto předdefinovaných scénářů měli testéři také možnost provádět vlastní testy podle svých specifických potřeb a představ, což umožnilo odhalit potenciální problémy, na které standardní scénáře nemusely upozornit.

#### Založení firmy, zařízení a akce

1. Přihlášení do systému: Uživatel se přihlásí do systému zadáním čtyřmístného PIN kódu.
2. Vytvoření firmy: Uživatel vyplní všechny potřebné údaje k založení nové firmy a uloží.
3. Vytvoření akce: Uživatel vybere nově vytvořenou firmu a zadá název a popis akce.
4. Přidání zařízení k firmě: Uživatel přidá zařízení, zadá jeho atributy a vybere typ.
5. Přidání licence k zařízení: Uživatel přidá licenci k zařízení, zadá potřebné informace o licenci a uloží.

#### Správa zařízení a licencí

1. Prohlížení zařízení: Uživatel si zobrazí seznam zařízení dle kategorie (např. počítače) a filtruje dle stavu.

2. Detail zařízení: Uživatel si otevře detail zařízení, zkontroluje a případně upraví jeho atributy.
3. Přidání licence k zařízení: Uživatel přidá další licenci k již existujícímu zařízení.
4. Správa expirujících licencí: Uživatel prochází seznam expirujících licencí a aktualizuje je.

### ■ Výsledky a doporučení

Během uživatelského testování byla získána řada cenných zpětných vazeb, které pomohly identifikovat klíčové aspekty použitelnosti aplikace a oblasti, které vyžadují další zlepšení. Po dokončení každého scénáře měli testeři možnost volně pracovat s aplikací, což jim umožnilo navrhnout jakákoli zlepšení nebo poskytnout zpětnou vazbu ohledně intuitivnosti aplikace. Podrobnější návrhy na zlepšení a rozšíření funkcionality aplikace jsou dále popsány v sekci *Návrhy na zlepšení a rozšíření funkcionality aplikace* (viz 7.2.2).

Zpětná vazba byla sbírána prostřednictvím osobních rozhovorů s testery, což umožnilo hlubší diskusi a okamžité objasnění jakýchkoli nejasností. Během tohoto procesu měli testeři možnost zkoumat nejen funkce definované scénáři, ale také další funkce aplikace, což poskytlo komplexní pohled na celkovou funkčnost systému.

Funkčnost všech hlavních funkcí definovaných ve scénářích byla ověřena bez problémů. Některé drobné problémy byly identifikovány mimo rámec definovaných scénářů, což vedlo k návrhům na zlepšení.

### ■ Shrnutí získaných poznatků z testování:

- Rozmístění tabulek a logická struktura hlavní stránky byly hodnoceny kladně pro svou přehlednost a intuitivnost.
- Tlačítko "Upravit firmu" na hlavní stránce bylo málo využíváno, uživatelé dávali přednost úpravám přímo v detailu firmy.
- Proces vytváření a zobrazování detailů akcí byl považován za jednoduchý a bez problémů.
- Uživatelé ocenili rozložení a přístupnost tlačítek po otevření detailu firmy.
- Filtrace zařízení podle stavu fungovala správně, avšak byla zaznamenána potíž s návratem k zobrazení všech zařízení bez nutnosti návratu na hlavní stránku.

### ■ Návrhy na zlepšení a rozšíření funkcionality aplikace:

- Přidat jasnější vysvětlení funkce tlačítka na hlavní stránce "Zobrazit licence", aby bylo zřejmé, co očekávat od této funkce.

- Umožnit návrat k zobrazení všech zařízení bez nutnosti opuštění aktuálního filtru nebo návratu na hlavní stránku.
- Zvětšit pole pro poznámky při přidávání zařízení pro lepší viditelnost.
- Opravit chování ChoiceBoxu při editaci zařízení, aby po uložení změn reflektoval aktuální stav a nezůstával v režimu úprav.
- Změna správy licencí: Uživatel označil, že licence obvykle nejsou mazány, ale spíše přestanou být aktivní. Doporučuje se implementovat funkci, která umožní licencím zůstat uloženým v systému, ale nezobrazovat je jako aktivní. Tímto způsobem bude možné zachovat historii licencí.

### 7.3 Závěrečné shrnutí

Na základě uživatelského testování bylo potvrzeno, že aplikace splňuje požadavky na funkčnost a je použitelná pro zamýšlené účely. Základní navigace a interakce byly hodnoceny kladně, uživatelé mohli efektivně a rychle dosáhnout svých cílů, což odráží dobře strukturované uživatelské rozhraní a logické uspořádání funkcí.

Byly však identifikovány některé oblasti vyžadující zlepšení, zejména v oblasti správy licencí a návratu k zobrazení všech zařízení. Tyto připomínky budou využity k dalšímu zlepšování a rozšiřování funkcionality aplikace.

V následující kapitole se zaměříme na aktuální stav aplikace a možné směry jejího budoucího vývoje, včetně úprav a vylepšení na základě zpětné vazby od uživatelů.



## Kapitola 8

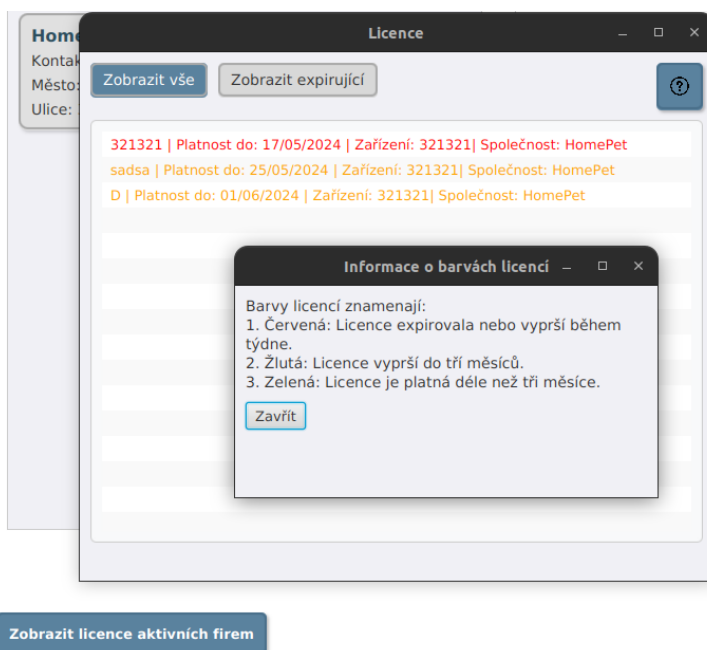
### Aktuální stav a budoucí rozvoj

Po důkladném testování aplikace a analýze výsledků provedených uživatelských testů jsem v této kapitole provedl úpravy a zlepšení na základě jejich návrhů. Návrhy na zlepšení a rozšíření funkcionality aplikace jsou uvedeny v kapitole 7.2.2.

#### 8.1 Implementace návrhů na zlepšení

##### 8.1.1 Vysvětlení funkce tlačítka Zobrazit licence

Na hlavní stránce aplikace jsem přidal tlačítka s ikonou otazníku vedle tlačítka "Zobrazit licence". Po kliknutí na tento otazník se zobrazí okno s vysvětlením, že tlačítka zobrazuje seznam všech licencí, včetně jejich stavu a expirace, jak je znázorněno na obrázku 8.1.



Obrázek 8.1: Vysvětlení funkce tlačítka "Zobrazit licence"

### 8.1.2 Návrat k zobrazení všech zařízení

V comboboxu pro filtrování zařízení jsem přidal možnost "Všechny statusy", která umožňuje uživatelům rychlý návrat k zobrazení všech zařízení bez nutnosti opuštění aktuálního filtru nebo návratu na hlavní stránku. Tato možnost je umístěna na viditelném místě v horní části stránky se seznamem zařízení.

### 8.1.3 Zvětšení pole pro poznámky při přidávání zařízení

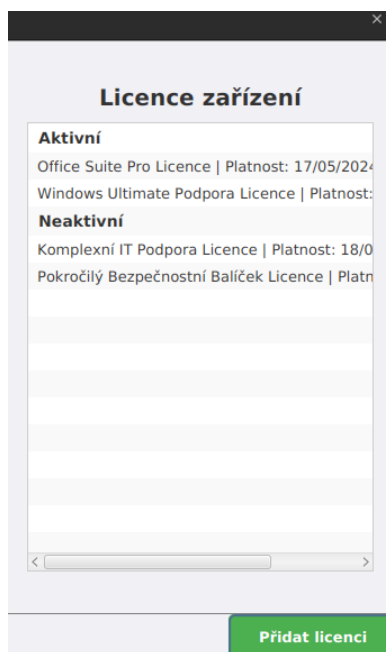
Při přidávání nového zařízení jsem zvětšil pole pro poznámky, aby byla lepší viditelnost a uživatelé měli více prostoru pro záznam detailů.

### 8.1.4 Oprava chování ChoiceBoxu při editaci zařízení

Opravil jsem chování ChoiceBoxu při editaci zařízení. Po uložení změn nyní ChoiceBox správně reflektuje aktuální stav a nezůstává v režimu úprav.

### 8.1.5 Změna správy licencí

Implementoval jsem funkci, která umožňuje licencím zůstat uloženým v systému, ale nezobrazovat se jako aktivní. To znamená, že neaktivní licence se již nezobrazují mezi aktivními, což umožňuje uchovat historii licencí bez jejich odstranění, jak je znázorněno na obrázku 8.2.



Obrázek 8.2: Zobrazení aktivních a neaktivních licencí

## 8.2 Shrnutí a uzavření aktuálního stavu

Aplikace umožňuje uživatelům evidovat firmy a přidružené události, jako jsou různé typy akcí a poznámky. Uživatelé mohou zadávat detaily o firmách, včetně kontaktních informací a dalších relevantních údajů. Ke každé firmě mohou být přidávána zařízení, jako jsou počítače, servery, tiskárny atd., a k těmto zařízením mohou být přidávány a spravovány licence. Aplikace umožňuje vytvářet, upravovat a odstraňovat licence, stejně jako zobrazovat a filtrovat seznamy licencí pro jednotlivé firmy. Veškeré operace běží lokálně na počítači správce a jsou chráněny čtyřmístným PIN kódem pro zajištění bezpečnosti. Aplikace podporuje správu firem a zařízení s možností zaznamenávání poznámek k jednotlivým entitám, což zajišťuje komplexní evidenci a správu informací.

## 8.3 Budoucí rozvoj

V rámci budoucího rozvoje této aplikace vidím několik klíčových oblastí, které by mohly přinést významná zlepšení. Jedním z hlavních kroků by byl přechod na framework Spring Boot a použití serverové databáze. Tento přechod by umožnil lepší škálovatelnost a možnost nasazení aplikace na cloudové platformy, což by usnadnilo online monitoring a správu zařízení.

Dalším zlepšením by mohlo být zavedení knihovny Lombok, která by zjednodušila práci s modelovými třídami tím, že automaticky generuje metody jako gettery, settery a toString. Použití Lomboku by nejen snížilo množství kódu, ale také zvýšilo jeho čitelnost a udržitelnost.

Kód aplikace by mohl být dále vylepšen refaktorováním a sjednocením validačních pravidel. Zavedení jednotných validačních pravidel a zvýšení pokrytí kódu testy by vedlo ke zlepšení kvality a spolehlivosti aplikace. V tomto směru vidím potenciál pro zvýšení efektivity vývoje a snížení chybovosti.

### 8.3.1 Funkční rozšíření a přechod na klient-server architekturu

Další klíčovou oblastí pro budoucí rozvoj je funkční rozšíření aplikace. Přechod na klient-server architekturu by mohl přinést několik výhod, ale je důležité zvážit, zda to má pro konkrétní uživatele aplikace smysl.

Přechod na klient-server architekturu by umožnil:

- **Centralizovanou správu dat:** Uživatelé by měli přístup k datům odkudkoli, což by zlepšilo spolupráci a flexibilitu.
- **Lepší bezpečnost:** Centralizované zabezpečení může být snazší spravovat a aktualizovat.
- **Rozšíření funkcionality:** Možnost integrace s dalšími systémy a službami, jako jsou API pro získávání externích dat nebo zasílání notifikací.

Tyto návrhy představují mé pohledy na možný budoucí rozvoj aplikace a poskytují směry, kterými by se další práce mohla ubírat.



## Kapitola 9

### Závěr

Tato bakalářská práce se zaměřila na návrh a implementaci aplikace pro externí správu počítačových sítí, speciálně navržené pro potřeby externích správců, kteří se věnují evidenci a správě IT infrastruktury malých firem. Práce byla zahájena na základě pečlivé analýzy a definice klíčových požadavků, které byly transformovány do komplexního doménového modelu a následně do databázové struktury, zajišťující správu dat.

Vývoj aplikace byl realizován pomocí programovacího jazyka Java s využitím frameworku JavaFX pro frontend, což umožnilo vytvoření intuitivního a uživatelsky přívětivého rozhraní. Vývoj a správa aplikace byly podpořeny nástroji jako IntelliJ IDEA a DB Browser for SQLite, které umožnily rychlé identifikování a opravu chyb během testování.

Na základě zpětné vazby od testerů jsem provedl několik úprav a vylepšení aplikace. Tato zpětná vazba poukázala na oblasti, které vyžadovaly zlepšení, a tyto návrhy jsem následně implementoval.

Vzhledem k těmto úpravám a analýze potřeb trhu se nabízí řada možností pro budoucí rozvoj aplikace. Mezi předpokládané inovace patří například přechod na framework Spring Boot, použití serverové databáze a zavedení pokročilých validačních pravidel.

Celkově projekt ukázal, že navržené řešení může významně přispět k zlepšení správy IT infrastruktury v malých podnicích. Implementovaná aplikace nabízí solidní základ pro další rozvoj a adaptaci na změny v technologickém prostředí, což přináší externím správcům efektivní nástroj pro správu IT prostředí jejich klientů. Cíle práce uvedené v kapitole 1.1 byly splněny.





## Bibliografie

1. TERRA, John. *What Is a Computer Network?* [<https://www.simplilearn.com/what-is-computer-network-article>]. 2023. [cit. 2023-11-12]. [online].
2. AKSHARA<sub>A</sub>. *Difference between LAN and WLAN* [<https://www.geeksforgeeks.org/difference-between-lan-and-wlan/>]. 2023. [cit. 2023-12-03].
3. COMPTIA. *What Is WAN? How Wide Area Networks Function* [<https://www.comptia.org/content/guides/what-is-a-wide-area-network>]. 2023. [cit. 2023-12-10].
4. ROUSE, Margaret. *What Does Network Administrator Mean?* [<https://www.techopedia.com/definition/8548/network-administrator>]. 2023. [cit. 2023-11-12].
5. MICHIGANTECH. *What is Computer Network and System Administration?* [<https://www.mtu.edu/applied-computing/undergraduate/cnsa/what/>]. 2023. [cit. 2023-11-12].
6. CHICK, Douglas. *What All Network Administrators Know*. The Network Administrator. Com, 2003.
7. CHOUFFANI, Reda. *What is IT asset management (ITAM)?* [<https://www.techtarget.com/searchcio/definition/IT-asset-management-information-technology-asset-management>]. 2023. [cit. 2023-12-04].
8. MICROSOFT. *Porovnání řešení Microsoft 365 a Office* [<https://www.microsoft.com/cs-cz/microsoft-365/buy/compare-all-microsoft-365-products>]. 2024. [cit. 2024-05-05].
9. NRB. *Definice MSP* [<https://www.nrb.cz/podnikatele/dalsi-informace-pro-podnikatele/mali-a-stredni-podnikatele/>]. 2024. [cit. 2024-05-05].
10. CALLIE MALVIK. *What Does a Network Administrator Do? A Behind-the-Scenes Look* [<https://www.rasmussen.edu/degrees/technology/blog/what-does-a-network-administrator-do/>]. 2024. [cit. 2024-03-05].
11. PEARSON. *An Ideal Network Administrator* [<https://www.pearsonitcertification.com/articles/article.aspx?p=1646444>]. 2024. [cit. 2024-10-05].

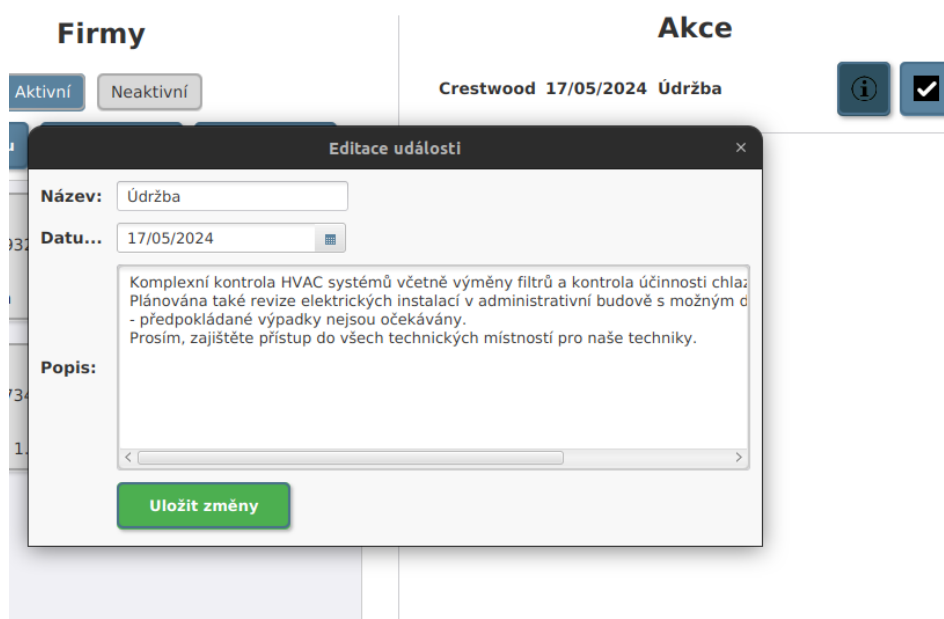
12. IS, MUNI. *Požadavky a Use Case Diagramy* [[https://is.muni.cz/el/1433/podzim2017/PA116/um/prezentace\\_z\\_prednasek/DUM\\_04\\_UC.pdf](https://is.muni.cz/el/1433/podzim2017/PA116/um/prezentace_z_prednasek/DUM_04_UC.pdf)]. 2024. [cit. 2024-08-05].
13. DAVID HARTINGER. *Lekce 4 - UML - Doménový model* [<https://www.itnetwork.cz/navrh/uml/uml-domenovy-model-diagram>]. 2024. [cit. 2024-08-05].
14. TECH ON THE NET. *SQLite: System Tables* [[https://www.techonthenet.com/sqlite/sys\\_tables/index.php](https://www.techonthenet.com/sqlite/sys_tables/index.php)]. 2024. [cit. 2024-08-05].
15. B6B36SMP, Sběr a modelování požadavků. *Přednáška 4 a 5 - Případy užití* [<https://moodle.fel.cvut.cz/local/kos/pages/course/info.php?id=3878>]. 2024. [cit. 2024-08-05].
16. SPRING BOOT. *Spring Boot Reference Guide* [<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>]. 2024. [cit. 2024-08-05].
17. ORACLE. *JavaFX 8* [<https://docs.oracle.com/javase/8/javafx/api/toc.htm>]. 2024. [cit. 2024-06-05].
18. DOCUMENTATION. *SQLite* [<https://www.sqlite.org/docs.html>]. 2024. [cit. 2024-06-05].
19. FIGMA. *Figma design-tools* [<https://help.figma.com/hc/en-us/categories/360002042553-Figma-design>]. 2024. [cit. 2024-08-05].
20. PETR ŠTECHMÜLLER. *Lekce 1 - IntelliJ IDEA - Úvod do vývojového prostředí* [<https://www.itnetwork.cz/java/intelij-idea-netbeans-eclipse-pokrocila-prace/intelij-idea-uvod-do-vyvojoveho-prostredi>]. 2024. [cit. 2024-09-05].
21. TUTORIALSPPOINT. *JavaFX - Environment* [[https://www.tutorialspoint.com/javafx/javafx\\_environment.htm](https://www.tutorialspoint.com/javafx/javafx_environment.htm)]. 2024. [cit. 2024-09-05].
22. JAVA. *Java 8 Information*) [<https://www.java.com/en/download/help/java8.html>]. 2024. [cit. 2024-09-05].
23. JAVA. *Kinsta* [<https://kinsta.com/knowledgebase/what-is-github/1>]. 2024. [cit. 2024-09-05].
24. SQ LITE. *DB Browser for SQLite* [<https://sqlitebrowser.org/>]. 2024. [cit. 2024-09-05].
25. EXPERIENCE UX. *What is usability testing?* [<https://www.experienceux.co.uk/faqs/what-is-usability-testing/>]. 2024. [cit. 2024-10-05].



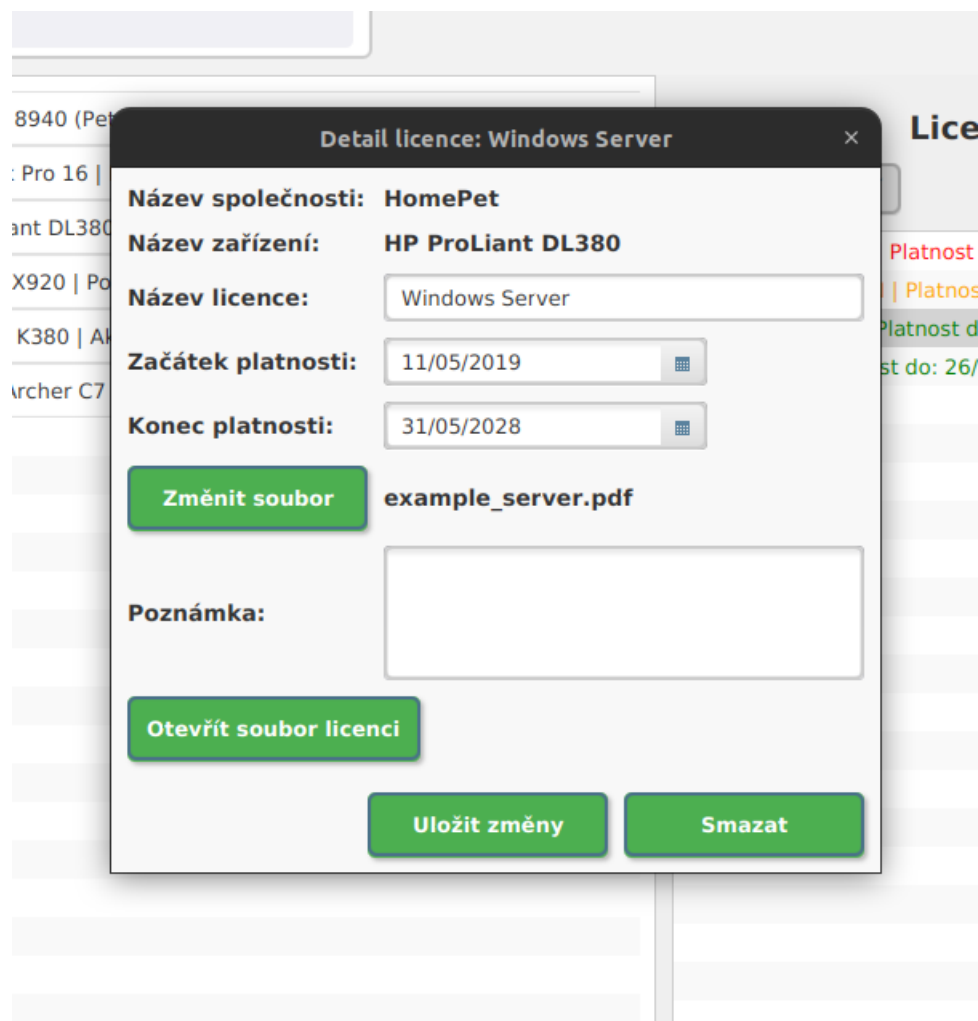
# Příloha A

## Přílohy

### A.1 Obrazovky aplikace



Obrázek A.1: Obrazovka detailu akce



Obrázek A.2: Obrazovka detailu licence

**Firmy** **Akce**

**Přidat novou firmu** ×

**Název firmy: \***

**Kontakt: \***

**Další kontakty:**

**Detaily firmy:**

**Ulice: \***

**Město: \***

**PSČ: \***

**Země: \***

**Aktivní:**  **Aktivní**

**Přidat**

Obrázek A.3: Okno přidání firmy

Pro 16 | Neaktivní | OS: macOS | Seriové číslo: A1B2C3D4E!

Vše Expirující

ant DL380 | Ak

IX920 | Poško

K380 | Aktiv

Archer C7 | Ak

se | Platnost

VPN | Platno:

er | Platnost c

tnost do: 26,

**Přidat zařízení**

**Název:** Database Server

**Status:** Neaktivní

**Typ:** Počítač

**Sériové číslo:** Sériové číslo

**Operační systém:** Vyberte OS

**CPU:** CPU

**RAM:** RAM GB

**Úložiště:** Úložiště GB

**Poznámky:** Poznámky

**Přidat**

Obrázek A.4: Okno přidání zařízení