



FAKULTA APLIKOVANÝCH VĚD  
ZÁPADOČESKÉ UNIVERZITY  
V PLZNI

KATEDRA INFORMATIKY  
A VÝPOČETNÍ TECHNIKY



## Diplomová práce

# System pro správu pracovních stanic v prostředí KIV

Ondřej Drtina







FAKULTA APLIKOVANÝCH VĚD  
ZÁPADOČESKÉ UNIVERZITY  
V PLZNI

KATEDRA INFORMATIKY  
A VÝPOČETNÍ TECHNIKY

## **Diplomová práce**

# **System pro správu pracovních stanic v prostředí KIV**

Bc. Ondřej Drtina

**Vedoucí práce**

Ing. Ladislav Pešička

© Ondřej Drtina, 2023.

Všechna práva vyhrazena. Žádná část tohoto dokumentu nesmí být reprodukována ani rozšiřována jakoukoli formou, elektronicky či mechanicky, fotokopírováním, nahráváním nebo jiným způsobem, nebo uložena v systému pro ukládání a vyhledávání informací bez písemného souhlasu držitelů autorských práv.

**Citace v seznamu literatury:**

DRTINA, Ondřej. *Systém pro správu pracovních stanic v prostředí KIV*. Plzeň, 2023. Diplomová práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra informatiky a výpočetní techniky. Vedoucí práce Ing. Ladislav Pešička.

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd  
Akademický rok: 2022/2023

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Ondřej DRTINA**  
Osobní číslo: **A20N0077P**  
Studijní program: **N3902 Inženýrská informatika**  
Studijní obor: **Softwarové inženýrství**  
Téma práce: **Systém pro správu pracovních stanic v prostředí KIV**  
Zadávající katedra: **Katedra informatiky a výpočetní techniky**

## Zásady pro vypracování

1. Prozkoumejte možnosti vzdálené správy pracovních stanic, zejména s přihlédnutím k potřebám pracoviště KIV.
2. Vyberte vhodné technologie pro potřeby vzdálené správy stanic.
3. Navrhněte systém pro vzdálenou správu stanic na pracovišti KIV. Systém bude možné spravovat z webového prohlížeče a dále i z mobilního zařízení s OS Android.
4. Navržený systém realizujte, ověřte jeho funkčnost a navrhněte další vhodná rozšíření.

Rozsah diplomové práce: **doporuč. 50 s. původního textu**  
Rozsah grafických prací: **dle potřeby**  
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

dodá vedoucí diplomové práce

Vedoucí diplomové práce: **Ing. Ladislav Pešička**  
Katedra informatiky a výpočetní techniky

Datum zadání diplomové práce: **9. září 2022**  
Termín odevzdání diplomové práce: **18. května 2023**

L.S.

---

**Doc. Ing. Miloš Železný, Ph.D.**  
děkan

---

**Doc. Ing. Přemysl Brada, MSc., Ph.D.**  
vedoucí katedry

V Plzni dne 11. října 2022

# Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného akademického titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Západočeská univerzita v Plzni má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V Plzni dne 18. května 2023

.....

Ondřej Drtina

V textu jsou použity názvy produktů, technologií, služeb, aplikací, společností apod., které mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

## Abstrakt

Předmětem této diplomové práce je tvorba řešení pro správu strojů v prostředí ZČU KIV. Hlavním cílem práce bylo vytvořit nové řešení pro správu strojů, jež by bylo vhodné pro nasazení v univerzitním prostředí. V úvodu práce jsou objasněny základní termíny, jejichž znalost je nezbytná pro pochopení následujících částí této práce. Následuje analýza několika vybraných řešení pro správu strojů. Dále práce obsahuje kapitolu popisující funkcionality, jež budou v novém řešení implementovány. Následuje popis návrhu nového systému, na který navazuje text popisující implementaci navrženého webového řešení a klienta pro platformu Android. Práce rovněž obsahuje popis testovacích scénářů a automatických testů, jež byly použity k ověření funkcionality implementovaného řešení. Práce dále pojednává o rozšířeních, která by v budoucnu mohla vylepšit funkcionality vytvořené webové aplikace, resp. mobilního klienta.

## Abstract

The subject of this master thesis is creating a software solution for managing workstations in the DCSE environment. The main objective of this thesis was to create a new solution for workstation management that would be suitable for deployment in a university environment. The work's introduction explains the basic terms necessary for understanding the following parts of this thesis. An analysis of several selected workstation management solutions follows. Furthermore, the thesis contains a chapter describing functionalities that will be implemented in the new solution. The thesis continues with a description of the new system's design, followed by a description of the implementation of the designed web solution and client for the Android platform. The work also describes test scenarios and automated tests, which were used to verify the implemented solution's functionality. Additionally, the thesis discusses extensions that could be later implemented to improve the functionality of the created web application and mobile client.

## Klíčová slova

správa strojů • SSH • Spring Boot • Android • Docker • Kerberos • vzdálená správa



## Poděkování

Rád bych tímto poděkoval panu Ing. Ladislavu Pešíčkovi za vedení diplomové práce a za odborné připomínky, které vedly ke zlepšení kvality této práce.



# Obsah

<b>1 Úvod</b>	<b>7</b>
<b>2 Technologie</b>	<b>9</b>
2.1 ISO/OSI model . . . . .	9
2.1.1 Fyzická vrstva . . . . .	9
2.1.2 Linková vrstva . . . . .	10
2.1.3 Síťová vrstva . . . . .	11
2.2 MAC adresa . . . . .	11
2.2.1 Změna MAC adresy . . . . .	11
2.2.2 Složky identifikátoru . . . . .	12
2.3 IP adresa . . . . .	13
2.3.1 Verze IP protokolu . . . . .	13
2.3.2 Veřejná vs. privátní IP adresa . . . . .	14
2.4 Broadcast IP adresa . . . . .	14
2.5 SSH . . . . .	15
2.5.1 Autentizace . . . . .	15
2.5.2 Požadovaná softwarová konfigurace . . . . .	16
2.6 Powershell . . . . .	17
2.6.1 Základní informace . . . . .	17
2.6.2 Přístup k systémovým komponentám . . . . .	18
2.6.3 Využití v práci . . . . .	19
2.7 Chocolatey . . . . .	19
2.7.1 Základní informace . . . . .	19
2.7.2 Alternativní nástroje . . . . .	20
2.7.3 Využití v práci . . . . .	21
2.8 SSO . . . . .	21
2.8.1 Protokol Kerberos . . . . .	22
2.9 Wake-on-LAN . . . . .	24
2.9.1 Struktura rámce „Magic Packet“ . . . . .	26
2.9.2 Zapnutí zařízení umístěných mimo segment . . . . .	26

---

2.10	TCO . . . . .	26
2.10.1	Oblasti využití . . . . .	27
2.10.2	Zahrnuté položky . . . . .	27
<b>3</b>	<b>Průzkum existujících řešení</b>	<b>29</b>
3.1	Hodnocené vlastnosti aplikací . . . . .	29
3.1.1	Zapnutí stroje . . . . .	29
3.1.2	Zjištění dostupnosti . . . . .	30
3.1.3	Informace o stroji . . . . .	30
3.1.4	Aktualizace operačního systému . . . . .	30
3.1.5	Multiplatformní server . . . . .	30
3.1.6	Uživatelské role . . . . .	30
3.1.7	Správa multiplatformních stanic . . . . .	31
3.1.8	Spuštění skriptů . . . . .	31
3.1.9	Plán obsazení stroje . . . . .	31
3.1.10	Autentizace externím zdrojem . . . . .	31
3.1.11	Cena . . . . .	32
3.2	Výběr aplikací pro analýzu . . . . .	32
3.2.1	Skóre webu G2 . . . . .	32
3.2.2	Hodnocení uživatelů . . . . .	33
3.3	NinjaOne . . . . .	34
3.3.1	Hodnocená kritéria . . . . .	34
3.4	Atera . . . . .	36
3.4.1	Hodnocená kritéria . . . . .	37
3.5	LogMeIn Central . . . . .	38
3.5.1	Hodnocená kritéria . . . . .	39
3.6	Zhodnocení funkcionality stávajících řešení . . . . .	41
3.7	Chybějící funkcionality existujících řešení . . . . .	41
<b>4</b>	<b>Výběr vhodné funkcionality</b>	<b>43</b>
4.1	Funkce plánovaného řešení . . . . .	43
4.2	Mobilní aplikace . . . . .	46
<b>5</b>	<b>Návrh webového řešení</b>	<b>47</b>
5.1	Přenositelnost řešení . . . . .	47
5.1.1	Software vyžadovaný řešením . . . . .	47
5.1.2	Docker . . . . .	47
5.2	Ukládaná data . . . . .	49
5.2.1	Počítačové stanice . . . . .	49
5.2.2	Uživatelé . . . . .	49

---

5.2.3	Forma uložení dat . . . . .	50
5.3	Technologie . . . . .	50
5.4	Identifikace uživatelů . . . . .	52
<b>6</b>	<b>Návrh mobilního klienta</b>	<b>53</b>
6.1	Volba platformy . . . . .	53
6.2	Knihovna Retrofit . . . . .	54
6.3	Obrázky třetích stran . . . . .	55
6.4	Uložení dat . . . . .	55
<b>7</b>	<b>Implementace navrženého řešení</b>	<b>57</b>
7.1	Webové řešení . . . . .	58
7.1.1	Využitý framework . . . . .	58
7.1.2	Architektura řešení . . . . .	58
7.1.3	API . . . . .	62
7.1.4	Vykonávání skriptů . . . . .	62
7.1.5	Persistence dat . . . . .	64
7.1.6	Systémové požadavky . . . . .	65
7.2	Mobilní aplikace . . . . .	67
7.2.1	Dělení tříd programu . . . . .	67
7.2.2	Persistence dat . . . . .	69
7.2.3	Systémové požadavky . . . . .	70
<b>8</b>	<b>Ověření funkčnosti produktu</b>	<b>71</b>
8.1	Testovací zařízení . . . . .	71
8.1.1	Webová aplikace . . . . .	71
8.1.2	Mobilní klient . . . . .	72
8.1.3	Server aplikace . . . . .	73
8.2	Testovací scénáře - práce s uživateli . . . . .	73
8.2.1	Přihlášení běžného uživatele . . . . .	73
8.2.2	Přihlášení administrátora . . . . .	75
8.2.3	Podání žádosti o registraci uživatele . . . . .	76
8.2.4	Schválení žádosti o registraci uživatele . . . . .	77
8.2.5	Přidání běžného uživatele administrátorem . . . . .	78
8.2.6	Změna role . . . . .	79
8.3	Testovací scénáře - evidence strojů . . . . .	79
8.3.1	Vytvoření seznamu zařízení . . . . .	79
8.3.2	Přidání zařízení administrátorem . . . . .	80
8.3.3	Podání žádosti o přidání zařízení . . . . .	81
8.3.4	Editace údajů zařízení . . . . .	82

8.3.5	Odstranění zařízení . . . . .	83
8.3.6	Obnovení seznamu strojů . . . . .	83
8.4	Testovací scénáře - ovládání strojů . . . . .	84
8.4.1	Zapnutí zařízení . . . . .	84
8.4.2	Aktualizování systému Windows . . . . .	85
8.4.3	Aktualizování Chocolatey balíčků . . . . .	88
8.4.4	Získání seznamu lokálních uživatelů . . . . .	89
8.4.5	Získání informací o stroji . . . . .	91
8.4.6	Vypnutí zařízení . . . . .	92
8.5	Automatické testy . . . . .	92
<b>9</b>	<b>Možná rozšíření produktu</b>	<b>95</b>
9.1	Získávání rozvrhových dat . . . . .	95
9.2	Logování událostí . . . . .	96
9.3	Implementace IPv6 adres . . . . .	96
9.4	Rozšíření správy Linux . . . . .	96
9.5	Rozšíření Android klienta . . . . .	97
9.6	Spouštění uživatelských skriptů . . . . .	97
9.7	Vytvoření iOS klienta . . . . .	97
<b>10</b>	<b>Závěr</b>	<b>99</b>
<b>A</b>	<b>Instalační příručka webové aplikace</b>	<b>101</b>
A.1	Nahrání SSH klíče na spravovaný stroj . . . . .	101
A.2	Instalace PSWindowsUpdate na spravovaný stroj . . . . .	102
A.3	Instalace Chocolatey na spravovaný stroj . . . . .	102
A.4	Spuštění serveru pomocí Docker . . . . .	103
<b>B</b>	<b>Instalační příručka mobilní aplikace</b>	<b>105</b>
<b>C</b>	<b>Uživatelská příručka webové aplikace</b>	<b>107</b>
C.1	Evidence strojů . . . . .	107
C.1.1	Vytvoření seznamu strojů . . . . .	107
C.1.2	Přidání stroje do seznamu . . . . .	108
C.1.3	Export seznamu strojů do PDF / CSV . . . . .	109
C.1.4	Import seznamu strojů z CSV . . . . .	109
C.1.5	Rozhodnutí žádosti o registraci stroje . . . . .	110
C.2	Ovládání strojů . . . . .	111
C.2.1	Zapnutí zařízení . . . . .	111
C.2.2	Zjištění dostupnosti stroje . . . . .	112
C.2.3	Správa aktualizací Windows . . . . .	112

---

C.2.4	Správa balíčků Chocolatey . . . . .	112
C.2.5	Správa uživatelů Windows . . . . .	113
C.2.6	Vypnutí / restart zařízení . . . . .	115
C.3	Správa uživatelů . . . . .	115
C.3.1	Přihlášení uživatele . . . . .	115
C.3.2	Podání žádosti o registraci . . . . .	116
C.3.3	Rozhodnutí žádosti o registraci uživatele . . . . .	116
<b>D</b>	<b>Uživatelská příručka mobilní aplikace</b>	<b>119</b>
D.1	Služby poskytované aplikačním serverem . . . . .	119
D.1.1	Přihlášení uživatele . . . . .	119
D.1.2	Zapnutí zařízení . . . . .	120
D.1.3	Správa aktualizací Windows . . . . .	121
D.1.4	Správa balíčků Chocolatey . . . . .	122
D.1.5	Správa uživatelů Windows . . . . .	124
D.1.6	Zobrazení informací o stroji . . . . .	125
D.1.7	Vypnutí / restart zařízení . . . . .	126
D.2	Služby lokální sítě . . . . .	126
D.2.1	Vytvoření seznamu zařízení . . . . .	126
D.2.2	Přidání zařízení do seznamu . . . . .	126
D.2.3	Zapnutí zařízení . . . . .	128
<b>E</b>	<b>Obsah příloženého archivu</b>	<b>129</b>
	<b>Bibliografie</b>	<b>131</b>
	<b>Seznam zkratk</b>	<b>139</b>
	<b>Seznam obrázků</b>	<b>143</b>
	<b>Seznam tabulek</b>	<b>145</b>
	<b>Seznam výpisů</b>	<b>147</b>





Mnoho organizací využívá počítače, jejichž softwarové vybavení je žádoucí pravidelně aktualizovat. Důvodů pro udržování software v aktuální verzi je mnoho. Programy mohou prostřednictvím aktualizace získat novou funkcionalitu, vylepšit uživatelské rozhraní a podobně. Zřejmě nejdůležitějším faktorem, jenž by měl administrátory počítačových systémů přimět k aplikování aktualizací software, je možná oprava bezpečnostních problémů. Pokud však instituce vlastní velké množství strojů, může být obtížné vykonávat činnosti spojené s aktualizacím procesem ručně. Proces aktualizace je navíc mnohdy časově náročný a je žádoucí vykonávat ho v době, kdy počítač není aktivně využíván uživatelem.

Za účelem usnadnění údržby a obsluhy strojů v institucích, jež vlastní velké množství pracovních stanic, byly vyvinuty aplikace umožňující vzdálenou správu počítačů. Systémy pro vzdálenou správu strojů existují již delší dobu a často jsou aplikována IT oddělením ve firmách, kde usnadňují nezbytnou softwarovou údržbu počítačů. Software usnadňující správu strojů je žádoucí využít i na univerzitách. Existující řešení pro vzdálenou správu určené pro firmy však nemusí být vhodné nasadit v univerzitním prostředí, ať již z hlediska ceny řešení či nemožnosti jejich interakce s dalšími systémy univerzity.

Jednou z typických funkcí, kterou často poskytují nástroje pro správu strojů, je vzdálené zapnutí počítače. Uvedená funkcionalita může vést ke snížení spotřeby elektrické energie v organizacích, kde zaměstnanci využívají vzdálený přístup k počítačům. Bez možnosti vzdáleného zapnutí stanice se pravděpodobně mnoho zaměstnanců rozhodne nechat stroj zapnutý permanentně pro případ, že by k němu potřebovali přistoupit. Při zavedení možnosti vzdáleného zapnutí pracovní stanice by zaměstnanci mohli počítač zapínat pouze v případě, že jej skutečně potřebují použít, a po vykonání potřebných operací by ho mohli opět vypnout.

Některé z produktů vzdálené správy nabízejí možnost získat informace o aktuální hardwarové konfiguraci strojů a mohou tedy sloužit pro účely inventarizace v rámci institucí. Například pokud má organizace k dispozici seznam komponent, jež mají firemní počítače obsahovat, může být pomocí software pro vzdálenou správu vyhodnocena kompletnost stanic. Uvedenou funkcionalitu lze tedy snadno

využít pro detekci krádeže či selhání části hardwarové výbavy počítače.

V neposlední řadě mohou být některé systémy vzdáleného řízení použity pro získání informací o uživatelích počítačů. Mezi žádané informace může patřit jméno právě přihlášeného uživatele, které může být potřebné získat například v případě, že stanice vykonává škodlivou či nelegální činnost.

Výhody řešení pro vzdálenou správu jsou zjevné, avšak nasazení systémů pro vzdálenou správu v univerzitním prostředí může provázet řada problémů. Komplikace může představovat zejména cena licence, kterou je často nutno uhradit ve spojitosti s užíváním produktu pro správu. Většina systémů pro vzdálenou správu navíc vyžaduje periodickou platbu, jejíž výše se mnohdy odvíjí od počtu spravovaných zařízení. Dalším potenciální problémem může být platformní závislost produktů pro správu či jejich neschopnost interagovat s dalšími univerzitními systémy.

Na základě uvedených poznatků jsem se rozhodl vytvořit vlastní software umožňující správu strojů, jenž bude určen zejména pro využití v univerzitním prostředí. Cílem práce je vytvoření komplexního nástroje pro vzdálenou správu, jenž bude dostupný zdarma a bude snadno nasaditelný v prostředí ZČU. V rámci projektu bude vytvořen aplikační server, který umožní provádění většiny z výše uvedených funkcionalit, jež poskytují stávající řešení pro správu. Součástí řešení bude podpůrná webová aplikace, prostřednictvím které bude uživateli umožněno spravovat stroje pomocí přehledného grafického rozhraní. Jelikož nové řešení by mělo být využíváno primárně v prostředí ZČU, budou se uživatelé řešení pro správu prokazovat přihlašovacími údaji, jež používají v rámci univerzity (Orion login).

Součástí plánovaného řešení pro správu počítačů bude i mobilní klient pro operační systém Android. Komunikace mobilního klienta a aplikačního serveru bude řešena prostřednictvím programového rozhraní, což zajistí snížení nároků na internetové připojení oproti využití webové aplikace. Program pro OS Android bude kromě služeb vyžadující kontaktování aplikačního serveru poskytovat i funkcionality, jež umožní zapnout počítače v rámci lokální sítě.

Předmětem následujících podkapitol je stručný popis technologií, na které je v práci odkazováno a jejichž znalost je pro pochopení práce nezbytná.

## 2.1 ISO/OSI model

Jedná se o koncepční model, který byl vytvořen v 70. letech pro účely demonstrace možného propojení počítačových systémů s odlišnou architekturou. Model vyvinula organizace ISO v rámci její snahy o standardizaci komunikace v počítačových sítích. Název modelu je zkratkou anglického spojení „Open System Interconnection“, jež lze volně přeložit jako „propojování otevřených systémů“ [For00].

ISO/OSI model dělí proces komunikace mezi stroji do sedmi základních úrovní [23w], jež jsou v modelu označovány jako „vrstvy“. Každá komunikační vrstva modelu si může vyměňovat informace pouze s vrstvou, která se nachází bezprostředně pod či nad danou vrstvou. Všechny vrstvy modelu, vyjma krajních vrstev, tedy mohou komunikovat pouze se dvěma sousedními vrstvami [For00]. Krajní vrstvy modelu komunikují se zařízeními, které ISO/OSI model implementují.

Každá z vrstev má svou jasně definovanou funkci a způsob reprezentace dat, se kterými pracuje. Pro označení jednotky dat vrstvy se často používá anglický výraz „Protocol Data Unit“ či jeho zkratka „PDU“. Tabulka 2.1 zachycuje názvy jednotlivých vrstev modelu a označení jednotek dat, se kterými vrstvy pracují. V uvedené tabulce jsou u každé vrstvy také uvedeny příklady protokolů, jež na dané vrstvě pracují. Většina technologií, které jsou v rámci této práce zmiňovány, pracuje na vrstvách L1 až L3. Bližší popis prvních tří vrstev ISO / OSI modelu je předmětem kapitol 2.1.1 - 2.1.3. Popis ostatních vrstev ISO / OSI modelu poskytuje například web, jež vytvořil Roman Pramberger [Pra23].

### 2.1.1 Fyzická vrstva

První vrstva ISO / OSI modelu se stará o přenos digitálních signálů (bitů) po fyzickém médiu mezi odesílatelem a příjemcem dat. Fyzickým médiem přenosu mohou

vrstva	název vrstvy	jednotka dat	protokoly
L7	aplikační	data	HTTP, NFS, SSH
L6	prezentační	data	AFP, XDR, ICA
L5	relační	data	NetBIOS, RPC, SCP
L4	transportní	segment	TCP, UDP, RDP
L3	síťová	paket	IPv4, IPv6, ICMP
L2	linková	rámec	PPP, ATM, Ethernet
L1	fyzická	bit	Ethernet

Tabulka 2.1: Vrstvy ISO/OSI modelu

být například optická vlákna, koaxiální kabely či bezdrátové vysílání. Fyzická vrstva se zabývá technickými parametry komunikačního média, například rychlostí přenosu, metodami modulace přenosu a tvarem konektoru.

Na této úrovni ISO / OSI se rovněž udává způsob reprezentace obou možných hodnot bitů (0 a 1) [For00]. Definování způsobu reprezentace bitu zahrnuje například určení úrovně napětí, která odpovídá dané hodnotě bitu, a dobu trvání, po kterou bude tato hodnota napětí udržena.

## 2.1.2 Linková vrstva

Daná vrstva modelu zajišťuje sdružování bitů do logických celků, jež jsou označovány jako „rámce“, a rovněž řeší přenos těchto bloků dat mezi dvěma přímo propojenými stanicemi. Pokud je mezi odesílatelem a příjemcem dat mezilehlý prvek, je přenos řešen na vyšších vrstvách OSI modelu. Jednou z úloh linkové vrstvy je zajištění detekce začátku a konce rámce. Linková vrstva dále pomocí kontrolních součtů zajišťuje detekci chyb vzniklých na fyzické úrovni, případně rovnou aplikuje algoritmy na opravu zjištěných chyb [For00].

Linková vrstva se v kontextu počítačových sítí dělí na dvě subvrstvy s označením „Media Access Control (MAC)“ a „Logical Link Control (LLC)“ [Pri21]. První z jmenovaných podvrstev (MAC) je zodpovědná za řízení přístupu k fyzickému síťovému médiu a zaručuje, že zařízení účastníci se přenosu mají unikátní MAC adresu. Bližší popis MAC adresy je předmětem kapitoly 2.2.

Cílem subvrstvy LLC linkové vrstvy je zajištění spolehlivého přenosu dat mezi dvěma propojenými zařízeními v jedné síti. Subvrstva LLC například zajišťuje provádění kontrolních součtů nad přenášenými daty a opakování přenosu v případě, že je detekována ztráta paketů.

## 2.1.3 Síťová vrstva

Primární úlohou síťové vrstvy modelu je zajištění směrování datových bloků v rámci sítě, jež jsou na úrovni síťové vrstvy označovány jako „pakety“. Síťová vrstva pro optimální zvolení trasy paketů potřebuje znát umístění jednotlivých uzlů sítě, tzv. „topologii sítě“.

Ke směrování dat je možno využít dva základní přístupy, jež se liší dle schopnosti reagovat na změny v síti. Změnu v oblasti síťových zařízení představuje například přidání nového stroje do sítě či odebrání zařízení ze sítě. Existuje neadaptivní (statické) směrování, které se nikterak nesnaží přizpůsobit změnám v síti. Hlavní výhodou neadaptivního směrování je absence nutnosti průběžné detekce lokace uzlů v síti a s tím spojené aktualizace směrovací tabulky [23f].

Pro směrování paketů v síti je možno využít i adaptivní (dynamické) směrování, které se snaží reagovat na nastalé události v síti. Oproti neadaptivnímu přístupu vyžaduje dynamické směrování průběžnou aktualizaci informací o lokaci a stavu jednotlivých uzlů. Získané informace o uzlech jsou využity pro aktualizaci směrovací tabulky, která slouží pro hledání optimální cesty mezi uzly. Průběžná aktualizace tabulek vyžaduje využití protokolů, například: OSPF, RIP, BGP. Výhodou adaptivního směrování oproti statickému směrování může být nalezení více optimální trasy dat mezi uzly. Nevýhodou dynamického směrování je zejména vysoká režie, která souvisí s nutností průběžně aktualizovat směrovací tabulky [Pet23].

## 2.2 MAC adresa

MAC je zkratka anglického výrazu „Media Access Control“ a představuje jednoznačný identifikátor síťových zařízení, jež pracují na druhé vrstvě modelu ISO/OSI (viz kapitola 2.1). Tento identifikátor zařízení se skládá ze 48 bitů, přičemž standard udává, že MAC adresy mají být zapisovány v hexadecimální soustavě. Běžně je uvedený identifikátor reprezentován jako 12 hodnot, jež jsou od sebe odděleny pomlčkou či dvojtečkou. Dle standardu ISO/IEC 15802-1, jež definuje MAC adresy, by však jako oddělovač hodnot měla být použita pouze pomlčka [95].

### 2.2.1 Změna MAC adresy

Ačkoliv měla MAC adresa sloužit jako jednoznačný identifikátor, který není možno uživatelem modifikovat, existují techniky, které umožňují tuto restrikcii obejít. Záměna MAC adres je běžně označována jako „MAC spoofing“ a motivací k provedení záměny může být vícero. Často je například přístup k síťovým prostředkům řízen na základě MAC adres a ke zdroji dat jsou připuštěny pouze stroje, jež mají MAC adresy definované správcem zdroje dat. Hypotetický útočník by tedy mohl změnit svou MAC adresu na některou z povolených a získat data, jež by mu s původní

MAC adresou nebyla přístupna [Ibe23]. V některých institucích se dle MAC adres řídí i přístup do internetu. Přístup do sítě je povolen pouze zařízením, jež mají adresu schválenou administrátorem. Motivací k záměně MAC adresy tedy může být i neoprávněný přístup do sítě internet.

Změnu MAC adresy je možno snadno provést přímo v operačním systému Windows pomocí integrovaného nástroje „Správce zařízení“. V daném nástroji je následně potřeba zobrazit položky seznamu „Síťové adaptéry“ a provést dvojí klepnutí na název zařízení, jež má MAC adresu, která má být změněna. Dalším krokem je přechod na záložku „Upřesnit“, kde je poté potřeba nalézt položku, která umožňuje změnu MAC adresy. Často je v konfiguraci zařízení MAC identifikátor označován jako „Síťová adresa“, konkrétní označení se však může lišit. Závěrem je potřeba zadat do pole označeného „Hodnota:“ požadovanou MAC adresu a provést potvrzení stisknutím tlačítka s nápisem „OK“. Aby se změny projeví, je nutné restartovat síťový adaptér, u kterého byla změna adresy provedena. Obrázek 2.1 zachycuje změnu MAC adresy v operačním systému Windows 11 Pro.

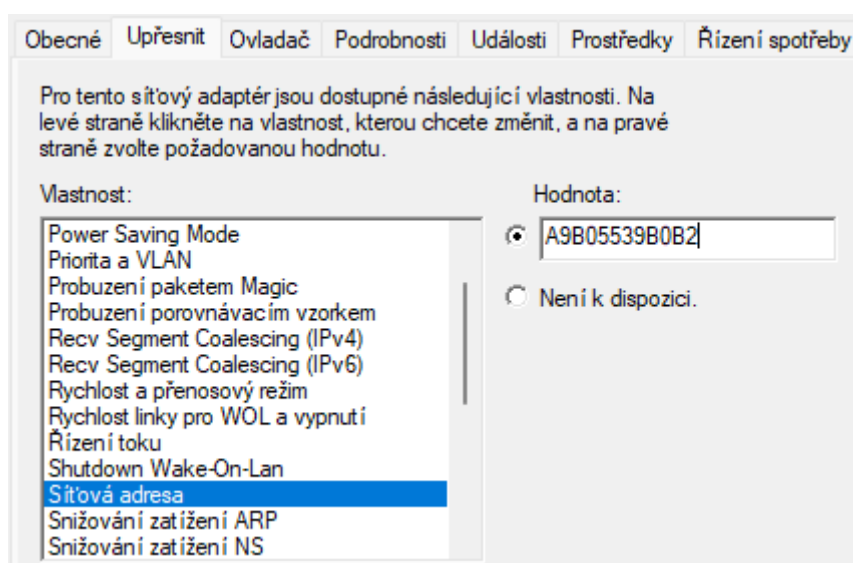
Nutno uvést, že změna MAC adresy na úrovni operačního systému není trvalá a při reinstalaci operačního systému nebo ovladače zařízení síťové karty dojde k obnovení původní adresy, která byla zařízení přidělena výrobcem. K permanentní změně adresy by bylo potřeba modifikovat firmware síťového zařízení.

### 2.2.2 Složky identifikátoru

MAC adresa je tvořena 48 bity, které lze rozdělit do dvou skupin. Prvních 24 bitů (3 bajty) adresy se označuje zkratkou OUI (z anglického „Organizationally Unique Identifier“) a představují identifikátor výrobce zařízení. Těchto charakteristických 24 bitů je výrobcům přidělováno organizací IEEE [SZ14]. Výrobci mohou mít v některých případech přidělený více než jeden identifikátor OUI, ale jeden identifikátor nemůže být přidělen více výrobcům. Příklady OUI známých výrobců jsou:

- Hewlett-Packard: 00-0B-CD,
- Intel Corporation: 00-1B-21,
- Apple, Inc.: 00-17-F2,
- Broadcom: 00-0A-F7,
- Dell Inc.: 00-1E-C9,

Druhou polovinu MAC adresy tvoří sekvence 24 bitů, která představuje sériové číslo síťového zařízení. Sekvence bitů reprezentující sériové číslo zařízení může být výrobcem náhodně vygenerována nebo pro její vytvoření může výrobce využít libovolný algoritmus.



Obrázek 2.1: Změna MAC adresy v operačním systému Windows 11 Pro

## 2.3 IP adresa

Představuje unikátní identifikátor, který je přiřazen každému síťovému zařízení, které pro komunikaci využívá IP protokol. IP adresa může být každému zařízení přiřazena staticky či dynamicky. Statická IP adresa musí být zařízení v síti přiřazena manuálně. Je vhodné, aby manuální změnu IP adresy prováděl pouze správce sítě či jiná pověřená osoba. Pokud je stroji přidělena nevhodná IP adresa, může docházet ke kolizím v rámci sítě či nemusí fungovat připojení počítače do sítě internet [For00].

Dynamické přiřazení IP adresy zařízení probíhá prostřednictvím protokolu DHCP, jenž přidělí nově připojenému zařízení do sítě IP adresu, která dosud není využívána žádným jiným zařízením v síti. Výhodou využití DHCP pro koncové uživatele je automatizace procesu přidělení IP adresy stroji a s tím související absence nutnosti manuálního přiřazení IP adresy.

### 2.3.1 Verze IP protokolu

Délka uvedeného identifikátoru se liší v závislosti na verzi IP protokolu, který je pro komunikaci využit. V současnosti se v běžných sítích používá protokol IP ve verzi 4 a 6. V případě IPv4 je adresa tvořena čtyřmi „oktety“ (tj. osmice bitů), jež jsou od sebe odděleny tečkou („.“). Celková délka IPv4 adresy je tedy 32 bitů.

Jelikož každý oktet 32 bitové adresy může v desítkové soustavě nabývat hodnot 0-255, je možno v protokolu IP verze 4 vygenerovat pouze cca 4,3 bilionu unikátních IP adres. Zejména kvůli rychlému nárůstu síťových zařízení a hrozbě vyčerpání IP adres byl vytvořen protokol IPv6, který využívá 128 bitové adresy. Adresa zaří-

zení je v IPv6 složena z osmi skupin hexadecimálních číslic, přičemž každá skupina reprezentuje 16 bitů. Skupiny jsou od sebe vzájemně odděleny dvojtečkou („:“). Vyjma většího adresního prostoru IPv6 oproti IPv4 přináší i další výhody, například automatickou konfiguraci síťových zařízení po jejich připojení do sítě bez využití DHCP protokolu [23n].

### 2.3.2 Veřejná vs. privátní IP adresa

IP adresa zařízení může být privátní či veřejná. Soukromé adresy mohou být přiděleny například zařízením v domácí síti, kde spolu zařízení v rámci sítě mohou komunikovat na základě znalosti privátní IP adresy zařízení. Zároveň však platí, že zařízení se soukromou IP adresou není možné kontaktovat z vnější sítě (internet). Výhodou nedostupnosti zařízení z vnější sítě může být zvýšená bezpečnost, jelikož hypotetický útočník nemůže k zařízení získat přístup pouze na základě znalosti IP adresy cílového stroje. Nevýhodou privátní IP adresy je omezení či dokonce nemožnost provozování určitých služeb. Zařízení s neveřejnou IP adresou například nemůže sloužit jako webový či mail server, jež má být dostupný z internetu, pokud nejsou provedena dodatečná nastavení dalších síťových prvků [Far22].

Veřejnou IP adresu je možné získat od poskytovatele internetu zpravidla za určitý poplatek. Za výhodu a potenciálně zároveň nevýhodu veřejné IP adresy lze považovat možnost snadného přístupu k zařízení ze sítě internet. Přímý přístup ke stroji na základě IP adresy je možno využít například pro pohodlnou vzdálenou obsluhu počítače pomocí protokolu RDP. Stroj s veřejnou IP lze také například snadno využít jako testovací server. Zařízení s veřejnou IP adresou mohou být potenciálně snadným terčem pro kybernetické útočníky, je tedy nutné přikládat patřičnou váhu zabezpečení stroje.

## 2.4 Broadcast IP adresa

Jedná se o speciální IP adresu síťových prvků, která je využívána odesílatelem dat, pokud mají data obdržet všechny stroje nacházející se v rámci subnetu, kterému daná broadcast adresa náleží. Oproti klasické IP adrese se liší zejména tím, že umožňuje adresovat více příjemců (strojů), přičemž odesílatel nemusí znát IP adresy každého z cílových zařízení. Běžná IP adresa má vždy náležet pouze jednomu stroji a odesílatel zpravidla musí znát IP adresu konkrétního stroje, kterému mají být data zaslána.

Výpočet broadcastové IP adresy se skládá z následujících pěti základních kroků, jež je nutné provést v uvedeném pořadí [Bou08a]:

1. převod IP adresy a síťové masky do binární soustavy,



2. nalezení síťové adresy využitím bitového součinu AND mezi binární IP adresou a síťovou maskou,
3. získání broadcast adresy subnetu, ve kterém se klient nachází, provedením negace (logická operace NOT) nad síťovou maskou,
4. provedení bitového součtu (logická operace OR) mezi broadcast adresou subnetu a síťovou adresou,
5. převod výsledku zpět do desítkové soustavy.

K získání broadcastové adresy je možno použít několik volně dostupných nástrojů, které provedou potřebné výpočty a uživateli poskytnou výslednou adresu. Jedním z nástrojů, které umožňují výpočet adresy uvedeného typu je následující webová aplikace: <https://jodies.de/ipcalc>.

## 2.5 SSH

SSH (z anglického „Secure Shell“) je označení pro program a síťový komunikační protokol, jenž je určený pro dosažení bezpečné komunikace mezi dvěma počítači. Daný protokol se běžně používá pro vzdálené přihlášení a následnou interaktivní práci se stroji pomocí terminálu, resp. příkazového řádku [Luc18]. SSH protokol je dále možno využít například pro neinteraktivní spouštění skriptů, při kterém uživatel pouze provede spuštění skriptu a nečeká na jeho dokončení.

### 2.5.1 Autentizace

Pro autentizaci uživatelů zavádí daný protokol tzv. „SSH klíče“, jež lze definovat jako množinu dvou kryptografických klíčů. Jeden z těchto klíčů se označuje jako veřejný a vlastník klíče ho umísťuje na stroje, ke kterým chce vzdáleně přistupovat. Lokace souboru s veřejnými klíči se liší v závislosti na operačním systému hostitelského stroje. Ve výchozím nastavení jsou veřejné klíče na Linuxových systémech ukládány do souboru `~/.ssh/authorized_keys`. V operačním systému Windows jsou veřejné klíče uživatelů, jež mají vzdálený přístup ke stroji, umístěny v rámci souboru `C:\ProgramData\ssh\administrators_authorized_keys`.

Další z páru klíčů je privátní a jeho vlastník by ho tedy neměl nikomu předávat. Privátní klíč účastník používá pro prokázání své proklamované identity vůči serveru, na kterém je umístěn jeho veřejný klíč. Pokud hostitel, ke kterému se chce uživatel připojit, vyhodnotí, že privátní klíč uživatele náleží k některému z veřejných klíčů, jež se na serveru vyskytují, bude přístup povolen [Luc18]. V opačném případě hostitelské zařízení přístup zamítne a uživatel pokoušející se připojit obdrží chybovou hlášku.

Server k vyhodnocení sounáležitosti uloženého veřejného klíče a privátního klíče uživatele používá následující algoritmus. Jakmile je detekován pokus o klienta o připojení k hostiteli, vygeneruje hostitelský server náhodná data, která zašifruje veřejným klíčem klienta a následně je zašle zájemci o připojení. Klientský stroj poté provede rozšifrování obdržených dat pomocí soukromého klíče a provede odeslání nezašifrovaného výsledku zpět na server. Pokud klient provedl rozšifrování dat pomocí odpovídajícího privátního klíče, potvrdí server totožnost klienta a umožní vytvoření spojení [Mah17].

Veškerá další SSH komunikace, jež proběhne mezi serverem a klientem po autentizaci klienta, je zašifrována pomocí symetrického šifrovacího algoritmu. Volba šifrovacího algoritmu je předmětem dohody účastníků komunikace, přičemž výměna klíče probíhá prostřednictvím Diffie-Hellman algoritmu [Luc18].

Ve výchozím nastavení protokol SSH očekává připojení na TCP portu číslo 22. Další ochranou, která může zajistit vyšší bezpečnost hostitelského stroje vůči kybernetickým útočníkům, je tedy změna výchozího portu na jiný, který není běžně využíván. Díky změně portu na jiný než výchozí by potenciální útočník pro úspěšné získání přístupu k hostitelskému stroji musel vyjma dalších informací znát navíc číslo nového portu hostitele, na kterém je služba SSH provozována.

### 2.5.2 Požadovaná softwarová konfigurace

Aby bylo umožněno připojení ke vzdálenému stroji, je nutné, aby byl na daném hostitelském zařízení provozován tzv. „SSH server“. Instalace SSH serveru je možná na Windows, MacOS i Linuxu. Konkrétní postup zprovoznění serveru závisí na operačním systému hostitele a jeho programové výbavě, ale obecně ho lze rozdělit do několika základních kroků.

Pro provoz serveru může být nejprve nutné nainstalovat některý software, jenž plní funkci serveru a je dostupný pro platformu cílového stroje. Zřejmě nejpoužívanějším SSH serverem je OpenSSH, který je možno provozovat na operačních systémech Windows, Linux i MacOS. Jelikož je SSH server v operačním systému mnohdy předinstalován, je možné v některých případech instalaci vynechat a provést pouze aktivaci dané součásti systému.

Po instalaci či aktivaci SSH serveru může být vyžadována konfigurace firewallu, ve kterém je nutno povolit port, na kterém pracuje služba SSH. Výchozím portem služby SSH je port číslo 22 (TCP), ale je možné provést změnu tohoto portu.

Aby bylo klientům umožněno připojení k serveru pomocí privátního klíče, je potřeba umístit na server veřejné klíče klientů. Lokace klíčů závisí na operačním systému a jeho konfiguraci. Jak již bylo uvedeno, v Linuxu se veřejné klíče ve výchozím nastavení nachází v souboru `~/.ssh/authorized_keys` a v případě Windows se jedná o soubor `C:\ProgramData\ssh\administrators_authorized_keys`. Po

umístění klíčů na server již stačí spustit SSH server, případně nastavit jeho pravidelné spuštění po startu systému.

Pro připojení k SSH serveru je nutné, aby na klientském stroji byl přístupný některý z programů umožňujících připojení k serveru. U většiny distribucí Linuxu není nutné pro připojení k hostiteli instalovat další programy, jelikož distribuce obsahují nástroj „ssh“, který slouží k uvedenému účelu. Nástroj „ssh“ se nachází i v operačním systému MacOS. Pro připojení k serveru ze stroje s operačním systémem Windows je však potřebné dainstalovat některý z programů třetí strany. Známymi SSH klienty jsou například: PuTTY, WinSCP a Bitvise SSH Client. Příklady známých SSH klientů jsou uvedeny v tabulce 2.2.

<b>název</b>	<b>platforma</b>	<b>domovská stránka</b>
PuTTY	Windows, MacOS, Linux	<a href="http://www.putty.org">www.putty.org</a>
MobaXterm	Windows	<a href="http://www.mobaxterm.mobatek.net">www.mobaxterm.mobatek.net</a>
SmarTTY	Windows	<a href="http://www.sysprogs.com/SmarTTY">www.sysprogs.com/SmarTTY</a>
Termius	Windows, MacOS, Linux	<a href="http://www.termius.com">www.termius.com</a>
Bitvise SSH Client	Windows	<a href="http://www.bitvise.com/ssh-client">www.bitvise.com/ssh-client</a>

Tabulka 2.2: Příklady SSH klientů

## 2.6 Powershell

Powershell představuje multiplatformní řešení pro automatizaci úloh. Uvedený nástroj je standardní součástí aktuální verze operačního systému Windows, přičemž je možno daný produkt nainstalovat i na operační systémy MacOS a Linux [PS17].

### 2.6.1 Základní informace

Powershell se skládá ze tří komponent, kterými jsou: příkazový řádek, skriptovací jazyk a automatizační platforma. Pro účely této práce jsou podstatné první dvě jmenované komponenty, které jsou popsány v následujícím textu. Automatizační platforma je určena zejména pro interakce s cloudovými službami, jako jsou například Microsoft Azure, AWS a Google Cloud [22m]. Interakce s cloudovými službami není v rámci této práce vyžadována.

Příkazový řádek Powershellu nabízí moderní rozhraní, jež umožňuje doplňování rozepsaných příkazů tabulátorem a poskytuje přehlednou historii vykonaných příkazů. Navazující příkazy je možno mezi sebou řetězit a docílit tak vytvoření Powershell skriptu. Posloupnost příkazů může být využita k vykonávání různorodých činností, například pro získání informací o počítači, manipulaci se soubory a dalším aktivitám [PS17].

Někteří uživatelé mohou Powershell snadno zaměnit s klasickou příkazovou řádkou systému Windows, jelikož oba nástroje umožňují vykonávání příkazů a poskytují vzhledově podobné uživatelské rozhraní. Klasická příkazová řádka Windows ovšem pracuje pouze s textovými řetězci, což značně limituje možnosti skriptování a automatizace úloh. Powershell na rozdíl od klasické příkazové řádky Windows pracuje s objekty platformy .NET framework, což vede k jednodušší manipulaci se zpracovávanými daty. Využití objektů .NET framework také přispívá k transparentnějšímu předávání dat mezi jednotlivými příkazy a usnadňuje práci s daty v rámci skriptů.

### 2.6.2 Přístup k systémovým komponentám

Powershell se od klasického příkazového řádku liší také množstvím komponent systému, ke kterým má nástroj přístup. Zatímco příkazový řádek má přístup pouze k velmi omezenému množství systémových nástrojů a aplikací, Powershell umožňuje uživateli práci téměř s každou komponentou systému Windows, mezi které patří například úložiště certifikátů, správa síťových zařízení a správa služeb. Přístup k součástem systému je v prostředí Powershell zjišťován pomocí tzv. „providerů“ a „cmdletů“ [PS17].

Provider zprostředkovává přístup k datům specifického typu, přičemž v rámci Powershellu plní obdobné služby jako ovladač zařízení v kontextu operačního systému. Každý Powershell provider definuje sadu abstraktních příkazů, jež umožňují manipulovat s komponentou systému, kterou daný provider spravuje. Konkrétní způsob implementace jednotlivých příkazů zůstává uživateli skryt.

Powershell obsahuje například provider umožňující správu registrů, jež poskytuje příkazy umožňující přidání, mazání a editaci registrových klíčů, resp. jejich hodnot. Informace poskytované providery jsou organizovány do hierarchických adresářových struktur, tzv. „stromů“. Stromová struktura umožňuje pracovat s daty providera obdobně jako se složkami a soubory, které jsou známé ze souborových systémů.

Cmdlet označuje základní funkční prvek Powershellu, jež umožňuje vykonání určité činnosti. Daný prvek má formu krátkého skriptu, jež obsahuje posloupnost příkazů vedoucí k vykonání požadované úlohy. Powershell cmdlety používají konvenci pojmenování, kde je první část názvu cmdletu tvořena slovesem a slouží k popisu charakteru akce, jež cmdlet provádí. Dle začátku pojmenování cmdletu je tedy možno dedukovat, zda může mít zavolání dané Powershell komponenty za následek modifikaci dat či zda komponenta data pouze získává. Cmdlety, jež pouze čtou informace, mají zpravidla v názvu anglická slovesa „Get“, „Select“, „Find“ a „Read“. Moduly měnící data často obsahují v názvu anglická slovesa „Format“ nebo „Convert“ [22a].

Druhá část názvu cmdletu je od první oddělena pomlčkou „-“ a blíže specifikuje konkrétní operaci, jež má cmdlet vykonat. Příklady cmdletů obsahuje tabulka 2.3.

název cmdletu	poskytovaná funkcionální
Get-Process	získání seznamu běžících procesů
Get-Service	získání seznamu spuštěných služeb
Get-NetIPAddress	získání seznamu IP adres přiřazených stroji

Tabulka 2.3: Příklady Powershell cmdletů

Cmdlety uvedené v rámci tabulky 2.3 jsou v rámci Powershellu vestavěné, avšak je umožněno i vytvoření uživatelských cmdletů. Uvedené komponenty musí být vytvořeny pomocí jakéhokoliv kompilovaného jazyka z rodiny .NET nebo PowerShell skriptovacího jazyka [22].

### 2.6.3 Využití v práci

PowerShell je v rámci projektu využíván pro vykonání vybraných operací na počítačích s OS Windows. Nově vytvořené řešení pro správu využívá možnosti PowerShellu například k provádění následujících akcí:

- získání informací o stroji (procesor, RAM, OS, BIOS),
- aktualizování software pomocí nástroje Chocolatey,
- aktualizování systému Windows pomocí modulu PSWindowsUpdate [Gaj22],
- vypnutí seznamu všech uživatelů stroje,
- vypnutí / restart stroje.

Příkazy jsou na vzdálených strojích spouštěny prostřednictvím technologie SSH (viz kapitola 2.5).

## 2.7 Chocolatey

Nástroj Chocolatey představuje zástupce balíčkovacích systémů, jež jsou dostupné pro operační systém Windows.

### 2.7.1 Základní informace

Software Chocolatey umožňuje uživatelům systému Windows snadno a rychle spravovat dostupný software z jednoho místa. Nástroj ve výchozím nastavení pro interakci s uživatelem využívá příkazovou řádku, avšak je možno software rozšířit o podporu grafického rozhraní prostřednictvím balíčku Chocolatey GUI [23].

Chocolatey poskytuje obdobné funkcionality, které jsou známé z Linuxových balíčkovacích systémů APT nebo Pacman. Prostřednictvím daného balíčkovacího systému je tedy možné aktualizovat stávající software stroje, přidat nové programy i odebrat již nainstalované aplikace. Balíčky software poskytované Chocolatey jsou ve výchozí konfiguraci získávány pouze z jediného zdroje, kterým je Chocolatey Community Repository (viz <https://community.chocolatey.org/>). Přítomnost jednoho zdroje balíčků v aplikaci Chocolatey může mít za následek dostupnost menšího množství software, jež lze na počítač nainstalovat. Využití jednoho oficiálního zdroje balíčků ovšem může přinášet výhody v oblasti bezpečnosti. Při využití pouze jednoho oficiálního repozitáře je omezena možnost instalace škodlivých programů, které by mohly poskytovat neoficiální zdroje a jež by mohly ohrozit bezpečnost stroje.

Výchozí zdroj balíčků Chocolatey poskytuje více než 9000 programů, které lze získat [23i]. V oblasti řešení správy software pro operační systém Windows Chocolatey počtem dostupných programů převyšuje alternativní řešení, jež jsou volně dostupná. Uvedený balíčkovací systém ovšem v počtu nabízených balíčků nemůže konkurovat Linuxovým řešením pro správu balíčků, které zpravidla počtem nabízených programů Chocolatey převyšují. Konkrétní počet balíčků, jež nabízí Linuxová řešení, závisí na Linuxové distribuci, resp. konfiguraci zdrojů software. Například výchozí repozitář distribuce Ubuntu, která je jednou z nejznámějších Linuxových distribucí, poskytuje více než 60 000 softwarových balíčků [23v].

Obdobně jako systém APT, zajišťuje Chocolatey instalaci všech závislostí, které jsou potřeba pro provoz instalovaného software. V praxi tedy může být při vykonávání příkazu k instalaci požadovaného balíčku nainstalován i další podpůrný software, jenž je potřebný pro správnou funkci instalovaného balíčku. Automatická instalace závislostí zajišťuje snadnou instalaci programu a eliminuje riziko opomenutí instalace závislosti, ke které by mohlo dojít při manuální instalaci požadované aplikace.

Nástroj Chocolatey může usnadnit mnoho úloh spojených se správou software, avšak instalace některých programů nemusí být prostřednictvím tohoto nástroje proveditelná. Například aplikace, jež vyžadují zakoupení licence, zpravidla nelze pomocí daného nástroje nainstalovat. Chocolatey nedisponuje žádnou formou platební brány, která by umožňovala příjem peněz od zákazníka a současně zajišťovala převod obnosu k poskytovateli aplikace. Dalším důvodem, proč nejsou některé aplikace v rámci Chocolatey dostupné, mohou být licenční podmínky vývojáře programu.

### 2.7.2 Alternativní nástroje

V tabulce 2.4 jsou uvedeny některé alternativní balíčkovací systémy pro systém Windows. Obecně platí, že každý z existujících balíčkovacích systémů poskytuje ob-

dobnou funkcionalitu jako Chocolatey. Všechny níže uvedené systémy pro správu software tedy umožňují instalaci nových programů, aktualizaci stávajících a odebrání již nepotřebných aplikací.

název	dostupné zdarma	domovská stránka vývojáře
Scoop	✓	<a href="http://www.scoop.sh">www.scoop.sh</a>
WinGet	✓	<a href="http://www.microsoft.com">www.microsoft.com</a>
Ninite Pro	✗	<a href="http://www.ninite.com">www.ninite.com</a>

Tabulka 2.4: Alternativní nástroje pro správu Windows software

Existují alternativní nástroje, přičemž jeden je spravován přímo vývojářem OS Windows - firmou Microsoft Corporation. Častou slabinou alternativních nástrojů je ovšem malé množství software, jež ostatní balíčkovací systémy nabízejí ve srovnání s Chocolatey. Alternativní programy může být vhodné použít zejména v případě, že pro některou z požadovaných aplikací neposkytuje Chocolatey instalační balíček.

### 2.7.3 Využití v práci

Chocolatey je provozován na většině laboratorních počítačích ZČU KIV, kde slouží ke správě mnoha instalovaných programů. Produkt této práce by mohl umožnit vzdáleně vyvolat aktualizace Chocolatey balíčků, čímž by se usnadnil a částečně zautomatizoval proces aktualizace software katedrálních strojů. Nově vytvořené řešení by rovněž mělo obsahovat funkcionalitu, která umožní získat seznam nainstalovaného Chocolatey software. Získané informace o nainstalovaných programech lze využít například ke zjištění, zda je softwarová výbava stroje kompletní.

Vytvořené řešení nebude vyžadovat instalaci software Chocolatey na spravovaný stroj, avšak bez programu Chocolatey nebudou dostupné možnosti správy programů nainstalovaných na stroji. Ostatní funkcionality aplikace, jako je například zjištění informací o stroji a získání seznamu přihlášených uživatelů, budou i nadále dostupné. V případě, že by bylo v budoucnu žádoucí změnit systém pro správu balíčků, se kterým bude řešení pracovat, bude potřeba provést změny pouze ve třídě `ChocoService`. Uvedená třída obsahuje logiku výkonu funkcí, jejichž řešení poskytuje v oblasti správy software.

## 2.8 SSO

Zkratka „SSO“ vychází z anglického „Single sign-on“. Jedná se o techniku, která umožňuje uživateli použít pro přihlášení k více systémům jednotné přihlašovací údaje. Primární výhodou pro uživatele je absence nutnosti pamatovat si přihlašovací

údaje ke každému systému zvlášť. Nevýhodou je fakt, že pokud by byly přihlašovací údaje odcizeny, mohl by útočník získat přístup ke všem systémům, jež původní vlastník údajů využíval.

V praxi mohou být pro zvýšení ochrany před neoprávněným přístupem zavedeny techniky, které zamezují přístup k systémům obsahujícím důležitá data pouze na základě přihlašovacích údajů uživatele. Pro zvýšení úrovně ochrany dat organizace, jež SSO využívá, může být například využita určitá forma dvoufázového ověření [23h]. Při využití dvoufázového ověření by útočník pro přístup k datům musel kromě přihlašovacích údajů získat i přístup k zařízení uživatele, jež je zapojeno do procesu dvoufázového ověření. Pro dvoufázové ověření může být využita SMS zasláná na mobilní číslo uživatele pokoušejícího se přihlásit či je možno využít specializovaných řešení, například: Microsoft Authenticator, Google Authenticator a LastPass Authenticator.

### 2.8.1 Protokol Kerberos

Pro autentizaci uživatelů pomocí techniky SSO je na ZČU využíván protokol Kerberos. Uvedený protokol byl vyvinut na univerzitě MIT [22g] a je založen na principu ověřovacích lístků, tzv. „ticketů“.

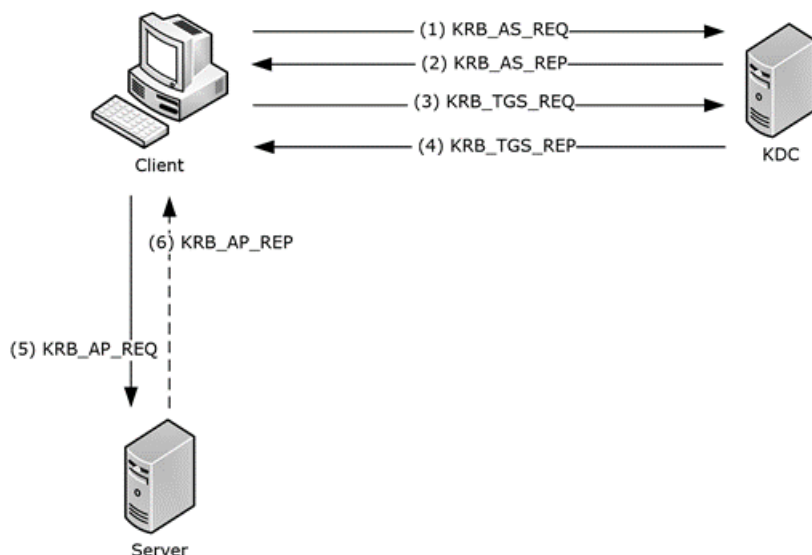
#### 2.8.1.1 Mechanismus fungování protokolu

Proces ověření identity osoby, jež se prokazuje chráněnému systému, je v protokolu Kerberos založen na věrohodném serveru třetí strany - „KDC“ serveru. Zmíněný server se skládá ze dvou hlavních komponent, které jsou označovány jako AS a TGS. Při pokusu o získání ticketu je nejprve kontaktována autentizační komponenta (AS), které je poskytnuto přihlašovací jméno uživatele. Autentizační komponenta KDC serveru provede porovnání zasláného uživatelského identifikátoru s údaji přítomnými v databázi nacházející se na serveru. Pokud jsou poskytnuté údaje správné, obdrží uživatel tzv. „session key“ a token označovaný jako „TGT“, jež jsou následně použity pro komunikaci s TGS komponentou.

Dalším nutným krokem, jenž musí klient vykonat pro získání ověřovacího lístku, je zaslání zašifrovaného TGT tokenu komponentě zajišťující přidělování ověřovacích lístků (TGS). Zašifrování TGT tokenu musí být vykonáno pomocí klíče („session key“), jenž byl v předchozím kroku získán od autentizační součásti systému. Komponenta TGS provede dešifrování zasláného TGT tokenu a pokud je poskytnutý token validní, obdrží uživatel pověřovací lístek („service ticket“) a šifrovací klíč, jenž bude použit pro komunikaci se službou chráněnou systémem Kerberos [19]. Klient má na základě pověřovacího lístku možnost přistoupit k chráněnému systému po určitou dobu [23z].



Princip ověření pomocí protokolu Kerberos je graficky znázorněn na obrázku 2.2. Vysvětlivky ke zprávám, jež jsou přítomné v uvedeném nákresu, obsahuje tabulka 2.5.



Obrázek 2.2: Princip funkce protokolu Kerberos [21]

typ zprávy	účel / popis zprávy
KRB_AS_REQ	požadavek na získání TGT (klient -> KDC)
KRB_AS_REP	zaslání session key a TGT (KDC -> klient)
KRB_TGS_REQ	poskytnutí TGT zašifrovaného session key (klient -> KDC)
KRB_TGS_REP	zaslání ticketu a šifrovacího klíče (KDC -> klient)
KRB_AP_REQ	poskytnutí zašifrovaného ticketu (klient -> server)
KRB_AP_REP	nepovinná odpověď značící validitu ticketu (server -> klient)

Tabulka 2.5: Popis zpráv používaných při komunikaci prostřednictvím Kerberos

### 2.8.1.2 Konfigurace v rámci ZČU

Jelikož Kerberos umožňuje nastavit mnoho parametrů souvisejících s procesem autentizace, může mít každá organizace využívající tento protokol zavedena jiná pravidla. Následující část textu se věnuje konfiguraci protokolu Kerberos, která je aplikována v prostředí ZČU. Konfigurace a údržba systému autentizace na ZČU je úlohou Centra informatizace a výpočetní techniky (CIV).

Platnost pověřovacího lístku získaného uživatelem odpovídá běžné pracovní době (8 hodin), resp. výchozí platnosti definované protokolem Kerberos [23z]. Vlastník pověření má možnost průběžně obnovovat platnost až do doby 14 dnů.

V prostředí ZČU existují celkem 4 KDC servery, jejichž adresa a typ jsou uvedeny v tabulce 2.6. Předmětem tabulky 2.6 je rovněž i seznam protokolů a portů, prostřednictvím kterých servery poskytují služby. Jeden ze serverů je označený jako „master“ a poskytuje služby spojené s administrací uživatelských kont. Tři servery s označením „replika“ poskytují služby pro zajištění autentizace uživatelů a vydávání pověřovacích lístků [13].

adresa serveru	typ	protokol a port
kerberos-adm.zcu.cz	master	TCP 749, TCP/UDP 88
kerberos1.zcu.cz	replika	TCP/UDP 88, UDP 4444, UDP 9878
kerberos2.zcu.cz	replika	TCP/UDP 88, UDP 4444, UDP 9878
kerberos3.zcu.cz	replika	TCP/UDP 88, UDP 4444, UDP 9878

Tabulka 2.6: KDC servery na ZČU


Mezi systémy, které jsou v rámci ZČU chráněny systémem Kerberos, patří například „Portál ZČU“ (<https://portal.zcu.cz>) a poštovní služba „WebMail“ (<https://webmail.zcu.cz>). Pro přihlášení k uvedeným službám se uživatel musí prokázat svým Orion loginem a příslušným heslem. Grafická podoba webové stránky, jež vybízí uživatele k prokázání identity pomocí Orion loginu, je vyobrazena na snímku 2.3.

Ověření uživatele pomocí Orion loginu je možno implementovat i v projektech třetí strany, jelikož CIV poskytuje konfigurační soubor `krb5.conf`. Uvedený soubor je možno získat na adrese <https://www.download.zcu.cz/public/config/krb5/krb5.conf>. Cílová lokace souboru `krb5.conf` se liší v závislosti na řešení, do kterého má být autentizace prostřednictvím protokolu Kerberos implementována. Konfigurační soubor protokolu Kerberos obsahuje veškerá data, jež jsou potřebná pro ověření identity uživatele. Obsahem souboru jsou například informace o KDC serverech, seznam využívaných Kerberos knihoven a konfigurace logování. Detailní popis konfiguračního souboru poskytují vývojáři produktu Kerberos na adrese: [https://web.mit.edu/kerberos/krb5-1.12/doc/admin/conf\\_files/krb5\\_conf.html](https://web.mit.edu/kerberos/krb5-1.12/doc/admin/conf_files/krb5_conf.html).

## 2.9 Wake-on-LAN

Jedná se o standard umožňující vzdálené zapnutí či probuzení počítačové stanice pomocí speciálního síťového rámce, jenž je označován jako Magic Packet - viz kapitola 2.9.1. Funkce Wake-on-LAN musí být podporována základní deskou, resp. síťovou

**Orion WebAuth**



**ZÁPADOČESKÁ  
UNIVERZITA  
V PLZNI**

**Orion login:**

**Heslo (password):**

[Potřebujete pomoc?](#) | [Need Help?](#)

**Kde to jsem? Kam jsem se to zase dostal?**

Webový server, na který se snažíte přihlásit, byl zařazen do domény jednotného přihlášení (single sign-on, SSO), a vyžaduje ověření vaší identity platným uživatelským jménem a heslem. Stránka, na které se právě nacházíte, je vstupním bodem k webovým serverům ZČU zařazeným pod systém jednotného přihlašování. To znamená, že svoji identitu prokážete skrze tuto stránku prvnímu serveru, na který chcete přistupovat, a tím jste automaticky přihlášen(a) i k ostatním serverům v doméně.

**Výhody**

Větší pohodlí pro uživatele (heslo zadávají jen jednou) a technicky vyšší bezpečnost: mezi prohlížečem a webovým serverem se neposílá heslo, ale jen autentizační token. Platnost tokenu je navíc časově omezena.

**Důležitá upozornění!**

 Nikdy nezadávejte Orion jméno a heslo do webových formulářů, pokud se nejedná o přihlašovací stránku systému WebAuth (tato stránka) nebo oficiální webový nástroj pro změnu hesla. Obě aplikace jsou provozovány na stroji *shib.zcu.cz!!!*

Po zadání hesla se zpřístupní všechny servery, včetně těch, se kterými právě nepracujete. Je zde větší riziko zneužití přístupových práv uživatele, odejde-li od počítače. Pro bezpečné odhlášení je potřeba ukončit webový prohlížeč.

Obrázek 2.3: Ověření uživatele v rámci ZČU

kartou stanice, jež má být vzdáleně zapnuta. Může být vyžadována aktivace dané funkce v BIOSu počítače, neboť funkce může být ve výchozím nastavení vypnutá.

Důvodem pro vypnutí funkce u notebooků či obdobných zařízení může být zvýšení výdrže zařízení při běhu na baterii. Pro využití uvedené technologie musí být síťová karta zařízení neustále napájena, aby mohla reagovat na síťový rámec, jenž má počítač zapnout.

Funkci Wake-on-LAN je obvykle možno bez zdlouhavé konfigurace provozovat v případě, že se zařízení, jež odesílá speciální síťový rámec, nachází ve stejném subnetu sítě jako počítač, jenž má být zapnut. Rámec zdrojové zařízení odešle do sítě ve formě broadcastu, obdrží ho tedy všechna zařízení v subnetu [Bou08b]. Zapnutím ovšem zareaguje pouze zařízení, jež má MAC adresu odpovídající obsahu obdrženého rámce.

Zapnutí počítače, jenž se nachází v rámci jiného subnetu než zařízení odesílající Magic Packet, mohou provázet různé komplikace. Více informací o tzv. „cross-

segment Wake-on-LAN“ obsahuje kapitola 2.9.2.

### 2.9.1 Struktura rámce „Magic Packet“

Magic Packet lze definovat jako síťový rámec, jehož prvních 6 bajtů má hodnotu 255 (FF v hex formátu). Následuje MAC adresa zařízení, která je 16x opakována. Jelikož MAC adresa má 6 bajtů, činí velikost Magic Packetu celkem 102 bajtů, viz následující výpočet [98]:

- 6x bajt s hodnotou FF
- 16x MAC: 6 bajtů x 16 = 96 bajtů
- 6 bajtů + 96 bajtů = 102 bajtů

### 2.9.2 Zapnutí zařízení umístěných mimo segment

Technologii Wake-on-LAN je možno obvykle bez potíží provozovat, pokud se odesílatel speciálního rámce nachází v rámci stejného segmentu sítě jako jeho příjemce (viz začátek sekce 2.9). Pokud se každý ze subjektů (odesílatel či příjemce paketu) nachází v rámci jiného subnetu, je možné využít modifikaci Wake-on-LAN, která je označována jako „cross-segment Wake-on-LAN.“

Struktura Magic Packetu, který se využívá pro zapnutí stroje v rámci jiného segmentu, je shodná se strukturou paketu, jež byla popsána v rámci kapitoly 2.9.1. Rozdílná je ovšem cílová adresa, na kterou je paket odeslán. Pokud má být zapnuto zařízení nacházející se v odlišné části sítě, musí být Magic Packet odeslán na broadcastovou adresu podsítě, ve které se příjemce nachází [Bou08b].

Cross-segment Wake-on-LAN vyžaduje konfiguraci síťových prvků, které se nacházejí mezi příjemcem a odesílatelem Magic Packetu. Síťová zařízení, která subnety spojují, ve výchozím nastavení zpravidla blokují přeposílání broadcastových zpráv. Hlavním důvodem blokování broadcastových zpráv je snaha snížit riziko spojené s vysokým množstvím broadcastových zpráv. Pokud by některé ze zařízení, jež zpracovávají síťový provoz, napadl útočník, mohl by v případě povolených broadcastových zpráv snadno zahltnout celou síť.

Je zřejmé, že povolení přeposílání broadcastových zpráv na síťových zařízeních není vždy žádoucí. V mnoha případech však ani není možné provést konfiguraci síťových zařízení nacházejících se mezi stroji, jelikož k nim není umožněn přístup.

## 2.10 TCO

Zkratka TCO vznikla ze anglického spojení „Total Cost of Ownership“, jež se do češtiny běžně překládá jako „Celkové náklady spojené s vlastnictvím“. Jedná se o me-

triku, která se snaží vyjádřit celkové náklady spojené s pořízením a následným provozem produktu po celou dobu jeho využívání [23e].

Kalkulace TCO je žádoucí zejména při výběru produktů, které mají dlouhou plánovanou životnost. Náklady spojené s následnou údržbou pořízeného řešení mohou být i mnohonásobně vyšší než jsou počáteční pořizovací náklady. Zejména ve firemní sféře je nutné pečlivě odhadnout a uvážit všechny budoucí náklady ještě před nasazením řešení, jelikož systém zpravidla ovlivní více zaměstnanců.

Využití uvedené metriky je u podniků běžnou praxí a firmy TCO kalkulují například při výběrových řízeních na různorodé služby či produkty. Zjištění TCO však může být žádoucí i pro jednotlivce, kteří plánují pořídit službu či majetek, jež plánují využívat po delší dobu.

### 2.10.1 Oblasti využití

Metrika TCO se využívá v mnoha odvětvích. Velmi často je důležitost metriky vydvihována ve spojitosti se softwarem. Je běžné, že výrobci softwarových řešení prodají zákazníkům produkt za určitou počáteční částku a následně účtují pravidelné poplatky za užívání výrobku. Pravidelných poplatků může být samozřejmě vícero a závisí na konkrétním produktu (službě), resp. podmínkách poskytovatele daného řešení.

Vyjma oblasti informačních technologií se výpočet metriky TCO využívá v automobilismu, kdy se daná metrika využívá k odhadu celkových nákladů na vlastnictví automobilu. Často je TCO kalkulováno také ve spojitosti s nemovitostmi.

### 2.10.2 Zahrnuté položky

Předmětem následujících podkapitol je popis složek TCO, které se vyskytují zejména při kalkulaci očekávaných nákladů na provoz produktů z oblasti informačních technologií. Položky, jež je nutno zahrnout při kalkulaci TCO jsou z velké části specifické a odvíjejí se od segmentu trhu, do kterého zkoumaný produkt spadá [22j].

#### Přímé náklady na pořízení

Tato součást TCO vyjadřuje finanční obnos, který je nutno vynaložit ještě předtím, než se začne daný produkt užívat. Daná součást TCO vždy zahrnuje pořizovací cenu produktu, případně může zahrnovat i náklady na jeho přepravu (pokud existují). Může se jednat například o cenu za pořízení licence určitého programu.

#### Údržba

Náklady spojené s pravidelnou údržbou představují finance, jež je nutno vynaložit, aby zakoupené řešení pracovalo dle očekávání po celou dobu své životnosti. V pří-

padě produktů z oblasti informačních technologií může tato položka zahrnovat například náklady na zálohování dat.

### **Opravy**

Opravy jsou náklady, které souvisejí s uvedením systému do opětovného provozu po jeho poruše. Náklady na opravu zahrnují ceny náhradních dílů, jež jsou potřebné k provedení opravy a cenu práce techniků, kteří opravu provádějí. Mohou být zahrnuty i další poplatky, které jsou specifické pro opravované zařízení.

### **Pravidelné poplatky**

Představují množinu poplatků, jež je nutno pravidelně uhradit, aby systém pracoval, resp. aby jeho provoz byl z právního hlediska legální. Druh a výše těchto poplatků je velmi různorodá a závisí na konkrétním zařízení (službě), jež je spravována. V případě softwarových řešení je například mnohdy nutné pravidelně platit obnovu licence software.

Podmnožinu pravidelných poplatků mohou tvořit i náklady na energie, jež jsou potřebné pro provoz systému. V případě provozu IT řešení je zdrojem energie téměř výhradně elektrická síť, provozovatel řešení tedy hradí poplatky poskytovateli elektrické energie. Je také možné, že produkt pro svůj provoz nepotřebuje téměř žádnou energii, a tak může být odhad spotřeby energie ve výpočtu TCO zanedbán. Náklady na energie lze zanedbat například u výpočetně nenáročných programů, které budou spuštěny na strojích, které jsou primárně určeny k výkonu jiných činností.

### **Konec životního cyklu**

V rámci výpočtu TCO metriky by měly být započítány i náklady spojené s ukončením životního cyklu produktu. Výše finančního obnosu, který je nutno vynaložit na ukončení užívání produktu či služby, se může odvíjet například od charakteru produktu. V případě ukončení užívání nehmotných věcí (např. software) by v TCO měly být započítány zejména náklady na přechod na alternativní software, jenž bude plnit funkci původního. Výše nákladů na migraci společnosti na alternativní program závisí zpravidla na velikosti firmy, resp. množství uživatelů produktu.

# Průzkum existujících řešení

## 3

Tvorbě mobilní a webové aplikace zaměřené na správu stanic v rámci univerzity předcházeli průzkum vybraných existujících softwarových řešení, které umožňují vzdálenou správu počítačových stanic. V rámci průzkumu byly analyzovány některé z neznámějších produktů, jež jsou na trhu dostupné a řadí se do kategorie vzdálené správy. U každého uvedeného software proběhlo vyhodnocení vlastností uvedených v kapitole 3.1.

## 3.1 Hodnocené vlastnosti aplikací

Před samotnou analýzou existujících systémů pro správu počítačů jsem vytvořil seznam funkcí, které bych u aplikací, jež spadají do dané kategorie, očekával. Následně jsem přítomnost jednotlivých funkcí ověřil u každého zkoumaného systému. Následující podkapitoly obsahují bližší popis jednotlivých požadavků kladených na hodnocené aplikace.

### 3.1.1 Zapnutí stroje

Aplikace je schopna cílovou stanici zapnout či ji probudit ze spánku. Uvedená akce vyžaduje kompatibilitu stanice s funkcí Wake-on-LAN a může být nutné provést manuální aktivaci této funkce. Předmětem kapitoly 2.9 je popis funkce Wake-on-LAN.

Zapnutí, resp. probuzení počítače je provedeno pomocí tzv. Magic Packetu. Jedná se o síťový rámec, jehož velikost činí 102 bajtů. Bližší popis struktury Magic Packetu je uveden v kapitole 2.9.1.

Může být také žádoucí, aby software umožňoval naplánovat automatické zapnutí PC v čas určený uživatelem.

### 3.1.2 Zjištění dostupnosti

Kritérium je splněno v případě, že daný zkoumaný produkt umožňuje zjistit, zda je cílový stroj zapnutý. Dostupnost stroje se v počítačových sítích běžně testuje pomocí programu „ping“, který je dostupný ve všech majoritních operačních systémech (Windows, MacOS, Linux).

### 3.1.3 Informace o stroji

Udává, zda analyzovaná aplikace umožňuje získat alespoň základní informace o spravovaném zařízení. Zjištění modelu procesoru stanice a jeho taktu, velikosti paměti RAM, modelu grafické karty apod. může být žádoucí z více důvodů.

Uživatel systému si může díky získaným informacím snadno udělat představu o výkonnosti stanice. Funkcionalita získání parametrů stanic však může být využitelná i pro účely inventarizace. Pokud je k dispozici inventární seznam obsahující očekávané komponenty počítače, lze jejich skutečnou přítomnost zkontrolovat pomocí této funkce.

Případně pokud je k dispozici seznam hardware, kterým má určitá stanice disponovat, lze funkci využít k průběžnému ověřování kompletnosti stanice.

### 3.1.4 Aktualizace operačního systému

Software umožňuje vzdáleně spustit aktualizace systému na cílové stanici. Proces aktualizace operačního systému je mnohdy zdlouhavý a je žádoucí vykonávat ho v době, kdy počítač není aktivně využíván.

### 3.1.5 Multiplatformní server

Software splňuje tento požadavek, pokud umožňuje spravovat podřízené stanice z více operačních systémů. Může se jednat o webovou aplikaci, která je přizpůsobena pro běh na různých operačních systémech a webových prohlížečích. Kritérium je uspokojeno i v případě, že řešení nabízí klasický desktopový program, který lze nainstalovat na více platformě.

Pokud aplikace nabízí instalační balíčky pro více platformě, budou u dané aplikace uvedeny názvy podporovaných platformě.

### 3.1.6 Uživatelské role

Systém umožňuje přiřazení role uživatelům, přičemž každá role má v systému jiná práva přístupu ke strojům. Například pokud bude v aplikaci figurovat administrátor univerzitních strojů, měl by mít právo spravovat údaje veškerých zařízení, která



jsou v systému uvedena. Oproti tomu běžný uživatel aplikace by mohl mít přístup pouze ke strojům, jež mu byly přiděleny administrátorem či jinou nadřazenou rolí.

### 3.1.7 Správa multiplatformních stanic

Zařízení spravovaná řešením mohou mít rozdílné operační systémy. Kritérium hodnocení je splněno, pokud program umožňuje obsluhu strojů, které mají odlišné operační systémy. Pokud aplikace splňuje daný požadavek, jsou uvedeny konkrétní operační systémy, jejichž správa je umožněna.

### 3.1.8 Spuštění skriptů

Systém umožňuje na cílové stanici spustit uživatelem dodané příkazy ve formě skriptů. Pomocí skriptů lze automatizovat rozličné konfigurační a instalační operace. Typickým využitím skriptů může být přidání dalších uživatelů do systému, instalace nového software apod.

### 3.1.9 Plán obsazení stroje

Aplikace je schopna získat informace o plánovaném obsazení strojů z externích zdrojů. Konkrétní implementace této funkce závisí na typu instituce, pro kterou je analyzovaný systém určen. V případě univerzity se může jednat o napojení na rozvrhové služby, jež spravují rozvrhy učeben (např. IS/STAG). U jiných organizací může být daná funkce realizovaná například napojením na Google Kalendář či jinou službu spravující kalendář akcí dané organizace.

### 3.1.10 Autentizace externím zdrojem

Softwarové řešení umožňuje ověření identity uživatele pomocí již existujícího externího zdroje. Zpravidla je žádoucí využít autentizační systém, jež je v dané organizaci použit i pro ověření uživatele v ostatních aplikacích organizace. Výhoda je zřejmá - uživatel si nepotřebuje pamatovat více přihlašovacích údajů a šetří čas, jenž by strávil přihlašováním do každé aplikace podniku zvlášť.

Technologie umožňující využití jednoho uživatelského jména a hesla napříč více aplikacemi je označována jako SSO (Single Sign-On). Protokolů umožňujících ověření uživatele je samozřejmě vícero. Příkladem protokolu umožňujícího ověření je Kerberos, který je využíván zejména v univerzitním prostředí. Popis protokolu Kerberos je předmětem kapitoly 2.8.1.

Vzhledem k různorodosti autentizačních protokolů je u každého zkoumaného produktu uveden název konkrétního protokolu, pomocí kterého ověření probíhá.

### 3.1.11 Cena

Parametr hodnocení udává, zda je zkoumaný software nabízen zdarma či zda se jedná o zpoplatněné řešení. Placený program může být nabízen za jednorázový poplatek. Ve zkoumaném segmentu trhu je však běžné, že software je poskytován za pravidelný poplatek.

Výše poplatků je u jednotlivých produktů daného segmentu rozdílná a může záviset na řadě faktorů. Vývojář řešení pro správu strojů může cenu licence přizpůsobit typu organizace, počtu spravovaných strojů apod. Vzhledem k uvedeným informacím je zřejmé, že u mnoha aplikací se mohou celkové náklady provozu software významně lišit. Například pokud bude placené řešení pro správu stanic používat soukromá osoba, může jí být poskytnuta levnější licence než firmě, jež vlastní desítky počítačů užívaných pro komerční účely.

Z uvedených důvodů je vhodné před nasazením systému zvážit tzv. TCO (Total Cost of Ownership), jenž udává celkové náklady, které bude nutné vynaložit v souvislosti s provozem daného software. Více informací o dané metrice lze nalézt v kapitole 2.10.

## 3.2 Výběr aplikací pro analýzu

Předmětem kapitoly je popis kritérií, na základě kterých byl sestaven seznam existujících řešení, jejichž schopnosti byly analyzovány.

### 3.2.1 Skóre webu G2

Produkty pro analýzu byly zvoleny zejména kvůli vysokému skóre, kterého dosahují na webu společnosti G2 (<https://company.g2.com/>). Uvedená firma sbírá recenze na rozličné produkty a služby od reálných uživatelů. Jakýkoliv vlastník produktu se může prokázat svým LinkedIn účtem a přidat na web vlastní hodnocení. Popsaný způsob hodnocení, kdy uživatelé hodnotí služby přímo a výsledek je poskytnut dalším uživatelům, je mnohdy označován jako „peer-to-peer review“. Společnost sbírá hodnocení komerčních i volně dostupných produktů.

Skóre G2, jehož produkt dosáhne, závisí na několika faktorech - např.:

- recenze uživatelů na webu G2,
- obsažení všech důležitých funkcí (dle týmu G2),
- ohlasy na produkt na sociálních sítích,
- popularita produktu.

Jelikož složky výpočtu uvedené metriky jsou proměnlivé, může se skóre G2 každého produktu v čase měnit. Další informace o uvedené metrice a jejím výpočtu lze nalézt na webu společnosti G2 - <https://research.g2.com/methodology/scoring>.

Volba nástrojů na základě zmíněné metriky mi přijde vhodná, jelikož dosažené skóre závisí na experech i běžných uživateli. Běžným uživatelem může být kdokoliv, kdo si daný software nainstaluje a poskytne recenzi. Experty v tomto případě představují členové týmu G2, kteří kontrolují, zda program poskytuje požadované funkce. Očekávaná funkcionalita programu je stanovena dle kategorie, do které se daný software řadí. V případě mnou zkoumaných nástrojů se jedná o kategorii RMM (Remote Monitoring & Management).

### 3.2.2 Hodnocení uživatelů

Při volbě aplikací k průzkumu byly brány v potaz i uživatelské recenze produktů, jež se nacházejí na webu SourceForge (<https://sourceforge.net/>). Web se primárně zabývá hostováním software s otevřeným zdrojovým kódem. Spuštění webu se datuje do roku 1999 a počet hostovaných projektů je nyní více než 502 000 [23a]. Popularitu webu dokazuje i fakt, že k němu přistupuje více než 2,6 milionu uživatelů každý den.

Z uvedených hodnot je zřejmé, že si web získal popularitu napříč vývojáři i běžnými uživateli, jež zde stahují software. Kromě hostování umožňuje SourceForge i hodnocení produktů, přičemž hodnocené produkty mohou být i komerční, s uzavřeným kódem.

Hodnotitel při vytvoření recenze poskytne informace, mezi které patří například:

- typ uživatele (administrátor, běžný uživatel),
- doba užívání produktu,
- periodičita užívání produktu (každý den, každý týden...).

Recenze publikované na webu jsou veřejně dostupné (i bez registrace). Hodnocení produktů lze filtrovat na základě uvedených kritérií (typ uživatele atd.). V případě, že je hodnocen produkt, který je využíván zejména ve firemní sféře, lze využít i filtrování na základě velikosti organizace, ve které hodnotitel působí.

Zohlednění recenzí z uvedeného webu při výběru aplikací pro analýzu mi přijde vhodné, jelikož web má rozsáhlou uživatelskou základnu a mnoho možností filtrace recenzí. Lze tedy například vyfiltrovat pouze recenze od uživatelů, kteří působí v organizaci, jejíž velikost odpovídá univerzitě atd.

## 3.3 NinjaOne

Dle webů SourceForge [23c] a G2 [23b] se jedná o nejoblíbenější aplikaci ve své kategorii. Na webu G2 dosahuje průměrného hodnocení 4,8 z 5 možných bodů. V případě webu SourceForge je průměrné hodnocení programu rovněž 4,8 z 5 možných bodů.

Domovská stránka produktu: <https://www.ninjaone.com/>.

### 3.3.1 Hodnocená kritéria

#### Zapnutí stroje

Program umožňuje vzdálené zapnutí cílového stroje pomocí technologie Wake-on-LAN.

#### Zjištění dostupnosti

Aplikace zobrazuje u každého stroje jeho uptime<sup>1</sup> a je schopna detekovat jeho odpojení od sítě.

#### Informace o stroji

Produkt umožňuje získat informace o procesoru ve stanici, velikosti osazené paměti RAM a kapacitu přítomných disků. Aplikace je schopna i reportovat informace o aktuálním vytížení systému. Tedy je možno získat aktuální vytížení CPU, obsazenost paměti RAM v daném čase apod.

NinjaOne navíc umožňuje přistupovat k terminálu (příkazové řádce) vzdálených stanic, více informací o hardwaru spravovaného stroje lze tedy získat i terminálovými příkazy. Množství dat, jež lze získat pomocí terminálu, samozřejmě závisí na operačním systému, kterým spravovaná stanice disponuje.

#### Aktualizace operačního systému

Produkt umožňuje na dálku aktualizovat systém spravovaných stanic. NinjaOne administrátora upozorní, pokud má některé ze spravovaných zařízení dostupné aktualizace a umožní spuštění aktualizacího procesu. Není nutné instalovat veškeré dostupné aktualizace, ale je možno zvolit pouze jejich podmnožinu. Je tedy například možné nainstalovat pouze kritické aktualizace a instalaci volitelných odložit. Využití této možnosti může být výhodné, zejména pokud potřebuje uživatel se strojem začít co nejdříve pracovat a nemá čas na instalaci všech aktualizací.

---

<sup>1</sup> doba, která uplynula od posledního zapnutí zařízení

NinjaOne kromě instalace aktualizací operačního systému umožňuje aktualizovat i software třetích stran od vybraných společností. Mezi podporované aplikace patří produkty od firmy Adobe, Google, Cisco atd. Rovněž je umožněna instalace aktualizací vybraných antivirových řešení.

### **Multiplatformní server**

NinjaOne umožňuje podřízené stroje spravovat pomocí webové aplikace, jež funguje na všech majoritních desktopových operačních systémech (Windows, macOS i Linux). Stroje je možno spravovat i z mobilních zařízení, jelikož je nabízena aplikace pro Android i iOS.

### **Uživatelské role**

Program umožňuje definovat uživatelské role a každé roli přidělit pouze určitá oprávnění. Privilegia uživatele se následně odvíjí od role, jež je mu přidělena.

U rolí se dá specifikovat typ akcí, jež daná role může provádět (přidat / smazat / pouze zobrazit). Rovněž je možno specifikovat oblast správy, na kterou se práva vztahují. Tzn. role může mít právo přidat nový stroj, ale už nemusí mít právo pro přidání či spuštění skriptu na daném stroji.

### **Správa multiplatformních stanic**

Software umožňuje spravovat zařízení, která disponují operačním systémem Windows, Linux i MacOS. V případě distribucí Linuxu je zaručena kompatibilita s distribucemi Ubuntu, Fedora, Debian, RedHat a CentOS. Je podporováno i monitorování Hyper-V serverů. Správa zařízení je podmíněna instalací podpůrného software na spravovaná zařízení [23g].

### **Spuštění skriptů**

Software umožňuje přistupovat k terminálu spravovaných zařízení a spouštět skripty, jež vykonávají požadovanou činnost. Funkcionalita je podporována pro všechny 3 majoritní operační systémy.

### **Plán obsazení stroje**

Softwarové řešení neposkytuje vytvoření rozvrhu plánovaného obsazení stroje ani neumožňuje získat tato data od služeb třetích stran.

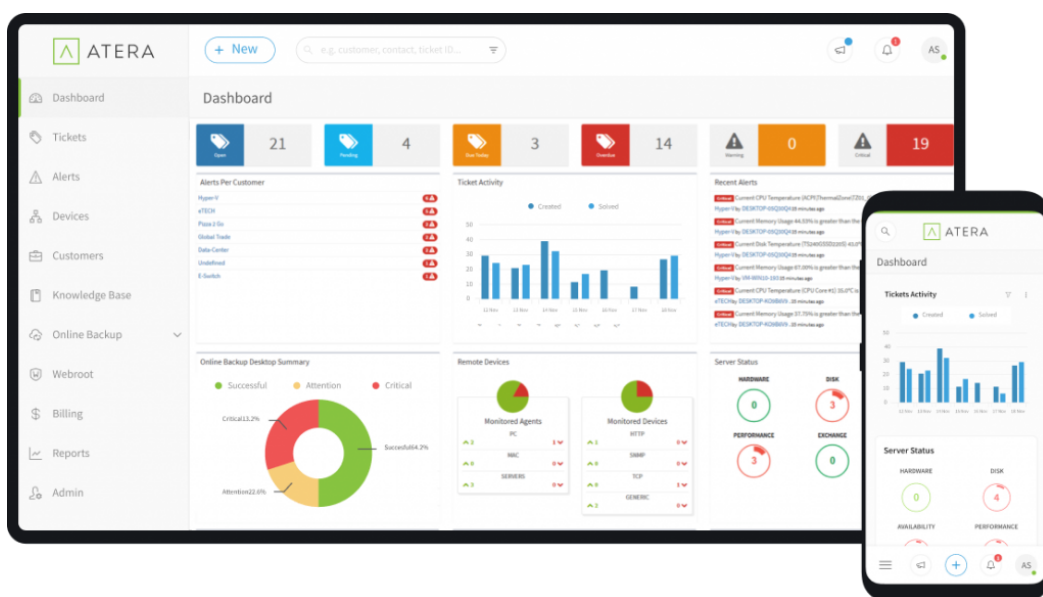
## Autentizace externím zdrojem

Aplikace umožňuje přihlášení pomocí služeb třetích stran. Podporováno je přihlášení pomocí několika SSO poskytovatelů - např. Azure, Okta, OneLogin a Duo [23m].

## Cena

Produkt nabízí zdarma 14denní testovací dobu, následně je účtován měsíční poplatek za užívání produktu. Cena licence je pro každého zákazníka (společnost) individuální a výrobce řešení neposkytuje veřejné informace o její výši. Je však známo, že cena se odvíjí od množství aktivních funkcí programu a počtu strojů, které chce společnost (zákazník) spravovat.

## 3.4 Atera



Obrázek 3.1: Ukázka uživatelského rozhraní Atera

Druhý nejvyužívanější produkt své kategorie, dle webů SourceForge [23c] a G2 [23b]. Ukázku GUI daného produktu je možno vidět na obrázku 3.1.

Domovská stránka produktu: <https://www.atera.com/>.

## 3.4.1 Hodnocená kritéria

### Zapnutí stroje

Aplikace umožňuje vzdálené zapnutí stroje. Zapnutí samozřejmě funguje pouze za předpokladu, že je technologie Wake-on-LAN podporována a povolena spravovanou stanicí. Výrobce udává, že pro využití funkce Wake-on-LAN se musí server i spravovaná stanice nacházet v rámci stejného subnetu sítě [23ab].

### Zjištění dostupnosti

Dostupnost zařízení je periodicky kontrolována, aktuální stav zařízení je znázorněn barevným obrazcem nacházejícím se vedle názvu zařízení. Pokud je obrazec zelený, pak je zařízení dostupné a je možno se k němu připojit. Červený obrazec značí nedostupnost zařízení.

### Informace o stroji

Produkt umožňuje získat informace o procesoru, RAM, disku a dalších komponentách stanice. V případě notebooků je možno dokonce získat počet cyklů baterie a její aktuální kapacitu.

Stejně jako předchozí hodnocený produkt, umožňuje Atera získat informace o aktuálním vytížení systému. Získaná data o taktu CPU a zabrané paměti RAM se průběžně archivují. Z dat jsou následně vytvořeny grafy, ze kterých lze vyčíst vytížení systému v průběhu času.

### Aktualizace operačního systému

Atera umožňuje aktualizovat operační systém spravované stanice. Instalaci aktualizací je možno vyvolat nad spravovanou stanicí jednorázově či nastavit automatickou periodickou kontrolu a instalaci aktualizací. Stejně jako u konkurenčních produktů, je možno instalovat i podmnožinu dostupných aktualizací.

Software by v budoucnu měl umožňovat instalaci aktualizací vybraných produktů třetích stran. Uvedená funkcionality je prozatím pouze v betaverzi.

### Multiplatformní server

Softwarové řešení poskytuje webovou aplikaci, kterou je možno provozovat na všech běžných desktopových operačních systémech (Windows, Linux, MacOS).

Poskytovány jsou i aplikace pro mobilní operační systémy Android a iOS, nejsou ovšem kladně hodnoceny komunitou. Verze pro iOS dosahuje průměrného hodnocení 2 z 5 bodů, verze pro Android pak 2,2 / 5. V obou případech si uživatelé stěžují na neodladěnost aplikace.

### **Uživatelské role**

Systém podporuje vytváření nových rolí, přičemž každé roli je možno přiřadit jiná práva přístupu ke spravovaným strojům.

### **Správa multiplatformních stanic**

Výrobce produktu propaguje bezproblémovou správu stanic s Windows, MacOS. Podpora Linuxových distribucí je implementována, avšak zatím je pouze ve fázi testování [23d]. Pro umožnění správy zařízení Atera vyžaduje instalaci programu na spravované stroje [23k].

### **Spuštění skriptů**

Produkt podporuje spuštění skriptů na spravovaných stanicích. Skripty lze spustit jednorázově nebo naplánovat jejich periodické spuštění v určitý čas.

### **Plán obsazení stroje**

Atera integruje práci s Google Kalendářem a Office 365 kalendářem. Nicméně s uvedenými službami je možno interagovat pouze při vytváření ticketů v programu a nelze jejich prostřednictvím definovat obsazenost stroje.

### **Autentizace externím zdrojem**

Produkt podporuje ověření uživatele na základě Microsoft Azure Active Directory [23y].

### **Cena**

Řešení nabízí bezplatnou 30denní testovací verzi, po uplynutí zkušební doby je účtován měsíční poplatek.

Výše měsíčního poplatku se u tohoto produktu odvíjí od množství odemčené funkcionality a lokace + počtu techniků, kteří zařízení spravují. V případě, že firma disponuje IT oddělením, kde pracují zaměstnanci na plný úvazek, pohybuje se cena licence v rozmezí \$149 - \$199 za jednoho technika.

Pokud by firma používala pro správu IT externistu, který nepracuje pouze pro jednu společnost, činí měsíční poplatek \$99 - \$169 technik / měsíc.

## **3.5 LogMeIn Central**

Domovská stránka produktu: <https://www.logmein.com/central>.



## 3.5.1 Hodnocená kritéria

### Zapnutí stroje

Program podporuje vzdálené zapnutí stanic pomocí Wake-on-LAN. K zapnutí stroje dojde po klepnutí na tlačítko s nápisem „Switch On“, které se nachází vedle názvu každé vypnuté stanice.

Funkcionalita nezávisí na operačním systému, je možno vzdáleně zapnout stroje s jakýmkoliv podporovaným systémem. Cílová stanice samozřejmě musí být připojena k síti drátovým připojením, nikoliv WiFi. Vývojář produktu rovněž uvádí, že pro využití funkce Wake-on-LAN se musí server aplikace a spravovaná stanice nacházet ve stejné síti [22e].

### Zjištění dostupnosti

Produkt u každého přidaného stroje zobrazuje aktuální stav jeho dostupnosti. Pokud se zařízení stane nedostupným, je změna stavu programem detekována.

### Informace o stroji

Aplikace zobrazuje model osazeného procesoru, takt procesoru i velikost osazené paměti RAM. Software je schopný, stejně jako ostatní analyzované produkty, zobrazit aktuální využití systému - aktuální takt procesoru a velikost obsazené RAM.

Vyjma uvedených základních informací je software schopný získat data o základní desce počítače a síťových rozhraních. Produkt v rámci dražších variant předplatného nabízí i periodické ukládání hardwarové konfigurace stanice [22c]. Získané informace je následně možno využít pro kontrolu, že se konfigurace stroje v čase nezměnila.

### Aktualizace operačního systému

Softwarové řešení nabízí aktualizaci operačního systému spravovaných stanic. Funkcionalita ovšem není přítomna v nejnižší verzi předplatného produktu, jedná se o příplatkový doplněk.

Jsou podporovány i aktualizace vybraných produktů třetích stran - např. Notepad++, Adobe Reader a Mozilla Firefox. Umožněny jsou i aktualizace ostatních produktů výrobce, mezi které patří LogMeIn Rescue a LogMeIn Hamachi.

### Multiplatformní server

Aplikaci Central je možno nainstalovat na Windows 7 a novější, MacOS 10.13 či novější a Windows Server 2008R2. Linux, ChromeOS a Windows RT nejsou podporovány [22b].

Software kritérium splňuje - aplikaci pro ovládání podřízených strojů lze provozovat na více platformách. Nicméně absence podpory Linuxových strojů je u aplikací podobného charakteru neobvyklá a ostatní hodnocené produkty správu z Linuxu umožňují.

#### **Uživatelské role**

Systém umožňuje u každého uživatele specifikovat oblasti správy zařízení, do kterých bude mít osoba přístup. Rovněž je možno specifikovat i typ akcí, které může daný uživatel v oblasti vykonávat (vytváření, mazání atp.).

#### **Správa multiplatformních stanic**

Produkt umožňuje spravovat cílová zařízení s operačním systémem Windows 7 či novějším, MacOS 10.13 nebo novějším a Windows Server 2008R2. Zařízení s jiným operačním systémem nejsou podporována. Pro zpřístupnění správy zařízení je vyžadováno nainstalovat na spravovaná zařízení podpůrný software (tzv. „agenta“) [22f]. Seznam operačních systémů, na které je možno nainstalovat agenta, je tedy shodný se seznamem systémů, ze kterých lze podřízené stroje ovládat.

#### **Spuštění skriptů**

Produkt umožňuje na spravovaných zařízeních spouštět skripty s libovolným obsahem. Jediné omezení je kladeno na velikost souborů, které jsou ve skriptu referencovány. Maximální velikost takového souboru je 1GB.

Central umožňuje jednorázové provedení skriptů i jejich automatické opakované spuštění.

#### **Plán obsazení stroje**

Central nenabízí vytvoření rozvrhu obsazení stroje žádnou formou. Rovněž není možno využít kalendářové služby třetích stran.

#### **Autentizace externím zdrojem**

Produkt umožňuje ověření uživatele pomocí SSO (Single sign on). Jedná se o techniku, kdy je uživateli umožněno přihlásit se k více systémům pomocí jednotných přihlašovacích údajů. Podporováno je například přihlášení pomocí Microsoft Azure Active Directory [22k].

## Cena

Produkt nabízí 14denní testovací verzi, jež je zdarma dostupná. Cena licence je, na rozdíl od ostatních analyzovaných produktů, jednotná pro všechny zákazníky a odvíjí se pouze od počtu spravovaných strojů [230]. Minimální počet strojů, pro který lze licenci zakoupit, je 25 a cena v takovém případě činí \$80. Cena je uváděna za měsíční období, jedná se tedy o pravidelný poplatek. Nejvyšší počet spravovaných strojů, pro který je produkt veřejnosti poskytován, je 250 a pravidelný poplatek činí \$250.

Uvedené ceny zahrnují pouze produkt o základní funkcionalitě, firma nabízí za příplatek rozšíření v podobě balíčků. Nabízen je například balíček související se zabezpečením, který umožní vzdálenou aktualizaci operačního systému a antiviru. Cena těchto balíčků se odvíjí od počtu stanic, pro který byl produkt zakoupen. Čím více stanic je spravováno, tím dražší je přikoupení balíčků.

## 3.6 Zhodnocení funkcionality stávajících řešení

Předmětem tabulky 3.1 je zhodnocení přítomnosti analyzovaných funkcí u jednotlivých programů.

	NinjaOne	Atera	LogMeIn Central
zapnutí stroje	✓	✓	✓
zjištění dostupnosti	✓	✓	✓
informace o stroji	✓	✓	✓
aktualizace operačního systému	✓	✓	✓
multiplatformní server	✓	✓	✓
uživatelské role	✓	✓	✓
správa multiplatformních stanic	✓	✓	✓
spuštění skriptů	✓	✓	✓
plán obsazení stroje	✗	✗	✗
autentizace externím zdrojem	✓	✓	✓
dostupné zdarma	✗	✗	✗

Tabulka 3.1: Zhodnocení funkcionality testovaných produktů

## 3.7 Chybějící funkcionality existujících řešení

Na základě provedené analýzy produktů jsem se rozhodl vytvořit nové řešení pro univerzitní správu strojů. Existující programy pro správu se zaměřují spíše na korpo-

race a tudíž mnohdy neobsahují funkcionalitu, jež je žádoucí ve školství. Zkoumané produkty disponují celou řadou funkcionalit užitečných pro korporace, avšak nejsou zcela vyhovující pro naše specifické potřeby a nasazení na univerzitní půdě.

Žádný z analyzovaných produktů navíc nesplňuje žádoucí funkci, jež jsem definoval jako „Plán obsazení stroje“. Ve firemním prostředí je obvyklé, že každý z uživatelů má svůj vlastní stroj a nedochází ke sdílení zařízení. Není tedy důvod, proč by konkurenční produkty obsahovaly nějakou formu rozvrhu obsazenosti stroje. Nicméně v univerzitním prostředí, kdy jsou počítače využívány během výuky, je sdílení strojů běžné a uvedená funkcionalita je žádoucí.

Spolupráce s rozvrhovou aplikací je žádoucí například kvůli možnosti automaticky aktualizovat software určité učebny v době, kdy není obsazena. V případě poruchy hardware některého ze strojů může být funkce využitelná i pro plánování fyzických zásahů počítačového technika. Řešení kompatibilní s rozvrhovým systémem by mohlo technikovi ušetřit čas strávený zjišťováním, zda je místnost, ve které se porouchaný počítač nachází, volná. V neposlední řadě by se rozvrhová data dala využít ke kontrole akcí uživatele řešení a případnému upozornění na potenciálně nežádoucí akci. Např. pokud by se technik pokoušel spustit aktualizace operačního systému v laboratoři, v níž právě probíhá výuka, mohl by systém zobrazit upozornění a požádat o potvrzení akce.

Cena stávajících řešení je dalším důvodem, proč bych chtěl vytvořit nový systém pro správu. Existující řešení jsou sice komplexní, ale požadují platbu na měsíční bázi, přičemž cena roste spolu s dostupnými funkcemi.

Některé funkcionality, jež poskytují placená řešení pro správu strojů, nabízí i zdarma dostupné nástroje. Například zapnutí stroje umožňuje aplikace WakeMe-OnLan (viz [https://www.nirsoft.net/utils/wake\\_on\\_lan.html](https://www.nirsoft.net/utils/wake_on_lan.html)). Nástroje dostupné zdarma jsou však jednoúčelové a chybí komplexní řešení, jež by bylo snadno aplikovatelné v univerzitním prostředí.

Předmětem následujících podkapitol je popis funkcí, jež bych chtěl implementovat ve svém řešení.

# Výběr vhodné funkcionality

## 4

Tato kapitola se zabývá popisem funkcí, které plánuji ve svém řešení implementovat. U všech zmíněných funkcí je naznačen zvažovaný způsob implementace v rámci plánovaného softwarového řešení.

## 4.1 Funkce plánovaného řešení

### Zapnutí stroje

Prostřednictvím nově vytvořené aplikace by mělo být možno zapnout zařízení, jež se nachází na totožném segmentu sítě jako stroj, na kterém je provozován server řešení. Předpokládá se, že na serveru bude provozována obslužná webová aplikace, ke které bude možno přistoupit z jiných počítačů či mobilních telefonů a zajistit tak zapnutí stroje pomocí zařízení, jež se nachází mimo segment sítě. Server po obdržení požadavku na zapnutí stroje pošle na cílový stroj tzv. „Magic packet“, viz kapitola 2.9.1.

### Zjištění dostupnosti

Je žádoucí, aby byl uživatel prostřednictvím řešení schopen zjistit stav dostupnosti stroje. Testování přítomnosti stroje na síti by mohlo být v rámci jazyka Java prováděno pomocí metody `isReachable`, jež nabízí třída `InetAddress`. Metoda `isReachable` požaduje pouze jeden argument, kterým je čas v milisekundách, jenž je vyhrazen pro kontaktování vzdáleného stroje. Pokud dojde k obdržení odpovědi od vzdáleného stroje v rámci času specifikovaného argumentem metody, je vzdálený počítač dostupný. Uvedená Java metoda pro zjištění dostupnosti stroje primárně využívá „Echo request“ protokolu ICMP. V případě, že dotazované zařízení neodpoví na „Echo request“, pokusí se metoda ověřit dostupnost cílového stroje otevřením TCP spojení na portu 7 [10].

## Informace o stroji

Práce by měla implementovat funkcionality, která umožní uživateli získat základní informace o stroji. Získaná data by měla zahrnovat zejména datum instalace operačního systému, verzi systému BIOS, model procesoru, velikost obsazené paměti RAM a případně další údaje stanice. Pro získání potřebných dat o stroji bude vytvořen PowerShell skript, pro jehož vzdálené spuštění je vyžadováno využití SSH.

## Aktualizace operačního systému

Přítomnost dané funkcionality je žádoucí, jelikož aktualizace operačního systému je mnohdy časově náročný proces a může uživateli znepříjemňovat práci s počítačem. Aby se omezilo riziko samovolného aktualizování stroje v průběhu práce uživatele, je vhodné periodicky spouštět proces aktualizace operačního systému v době, kdy počítač není aktivně využíván.

Pro správu aktualizací systému Windows v rámci řešení bude využito PowerShell rozšíření s názvem „PSWindowsUpdate“. Dané rozšíření bude v nově vytvořeném systému použito i pro získání seznamu provedených aktualizací. Seznam několika posledních provedených aktualizací systému může být žádoucí získat například v případě, že některá z nainstalovaných aktualizací způsobí nestabilitu systému. Produkt PSWindowsUpdate lze získat ze stránky <https://www.powershellgallery.com/packages/PSWindowsUpdate>.

## Multiplatformní server

Jedním z klíčových výsledků této práce bude server, který bude možno provozovat na vícero platformách. Se serverem bude možno interagovat prostřednictvím platformně nezávislé webové aplikace a mobilní aplikace pro systém Android. Pro zajištění snadného nasazení celého řešení a zajištění multiplatformního návrhu bude využit nástroj Docker.

## Uživatelské role

Uživatelé aplikace budou řazeni do skupin, přičemž každá skupina uživatelů bude mít v systému jiná práva přístupu ke spravovaným strojům. Nově vytvořený produkt bude zavádět minimálně dvě role. Nezbytná je přítomnost role administrátora, jenž bude mít přístup ke všem seznamům, resp. strojům v systému. Dále bude přítomna role běžného uživatele, jejíž vlastník bude moci přistupovat pouze ke strojům, jež mu byly přiděleny administrátorem.

Stroje v systému budou tříděny do seznamů a každá z rolí bude oprávněna vytvořit seznam. Přímé přidání nového stroje do některého ze seznamů bude oprávněn

provést pouze administrátor. Běžný uživatel může podat pouze žádost o přidání stroje do některého ze seznamů, jež sám vytvořil. Každá z žádostí, které jsou podány běžným uživatelem, musí být následně schválena či zamítnuta administrátorem systému.

## Správa multiplatformních stanic

Softwarový produkt bude určen primárně na správu zařízení s operačním systémem Windows, avšak bude poskytovat i některé funkce umožňující správu Linuxových distribucí.

## Spuštění skriptů

Aplikace by měla umožňovat spuštění skriptů dodaných uživatelem na spravovaných počítačových stanicích. Jelikož schopnost spouštět vlastní skripty může být snadno zneužita ke škodlivé činnosti, bude využití této funkcionality umožněno pouze privilegovaným uživatelům systému.

Možnost spuštění vlastních skriptů umožní snadné rozšíření funkcionality poskytované systémem. Dodané skripty může uživatel využít například pro obsluhu vlastního software, pro které řešení neposkytuje vestavěnou funkcionality.

## Export a import informací

Nově vytvořené řešení by mělo umožňovat export informací o strojích, jež jsou v rámci řešení pro správu strojů spravovány. Rovněž je žádoucí, aby softwarový produkt implementoval funkcionality umožňující následný import dat zpět do systému.

Vytvořený produkt bude umožňovat export dat všech zařízení, jež se nachází v rámci seznamu zařízení. Data o strojích budou ukládána do souboru typu csv, který bude mít následující hlavičku:

- název - název stroje definovaný uživatelem,
- MAC - MAC adresa spravovaného stroje,
- IP - IP adresa spravovaného stroje,
- hostname - hostname spravovaného stroje.

Import dat bude mít aditivní charakter. Pokud budou importována data do seznamu, který již některé stroje obsahuje, nedojde k nahrazení těchto strojů.

## Autentizace externím zdrojem

V rámci nově vytvořeného řešení bude ověření identity uživatele prováděno externím systémem. Využití externího nástroje pro autentizaci je vhodné, jelikož si uživatel nebude muset pamatovat další přihlašovací údaje. Nově vytvořený systém správy strojů rovněž nebude muset uchovávat hesla uživatelů a řešit s tím spojené bezpečnostní aspekty, například vhodné metody šifrování hesel.

V univerzitním prostředí, ZČU nevyjímaje, je pro ověření identity uživatelů často používán protokol Kerberos. Uvedený protokol bude využit v novém systému správy strojů pro autentizaci, uživatel se tedy bude v rámci systému prokazovat svým Orion loginem a příslušným heslem. Bližší popis protokolu Kerberos je předmětem kapitoly 2.8.1.

## 4.2 Mobilní aplikace

Součástí řešení bude i mobilní klient pro platformu Android, který bude poskytovat podmnožinu funkcí uvedených v kapitole 4.1. V případě mobilní aplikace je žádoucí umožnit komunikaci se serverem prostřednictvím programového rozhraní (API), jež zajistí snížené nároky na datové připojení oproti využití webové aplikace. Jelikož mobilní klient musí vyžadovat pro správu univerzitních strojů kontakt se serverem, měla by mobilní aplikace umožňovat uživateli uložit IP adresy a porty používaných serverů. Z aplikačního serveru budou získávána data související se spravovanými stanicemi, mezi které patří například uživatelské označení počítače a IP adresa stroje. Klient bude na aplikační server odesílat požadavky, kterými zajistí provedení určité akce nad specifickým strojem. Požadovanou akcí může být například zapnutí stroje či získání seznamu právě přihlášených uživatelů na určitém stroji.

Mobilní verze řešení by mohla poskytovat i funkcionalitu, jež by umožňovala zapnutí strojů v síti, ke které je uživatel připojen. Pro zapnutí stroje, resp. poslání speciálního „Magic packetu“ ze systému Android je irrelevantní, zda je odesílatel paketu připojen pomocí Ethernetu či bezdrátového připojení. Připojení běžných mobilních zařízení k síti pomocí Ethernetu není běžné, ale může být využito u specializovaných přístrojů, kterými jsou například TV boxy či platební terminály.



# Návrh webového řešení

## 5

Podkapitoly níže uvedené popisují zvažované způsoby implementace určitých částí mnou vyvíjeného systému pro správu počítačů určeného do univerzitního prostředí, zejména ZČU KIV.

### 5.1 Přenositelnost řešení

Je žádoucí, aby server webové aplikace bylo možné provozovat na co možná nejširším spektru dostatečně výkonných zařízení bez ohledu na jejich softwarovou výbavu. Pro usnadnění spuštění serveru aplikace bylo nutné zajistit co největší nezávislost na softwarovém vybavení hostujícího stroje.

#### 5.1.1 Software vyžadovaný řešením

Množství softwarového vybavení, jež je potřebné pro běh serveru, se odvíjí od technologií, se kterými server pracuje. Aplikační server pro svou funkcionalitu bude vyžadovat minimálně dvě základní komponenty, kterými jsou webový a databázový server. Primárním účelem webového serveru je doručení webových stránek či jiného obsahu uživatelům. K vytvoření webového serveru lze využít mnoho technologií, například: Apache, Nginx, lighttpd, Apache Tomcat a Microsoft IIS.

Databázový server bude v aplikaci nutné použít, jelikož aplikace musí uchovávat informace o strojích, jež jsou spravovány. Pro uložení dat bude vhodné využít některou z relačních databází, důvody jsou uvedeny v sekci 5.2.3. Příklady relačních databází jsou: MySQL, Microsoft SQL Server, Oracle Database, PostgreSQL a IBM DB2.

#### 5.1.2 Docker

Po instalaci webového a databázového serveru je třeba provést jejich konfiguraci, která může být mnohdy obtížná a časově náročná. Proces instalace a konfigurace navrhovaného řešení by mohl být zjednodušen využitím nástroje Docker, jehož

využití by zároveň zvýšilo nezávislost na platformě a s tím spojenou přenositelnost řešení.

Docker je nástroj, který umožňuje součásti aplikace distribuovat v tzv. „kontejnerech“, přičemž kompletní řešení se může skládat z několika kontejnerů. Obsahem kontejneru jsou všechny komponenty, jež daná aplikace potřebuje k běhu. Součástí kontejneru může být například zkompileovaný kód, knihovny, konfigurační nastavení a další vyžadované součásti, jež závisí na konkrétní aplikaci.

Kontejner je vytvořen z tzv. „image“, který lze definovat jako předlohu aplikace. Při procesu vytváření kontejneru je nutné poskytnout konfiguraci, na základě které bude image upravena a vytvořen výsledný kontejner [Tur14]. Například pro spuštění webového serveru Apache Tomcat lze využít image označený „tomcat“ a využít příkazy, které provedou úpravu image [23aa]. Požadovanou úpravou image webového serveru je typicky nahrání vlastní webové aplikace a vykonání skriptu, jež provede spuštění serveru.

Příkazy modifikující image se zapisují do souboru s označením „Dockerfile“, který se nachází ve složce s aplikací, jež má být distribuována pomocí systému Docker. Pro přidání souborů do kontejneru se používá příkaz `ADD`, jež vyžaduje dva argumenty. Prvním argumentem je původní umístění souboru, jež má být do kontejneru vložen, a druhým argumentem je požadované umístění souboru v rámci kontejneru. Dalším často používaným příkazem je `CMD`, který slouží pro spuštění příkazu v prostředí kontejneru po jeho spuštění. Parametry příkazu `CMD` jsou uvedeny ve formě pole obsahujícího textové řetězce, přičemž první řetězec obsahuje cestu ke spouštěnému programu. Následující textové řetězce v poli představují argumenty příkazu, jež je spouštěn v rámci kontejneru [Tur14]. Kompletní dokumentace příkazů, jež umožňují modifikaci, je dostupná na webu vývojáře: <https://docs.docker.com/engine/reference/builder/>.

Sestavení celého distribuovaného řešení probíhá pomocí systému „Docker compose.“ Konfigurace kompletního řešení je uvedena v souboru `docker-compose.yml`, který je obvykle umístěn v kořenové složce projektu určeného k distribuci. Uvedený soubor se může referovat na více aplikací (Dockerfile), jež mají být v rámci projektu sestaveny. V případě, že je dostupný soubor `docker-compose.yml`, stačí pro sestavení projektu na hostujícím stroji vykonat příkaz `docker compose up`. Dokumentace systému Docker compose je dostupná na stránce vývojáře produktu: <https://docs.docker.com/compose/>.

Navrhované řešení by mělo být nasaditelné prostřednictvím systému Docker, jelikož jsou dostupné image systémů relační databáze. Například databáze MySQL je dostupná v rámci image „mysql“ [23u]. Jak již bylo uvedeno, je rovněž poskytován image pro webový server Apache Tomcat, který nese označení „tomcat“. Docker tedy poskytuje image pro všechny základní komponenty systému, které v navrhovaném řešení figurují.

Docker je dostupný pro všechny majoritní desktopové operační systémy (Windows, macOS i Linux). Stažení software je možné z oficiálních webových stránek výrobce: <https://www.docker.com/products/docker-desktop/>.

## 5.2 Ukládaná data

Plánované řešení musí v persistentním úložišti uchovávat informace o spravovaných strojích a uživateli, jež mají mít do systému správy přístup.

### 5.2.1 Počítačové stanice

U spravovaných strojů musí být uchována minimálně jejich MAC adresa a IP adresa. MAC adresa je potřebná, aby bylo možné spravovaný počítač vzdáleně zapnout či probudit pomocí technologie Wake-on-LAN (viz kapitola 2.9). IP adresu je nutné uchovávat pro umožnění vzdálené správy a provádění operací v rámci operačního systému, který je na spravovaném stroji provozován. Pro vzdálené spuštění skriptů, jež zajišťují provedení požadovaných akcí, bude využita technologie SSH. IP adresa je uchována, jelikož technologie SSH tento identifikátor vyžaduje pro vytvoření spojení s cílovým strojem. Bližší popis technologie SSH je předmětem kapitoly 2.5.

Jelikož je žádoucí, aby navrhovaný systém mohl být použit i pro účely evidence strojů, budou v rámci řešení uchovávány i další informace, které mohou pomoci s identifikací stroje. Jednou z volitelných informací, jež by mohly být stroji přiděleny, je název hostitele stroje. Daný identifikátor je v českých textech mnohdy označován anglickým originálem „hostname“. Název hostitele je textový řetězec s maximální délkou 253 znaků, jež představuje jednoznačný identifikátor stroje v rámci počítačové sítě.

### 5.2.2 Uživatelé

Řešení, které je určeno primárně pro správu stanic v rámci univerzity, potřebuje uchovávat pouze minimum informací o uživateli. Je ovšem nutné, aby byli uživatelé systému identifikovatelní, pokud by bylo zjištěno zneužití systému a bylo potřeba nalézt viníka. K identifikaci v prostředí ZČU může sloužit Orion login (viz kapitola 5.4). Každý uživatel řešení by se tedy mohl do systému správy stanic přihlašovat svým Orion loginem a heslem, kterým se prokazuje v rámci ostatních univerzitních systémů.

Pokud by plánované řešení bylo napojeno na systém Kerberos (viz kapitola 2.8.1), nebylo by nutné uchovávat uživatelská hesla v rámci navrhovaného řešení. Ověření hesla externím zdrojem je vhodné, jelikož není nutné řešit šifrování spojené s uložením uživatelského hesla do vlastní databáze. K jednoznačné identifikaci uživatele

lze tedy využít pouze Orion login, na základě kterého lze poté v univerzitních systémech dohledat více informací o osobě, jež má daný login přidělen. Více informací není potřeba uchovávat a vzhledem k určení systému to ani není žádoucí.

### 5.2.3 Forma uložení dat

Data, jež bude řešení potřebovat ukládat, jsou řádně strukturována a mají mezi sebou jednoznačně definované vztahy. Každá uchovávaná informace má jednoznačně definovaný datový typ, který je pro všechny spravované stroje stejný. Vzhledem k uvedeným faktům bude vhodné pro ukládání dat zvolit některou z relačních databází, které umožňují uložení strukturovaných dat do tabulek a vyjádření vztahů mezi daty pomocí cizích klíčů.

Z předcházejících dvou kapitol (5.2.2 a 5.2.3) je zjevné, že návrh databázového modelu bude obsahovat minimálně dvě tabulky. Jedna tabulka bude vyčleněna pro uchovávání uživatelských dat, druhá tabulka bude obsahovat informace o spravovaných strojích. Další tabulky mohou být v modelu obsaženy za účelem rozšíření funkcionality systému.

Plánovaný systém bude rovněž poskytovat i možnost třídění uložených strojů do seznamů, pro tento účel budou v databázi další dvě tabulky. Jedna tabulka bude reprezentovat seznam strojů. Jelikož by v budoucnu mohlo být řešení rozšířeno o možnost přiřazení jednoho stroje do více seznamů, je v návrhu i tabulka pro realizaci vztahu M:N, jenž by mezi stroji a seznamy vznikl.

Pokud by například bylo žádoucí rozšířit plánovaný systém o možnost třídění uložených strojů do seznamů, bylo by vyžadováno rozšíření databázového modelu o další dvě tabulky. Jedna tabulka by reprezentovala seznam strojů a jelikož seznam může obsahovat více počítačů, jedná se o typický příklad relace typu M:N. Vztah typu M:N mezi stroji a seznamy by byl řešen pomocí rozkladové tabulky.

## 5.3 Technologie

Pro vytvoření webové, respektive serverové části řešení, bude vhodné využít některý z dostupných frameworků pro tvorbu webových aplikací.

Využití frameworku vede k rychlejšímu vývoji řešení, jelikož framework poskytuje hotová řešení pro mnoho běžných problémů, které se ve spojitosti s tvorbou webové aplikace vyskytují. Frameworky například poskytují standardizované nástroje, které umožňují snadné připojení aplikace k databázi [Cla23]. Využití standardizovaných nástrojů pro připojení k databázi je vhodné i z důvodu případné migrace aplikace na jiný typ databáze. Pokud by byl veškerý kód pro připojení k databázi psán programátorem, bez využití frameworku, bylo by pravděpodobně pro

přechod na jiný typ databáze nutno modifikovat větší množství kódu než při využití frameworku.

Frameworků pro tvorbu webových projektů je mnoho a volba konkrétního frameworku závisí především na preferenci vývojáře a dalších technologiích, které mají být v rámci řešení využity. Každý framework je založen na určitém programovacím jazyce a pokud je plánovaná webová aplikace závislá na některém z jazyků, je vhodné pro tvorbu aplikace využít framework psaný v daném jazyce. Pokud se shoduje jazyk frameworku a plánované aplikace, odpadá například nutnost provozu další technologie, ve které je aplikace napsána.

Vzhledem k výše uvedeným důvodům plánuji pro vytvoření webové části aplikace využít Java framework, jelikož webová aplikace bude psána v jazyce Java. Využití zmíněného jazyka je výhodné i z toho důvodu, že součástí řešení bude mobilní klient pro platformu Android. Oficiálním nástrojem pro nativní vývoj Android aplikací je Android Studio, jež umožňuje vývoj pomocí jazyka Kotlin a Java [23r]. Využití programovacího jazyka Java pro vývoj webového řešení tedy může být výhodné i z toho důvodu, že v rámci celého řešení bude využit pouze jeden jazyk. Sjednocení jazyka v rámci celého řešení může vést ke snazšímu debugování řešení, případně ke snazšímu rozšiřování funkcionality řešení v budoucnu.

Mezi populární Java frameworky, jež zjednodušují tvorbu webových řešení, se řadí například Spring, Play, Tapestry, Dropwizard a Apache Struts [Kri18]. Pro vytvoření serverové části řešení a obslužné webové aplikace plánuji využít nadstavbu frameworku Spring, která nese pojmenování Spring Boot [Stř23a]. Komponenty frameworku Spring Boot je možno konfigurovat pomocí nástroje Spring Initializr, jenž je dostupný na adrese <https://start.spring.io/> a získat tak framework, jenž je na míru přizpůsobený potřebám řešeného projektu.

Součástí nabízených modulů frameworku Spring Boot je například balík označený „Spring Boot Starter Web“, který poskytuje technologie pro tvorbu webových řešení. Uvedený balík obsahuje nástroje zjednodušující tvorbu RESTful rozhraní a nakonfigurovaný server Apache Tomcat, jehož je možno využít k nasazení webové obslužné aplikace [23q]. Modul Spring Web bude v nově vytvořeném řešení pro správu strojů využit zejména pro tvorbu aplikačního rozhraní a pro nasazení obslužné aplikace prostřednictvím Apache Tomcat. Další modul, který bude vhodné v plánovaném řešení využít, je šablonovací systém „Thymeleaf“. Komponenta Thymeleaf bude v projektu využita pro správu šablon obslužné webové aplikace. Jelikož je žádoucí, aby nově vytvořené řešení pro správu strojů umožňovalo autentizaci uživatele pomocí protokolu Kerberos, musí řešení implementovat modul „Spring Security“.

## 5.4 Identifikace uživatelů

V rámci Západočeské univerzity v Plzni je jednoznačným identifikátorem zaměstnanců univerzity tzv. Orion login. Uvedený identifikátor si každý zaměstnanec volí sám, obvykle ihned při svém nástupu na univerzitu. Jednotlivec následně dokazuje svou totožnost v univerzitních systémech znalostí Orion loginu a jím definovaného hesla [22h].

Za jednoznačný identifikátor zaměstnance se v rámci ZČU pokládá i zaměstnanecké číslo, což je řetězec skládající se z písmen a čísel. Ve své aplikaci neplánuji pro identifikaci uživatelů použít daný identifikátor, jelikož se může za dobu působení osoby na univerzitě změnit. Změna může nastat při přestupu na jinou katedru či fakultu.

Dalším údajem, dle kterého může být zaměstnanec ZČU jednoznačně identifikován, je číslo JIS karty. Uvedeným identifikátorem se však zaměstnanci neprokazují v ostatních univerzitních systémech (např. IS/STAG). V případě zavedení tohoto identifikátoru v nově vytvořeném systému by si uživatelé museli pamatovat další údaj, což není žádoucí.

Vzhledem k uvedeným skutečnostem bude pro autentizaci uživatelů v rámci navrhovaného systému využít Orion login a s ním spojené heslo.

Orion login uživatele se bez explicitní žádosti majitele nemění. Ověření pomocí Orion loginu je vhodné i proto, že ho k autentizaci používají i elektronické informační zdroje ZČU [22h]. Uživatel by tedy mohl pro přihlášení do navrhovaného systému použít stejné přihlašovací údaje jako do elektronických systémů ZČU.

Aplikace musí integrovat autentizaci uživatelů pomocí protokolu Kerberos, jelikož ověření v systému Orion je založeno na uvedeném protokolu [22d]. Implementace ověření uživatele by neměla být příliš složitá, jelikož mnoho webových frameworků obsahuje moduly pro práci se zmíněným protokolem, které integraci usnadňují.

Například pro framework Spring, který plánuji využít v serverové části aplikace, existuje rozšíření „Spring Security Kerberos“, jež plní uvedenou funkcionalitu. Ověření uživatele pomocí Kerberos protokolu je možné realizovat i na úrovni programovacího jazyka Java, bez využití externích frameworků. To je možné díky balíku JAAS (Java Authentication and Authorization Service). Navržená aplikace bude využívat rozšíření „Spring Security Kerberos“, jelikož se snadno integruje s ostatními bezpečnostními mechanismy frameworku Spring.

# Návrh mobilního klienta

## 6

V rámci této práce bude vytvořen mobilní klient umožňující obsluhu spravovaných zařízení, jež bude pro poskytnutí plné funkcionality vyžadovat připojení k aplikačnímu serveru. Předmětem následujících podkapitol je popis zamýšleného způsobu implementace mobilní aplikace.

## 6.1 Volba platformy

Mobilní klient bude vytvořen pro operační systém Android, jenž je vyvíjen firmou Google. Daná platforma byla zvolena zejména pro její rozšířenost na poli mobilních zařízení. Android je dlouhodobě nejpoužívanějším mobilním operačním systémem, což dokládají dostupná data z období let 2009 - 2022 udávající rozšíření mobilních OS [Tay23].

Volba platformy Android namísto iOS je vhodná i z hlediska množství a ceny vybavení, jež je vyžadováno pro vytvoření produktu. Nativní vývoj aplikací pro operační systém iOS vyžaduje iOS SDK, který je možno provozovat pouze na systému macOS. Instalaci systému macOS je oficiálně možno provést pouze na počítače od firmy Apple a pro vývoj nativní iOS aplikace je tak nutné vlastnit počítač vytvořený společností Apple [23p]. Rovněž je vhodné vlastnit i mobilní zařízení od společnosti Apple, jež by umožnilo testování vyvíjené aplikace. U mobilního operačního systému firmy Apple platí, obdobně jako u desktopového OS dané společnosti, že ho lze provozovat pouze na mobilních zařízeních uvedené firmy [23s]. Řádné otestování aplikace tedy vyžaduje zakoupení telefonu či tabletu od organizace Apple.

Existují multiplatformní řešení pro vývoj mobilního software, jež umožňují vytváření aplikací pro iOS i na Windows. Jedním z nástrojů, jež umožňují multiplatformní vývoj, je například Xamarin. Nástroje umožňující vývoj aplikace na více mobilních platformách současně ovšem často neposkytují možnost konfigurace komponent, jež jsou specifické pouze pro určitou mobilní platformu. Vývoj pro více platformách současně tak často neumožňuje využít plný potenciál všech operačních systémů, pro které je aplikace určena.

Nativní Android aplikaci je možno vytvořit pomocí vývojového prostředí Android Studio od organizace Google. Dané vývojové prostředí je dostupné na všechny majoritní operační systémy trhu, kterými jsou Windows, macOS a Linux [23r]. Android Studio umožňuje vývoj prostřednictvím jazyka Java a Kotlin [23r]. Android program bude vytvořen v jazyce Java, jelikož autor této práce má zkušenost zejména s uvedeným programovacím jazykem. Java je rovněž zvažovaným jazykem pro tvorbu webové části projektu, jenž bude v rámci práce vytvořen, a je žádoucí udržovat konzistentní programovací jazyk napříč celým projektem. Responzivní webovou aplikaci bude možno obsluhovat z počítačových webových prohlížečů (Edge, Firefox, Chrome) a moderních mobilních prohlížečů (Safari, Chrome). Webovou verzi řešení tedy budou moci využít i uživatelé systému Apple iOS.

## 6.2 Knihovna Retrofit

Mobilní klient využívá knihovnu Retrofit pro komunikaci s aplikačním serverem. Knihovna Retrofit slouží pro usnadnění komunikace se službami poskytujícími programové rozhraní typu REST. Uvedenou knihovnu vyvíjí společnost Square a je možno ji využít při tvorbě aplikací v programovacích jazycích Java a Kotlin. Retrofit definuje programové rozhraní, resp. jednotlivé volané endpointy, pomocí rozhraní a metod. Více informací o knihovně Retrofit je možné nalézt na „Github Pages“ vývojáře: <https://square.github.io/retrofit/>

Pro každý vzdálený endpoint, k jehož volání je využita knihovna Retrofit, musí být v projektu vytvořena metoda označená odpovídajícími anotacemi. Každá metoda umožňující volání API musí mít anotaci, jejíž označení odpovídá typu požadavku, který má být na server odeslán. Každá z metod zajišťující volání vzdáleného serveru tak musí být označena jednou z 8 anotací vyjadřující HTTP požadavek, mezi které patří například @GET, @POST a @PUT. Uvedené anotace specifikující typ požadavku vyžadují jeden argument, kterým cílová URL volaného endpointu.

Pokud některý z endpointů očekává v rámci URL proměnné parametry, musí být součástí metody specifikující daný endpoint i seznam parametrů tohoto koncového bodu. Pro specifikaci URL parametrů používá Retrofit anotaci @Path, po níž je ve složených závorkách umístěn název parametru v URL endpointu. Kód 6.1 reprezentuje příklad validní metody zajišťující volání endpointu. Uvedený příklad ukazuje Retrofit metodu volající vzdálený endpoint `/api/groups/{id}`, jenž vyžaduje HTTP metodu GET a má jeden proměnný parametr označený `id`. Na uvedeném příkladu lze pozorovat, že očekávaný parametr je typu Long a data získaná ze serveru mají reprezentovat množinu instancí třídy Device.



## Zdrojový kód 6.1: Ukázka anotací knihovny Retrofit

```
1 @GET("/api/groups/{id}")  
2 Call<ArrayList<Device>> getGroup(@Path("id") Long id);
```

## 6.3 Obrázky třetích stran

Veškeré ikony, jež byly v rámci mobilní aplikace použity, jsou dostupné prostřednictvím vývojového prostředí Android Studio pod licencí Apache 2.0. Text uvedené licence je k dispozici online na webu tvůrce licence: <https://www.apache.org/licenses/LICENSE-2.0.txt>.

## 6.4 Uložení dat

Data mobilního klienta budou uchovávána v SQLite databázi, jež bude uložena v mobilním zařízení. Databáze mobilního klienta řešení pro správu strojů musí obsahovat údaje o zařízeních, jež jsou spravována v rámci lokální sítě. Jelikož je žádoucí, aby jednotlivé stroje mohly být sdružovány do skupin, musí být v databázi přítomna tabulka reprezentující seznamy počítačů. Z uvedeného návrhu vychází, že SQLite databáze bude muset disponovat minimálně dvěma tabulkami.

Jelikož aplikace pro mobilní zařízení vyžaduje pro zpřístupnění většiny funkcionality kontaktování aplikačního serveru, bude součástí databáze i tabulka uchovávající data serverů. Uložení hostname / IP adresy a portu často používaných aplikačních serverů může být žádoucí pro eliminaci nutnosti zadávání údajů při každém připojení k serveru.

Žádné další informace není potřeba na klientském zařízení uchovávat, jelikož potřebné informace o vzdálených strojích budou získávány z aplikačního serveru. Komunikace klienta a serveru bude realizována prostřednictvím programového rozhraní, což zajistí nižší náročnost na datové přenosy oproti využití webové aplikace.

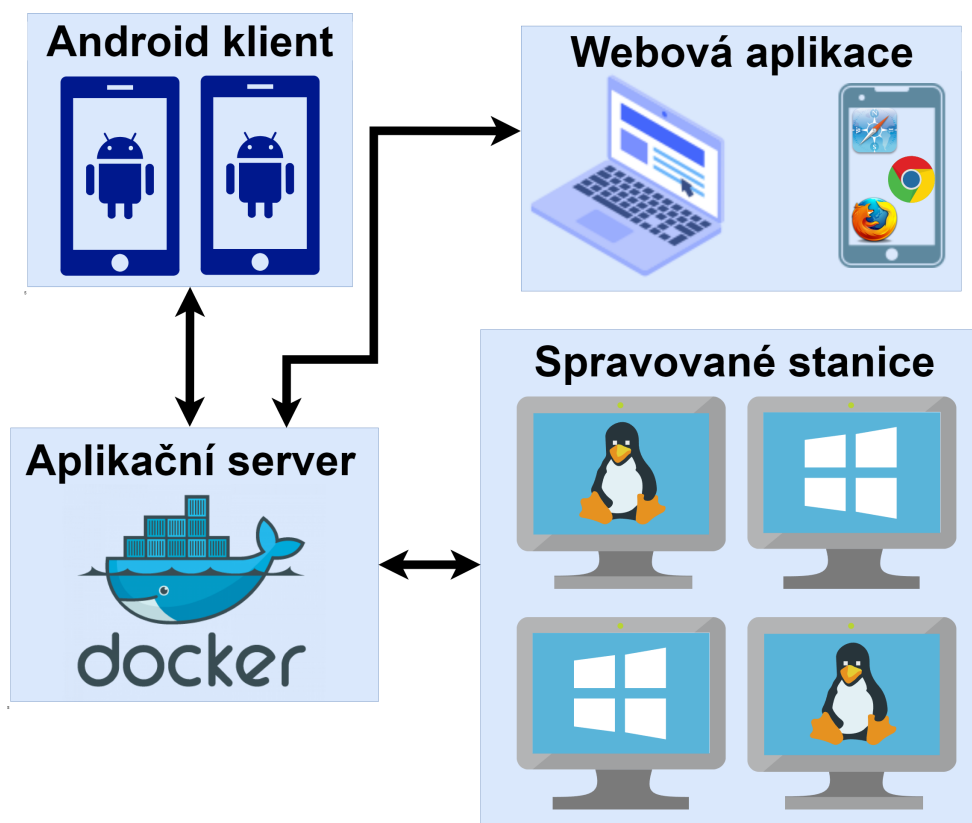


# Implementace navrženého řešení

## 7

V rámci řešení pro správu strojů byla vytvořena webová aplikace a mobilní klient pro platformu Android. Obě zmíněné komponenty komunikují s aplikačním serverem, jež zajišťuje provádění operací nad spravovanými stroji. Požadovanou operací může být například zapnutí stroje či zjištění aktuálně přihlášených uživatelů na dané stanici.

Princip fungování implementovaného systému zachycuje obrázek 7.1, jenž byl vytvořen pomocí webové aplikace draw.io (<https://app.diagrams.net/>).



Obrázek 7.1: Schéma vytvořeného systému

Následující popis implementace je rozdělen do dvou částí. Kapitola 7.1 se zabývá implementací aplikačního serveru a obslužné webové aplikace. Sekce 7.2 popisuje způsob implementace mobilního klienta, jenž byl v rámci projektu vytvořen.

## 7.1 Webové řešení

Následující podkapitoly popisují způsob implementace aplikačního serveru a obslužné webové aplikace, jež byla v rámci řešení vytvořena.

### 7.1.1 Využitý framework

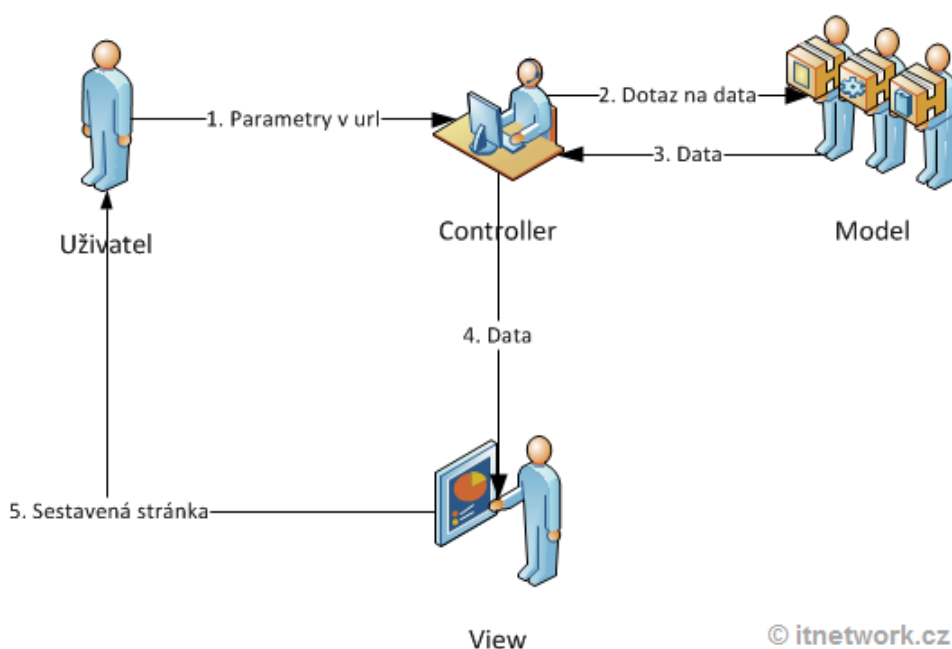
Pro vytvoření aplikačního serveru i obslužné webové aplikace byl využit framework Spring Boot. Uvedený framework poskytuje technologie vhodné pro vytvoření obou zmíněných komponent řešení. Například serverová část projektu využívá anotaci `@RestController`, která ve Spring Boot frameworku označuje třídy definující REST API. Pokud je daná anotace v některé ze tříd přítomna, zajistí framework mapování metod přítomných ve třídě na URL adresy specifikované programátorem. Programové rozhraní je v nově vytvořeném produktu pro správu stanic přítomno zejména za účelem poskytování dat mobilním zařízením, jež využívají mobilního klienta. Další informace týkající se programového rozhraní jsou uvedeny v podkapitole 7.1.3.

Jelikož je obslužná webová aplikace založena na MVC modelu, využívají se v rámci projektu některé anotace Spring Boot frameworku, jež jsou specifické pro zvolenou architekturu. Například třídy poskytující služby jsou v řešení značeny anotací `@Service`. Daná anotace zajistí, že framework po spuštění detekuje třídu poskytující určitou funkcionalitu, vytvoří její instanci a umožní její injektování do jiných částí kódu. Více informací o architektuře řešení obsahuje kapitola 7.1.2.

Využití frameworku Spring Boot vedlo ke zjednodušení kódu serveru i webové aplikace a přispělo k přehlednosti zdrojového kódu. Bližší informace zvažovaných frameworků jsou uvedeny v rámci kapitoly 5.3.

### 7.1.2 Architektura řešení

Při tvorbě webového řešení byla aplikována vrstvená architektura „MVC“, jejíž cílem je oddělit business logiku programu od jeho výstupu. Při využití uvedeného architektonického stylu je kód projektu členěn do 3 hlavních kategorií, které jsou označeny jako „model“, „view“ a „controller“. V česky psané literatuře jsou MVC komponenty mnohdy označovány jako „model“, „pohled“ a „kontroler“.



Obrázek 7.2: Popis komunikace v rámci architektury MVC [Stř23b]

Každá vrstva uvedeného architektonického návrhového vzoru poskytuje jinou funkcionalitu v rámci webové aplikace a možnosti komunikace mezi vrstvami jsou omezeny. Při odeslání klientského požadavku na server je v MVC zpravidla kontaktován kontroler, který zajistí provedení klientem požadované akce. Součástí požadavku od uživatele jsou parametry, na základě kterých kontroler detekuje konkrétní akci, jež má být vykonána. Pro vykonání požadované akce jsou využity funkcionality poskytované modelem, což zajišťuje oddělení aplikační logiky od prezentační vrstvy aplikace. Data získaná od modelu jsou následně využita k modifikaci pohledu, který zajišťuje prezentaci dat uživateli [Stř23b].

Obrázek 7.2 znázorňuje princip komunikace v rámci řešení využívající architekturu MVC. Následující kapitoly obsahují obecný popis jednotlivých MVC vrstev. Pro každou vrstvu jsou rovněž uvedeny příklady tříd z nově vytvořeného řešení, jež plní účely odpovídající vrstvy.

## Model

Třídy spadající do MVC vrstvy označované jako „model“, slouží v aplikacích pro správu dat, se kterými program pracuje. Vrstva může obsahovat metody pro práci s databází, definovat pravidla pro validaci dat a provádět operace, jež jsou potřebné pro vykonání operace požadované klientem. Daná skupina tříd typicky zahrnuje tzv. „business logiku“ aplikace [23t].

V nově vytvořeném webovém řešení do uvedené vrstvy architektury spadají například třídy, jež zajišťují práci s databází - tzv. „repozitáře“. Ve Spring Boot frameworku se takové třídy označují anotací `@Repository`. Příklady repozitářů v projektu jsou:

- `DeviceRepository` - práce s tabulkou „device“ relační DB,
- `GroupRepository` - práce s tabulkou „group“ relační DB,
- `UserRepository` - práce s tabulkou „user“ relační DB.

Vytvořený produkt využívá objektně orientované mapování („ORM“). Jedná se o techniku, která umožňuje namapovat objekt získaný z relační databáze na entitu, kterou je možno pomocí zvoleného objektně orientovaného jazyka dále zpracovávat. V rámci frameworku Spring Boot je třeba entity, jež reprezentují databázové objekty, označovat anotací `@Entity`. U třídy reprezentující databázový objekt je často spolu s anotací „`@Entity`“ uvedena i anotace `@Table`, jež specifikuje databázovou tabulku obsahující potřebná data entity. Třídami reprezentující entity jsou například:

- `Device` - reprezentuje spravované zařízení, obsahuje například MAC a IP adresu zařízení,
- `Group` - představuje skupinu spravovaných zařízení, jež musí být označena názvem a případně popisem,
- `User` - entita reprezentující uživatele produktu pro správu strojů, obsahuje například Orion login uživatele a jeho roli v systému.

### Pohled

Účelem kódu, který je řazen do pohledové vrstvy, je zajistit zobrazení výstupních dat aplikace uživateli. Daná vrstva je u webových řešení povětšinou realizována pomocí šablon, jež jsou psány s využitím značkovacího jazyka HTML. Kromě elementů specifických pro jazyk HTML obsahují šablony i speciální tagy zvoleného šablonovacího systému. Prostřednictvím tagů mohou být v rámci šablon vyjádřeny například cykly, podmínky a proměnné, jež jsou při využití šablony nahrazeny dynamickými daty aplikace [Stř23b]. V rámci projektu je využit šablonovací systém „Thymeleaf“, přičemž šablony se nacházejí v adresáři `src/main/resources/templates`. Mezi šablony, jež jsou v řešení využívány, patří například:

- `device_management` - šablona, která je využita pro zobrazení seznamu spravovaných zařízení,

- `action_log` - zobrazení logu operace prováděné nad strojem či stroji,
- `registration_approval` - zobrazení žádostí o registraci uživatelů (pouze admin),
- `device_approval` - šablona žádostí o přiřazení zařízení (pouze admin).

## Kontroler

Daná vrstva v rámci MVC architektury funguje jako řadič a představuje prostředníka mezi komponentou modelu a pohledu [23t]. Odpovědností kontroler vrstvy je reakce na události, na základě kterých vrstva patřičně aktualizuje model a výsledek poskytne pohledové vrstvě. Události, na které je třeba reagovat, jsou typicky vytvářeny uživatelem.

Příkladem události může být přechod na určitou webovou stránku, definovanou pomocí URL. Kontroler vrstva na základě parametrů uvedených v URL detekuje, k jaké části aplikace chce uživatel přistoupit, a zajistí volání odpovídající metody modelu. Po získání dat z modelové vrstvy jsou získané informace předány pohledové vrstvě, která zajistí zobrazení informací, typicky s využitím HTML šablony.

Konvence frameworku Spring Boot nařizuje označit všechny třídy, jež slouží jako kontrolery aplikace, anotací `@Controller`. Metody, jež se nachází uvnitř kontrolerů, musí specifikovat typ HTTP požadavku, na který mají reagovat, a endpoint, se kterým bude metoda provázána. Spring Boot pro uvedený účel poskytuje anotaci `@RequestMapping`, jež vyžaduje dva argumenty. Prvním argumentem uvedené anotace je požadovaný typ HTTP požadavku (např. GET, POST, PUT), druhý argument specifikuje požadovaný endpoint. Framework Spring Boot poskytuje i alternativní anotace, které očekávají pouze jeden argument, kterým je webový endpoint. Anotacemi vyžadující pouze jeden argument jsou například `@GetMapping`, `@PostMapping` a `@PutMapping`. Použití anotací, jež mají v názvu označení HTTP metody, vede ke zvýšení čitelnosti kódu a usnadňuje jeho budoucí údržbu. Více informací k uvedeným anotacím lze nalézt na oficiální stránce vývojáře frameworku Spring: <https://docs.spring.io/spring-framework/docs/current/javadoc-api/org.springframework.web.bind.annotation/>.

Ve vytvořeném webovém řešení se kontrolery vyskytují ve složce `controller` a jsou jimi například:

- `DeviceController` - poskytuje služby související se spravovanými stroji,
- `GroupController` - zprostředkovává práci se seznamy strojů,
- `UserController` - umožňuje správu uživatelů aplikace.

### 7.1.3 API

Součástí projektu je programové rozhraní, jež umožňuje práci s produktem bez využití webové aplikace. Rozhraní bylo integrováno především za účelem usnadnění komunikace mezi Android klientem a serverem. Využití API namísto grafického rozhraní webové aplikace vede k nižší datové spotřebě, jelikož předmětem síťové komunikace je pouze text. Omezit datovou náročnost je u mobilních zařízení žádoucí, protože uživatelé zařízení mají mnohdy datové tarify, jež limitují množství přenesených dat.

Programové rozhraní je řádně zdokumentováno pomocí Javadoc a může být využito k vytvoření alternativních klientů aplikace. Všechny endpointy programového rozhraní jsou chráněny vůči neoprávněnému přístupu na úrovni serveru a nehrozí tak narušení bezpečnosti projektu při využití klientů třetích stran. Pro ochranu přístupu k endpointům se v použitém frameworku využívá anotace `@PreAuthorize`, která umožňuje v rámci argumentu specifikovat podmínky, za kterých může být k danému endpointu přistoupeno. V nově vytvořeném řešení je přístup k funkcím systému řízen na základě rolí. Například funkce poskytující seznam uživatelů, kteří jsou přihlášení na určitém stroji, je přístupná pouze uživatelům s rolí administrátora. Metoda kontroleru, jež zajišťuje zjištění seznamu přihlášených uživatelů, obsahuje anotaci `@PreAuthorize(hasAuthority('ADMINISTRATOR'))`.

Třídy poskytující programové rozhraní jsou označeny Spring Boot anotací `@RestController`. V rámci projektu jsou API třídy umístěny ve složce `apiController` a jsou jimi například:

- `WindowsUpdateAPI` - rozhraní pro práci s Windows Update (spuštění aktualizace, zobrazení historie aktualizací),
- `CompWinUserAPI` - rozhraní pro správu uživatelů v rámci OS Windows (zobrazení právě přihlášených uživatelů, získání seznamu všech lokálních uživatelů),
- `GroupAPI` - zprostředkovává funkcionalitu související se seznamy strojů (získání všech skupin strojů uživatele, vyžádání obsahu určitého seznamu zařízení).

### 7.1.4 Vykonávání skriptů

Pro získání různorodých informací a obsluhu strojů jsou využívány skripty, jež jsou na spravovaných strojích spouštěny prostřednictvím technologie SSH. Pro spuštění příkazů na vzdálených strojích byl využit program „Parallel-SSH“, jenž je provozován v rámci Docker kontejneru s aplikačním serverem. Uvedený nástroj umožňuje paralelní spuštění příkazů na více vzdálených strojích současně, čímž



může značně zkrátit čas, jenž je nutný k vykonání příkazu na skupině strojů. Více informací o programu Parallel-SSH je možno nalézt na webu vývojáře: <https://parallel-ssh.org/>.

Logiku výkonu skriptů obsahuje třída `SshServiceImpl`, jež má 4 metody. Metody s označením `execCmdDev` a `execCmdGroup` slouží pro okamžité vykonání příkazu na stroji, resp. skupině strojů. Při využití těchto metod nástroj Parallel-SSH čeká na dokončení příkazu na spravovaných strojích a následně vrátí SSH log s výsledky operace. Popis parametrů metody `execCmdDev` je uveden v tabulce 7.1.

<b>parametr</b>	<b>popis</b>
<code>ip</code>	IP adresa stroje, na kterém má být příkaz vykonán
<code>user</code>	jméno uživatele, jež je použito pro SSH přihlášení
<code>cmd</code>	příkaz (skript), jež má být vykonán na spravovaném stroji
<code>normalize</code>	pokud <code>true</code> , bude odstraněna diakritika z SSH logu

Tabulka 7.1: Popis parametrů metody `execCmdDev`

Třída obsahuje ještě metody `execAsEventDev` a `execAsEventGroup`, jež jsou relevantní pouze pro obsluhu strojů s operačním systémem Windows. Uvedené metody zajistí vytvoření plánované úlohy v systému, jež je následně spuštěna a Parallel-SSH nečeká na dokončení této úlohy. Uvedené metody jsou využívány pro spuštění Chocolatey aktualizací, jelikož aktualizace programů může být zdlouhavá činnost a není žádoucí, aby uživatel musel mít otevřenou webovou aplikaci či mobilního klienta po celou dobu aktualizace. Parametry metody `execAsEventGroup` jsou uvedeny v tabulce 7.2.

parametr	popis
ips	ArrayList objekt s IP strojů, na kterých bude příkaz vykonán
user	jméno uživatele, jež je použito pro SSH přihlášení
cmd	příkaz (skript), jež bude vykonán na množině spravovaných strojů
logPath	cesta k logovacímu souboru (na spravovaném zařízení)
taskName	název, který bude přiřazen úloze (v rámci Windows)

Tabulka 7.2: Popis parametrů metody `execAsEventGroup`

Třídy, jež obsahují vytvořené skripty a volají metody umístěné v `SshServiceImpl`, se nachází ve složce `service\actionsWin` a `service\actionsLin`. Kód, jenž zajišťuje získání informací o stroji, je uveden v rámci ukázky 7.1. Uvedený kód se nachází v rámci metody `getInfoDev` třídy `BasicWinToolsServiceImpl`. Mezi získané informace patří například datum instalace operačního systému, architektura systému, verze systému BIOS, model procesoru, informace o doméně a velikost paměti RAM. Z ukázky lze pozorovat, že skript bude spuštěn pod účtem „Administrator“ a bude vykonán pouze na jednom zařízení, jelikož je volána metoda `execCmdDev`. Před navrácením informací od spravovaného stroje dojde k odstranění diakritiky, jelikož je na patřičné pozici použit argument `true`.

#### Zdrojový kód 7.1: Ukázka skriptu zajišťujícího získání informací o stroji

```
1 sshService.execCmdDev(deviceIP, "Administrator", "Get-ComputerInfo|Select-WindowInstallDateFromRegistry,WindowsProductName,WindowsRegisteredOrganization,WindowsRegisteredOwner,WindowsSystemRoot,BiosBIOSVersion,BiosCaption,BiosCurrentLanguage,BiosDescription,BiosFirmwareType,BiosListOfLanguages,BiosManufacturer,BiosName,BiosReleaseDate,BiosStatus,CsDNSHostName,CsDomain,CsNumberOfLogicalProcessors,CsNumberOfProcessors,CsProcessors,OsTotalVisibleMemorySize,OsFreePhysicalMemory,OsTotalVirtualMemorySize,OsFreeVirtualMemory,OsInUseVirtualMemory,OsArchitecture,OsLanguage,KeyboardLayout,TimeZone", true);
```

## 7.1.5 Persistence dat

Uchovávání persistentních dat je v systému řešeno pomocí relační databáze MySQL, pro kterou systém Docker poskytuje image označený jako „mysql“. V rámci databáze jsou uloženy data spravovaných strojů a uživatelů aplikace pro správu strojů. V modelu databáze se vyskytuje celkem 7 databázových tabulek, jejichž význam je uveden v tabulce 7.3. Návrh modelu databáze, ze kterého jsou patrné vztahy mezi tabulkami, je předmětem obrázku 7.3. Uvedený model byl vytvořen v nástroji MySQL Workbench, jež vyvíjí společnost Oracle. Více informací o produktu

MySQL Workbench je možno nalézt na oficiálních webových stránkách: <https://www.mysql.com/products/workbench/>.

název tabulky	ukládaná data
device	data spravovaných strojů (MAC adresa, IP adresa...)
group	data skupin strojů (název skupiny a její popis)
group_contains_device	realizace relace M:N mezi tabulkou device a group
user	informace o uživateli řešení (Orion login, jméno...)
role	dostupné uživatelské role (název role a její popis)
registration_request	registrační požadavky uživatelů (pro užívání aplikace)
device_request	požadavky na registraci stroje uživatelských strojů

Tabulka 7.3: Význam databázových tabulek (server aplikace)

## 7.1.6 Systémové požadavky

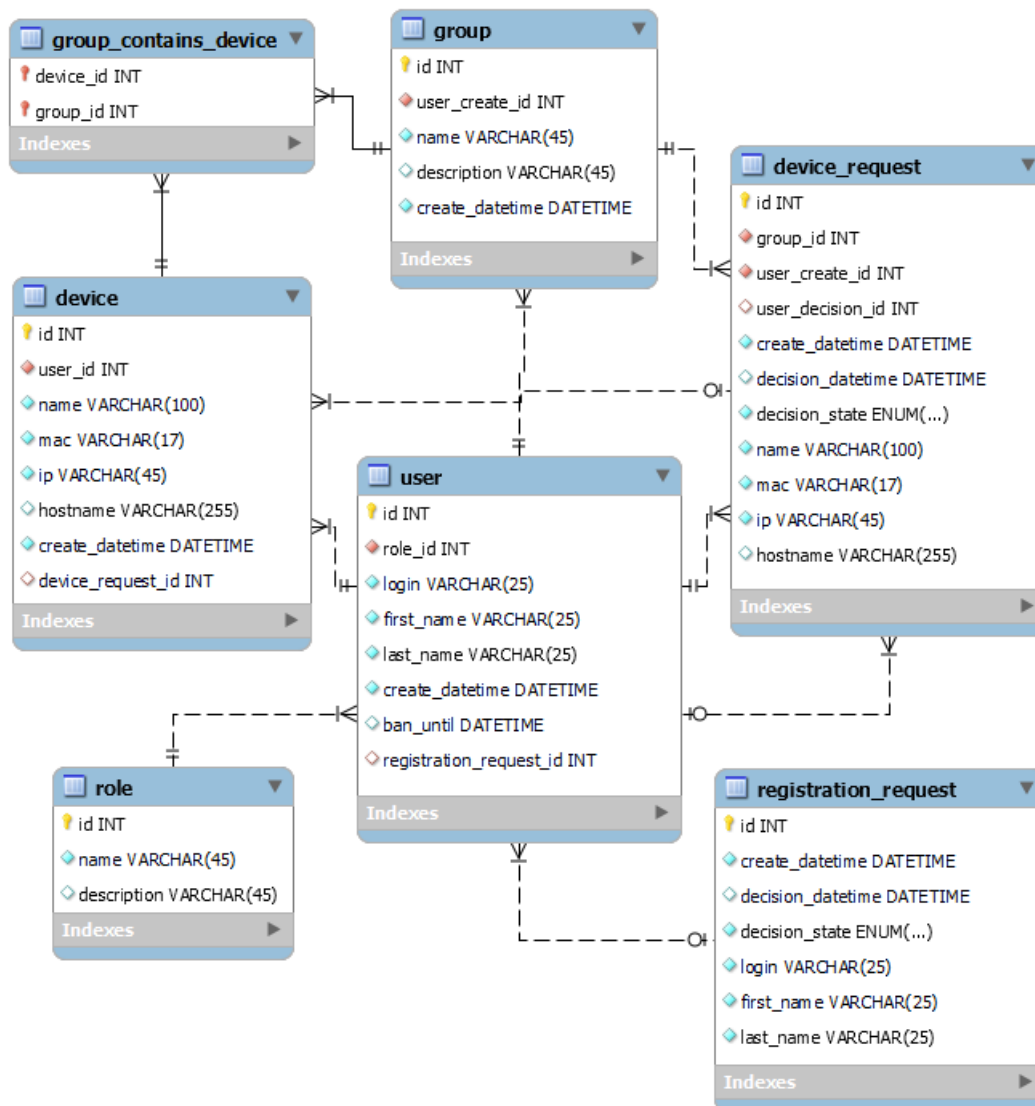
Provoz serveru webového řešení vyžaduje kontejnerovou technologii Docker. Uvedený nástroj pro správu izolovaných kontejnerů je možno nainstalovat na všechny majoritní platformy (Windows, macOS a Linux) [Tur14]. Vývojář programu Docker udává, že produkt vyžaduje alespoň 4GB operační paměti a 64bitový procesor, jenž má podporu pro virtualizaci. Uvedená velikost RAM však není vztahena k počtu kontejnerů, jež mají být pomocí Dockeru provozovány, přičemž množství obsazené paměti se zvyšuje spolu s počtem běžících kontejnerů. Vytvořené řešení pro správu vyžaduje provoz dvou kontejnerů, které zajišťují provoz webového serveru a MySQL databáze.

Během testování velikost paměti RAM, jež byla využita kontejnery, nepřesáhla 1GB (viz snímek 7.4). Provoz systému na stroji vybaveném pouze 4GB operační paměti lze na základě testu doporučit, avšak pouze za předpokladu, že hostující stroj nebude sloužit k dalším účelům.

Velikost požadované paměti RAM závisí i na hostujícím operačním systému a platformě kontejnerů, jež jsou prostřednictvím Dockeru provozovány. Nově vytvořené řešení pro správu využívá Linuxové kontejnery a pro jejich provoz na platformě Windows je vyžadován nástroj „WSL“. Jedná se o nástroj, který umožňuje běh Linuxového software v prostředí Windows [23]. Využití WSL zvyšuje hardwarovou náročnost, jelikož je založeno na principu virtualizace, a určité množství paměti RAM musí být vyhrazeno pro využití virtualizovaným strojem.

Více informací o Dockeru je uvedeno v kapitole 5.1.2, jež se zabývá návrhem webové části projektu.

## 7. Implementace navrženého řešení



Obrázek 7.3: Databázový model aplikace

```
C:\Windows\system32\cmd.exe - docker stats
```

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %
6edd36342f4b	magicwol-magicwolapp-1	0.14%	454.7MiB / 12.39GiB	3.58%
5c602ab81da6	magicwol-mysqldb-1	0.41%	392.2MiB / 12.39GiB	3.09%

Obrázek 7.4: Využití paměti RAM kontejnery aplikace

## 7.2 Mobilní aplikace

Předmětem této kapitoly je popis implementace mobilního klienta řešení. Výsledný mobilní klient je určen pro platformu Android a je distribuován formou instalačního balíku „apk“.

### 7.2.1 Dělení tříd programu

Třídy vytvořené v rámci Android projektu lze rozdělit do 4 základních skupin, jejichž popis je uveden níže. Každá programová třída je důkladně okomentována s využitím Javadoc.

#### Komunikace se serverem

Komunikace mobilního klienta s aplikačním serverem probíhá prostřednictvím programového rozhraní. Informace o službách, které poskytuje API serveru obsahuje kapitola 7.1.3. Android klient pro volání endpointů aplikačního serveru využívá knihovnu Retrofit, která umožňuje specifikovat endpointy programového rozhraní a jejich parametry pomocí anotací. Každá metoda, jež je v rámci projektu využita k volání API, obsahuje Retrofit anotaci specifikující typ HTTP požadavku. Například metoda, která zajišťuje volání endpointu očekávajícího HTTP metodu GET, musí být označena anotací @GET. Retrofit anotace specifikující HTTP požadavky očekávají jeden argument, kterým je cílová URL endpointu. Kompletní dokumentaci knihovny Retrofit lze nalézt na oficiálních stránkách vývojáře: <https://square.github.io/retrofit/>.

Rozhraní zajišťující komunikaci se serverovým API jsou v Android projektu umístěna ve složce `api`. Mezi realizovaná rozhraní patří například `WindowsUpdateAPI`, `CompWinUserAPI` a `WindowsUpdateAPI`. Názvy uvedených rozhraní jsou shodné s označením tříd, jež definují API na straně serveru (viz kapitola 7.1.3), což přispívá k čitelnosti kódu a usnadňuje orientaci v projektu.

#### Správa v rámci lokální sítě

Mobilní klient kromě možnosti správy vzdálených univerzitních strojů prostřednictvím serveru poskytuje i omezené možnosti správy strojů v lokální síti. Bez připojení k externímu serveru umožňuje mobilní aplikace uživateli vytvářet seznamy s lokálními stroji a může tak sloužit pro evidenci strojů. Za předpokladu, že se zařízení s klientem nachází na stejném segmentu sítě jako spravovaný stroj, umožňuje program i zapnutí daného stroje. Je zřejmé, že pro poskytnutí uvedené funkcionality musí program pracovat s databází, broadcastovou adresou sítě a rovněž musí ob-

sahovat logiku pro odesílání „Magic Packetu“. Uvedené služby v rámci Android projektu zajišťují třídy:

- `DatabaseLogic` - obsahuje metody umožňující správu lokální SQLite databáze strojů,
- `DeviceInfoLogic` - zajišťuje práci se síťovým zařízením (získání broadcast IP adresy, zjištění dostupnosti zařízení, kontrola validity IP adresy),
- `MagicPacketLogic` - zprostředkovává odeslání Magic Packetu na cílové zařízení.

### Třídy zajišťující reprezentaci dat

Uživatelské rozhraní programu pracuje pouze s jednou Aktivitou, přičemž v rámci této Aktivity se střídají Fragments. Vyobrazený fragment se mění v závislosti na funkcionalitě aplikace, jež uživatel zvolí z postranního navigačního panelu programu. Kód zajišťující změnu Fragmentů se nachází uvnitř třídy `MainActivity`. Detailní informace o Fragment objektech lze nalézt v oficiální Android dokumentaci: <https://developer.android.com/guide/fragments>.

Jednotlivé Fragments se v projektu nachází ve složce `fragment` a jsou jimi například:

- `UniListFrag` - zobrazení dostupných seznamů obsahujících vzdálené stroje,
- `UniDevFrag` - zobrazení zařízení v rámci konkrétního seznamu,
- `UniServerFrag` - správa oblíbených vzdálených serverů,
- `PrefFrag` - zobrazení nastavení aplikace.

### Spojovací třídy

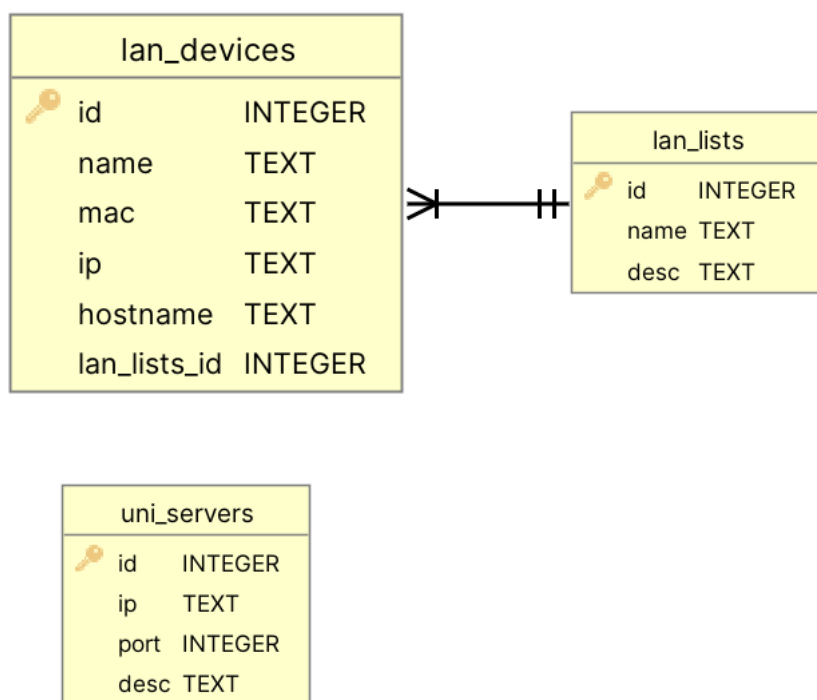
Při vývoji na platformě Android nejsou získaná data přímo prezentována uživateli. Pro zobrazení dat v uživatelském rozhraní je často nutné využít objekt, který je označován jako `Adapter`. Jedná se o prvek, který představuje prostředníka mezi zdrojem dat a uživatelským rozhraním. Cílem `Adapter` objektů je zajistit transformaci vstupních dat na formát, který lze využít v částech grafického rozhraní aplikace. Komponentami grafického rozhraní, ve kterých mohou být transformovaná data vyobrazena, jsou například `ListView` nebo `Spinner`. Více informací o objektech typu `Adapter` lze nalézt v oficiální dokumentaci OS Android: <https://developer.android.com/reference/android/widget/Adapter>.

V rámci řešení se vyskytují například následující spojovací třídy:

- UniListAdapter - konverze dat seznamů vzdálených strojů,
- UniDevAdapter - zpracování dat zařízení v konkrétním seznamu,
- UniServerAdapter - transformace dat spravovaných vzdálených serverů.

## 7.2.2 Persistence dat

Pro uchovávání dat v rámci mobilní aplikace byla využita SQLite databáze. Jedná se o relační databázi s otevřeným zdrojovým kódem, pro kterou Android SDK poskytuje nativní podporu [23x]. Databáze je v řešení potřebná pro uložení dat spravovaných lokálních strojů a informací o aplikačních serverech, ke kterým se uživatel opakovaně připojuje. Data vzdálených strojů nejsou v rámci SQLite databáze uchovávána a jsou získávána z aplikačního serveru pomocí API. V databázovém modelu se vyskytují celkem 3 databázové tabulky, přičemž jejich význam je uveden v tabulce 7.4. Grafický návrh modelu databáze obsahuje obrázek 7.5. Z modelu je patrné, že relační vztah existuje pouze mezi tabulkou uchovávající záznamy o strojích a tabulkou, jež definuje seznamy strojů. Zmíněné tabulky (lan\_devices a lan\_lists) jsou v relaci 1:N, což znamená, že jednomu seznamu náleží více strojů.



Obrázek 7.5: Model použité SQLite databáze

<b>název tabulky</b>	<b>ukládaná data</b>
lan_devices	informace o spravovaných strojích (MAC adresa, IP adresa...)
lan_lists	informace o skupinách strojů (název skupiny a její popis)
uni_servers	data oblíbených aplikačních serverů (IP adresa, port a popis)

Tabulka 7.4: Význam databázových tabulek (mobilní klient)

### 7.2.3 Systémové požadavky

Instalace a provoz mobilního klienta vyžaduje Android API verze 21 nebo vyšší. Android klienta je tak možno provozovat na mobilních zařízeních, jež využívají operační systém Android minimálně ve verzi 5.0. Program vyžaduje přidělení práv přístupu k internetu. Internetové připojení program využívá pro kontaktování aplikačního serveru, který poskytuje informace o vzdálených strojích a zprostředkovává vykonávání požadavků klienta. V případě používání programu pouze pro účely správy počítačů lokální sítě není internetové připojení využíváno. Aplikace je distribuována prostřednictvím apk balíčku, proces instalace programu je uveden v příloze B této práce.



# Ověření funkčnosti produktu

## 8

Stěžejní funkcionality webové části projektu i mobilní aplikace byly v průběhu vývoje testovány. Pro účely testování webové aplikace a mobilního klienta byly vytvořeny testovací scénáře, jež umožňují snadno ověřit funkcionality aplikace. Všechny uvedené testovací scénáře očekávají provoz řešení na lokálním stroji, proto se v rámci URL testovacích scénářů vyskytuje hostname `localhost`.

Modul pro správu uživatelů v rámci webového řešení je navíc pokryt JUnit testy, jež umožňují automatizované otestování očekávané funkčnosti daného modulu.

## 8.1 Testovací zařízení

V následujících podkapitolách jsou uvedeny parametry zařízení, na kterých probíhalo testování vytvořeného produktu. Sekce 8.1.1 popisuje konfigurace zařízení, jež byla využita k testování webové verze řešení. V oddílu 8.1.2 jsou uvedeny parametry zařízení s OS Android, na kterých probíhalo testování mobilního klienta. Kapitola 8.1.3 obsahuje parametry počítačů, na kterých byl testován provoz aplikačního serveru.

### 8.1.1 Webová aplikace

Vytvořená webová aplikace byla otestována na zařízeních, jejichž konfigurace je uvedena v tabulce 8.1. Web byl otestován na počítačích s různorodými operačními systémy, úhlopříčkou obrazovky a prohlížeči. Webová verze produktu byla otestována i na mobilních telefonech Poco M5s a iPhone 13 Pro, jejichž konfigurace je v níže umístěné tabulce uvedena.

Během testování finální verze produktu nebylo zaznamenáno žádné nestandardní chování webové aplikace. Aplikace pracovala na zařízeních uvedených v tabulce 8.1 dle očekávání. Při provádění testovacích scénářů webové verze aplikace, jež jsou uvedeny v kapitolách 8.2, 8.3 a 8.4, bylo dosaženo očekávaných výsledků testů.

typ zařízení	operační systém	prohlížeč	vel. displeje [palec]
notebook	Windows 10	Google Chrome	11,6
notebook	Windows 10	Microsoft Edge	11,6
desktop	Windows 11	Opera	24
desktop	Windows 11	Firefox	24
desktop	Debian 11	Firefox	24
telefon	Android 12	Google Chrome	6,43
telefon	iOS 16	Safari	6,06

Tabulka 8.1: Testované konfigurace - webová aplikace

## 8.1.2 Mobilní klient

Android program byl v průběhu vývoje testován zejména na zařízení Poco M5s, jež využívá operační systém Android 12 s nadstavbou výrobce MIUI 13.0.5.0. Zařízení disponuje procesorem MediaTek Helio G95 a 4GB RAM. Finální verze programu na zařízení funguje spolehlivě a během používání programu nebylo zaznamenáno žádné samovolné ukončení aplikace či jiné problémy, jež by mohly pokazit uživatelský dojem.

Jelikož bylo žádoucí vyzkoušet vytvořenou aplikaci na co možná největším množství zařízení s rozličnou konfigurací, rozhodl jsem se pro testování programu využít nástroj „AVD“. Uvedený nástroj umožňuje správu virtuálních zařízení s OS Android, přičemž lze specifikovat parametry těchto zařízení. Virtuální stroje vytvořené nástrojem AVD mohou mít rozdílnou úhlopříčku obrazovky, rozlišení displeje, velikost paměti RAM i jinou verzi OS Android. Specifikace virtuálních zařízení, na kterých byla finální verze aplikace testována, jsou uvedena v tabulce 8.2. Všechny uvedené virtuální stroje měly přidělenou operační paměť o velikosti 1536MB.

emulátor č.	Android	vel. displeje [palec]	rozliš. displeje [pixel]
1	13.0	5,5	1440x2560
2	12.0	5,0	1080x1920
3	9.0	6,0	1440x2880
4	5.0	5,0	1080x1920

Tabulka 8.2: Testované konfigurace virtuálních zařízení

Na veškerých strojích, jež jsou specifikovány v rámci tabulky 8.2, program pracoval bezproblémově a nebylo zaznamenáno žádné nestandardní chování aplikace. Z uvedené tabulky je patrné, že program byl otestován i na nejnovějším Androidu 13. Funkčnost aplikace byla ověřena i na Androidu 5.0, což je nejstarší verze Androidu, jež vytvořený mobilní program podporuje.

Při vykonávání testovacích scénářů určených pro testování Android aplikace, jež jsou podmnožinou scénářů uvedených v kapitolách 8.2 a 8.4, byly získány očekávané výsledky.

### 8.1.3 Server aplikace

Nasazení serveru prostřednictvím kontejnerové technologie Docker bylo otestováno na zařízeních, jejichž specifikace je uvedena v tabulce 8.3.

typ zařízení	operační systém	RAM [GB]	model procesoru
desktop	Windows 10	16	Intel Core i5-12500
desktop	Windows 10	16	Intel Xeon E3-1246 v3
desktop	Windows 11	64	AMD Ryzen 5 5600G
notebook	Windows 11	12	Intel Core i5-5200U
desktop	Debian 11	16	Intel Xeon E3-1246 v3

Tabulka 8.3: Testované konfigurace - aplikační server

Spuštění serveru bylo otestováno na více platformách (Windows i Linux), aby byla ověřeno, že je řešení multiplatformní. Server aplikace bylo možné spustit na všech konfiguracích uvedených v tabulce 8.3 a server na všech uvedených strojích pracoval dle očekávání.

## 8.2 Testovací scénáře - práce s uživateli

Následující podkapitoly obsahují popis testovacích scénářů, jejichž cílem je ověřit funkcionální část aplikace, jež pracuje s údaji uživatelů řešení pro správu.

### 8.2.1 Přihlášení běžného uživatele

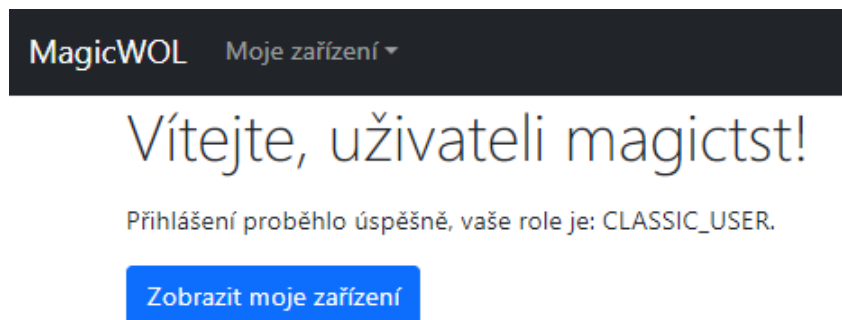
Daný scénář otestuje přihlášení uživatele a ověří, zda byl serverem označen jako běžný uživatel. Scénář je dostupný pro webovou aplikaci i Android klienta.

#### Webová aplikace

1. Přejít na webovou stránku `http://localhost:8080/login`.
2. Klepnutí do pole s nápovědou „Orion login“.
3. Vepsání validního Orion loginu běžného uživatele.
4. Klepnutí do pole s nápovědou „Orion heslo“.

5. Vepsání validního Orion hesla běžného uživatele.
6. Stisknutí tlačítka „Přihlásit“.

Test je splněný za předpokladu, že proběhlo přesměrování a nyní je vyobrazena stránka informující o přihlášení běžného uživatele. Možný výsledek testu je předmětem obrázku 8.1.

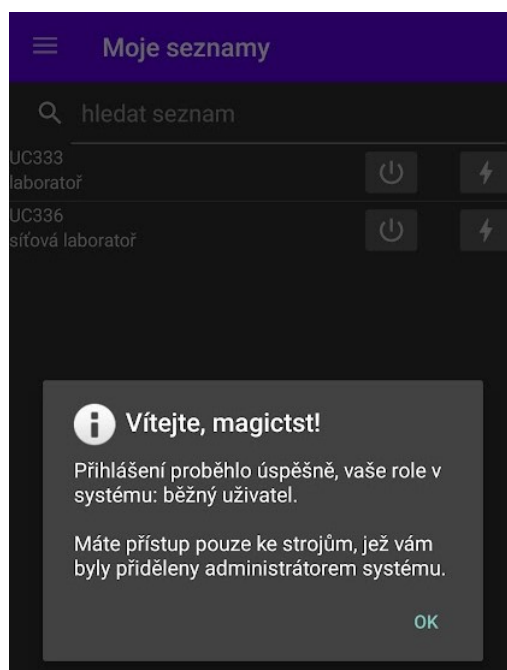


Obrázek 8.1: Přihlášení běžného uživatele do webové aplikace

### Android klient

1. Přejít na kartu „Přihlášení“, jež se nachází v sekci „Další možnosti“ postranní navigace aplikace.
2. Klepnutí do pole s nápovědou „Orion login“.
3. Vepsání validního Orion loginu běžného uživatele.
4. Klepnutí do pole s nápovědou „heslo“.
5. Vepsání validního Orion hesla běžného uživatele.
6. Zvolení odpovídajícího aplikačního serveru z nabídky serverů.
7. Stisknutí tlačítka „Přihlásit se“.

Splnění testu je podmíněno automatickým přechodem na kartu „Moje seznamy“ a zobrazením hlášky, jež informuje o úspěšném přihlášení běžného uživatele. Výsledek testu přihlášení běžného uživatele je možno vidět na obrázku 8.2.



Obrázek 8.2: Přihlášení běžného uživatele do mobilní aplikace

## 8.2.2 Přihlášení administrátora

Cílem daného scénáře je ověřit funkčnost přihlášení k administrátorským účtům. Scénář testuje, zda je uživatel vůči serveru úspěšně ověřen a zda má přidělenou očekávanou roli.

### Webová aplikace

1. Přejít na webovou stránku <http://localhost:8080/login>.
2. Klepnutí do pole s nápovědou „Orion login“.
3. Vepsání validního Orion loginu administrátora.
4. Klepnutí do pole s nápovědou „Orion heslo“.
5. Vepsání validního Orion hesla administrátora.
6. Stisknutí tlačítka „Přihlásit“.

Test je považován za splněný, pokud po provedení posledního kroku scénáře proběhne automatické přesměrování a uživateli je vyobrazena stránka, která informuje o úspěšném přihlášení administrátora.

### **Android klient**

1. Přejít na kartu „Přihlášení“ nacházející se v sekci „Další možnosti“ postranního navigačního panelu.
2. Klepnutí do pole s nápovědou „Orion login“.
3. Vepsání validního Orion loginu administrátora.
4. Klepnutí do pole s nápovědou „heslo“.
5. Vepsání validního Orion hesla administrátora.
6. Zvolení odpovídajícího aplikačního serveru z nabídky serverů.
7. Stisknutí tlačítka „Přihlásit se“.

Uvedený test je splněný za předpokladu, že po stisknutí tlačítka „Přihlásit se“ nastane přechod na kartu „Moje seznamy“ a je zobrazena hláška informující o přihlášení administrátora.

### **8.2.3 Podání žádosti o registraci uživatele**

Testovací scénář ověří, zda systém pro podávání žádostí o registraci uživatele funguje dle očekávání. V rámci testu je podána žádost o registraci a přítomnost požadavku je následně ověřena prostřednictvím administrátorského účtu. Daný testovací scénář je možné reprodukovat pouze v rámci webové aplikace.

### **Webová aplikace**

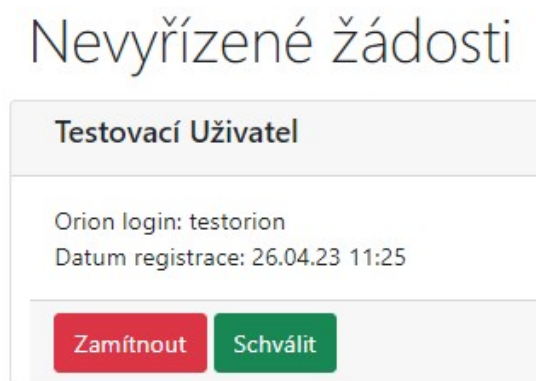
1. Přejít na webovou stránku `http://localhost:8080/login`.
2. Stisknutí tlačítka „Registrace“.
3. Klepnutí do pole s nápovědou „Jméno“.
4. Vepsání křestního jména testovacího uživatele.
5. Klepnutí do pole s nápovědou „Příjmení“.
6. Vepsání příjmení testovacího uživatele.
7. Klepnutí do pole s nápovědou „Orion login“.
8. Vepsání Orion loginu testovacího uživatele.
9. Stisknutí tlačítka „OK“.

10. Přihlášení administrátora do systému.

11. Přejít na stránku

<http://localhost:8080/management/registrationApproval>.

Test je splněný za předpokladu, že je po provedení posledního kroku scénáře vyobrazena stránka obsahující vytvořenou žádost o registraci. Možný výsledek testu znázorňuje obrázek 8.3.



Obrázek 8.3: Zobrazení registrační žádosti v rámci webové aplikace

## 8.2.4 Schválení žádosti o registraci uživatele

Předmětem uvedeného testovacího scénáře je schválení existující uživatelské registrační žádosti administrátorem. Požadovaným výchozím stavem testovacího scénáře je přihlášení administrátora do systému. Uvedený scénář je možno aplikovat pouze na webovou část řešení.

### Webová aplikace

1. Přejít na webovou stránku  
<http://localhost:8080/management/registrationApproval>.
2. Rozbalení nabídky „Žádosti“ vyskytující se v navigaci webu.
3. Volba položky „Schválení registrací“.
4. Zaznamenání uživatelských údajů (křestní jméno, příjmení, Orion login) uvedených u registrace, jež má být schválena.
5. Stisknutí tlačítka „Schválit“ nacházejícího se v kartě registrace, jež má být schválena.

6. Rozbalení roletkové nabídky „Správa uživatelů“ nacházející se v navigačním panelu aplikace.
7. Zvolení položky „Seznam uživatelů“, jež se nachází v roletkové nabídce.
8. Zaznamenání údajů uživatele, jehož účet byl vytvořen na základě provedené registrace.

Test je splněný, pokud se shoduje křestní jméno, příjmení a Orion login, jež byly získány v bodech 4 a 8.

### 8.2.5 Přidání běžného uživatele administrátorem

Testovací scénář ověří, zda může uživatel s administrátorským účtem přidat nového běžného uživatele do systému pro správu strojů. Nutným výchozím stavem je přihlášení uživatele s rolí administrátora do aplikace. Přidání uživatelů je možné pouze prostřednictvím webové aplikace a postup scénáře je tak aplikovatelný pouze na webovou aplikaci.

#### Webová aplikace

1. Přejít na webovou stránku  
`http://localhost:8080/management/userManagement`.
2. Rozbalení nabídky navigačního panelu, jež má označení „Správa uživatelů“.
3. Volba položky „Přidat uživatele“.
4. Klepnutí do pole s nápovědou „Jméno“.
5. Vepsání křestního jména testovacího uživatele.
6. Klepnutí do pole s nápovědou „Příjmení“.
7. Vepsání příjmení testovacího uživatele.
8. Klepnutí do pole s nápovědou „Orion login“.
9. Vepsání Orion loginu testovacího uživatele.
10. Zvolení role „Běžný uživatel“ nacházející se v roletkové nabídce.
11. Stisknutí tlačítka „OK“.

Přidání uživatele proběhlo úspěšně za předpokladu, že po provedení posledního kroku scénáře došlo k obnovení webové stránky a součástí stránky je nyní karta obsahující údaje nového uživatele (křestní jméno, příjmení a Orion login).



## 8.2.6 Změna role

Cílem testovacího scénáře je ověřit, zda je administrátor řešení schopný povýšit stávajícího běžného uživatele na administrátora. Předpokládaným výchozím stavem je přihlášení administrátora do webové aplikace. Změnu role uživatele je možno provést pouze pomocí webové aplikace. Za testovacího uživatele je v kontextu scénáře pokládán uživatel, jehož role má být změněna.

### Webová aplikace

1. Přejít na webovou stránku  
`http://localhost:8080/management/userManagement`.
2. Stisknutí tlačítka s ikonou tužky nacházejícího se v rámci karty testovacího uživatele.
3. Volba položky „Administrátor“ z roletkové nabídky rolí.
4. Stisknutí tlačítka „OK“.

Změna role je vyhodnocena jako úspěšná a test je splněn, pokud dojde k obnovení webové stránky a údaj „Role uživatele“ uvedený v kartě testovacího uživatele obsahuje hodnotu „ADMINISTRATOR“.

## 8.3 Testovací scénáře - evidence strojů

Testovací scénáře uvedené v rámci této kapitoly ověřují funkcionality aplikace související se správou informací o strojích, jež jsou v aplikaci uchovávány.

### 8.3.1 Vytvoření seznamu zařízení

Cílem testovacího scénáře je ověřit, zda je uživatel schopný vytvořit vlastní seznam zařízení. Testovací scénář předpokládá přihlášení běžného uživatele do aplikace. Uvedený testovací scénář je možno replikovat pouze ve webové verzi aplikace.

### Webová aplikace

1. Přejít na webovou stránku  
`http://localhost:8080/owner/deviceManagement`.
2. Rozbalení nabídky „Moje zařízení“ vyskytující se v horním navigačním panelu aplikace.
3. Volba položky „Vytvořit seznam“.

4. Klepnutí do pole s nápovědou „název seznamu“.
5. Vepsání názvu testovacího seznamu strojů.
6. Klepnutí do pole s nápovědou „popis seznamu“.
7. Vepsání popisu testovacího seznamu strojů.
8. Stisknutí tlačítka „OK“.

Vytvoření seznamu bylo úspěšné, pokud proběhne automatické obnovení webové stránky a v rámci vyobrazené stránky je přítomna karta obsahující údaje nového seznamu, jež byly specifikovány v bodech 4 a 6.

### 8.3.2 Přidání zařízení administrátorem

Předmětem daného scénáře je otestování funkcionality přidání nového zařízení do seznamu jiného uživatele. Přímé vložení nového zařízení do systému může provést pouze administrátor, a proto je požadovaným výchozím stavem přihlášení administrátora do aplikace. Testovací scénář předpokládá přítomnost alespoň jednoho seznamu zařízení - v testu „testovací seznam“.

#### Webová aplikace

1. Přejít na webovou stránku  
<http://localhost:8080/global/deviceManagement>.
2. Stisknutí tlačítka s ikonou znaménka „+“, které se nachází uvnitř karty testovacího seznamu.
3. Klepnutí do pole s nápovědou „název stroje“.
4. Vepsání názvu testovacího stroje, jenž má být vložen do seznamu.
5. Klepnutí do pole s nápovědou „MAC stroje“.
6. Vepsání MAC adresy testovacího stroje.
7. Klepnutí do pole s nápovědou „(volitelné) IP stroje“.
8. Vepsání IP adresy testovacího stroje.
9. Klepnutí do pole s nápovědou „(volitelné) hostname stroje“.
10. Vepsání hostname testovacího stroje.
11. Stisknutí tlačítka „OK“.

Test je splněn, pokud dojde k automatickému obnovení webové stránky a v rámci testovacího seznamu je uveden záznam obsahující údaje specifikované v bodech 4, 6, 8 a 10.

### 8.3.3 Podání žádosti o přidání zařízení

Cílem daného scénáře je ověřit spolehlivost funkcionality aplikace umožňující podání žádosti o přidání stroje běžným uživatelem. Výchozím stavem testovacího scénáře je přihlášení běžného uživatele do systému. Scénář předpokládá přítomnost alespoň jednoho seznamu strojů - „testovací seznam“. Spojení „testovací stroj“ v rámci scénáře označuje zařízení, jež má být přidáno do seznamu strojů přihlášeného uživatele.

#### Webová aplikace

1. Přejít na webovou stránku  
`http://localhost:8080/owner/deviceManagement`.
2. Stisknutí tlačítka s ikonou „+“ nacházejícího se v rámci karty testovacího seznamu strojů.
3. Klepnutí do pole s nápovědou „název stroje“.
4. Vepsání uživatelského označení testovacího stroje, který má být přidán.
5. Klepnutí do pole s nápovědou „MAC stroje“.
6. Vepsání MAC adresy testovacího stroje.
7. Klepnutí do pole s nápovědou „(volitelné) IP stroje“.
8. Vepsání IP adresy testovacího stroje.
9. Klepnutí do pole s nápovědou „(volitelné) hostname stroje“.
10. Vepsání hostname testovacího stroje.
11. Stisknutí tlačítka „OK“.
12. Přihlášení administrátora do systému.
13. Přejít na stránku `http://localhost:8080/management/deviceApproval`.

Funkcionalita aplikace pracuje očekávaným způsobem, pokud je po provedení posledního kroku testu vyobrazena webová stránka, jejíž součástí je požadavek na registraci stroje obsahující údaje uvedené v krocích 4, 6, 8 a 10.

### 8.3.4 Editace údajů zařízení

Testovací scénář testuje funkčnost komponenty aplikace, jež umožňuje změnit údaje registrovaného stroje (název, MAC adresa, IP adresa a hostname). Požadovaným výchozím stavem je přihlášení administrátora do aplikace. Scénář předpokládá přítomnost alespoň jednoho zařízení, jehož údaje mají být změněny - v testu „testovací zařízení“.

#### Webová aplikace

1. Přejít na webovou stránku  
`http://localhost:8080/owner/deviceManagement`.
2. Stisknutí tlačítka „zobrazit / skrýt stroje“ nacházejícího se v oblasti karty seznamu, jež obsahuje testovací zařízení.
3. Stisknutí tlačítka s ikonou tužky, které se nachází v řádce odpovídající testovacímu zařízení.
4. Klepnutí do pole s nápovědou „název stroje“.
5. Vepsání nového označení testovacího zařízení.
6. Klepnutí do pole s nápovědou „MAC stroje“.
7. Vepsání nové MAC adresy testovacího zařízení.
8. Klepnutí do pole s nápovědou „(volitelné) IP stroje“.
9. Vepsání nové IP adresy testovacího zařízení.
10. Klepnutí do pole s nápovědou „(volitelné) hostname stroje“.
11. Vepsání hostname testovacího zařízení.
12. Stisknutí tlačítka „OK“.

Pokud po provedení posledního kroku testovacího scénáře dojde k obnovení webové stránky a je vyobrazena hláška „Aktualizace dat zařízení proběhla úspěšně“, pracuje testovaná funkcionální dle očekávání.

### 8.3.5 Odstranění zařízení

Daný scénář je určen pro ověření funkcionality komponenty umožňující odstranění existujícího zařízení ze systému. Scénář pracuje se seznamem strojů vytvořeným uživatelem, a proto je požadovaným výchozím stavem přihlášení běžného uživatele do systému. Předpokládá se přítomnost zařízení, jež má být odstraněno - v testu značeno „testovací zařízení“.

#### Webová aplikace

1. Přejít na webovou stránku  
`http://localhost:8080/owner/deviceManagement`.
2. Stisknutí tlačítka „zobrazit / skrýt stroje“, které se nachází v kartě reprezentující seznam, jenž obsahuje testovací zařízení.
3. Stisknutí tlačítka s ikonou odpadkového koše, které se nachází v řádce odpovídající testovacímu zařízení.
4. Stisknutí tlačítka „Ano“, které je součástí vyobrazeného modálního okna.

Funkcionalita aplikace funguje očekávaným způsobem, pokud je po automatickém obnovení stránky vyobrazena hláška „Odstranění zařízení proběhlo úspěšně“ a seznam strojů neobsahuje záznam testovacího zařízení.

### 8.3.6 Obnovení seznamu strojů

Scénář ověří, zda jsou správně implementovány funkce umožňující export a import informací o spravovaných strojích.

#### Webová aplikace

1. Přejít na webovou stránku  
`http://localhost:8080/owner/deviceManagement`.
2. Stisknutí tlačítka s ikonou csv dokumentu, které se nachází v kartě reprezentující seznam, jehož data mají být exportována (zálohována) a uložena na disk stroje.
3. Stisknutí tlačítka s ikonou odpadkového koše nacházejícího se v rámci karty, jež reprezentuje seznam strojů exportovaný v předchozím kroku, čímž dojde k odstranění seznamu zařízení.
4. Stisknutí tlačítka „Ano“, které je součástí vyobrazeného modálního okna.

5. Vytvoření nového seznamu zařízení.
6. Stisknutí tlačítka pro import, které je součástí karty reprezentující seznam vytvořený v rámci kroku č. 5.
7. Stisknutí tlačítka „Vybrat soubor“ ve vyobrazeném modálním okně.
8. Volba csv souboru, jenž byl získán v rámci kroku č. 2 tohoto scénáře.
9. Stisknutí tlačítka „Importovat“ nacházejícího se v zápatí modálního okna.

Funkcionalita pracuje očekávaným způsobem, pokud je po provedení uvedených kroků vyobrazen výpis všech úspěšně importovaných zařízení a obsah tohoto seznamu odpovídá obsahu původního seznamu, jenž byl ze systému exportován. Možný výsledek testu je vidět na obrázku 8.4.

```
Do seznamu UC333import byly přidány položky:  
název: p01, MAC: 0B-58-61-0D-72-8D, IP: 147.228.63.101, hostname: uc333p01-kiv.fav.zcu.cz  
název: p02, MAC: 38-89-CC-4F-8A-42, IP: 147.228.63.102, hostname: uc333p02-kiv.fav.zcu.cz  
název: p03, MAC: 79-E6-5E-9C-76-C1, IP: 147.228.63.103, hostname: uc333p03-kiv.fav.zcu.cz  
název: p04, MAC: 8C-BF-01-71-D3-82, IP: 147.228.63.104, hostname: uc333p04-kiv.fav.zcu.cz  
název: p05, MAC: 0A-E0-B4-3B-F8-E6, IP: 147.228.63.105, hostname: uc333p05-kiv.fav.zcu.cz
```

Obrázek 8.4: Seznam importovaných informací o strojích

## 8.4 Testovací scénáře - ovládání strojů

Testy nacházející se v této kapitole mají za cíl ověřit funkcionality aplikace, které umožňují ovládání spravovaných strojů. Veškeré uvedené testovací scénáře, vyjma scénáře 8.4.1, očekávají ve výchozím stavu přihlášení administrátora. Ve všech uvedených scénářích figuruje pojem „testovací zařízení“, který označuje počítačovou stanici, jež má být v rámci testu kontaktována.

### 8.4.1 Zapnutí zařízení

Test ověří, zda je uživatel aplikace schopný vzdáleně zapnout stroj. Nutným výchozím stavem testovacího scénáře je přihlášení běžného uživatele (webová aplikace i mobilní klient). Test rovněž předpokládá přítomnost alespoň jednoho zařízení, které má být zapnuto - v testu označeno „testovací zařízení“. Zařízení, jež má být zapnuto, se musí nacházet na stejném segmentu sítě jako server aplikace. Kontrola zapnutí stroje je u daného scénáře prováděna manuálně testerem.

### Webová aplikace

1. Přejít na webovou stránku  
`http://localhost:8080/owner/deviceManagement`.
2. Stisknutí tlačítka „zobrazit / skrýt stroje“, jež se nachází v rámci karty seznamu obsahujícího testovací zařízení.
3. Stisknutí tlačítka s ikonou zapínacího tlačítka nacházejícího se ve sloupci „akce“ a řádce odpovídající testovacímu zařízení.

Pokud proběhlo obnovení webové stránky a zapnutí stroje, funkcionality aplikace funguje korektně a test je splněn.

### Android klient

1. Přejít na kartu „Moje zařízení“ nacházející se v sekci „Univerzita“ postranního navigačního panelu.
2. Klepnutí na kartu seznamu, jež obsahuje testovací zařízení.
3. Klepnutí na tlačítko s ikonou zapínacího tlačítka, které se nachází v rámci karty reprezentující testovací zařízení.

Test mobilního klienta je splněn, pokud po vykonání posledního kroku testovacího scénáře proběhne zapnutí testovacího stroje.

## 8.4.2 Aktualizování systému Windows

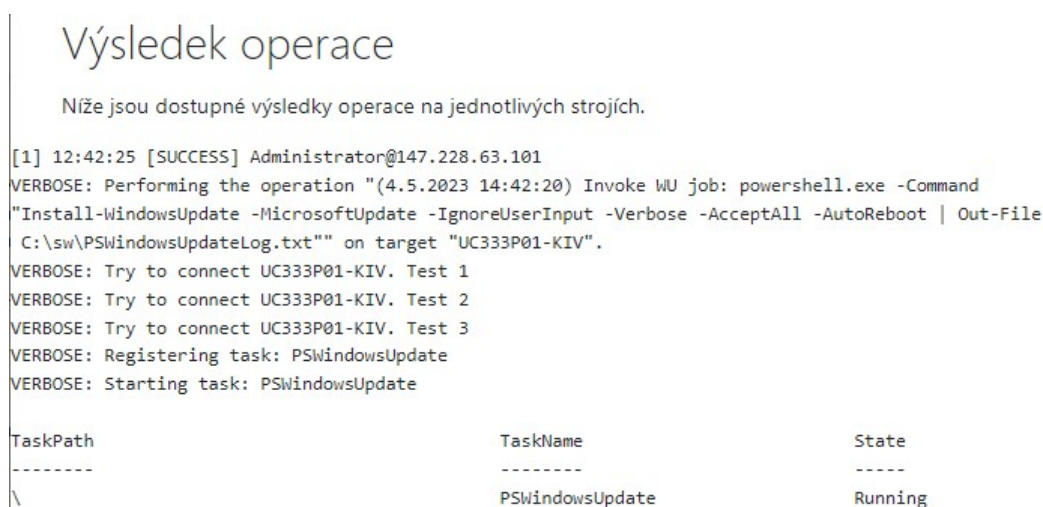
Cílem testovacího scénáře je ověřit, zda řešení pro správu strojů umožňuje vzdáleně spustit proces aktualizace operačního systému Windows.

### Webová aplikace

1. Přejít na webovou stránku  
`http://localhost:8080/owner/deviceManagement`.
2. Stisknutí tlačítka „zobrazit / skrýt stroje“, které se nachází v oblasti karty reprezentující seznam obsahující testovací zařízení.
3. Stisknutí tlačítka s logem systému Windows nacházejícího se v řádce odpovídající testovacímu zařízení.
4. Zvolení položky „Windows Update“ z roletkové nabídky.

5. Stisknutí karty s označením „Aktualizovat OS“, jež se nachází ve vyobrazeném modálním okně.

Test je splněný, pokud proběhlo obnovení stránky a součástí aktuální podoby stránky je SSH log informující uživatele o úspěšném naplánování nové úlohy v systému Windows. Možný výsledek testu je vidět na obrázku 8.5. Průběh aktualizace je logován do souboru `C:\sw\PSWindowsUpdateLog.txt`, jež je dostupný na testovacím zařízení. Umístění logovacího souboru je v případě zájmu možno změnit ve třídě `WindowsUpdateServiceImpl`.



Obrázek 8.5: Naplánování úlohy zajišťující aktualizace OS Windows

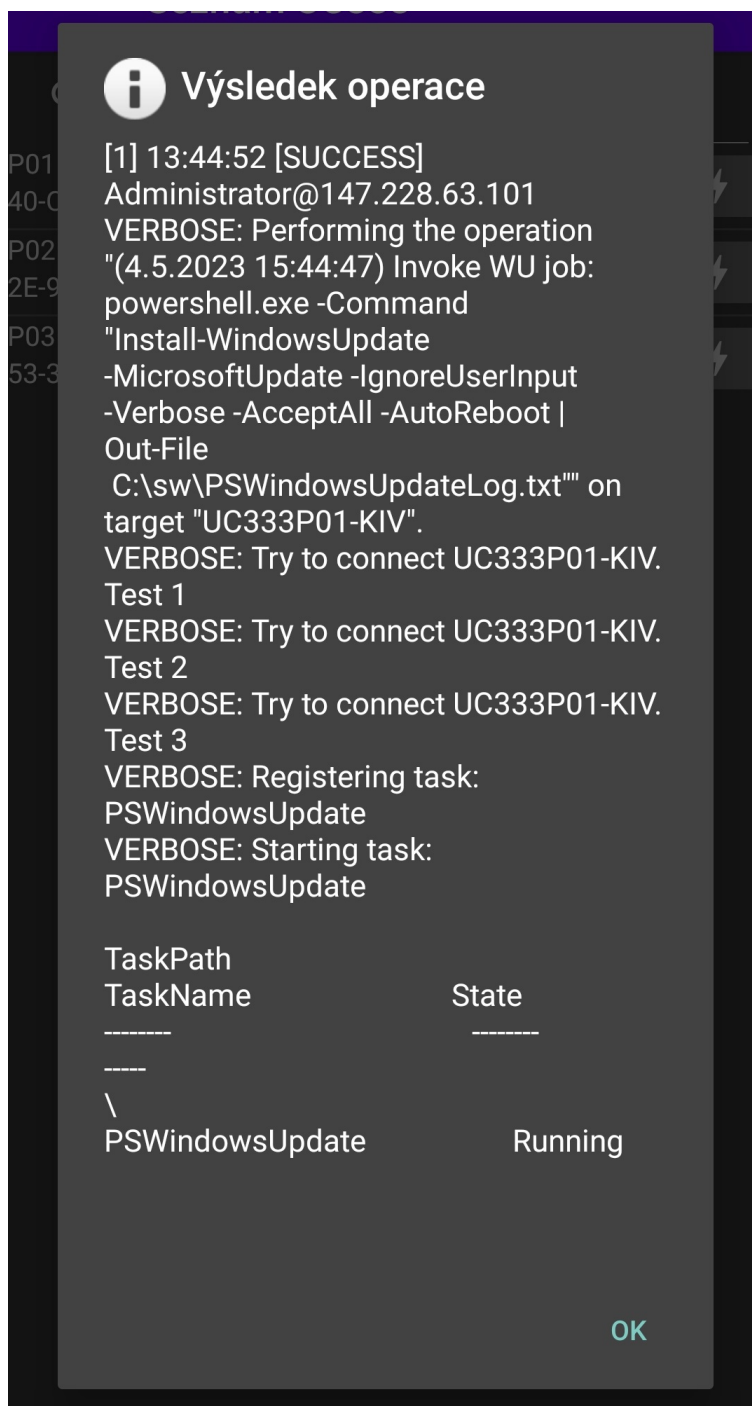
### Android klient

1. Přechod na kartu „Moje zařízení“ nacházející se v sekci „Univerzita“ postranního navigačního panelu.
2. Klepnutí na kartu seznamu, jež obsahuje testovací zařízení.
3. Klepnutí na tlačítko s ikonou blesku, které se nachází v rámci karty reprezentující testovací zařízení.
4. Zvolení položky „Windows“ z nabídky operačních systémů.
5. Zvolení kategorie „Windows Update“.
6. Zvolení akce „Aktualizovat OS“.

Testovaná funkcionální mobilní aplikace pracuje správně, pokud se zobrazí SSH log obsahující informace o naplánování úlohy zajišťující aktualizaci systému



Windows. Možný výsledek testu funkcionality mobilního klienta je předmětem obrázku 8.6.



Obrázek 8.6: Spuštění aktualizace OS Windows pomocí mobilního klienta

### 8.4.3 Aktualizování Chocolatey balíčků

Uvedený test ověří, zda je možné pomocí webové aplikace, resp. mobilního klienta, inicializovat proces aktualizace Chocolatey balíčků, jež jsou nainstalovány na testovacím zařízení.

#### Webová aplikace

1. Přejít na webovou stránku  
`http://localhost:8080/owner/deviceManagement`.
2. Stisknutí tlačítka „zobrazit / skrýt stroje“ nacházejícího se v kartě, která reprezentuje seznam obsahující testovací zařízení.
3. Stisknutí tlačítka s logem systému Windows, jež se nachází v řádce odpovídající testovacímu zařízení.
4. Zvolení položky „Správa software“ z roletkové nabídky.
5. Stisknutí karty s označením „Aktualizovat software“ nacházející se ve vyobrazeném modálním okně.

Test proběhl úspěšně, pokud po provedení posledního kroku testovacího scénáře dojde k automatickému obnovení webové stránky a aktualizovaná podoba stránky obsahuje SSH log informující o úspěšném naplánování úlohy. Stav aktualizace je předmětem logovacího souboru `ChocolateyLog.txt`, který se nachází na testovacím zařízení.

#### Android klient

1. Přejít na kartu „Moje zařízení“ nacházející se v sekci „Univerzita“ postranního navigačního panelu.
2. Klepnutí na kartu seznamu obsahujícího testovací zařízení.
3. Klepnutí na ikonu blesku nacházející se v kartě reprezentující testovací zařízení.
4. Zvolení položky „Windows“ z vyobrazené nabídky systémů.
5. Zvolení kategorie „Správa software“.
6. Zvolení akce „Aktualizovat software“.

Test proběhl úspěšně, pokud je po provedení testu vyobrazeno dialogové okno obsahující SSH log informující o naplánování úlohy zajišťující aktualizace Chocolatey balíčků.

## 8.4.4 Získání seznamu lokálních uživatelů

Cílem testu je ověřit, zda spolehlivě funguje funkcionalita umožňující získání seznamu všech lokálních uživatelů systému Windows. Testovací scénář vyžaduje vlastnění seznamu uživatelských účtů, jež na testovacím stroji existují. Uživatelské účty je možno získat například prostřednictvím nástroje „Správa počítače“, jež je součástí systému Windows 2000 a novější. Zobrazení seznamu lokálních uživatelů prostřednictvím daného nástroje znázorňuje obrázek 8.7.

### Webová aplikace

1. Přejít na webovou stránku  
`http://localhost:8080/owner/deviceManagement`.
2. Stisknutí tlačítka „zobrazit / skrýt stroje“, které se nachází v rámci karty, jež reprezentuje seznam s testovacím zařízením.
3. Stisknutí tlačítka s logem systému Windows nacházejícího se v řádce odpovídající testovacímu zařízení.
4. Zvolení položky „Správa uživatelů“ z roletkové nabídky.
5. Stisknutí karty s názvem „Zobrazit lokální uživatele“ nacházející se ve vyobrazeném modálním okně.

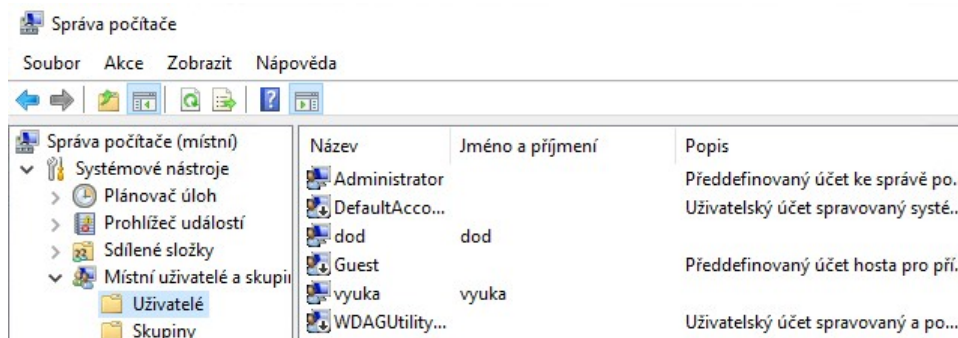
Funkcionalita webové aplikace pracuje očekávaným způsobem, pokud jsou po provedení posledního kroku součástí webové stránky názvy všech lokálních uživatelských účtů, jež na stroji skutečně existují. Ověření lze provést například pomocí nástroje „Správa počítače“, jež byl zmíněn v úvodu scénáře.

### Android klient

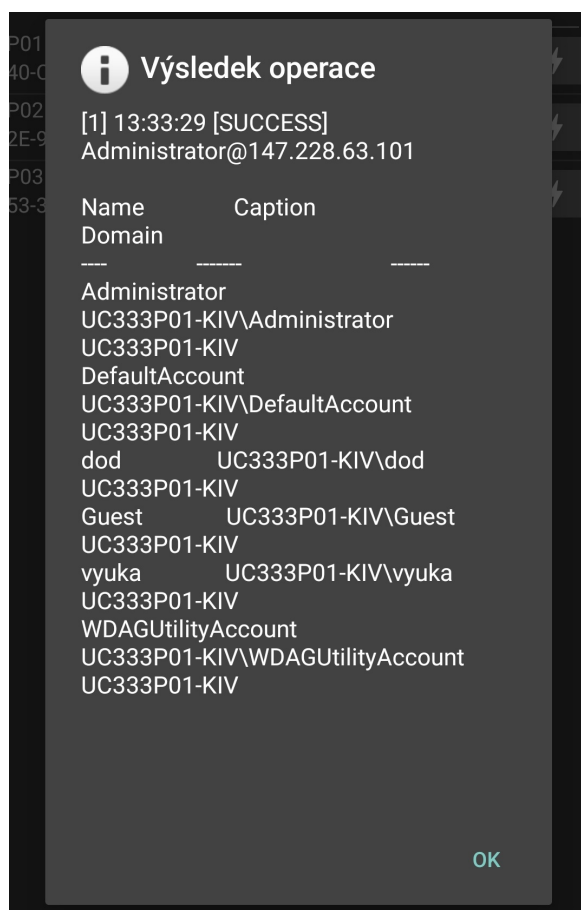
1. Přejít na kartu „Moje zařízení“, která se nachází v kategorii „Univerzita“ postranního navigačního panelu.
2. Klepnutí na kartu seznamu obsahujícího testovací zařízení.
3. Klepnutí na ikonu blesku, jež se nachází v kartě reprezentující testovací zařízení.
4. Zvolení operačního systému „Windows“ z vyobrazené nabídky.
5. Zvolení kategorie „Správa uživatelů“.
6. Zvolení akce „Zobrazit lokální uživatele“.

## 8. Ověření funkčnosti produktu

Test mobilní aplikace proběhl úspěšně, pokud je vyobrazen log SSH komunikace obsahující seznam všech uživatelských účtů, jež existují na testovacím zařízení. Možnou podobu výsledku testu funkcionality mobilní aplikace zachycuje obrázek 8.8.



Obrázek 8.7: Zobrazení lokálních uživatelských účtů v aplikaci Správa počítače



Obrázek 8.8: Získání lokálních účtů v mobilní aplikaci

## 8.4.5 Získání informací o stroji

Testovací scénář ověří, zda aplikace poskytuje správné informace o vybavení testovacího zařízení. Uvedený test vyžaduje znalost modelu procesoru, velikosti nainstalované paměti RAM a verzi nainstalovaného operačního systému na testovacím zařízení. Uvedené informace je možno zobrazit přímo v OS Windows klepnutím pravým tlačítkem na „Tento počítač“ a zvolením možnosti „Vlastnosti“.

### Webová aplikace

1. Přejít na webovou stránku  
<http://localhost:8080/owner/deviceManagement>.
2. Stisknutí tlačítka „zobrazit / skrýt stroje“, které je umístěno v oblasti karty, která reprezentuje seznam s testovacím zařízením.
3. Stisknutí tlačítka s logem systému Windows nacházejícího se v řádce odpovídající testovacímu zařízení.
4. Zvolení položky „Informace o stroji“ z roletkové nabídky.

Test byl splněn, pokud získané informace o modelu procesoru, velikosti paměti RAM a verzi operačního systému odpovídají reálné specifikaci stroje. Informace o hardware stroje se dají ověřit například pomocí nástroje HWiNFO, jenž lze bezplatně stáhnout z domovské stránky produktu: <https://www.hwinfo.com/download/>.

### Android klient

1. Přejít na kartu „Moje zařízení“, v sekci „Univerzita“ postranního navigačního panelu.
2. Klepnutí na kartu seznamu, jež obsahuje testovací zařízení.
3. Klepnutí na ikonu blesku nacházející se uvnitř záložky reprezentující testovací zařízení.
4. Zvolení operačního systému „Windows“ z vyobrazené nabídky.
5. Zvolení záložky „Informace o stroji“ z vyobrazených voleb.

Funkcionalita mobilní aplikace pracuje očekávaným způsobem, pokud je po provedení posledního kroku testu vyobrazen SSH log obsahující reálnou specifikaci testovacího zařízení.

## 8.4.6 Vypnutí zařízení

Test zjistí, zda aplikace umožňuje vypnutí vzdáleného stroje. Kontrola vypnutí stroje je u testovacího scénáře prováděna manuálně.

### Webová aplikace

1. Přejít na webovou stránku  
`http://localhost:8080/owner/deviceManagement`.
2. Stisknutí tlačítka „zobrazit / skrýt stroje“ umístěného v rámci karty, která reprezentuje seznam obsahující testovací zařízení.
3. Stisknutí tlačítka s logem systému Windows, které se nachází v řádce odpovídající testovacímu zařízení.
4. Zvolení položky „Vypnout“ z roletkové nabídky.

Pokud bylo při manuální kontrole zjištěno, že je testovací zařízení vypnuto, funguje funkcionality aplikace dle očekávání.

### Android klient

1. Přejít na kartu „Moje zařízení“, která se nachází v kategorii „Univerzita“ postranního navigačního panelu.
2. Klepnutí na kartu seznamu obsahující testovací zařízení.
3. Klepnutí na ikonu blesku, jež se nachází v rámci karty reprezentující testovací zařízení.
4. Zvolení operačního systému „Windows“ z vyobrazené nabídky.
5. Zvolení záložky „Vypnout“ z vyobrazených voleb.

Funkcionality pracuje dle očekávání, pokud se testovací zařízení po provedení posledního kroku scénáře vypnulo.

## 8.5 Automatické testy

Veškeré vytvořené třídy s JUnit testy se v rámci dodaného Spring Boot projektu nachází v cestě `\src\test\java\lav\drtinao\MagicWOL\service\users`. Automatizované testy jsou obsaženy ve třídách `UserServiceImplTest` a `RegistrationRequestImplTest`. Všechny testy využívají testovací framework Mockito, jehož domovská webová stránka je <https://site.mockito.org/>.

V rámci testovací třídy `UserServiceImplTest` se nachází JUnit testy, jež ověřují práci s již existujícími uživateli aplikace. Uvedená třída obsahuje například test, jež ověří, zda je vyvolána vlastní výjimka `LastPrivilegedException`, pokud je detekován pokus o vymazání posledního administrátora aplikace. Předmětem testovací třídy `RegistrationRequestImplTest` je ověření funkcionality modulu správy uživatelských registrací webové aplikace. Testovací třída registračního modulu obsahuje například test ověřující, zda je při pokusu o schválení již existujícího uživatele vyvolána vlastní výjimka `AlreadyPresentException`. Každý uživatel je v rámci systému jednoznačně identifikován Orion loginem.





# Možná rozšíření produktu

## 9

Podkapitola pojednává o funkcích, které by v řešení mohly být implementovány v rámci budoucích rozšíření.

### 9.1 Získávání rozvrhových dat

Vytvořené řešení pro správu univerzitních strojů by bylo vhodné v budoucnu rozšířit o možnost získání rozvrhových dat k učebnám, jež jsou v rámci řešení spravovány. Uvedená funkcionality by umožnila snadno zjistit obsazenost učebny přímo z aplikace pro správu strojů. Uživatel by tak přímo z aplikace zjistil, zda je vhodné na strojích v určité učebně vyvolat určitou akci (např. aktualizace OS) či nikoliv.

Pro správu rozvrhů používá Západočeská univerzita v Plzni systém IS/STAG [22i]. Získání rozvrhových dat je možno prostřednictvím REST API. Seznam všech dostupných služeb, jež poskytuje programové rozhraní systému IS/STAG provozovaném na ZČU, je dostupný online: [https://stag-ws.zcu.cz/ws/web?pp\\_local\\_e=cs&selectedTyp=REST&pp\\_reqType=render&pp\\_page=serviceList](https://stag-ws.zcu.cz/ws/web?pp_local_e=cs&selectedTyp=REST&pp_reqType=render&pp_page=serviceList).

Systém IS/STAG je využíván i na mnoha dalších českých univerzitách. Jmenovitě např.: Jihočeská univerzita v Českých Budějovicích, Univerzita Hradec Králové a Technická univerzita v Liberci. Výhoda využití shodného systému správy napříč univerzitami je zjevná. Pokud bude provedena integrace systému IS/STAG do vytvořeného řešení pro správu strojů, měla by daná implementace být ve velké míře využitelná i pokud bude produkt nasazen na ostatních jmenovaných univerzitách využívajících IS/STAG. Úplnou kompatibilitu integrace napříč univerzitami samozřejmě nelze zaručit a závisí na řadě faktorů.

Mezi omezeními, které je potřeba brát v potaz při přístupu k datům v systému IS/STAG, mohou být role v systému, resp. jejich přístupová práva. Každá univerzita může mít přístupová práva jednotlivých rolí nastavena odlišně a nelze tedy zaručit plnou kompatibilitu rozšiřujícího modulu pro získávání dat ze systému IS/STAG provozovaného na ZČU se systémem IS/STAG využívaným na jiných univerzitách.

## 9.2 Logování událostí

Je žádoucí rozšířit vytvořený produkt o evidování akcí, jež jsou v rámci systému pro správu strojů prováděny uživateli systému. Logování akcí vyvolaných na určitém stroji může být užitečné zejména pro administrátory projektu, kteří by na základě dostupných logů mohli detekovat pokus o provedení škodlivé činnosti, například neoprávněné získání uživatelských dat stroje.

V současné době je systém zabezpečen prostřednictvím autentizačního systému Kerberos a přístup do správy strojů je povolen pouze uživatelům, jejichž Orion login je schválen administrátory aplikace. Potencionální útočník by tedy pro přístup do systému správy strojů musel získat přístup k Orion účtu uživatele. Pokud by se útočník zmocnil účtu s běžnými privilegii, měl by přístup pouze k omezené množině strojů v systému.

Navíc i v případě, že by útočník získal přístup do systému, současné funkcionality aplikace mu neumožní provádění škodlivé činnosti. Vytvořené řešení pro správu umožňuje vyvolat pouze předdefinované, bezpečné akce, kterými jsou například aktualizování OS Windows, zobrazení aktuálně přihlášeného uživatele a zobrazení seznamu Chocolatey software.

Implementace logování bude nezbytná, pokud systém v budoucnu umožní spouštění vlastních skriptů uživatele. Spouštění skriptů by samozřejmě mohlo být snadno zneužito k výkonu škodlivé činnosti.

## 9.3 Implementace IPv6 adres

Vytvořené řešení pro správu strojů v současnosti pracuje s adresami typu IPv4. Produkt by mohl být rozšířen o možnost práce s adresami typu IPv6, což by umožnilo jeho využívání i v budoucnu.

V současné době jsou v prostředí ZČU KIV používány nástroje, jež identifikují stroje na základě IPv4 adres, a proto implementace IPv6 adresování nebyla při vývoji řešení pro správu strojů prioritou.

## 9.4 Rozšíření správy Linux

V současné době vytvořený produkt poskytuje, ve srovnání s poskytovanou funkcionalitou pro OS Windows, méně možností správy OS Linux. V dalších verzích produktu by mohlo být žádoucí implementovat obdobné funkcionality správy, jež produkt práce nabízí pro OS Windows. Správa OS Linux by tak mohla umožňovat například aktualizaci balíčků operačního systému prostřednictvím systému APT nebo Pacman.

Implementace více možností správy OS Linux je předmětem možného rozšíření, jelikož bylo potřeba řešit primárně správu stanic s operačním systémem Windows.

## 9.5 Rozšíření Android klienta

Android program by mohl být v budoucnu rozšířen o možnost práce s více aplikačními servery současně. V současné době klient umožňuje připojení pouze k jednomu aplikačnímu serveru, ze kterého jsou získávána data o strojích a jež je kontaktován za účelem provedení akce související se správou strojů. Cílový aplikační server, se kterým bude aplikace komunikovat, je volen při každém přihlášení uživatele. Při přechodu na jiný server se uživatel musí odhlásit od předchozího serveru a následně navázat spojení s dalším serverem aplikace.

Sdružování dat z více serverů by uživateli umožnilo správu počítačů, jež jsou registrovány na odlišných aplikačních serverech, bez nutnosti přepínání mezi servery aplikace. Implementace této funkce by vedla k ušetření času uživatele, jenž je vyžadován pro přepnutí serverů.

Implementace uvedené funkcionality nebyla při tvorbě aplikace prioritou, jelikož vytvořené řešení nutně nevyžaduje provoz více aplikačních serverů a při využití jednoho serveru není tato funkcionality relevantní. Provoz více serverů aplikace může být žádoucí v případě, že má být řešení využito pro funkci Wake-on-LAN a spravované sítě se nacházejí na více segmentech sítě.

## 9.6 Spouštění uživatelských skriptů

Řešení pro správu strojů by mohlo být v budoucnu rozšířeno o možnost spouštění skriptů vytvořených uživatelem aplikace. Funkcionality umožňující spuštění cizích skriptů na spravovaných strojích by ovšem mohla představovat potenciální nebezpečí. Pokud by došlo ke spuštění škodlivého skriptu, bylo by možné vážně poškodit software spravovaného stroje anebo neoprávněně získat uživatelská data.

Vzhledem k uvedeným důvodům by bylo vhodné, aby uvedená funkcionality byla dostupná pouze uživatelům systému s určitou privilegovanou rolí. Vlastní skripty by například mohli pouštět pouze administrátoři aplikace či uživatelé s jinou rolí, jež by pro tento účel byla speciálně vytvořena. Databáze pro tento účel obsahuje tabulku `role`, jež může být snadno rozšířena o roli umožňující spouštění skriptů.

## 9.7 Vytvoření iOS klienta

Vytvoření mobilního klienta pro operační systém iOS je žádoucí, aby bylo osobám využívajícím mobilní zařízení od společnosti Apple umožněno snadno přistupovat k řešení pro správu strojů. V současné době mohou uživatelé operačního systému

## 9. Možná rozšíření produktu

---

iOS k vytvořenému řešení přistupovat pomocí webového prohlížeče, jelikož vytvořený web je možno obsluhovat i z mobilního zařízení. Responzivní web byl testován na iPhone 13 Pro s iOS 16 a prohlížečem Safari. Web pracoval na uvedeném zařízení dle očekávání a umožnil obsluhu veškerých funkcionalit řešení, ke kterým je umožněn přístup i na počítači.

Cílem práce bylo navrhnout a vytvořit komplexní řešení pro správu počítačových stanic, jež je určené zejména pro nasazení v univerzitním prostředí. Při vývoji produktu měl být kladen důraz zejména na snadnou nasaditelnost řešení a jeho platformní nezávislost. Rovněž bylo žádoucí poskytnout v rámci řešení co nejvíce funkcí, jež by vedly k usnadnění správy univerzitních strojů. Funkcionalita nového produktu pro správu strojů byla navržena zejména s přihlédnutím k potřebám správy stanic na katedře KIV ZČU.

Před zahájením návrhu nového řešení byla provedena analýza existujících řešení určených pro správu vzdálených strojů. V rámci analýzy bylo zjištěno, že existující nástroje pro správu poskytují dostatečné množství funkcí, avšak vyžadují platbu na měsíční bázi. Výše požadované finanční částky za užívání produktu je zpravidla závislá na množství zpřístupněných funkcí produktu a počtu spravovaných stanic. Analyzovaná řešení navíc neposkytují Docker image či alternativu, jež by umožnila snadné nasazení řešení pro správu strojů.

Na základě provedené analýzy byl sestaven návrh nového řešení pro správu počítačových stanic a následně byla zahájena realizace návrhu. V rámci této práce byl vytvořen server aplikace, jež vykonává operace nad spravovanými stroji. S aplikačním serverem je uživateli umožněno komunikovat pomocí webové aplikace nebo mobilního klienta, které jsou rovněž součástí této práce. Jelikož aplikační server poskytuje zabezpečené programové rozhraní, mohou být v budoucnu vytvořeny alternativní klienti řešení pro správu strojů.

Vývoj aplikačního serveru a webové aplikace univerzitního řešení pro správu strojů probíhal ve vývojovém prostředí IntelliJ IDEA s využitím jazyka Java a frameworku Spring Boot. Mobilní klient řešení pro vzdálenou správu strojů byl vytvořen s pomocí nástroje Android Studio a využitím stejného programovacího jazyka, jež byl použit pro tvorbu webové části projektu. Platformní nezávislost a snadné nasazení vytvořeného produktu zajišťuje kontejnerová technologie Docker, prostřednictvím které je distribuována webová část projektu.

Webová komponenta vytvořeného řešení vyžaduje provoz MySQL databáze a webového serveru Apache Tomcat. Obě zmíněné komponenty jsou provozovány

v rámci Docker kontejnerů a není vyžadována ruční konfigurace těchto nástrojů. K webové aplikaci je možno přistoupit z velkého množství zařízení, jež mají internetový prohlížeč. K webové verzi produktu lze přistoupit i pomocí mobilních webových prohlížečů, kterými jsou například Safari od společnosti Apple a Google Chrome. Vytvořeného mobilního klienta je možno provozovat na zařízeních se systémem Android 5.0 či vyšší.

Komponenty aplikace pro správu byly důsledně otestovány. Za účelem ověření funkcionality webové aplikace a mobilního klienta bylo celkem vytvořeno 20 testovacích scénářů. Pro otestování webové aplikace byly navíc vytvořeny i automatické JUnit testy. Testování mobilní aplikace probíhalo na několika fyzických i virtuálních zařízeních. Během testů aplikace nebyly zaznamenány neočekávané pády programu či jiné problémy znepříjemňující práci s aplikací. Nasazení serverové části projektu bylo otestováno na fyzickém stroji s OS Linux i Windows, aby bylo ověřeno, že je řešení multiplatformní.

Vytvořený systém pro univerzitní správu strojů implementuje požadované funkcionality. Nově vytvořené řešení umožňuje například vzdálené zapnutí stroje, správu aktualizací operačního systému, práci s uživateli počítače a další funkcionality, jež byly popsány v rámci této práce. Součástí diplomové práce je i návrh dalších rozšíření, kterými lze systém v budoucnu dále rozvíjet.

# Instalační příručka webové aplikace



Nasazení vytvořeného webového řešení vyžaduje vykonání čtyř hlavních kroků, následující text je proto rozdělen do čtyř sekcí. Na stroje, jež mají být prostřednictvím řešení pro vzdálenou správu ovládány, je potřeba nahrát veřejný SSH klíč. V případě, že bude řešení využito pro správu aktualizací systému Windows, je potřeba na spravovaný stroj nainstalovat nástroj PSWindowsUpdate. Obdobně je potřeba provést instalaci software Chocolatey na stroje, u kterých má být umožněna správa instalovaného software pomocí uvedeného balíčkovacího nástroje. Posledním nutným krokem, jenž je nutné provést pro úspěšné nasazení systému, je spuštění serveru řešení. Obsluhu spravovaných zařízení zajišťuje server, který lze provozovat na jakémkoliv stroji, jenž je vhodný pro běh systému Docker.

## A.1 Nahrání SSH klíče na spravovaný stroj

Pro zpřístupnění základních funkcionalit správy spravovaných strojů, kterými jsou například získání seznamu uživatelů stroje a získání základních informací o stroji, stačí nahrát na dané zařízení veřejný SSH klíč. Výchozí veřejný klíč se nachází v dané složce s projektem v rámci souboru `public_key.txt`. Přidání veřejného klíče na cílový stroj je možno docílit editací souboru `authorized_keys`. Výchozí lokací souborů s veřejnými klíči je na Linuxových systémech

`~/.ssh/authorized_keys`. Veřejné klíče uživatelů s administrátorskými právy se na systému Windows nachází v souboru

`C:\ProgramData\ssh\administrators_authorized_keys`, který je nutno pro užívání řešení pro správu editovat. Veřejné klíče běžných uživatelů systému Windows se nachází v rámci souboru `authorized_keys`, jenž je ve složce `.ssh` domovského adresáře uživatele. Seznam veřejných klíčů běžných uživatelů není potřeba pro užívání řešení editovat. Uvedené cesty a názvy souboru veřejných klíčů jsou platné pouze za předpokladu, že je na klientských strojích provozován OpenSSH server s výchozí konfigurací.

Možný postup nahrání SSH klíče na spravovaný stroj:

1. Vykopírování obsahu dodaného souboru `public_key.txt`, jenž se nachází ve složce Docker Compose do schránky.
2. Vložení obsahu schránky na konec souboru, který obsahuje veřejné SSH klíče administrátorů:
  - Windows:C:\ProgramData\ssh\administrators\_authorized\_keys,
  - Linux:~/ .ssh/authorized\_keys.

Z bezpečnostních důvodů autor řešení doporučuje vygenerování vlastní dvojice SSH klíčů (privátní a veřejný), jež bude použita v rámci řešení pro správu strojů. Při využití vlastního veřejného SSH klíče musí být změněn i obsah souboru `id_rsa`, jenž obsahuje privátní klíč použitý na serveru.

## A.2 Instalace PSWindowsUpdate na spravovaný stroj

Pro umožnění správy aktualizací systému Windows je potřeba na počítače, jež jsou v rámci řešení spravovány, nutné nainstalovat program „PSWindowsUpdate“. Uvedený Powershell software je v projektu využit pro aktualizování strojů a získání historie aktualizací. Více informací o rozšíření lze nalézt na stránce <https://www.powershellgallery.com/packages/PSWindowsUpdate>.

Postup instalace probíhá z prostředí Powershell a je velmi přímočarý. Pro zahájení instalace spusťte prostředí Windows Powershell jako administrátor a vykonajte následující příkaz:

```
Install-Module -Name PSWindowsUpdate
```

Po vykonání příkazu můžete být upozorněni na skutečnost, že na stroji prozatím není nainstalován balíčkový systém NuGet, a současně vás instalátor vybídne k jeho instalaci. Pokud se tak stane, potvrďte instalaci NuGet volbou Y.

Počítač rovněž může vygenerovat upozornění informující o pokusu instalace Powershell modulu z neznámého zdroje. Pokud bude upozornění vyobrazeno, potvrďte instalaci volbou Y.

## A.3 Instalace Chocolatey na spravovaný stroj

Nově vytvořený nástroj pro správu zařízení umožňuje provádět aktualizace Chocolatey software a rovněž obsahuje funkcionalitu umožňující získat seznam Chocolatey balíčků, jež jsou nainstalovány na spravovaných strojích. Pro využití této



funkcionality je nutné, aby na spravovaném stroji byl nainstalován program Chocolatey. Instalace aplikace Chocolatey probíhá, stejně jako u produktu PSWindowsUpdate, prostřednictvím Windows Powershell. Pro zahájení instalace uvedeného balíčkovacího systému vykonajte s právy administrátora příkaz:

```
1 Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.
  ServicePointManager]::SecurityProtocol = [System.Net.
  ServicePointManager]::SecurityProtocol -bor 3072; iex ((
  New-Object System.Net.WebClient).DownloadString('https://
  community.chocolatey.org/install.ps1'))
```

Uvedený skript povolí instalaci programů z neznámých zdrojů a následně zahájí instalaci programu. Více informací o instalaci Chocolatey lze nalézt na stránkách vývojáře produktu <https://chocolatey.org/install>.

## A.4 Spuštění serveru pomocí Docker

Pro sestavení a následné nasazení serverové části projektu je potřeba mít nainstalovaný kontejnerizační systém Docker. Uvedený systém pro kontejnerizaci aplikací je dostupný pro všechny tři majoritní desktopové operační systémy a konkrétní postup instalace je uveden na webových stránkách vývojáře daného produktu: <https://www.docker.com/products/docker-desktop/>. Více informací systému Docker obsahuje kapitola 5.1.2 této práce.

Nově vytvořené řešení pro správu vyžaduje definici jednoho uživatele, jenž bude mít roli administrátora, a daný uživatel následně zajistí přidání dalších uživatelů do systému. Prvotní administrátor je specifikován v rámci SQL skriptu pojmenovaného 01.sql, jenž se nachází ve složce ini t serverové části řešení. Uvedený soubor na řádce 144 obsahuje údaje o prvním administrátorovi řešení. Patříčně upravte řádku tak, aby odpovídala údajům uživatele, jenž se do systému přihlásí jako první. Popis položek, jež je ve skriptu potřeba změnit, je následující:

- login - Orion login uživatele, jenž bude prvním administrátorem,
- first\_name - křestní jméno administrátora,
- last\_name - příjmení administrátora,
- create\_datetime - datum přidání uživatele do systému.

Po provedení požadovaných úprav provedte uložení skriptu a můžete pokračovat v procesu nasazení řešení pro správu strojů.

Po instalaci systému Docker a provedení požadovaných úprav SQL skriptu proveďte spuštění nástroje Docker. Po inicializaci systému Docker stačí v dodané složce

## *A. Instalační příručka webové aplikace*

---

s projektem označené jako `server` vykonat příkaz `docker compose up`, jenž zajistí sestavení a nasazení řešení. Po provedení příkazu počkejte na start kontejnerů Apache Tomcat a MySQL databáze. Pokud nasazení řešení pro správu strojů proběhlo úspěšně, je na lokálním stroji k aplikaci možno přistoupit pomocí odkazu `http://localhost:8080/user/login`.

# Instalační příručka mobilní aplikace



Proces zprovoznění mobilního klienta je potřeba zahájit přenosem instalačního souboru „magicwol.apk“ na cílové mobilní zařízení s operačním systémem Android. Zahájení instalace klienta správy strojů na zařízení proveďte klepnutím na název uvedeného instalačního souboru aplikace v některém ze souborových manažerů. Následně dokončete instalaci programu následováním vyobrazených instrukcí. Průběh instalace je zpravidla přímočarý, ale postup instalace se může na každém zařízení mírně lišit v závislosti na použitém správci souborů.

Při pokusu o instalaci mobilního klienta může být uživatel upozorněn na fakt, že zařízení má z bezpečnostních důvodů zakázanou instalaci aplikací, jež pocházejí z jiného zdroje než Google Play. Povolení instalace aplikací z neoficiálních zdrojů lze provést v nastavení zařízení. Většinou je volba povolující instalaci z neznámých zdrojů označena jako „Neznámé zdroje“ a nachází se v sekci nastavení zabezpečení zařízení. Konkrétní postup povolení neznámých zdrojů software ovšem závisí na nadstavbě operačního systému Android, kterou využívá výrobce mobilního zařízení.

Po dokončení procesu instalace aplikace by měl být mobilní klient „MagicWOL“ dostupný z hlavního menu mobilního zařízení.



# Uživatelská příručka webové aplikace

## C

Příloha obsahuje návody pro provedení typických akcí, jež může chtít uživatel v rámci aplikace vykonat. Uvedené akce jsou děleny do kategorií podle částí programu, ke kterým se vztahují. Detailní postupy pro provedení většiny činností, jež program umožňuje, jsou předmětem testovacích scénářů, které jsou uvedeny v kapitole 8.

## C.1 Evidence strojů

Tato kategorie popisuje typické akce související s evidencí strojů, jež může chtít uživatel v programu vykonat.

### C.1.1 Vytvoření seznamu strojů

Vytvoření nového seznamu strojů lze zahájit klepnutím na položku „Vytvořit seznam“, jež se nachází v nabídce „Moje zařízení“ horního navigačního panelu aplikace. Po volbě dané položky se zobrazí modální okno, v rámci kterého musí uživatel vyplnit požadované informace o novém seznamu zařízení (viz obrázek C.1). Po vyplnění požadovaných informací potvrďte přidání seznamu klepnutím na tlačítko s nápisem „OK“.

Kompletní instrukce pro přidání nového seznamu strojů jsou zahrnuty v testovacím scénáři 8.3.1.



Nový seznam strojů

název seznamu

popis seznamu

Zrušit OK

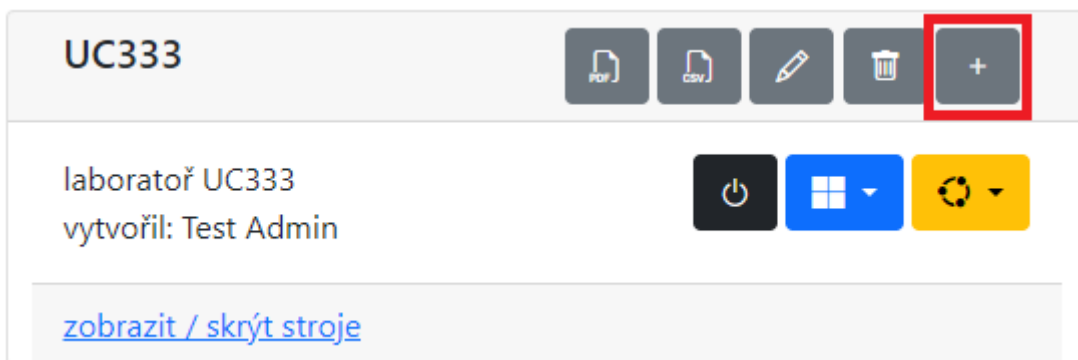
Obrázek C.1: Přidání seznamu zařízení

### C.1.2 Přidání stroje do seznamu

Vložení nového stroje do seznamu zahájíte klepnutím na tlačítko s ikonou znaménka +, jež se nachází uvnitř karty seznamu, do kterého má být nové zařízení přidáno (vyznačeno na snímku C.2). Po volbě položky se zobrazí modální okno, v němž vyplňte informace o novém stroji a potvrďte přidání stroje stiskem tlačítka „OK“.

V případě, že je přihlášená osoba administrátorem, nevyžaduje přidání zařízení do seznamu další akce. Pokud je osoba přidávající zařízení běžným uživatelem, podléhá přidání zařízení schválení některým z administrátorů aplikace.

Detailní dokumentaci postupu podání žádosti o registraci nového zařízení běžným uživatelem obsahuje testovací scénář 8.3.3. Kompletní instrukce přidání zařízení administrátorem aplikace pro správu strojů popisuje testovací scénář 8.3.2.

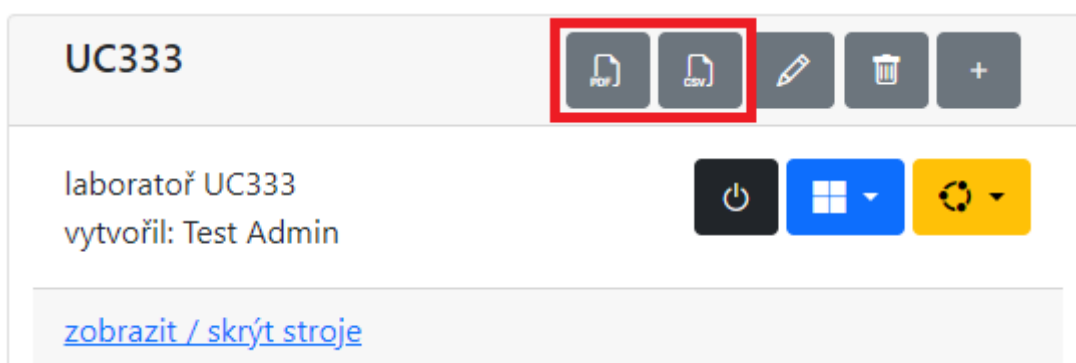


Obrázek C.2: Přidání zařízení do seznamu

### C.1.3 Export seznamu strojů do PDF / CSV

Export údajů zařízení přítomných v seznamu lze vykonat stisknutím jednoho z tlačítek s ikonou dokumentu, jež jsou dostupná v rámci karty reprezentující seznam strojů.

Tlačítko s nápisem pdf slouží pro export informací o strojích do dokumentu stejnojmenného formátu. Analogicky tlačítko označené nápisem csv slouží pro export dat do odpovídajícího formátu. Obě tlačítka umožňující export jsou vyznačena na obrázku C.3.



Obrázek C.3: Export údajů zařízení

### C.1.4 Import seznamu strojů z CSV

Informace o strojích, jež byly exportovány pomocí funkce popsané v kapitole C.1.3, lze do systému opětovně nahrát. K uvedenému účelu slouží funkce importu, kterou lze vyvolat stisknutím odpovídajícího tlačítka nacházejícího se v rámci karty reprezentující seznam strojů (viz obrázek C.4). Po stisknutí příslušné ikony se zobrazí modální okno, ve kterém následně vyberte csv soubor obsahující údaje o strojích a proveďte import stisknutím tlačítka s nápisem „Importovat“.



Obrázek C.4: Import údajů zařízení

Přidání strojů má aditivní charakter, tedy pokud cílový seznam již obsahuje některé položky, nedojde k jejich nahrazení. Ze vstupního souboru jsou přidány pouze položky, ke kterým neexistuje v systému alternativa - tzn. mají MAC adresu, IP adresu a hostname, jež doposud není přiřazeno žádnému zařízení v systému.

Kompletní instrukce pro export dat a jejich následný import do seznamu uvádí testovací scénář 8.3.6.

### C.1.5 Rozhodnutí žádosti o registraci stroje

Pokud má přihlášený uživatel roli administrátora, má pravomoc rozhodovat o schválení či zamítnutí požadavků na přidání nových zařízení, jež vytvořili běžní uživatelé. Administrátor aplikace může uživatelské žádosti o přidělení stroje zobrazit v sekci „Přidělení strojů“, jež se nachází v záložce „Žádosti“. V rámci dané části programu administrátor vidí všechny dosud nerozhodnuté žádosti spolu s detaily každé žádosti. Mezi informace, jež jsou uvedeny u každé žádosti, patří například jméno žadatele, Orion login žadatele a IP adresa stroje, jež má být přidán (viz snímek C.5). Každou žádost je možno schválit či zamítnout pomocí odpovídajících tlačítek, jež se nacházejí v zápatí každé z karet reprezentující žádost.



**Ondřej Drtina**

Orion login: drtinao  
Datum požadavku: 02.05.23 17:46  
Přidat do seznamu: Drtina PC  
Název stroje: PC kancelář  
MAC stroje: D0-06-AF-09-A1-F3  
IP stroje: 147.228.63.103  
Hostname stroje: un326p02-kiv.fav.zcu.cz

Zamítnout
Schválit

Obrázek C.5: Karta reprezentující žádost o registraci stroje

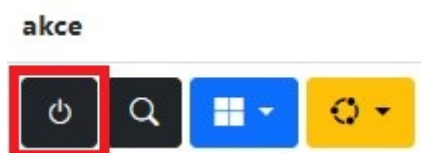
## C.2 Ovládání strojů

Tato sekce textu se zaměřuje na popis typických činností, jež může chtít uživatel řešení pro správu strojů provést v souvislosti s ovládáním spravovaných strojů.

### C.2.1 Zapnutí zařízení

Zapnutí konkrétního zařízení je možno provést klepnutím na ikonu zapínacího tlačítka, jež se nachází v řádce seznamu strojů, která reprezentuje dané zařízení. Zapínací tlačítko je vyznačeno na obrázku C.6. Analogicky lze shodným tlačítkem, které je obsaženo v rámci karty reprezentující seznam, zapnout všechna zařízení v seznamu.

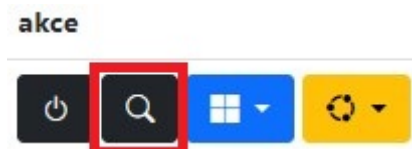
Popis veškerých kroků, jež je nutno vykonat k zapnutí zařízení, obsahuje testovací scénář 8.4.1.



Obrázek C.6: Tlačítko umožňující zapnutí stroje

## C.2.2 Zjištění dostupnosti stroje

Zda je určitý stroj zapnutý je možno zjistit stiskem tlačítka s ikonou lupy, které je umístěno v řádku seznamu strojů, jenž reprezentuje dané zařízení (viz obrázek C.7). Po klepnutí na dané tlačítko proběhne kontrola dostupnosti stroje a výsledek dostupnosti je zobrazen v horní části webu. Čas, jenž je vymezen pro kontaktování stroje je omezen na 5000 ms.



Obrázek C.7: Tlačítko pro kontrolu dostupnosti stroje

## C.2.3 Správa aktualizací Windows

Nabídku dostupných možností souvisejících se správou aktualizací systému Windows lze vyvolat stisknutím tlačítka s logem systému Windows a následnou volbou možnosti Windows Update (viz snímek C.8). Akce spojené s aktualizací operačního systému je možno vyvolat buď na úrovni seznamu či na úrovni jednotlivých zařízení.

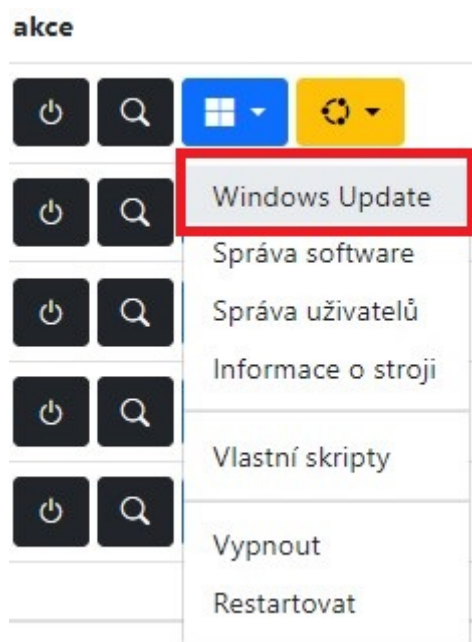
Po zvolení možnosti Windows Update bude vyobrazena nabídka všech akcí, jež je možno ve spojitosti s aktualizacemi operačního systému Windows vykonat. Aplikace například umožňuje vyvolat proces aktualizace nebo zobrazit seznam aktualizací, jež byly nedávno nainstalovány. Z vyobrazené nabídky zvolte požadovanou akci klepnutím na některou z karet reprezentujících dostupné volby.

Při vyvolání aktualizacího procesu dojde k naplánování úlohy v systému Windows, jež je následně ihned spuštěna. Uživatel nevidí průběh aktualizace v systému pro správu strojů, avšak průběh operace je logován do souboru `C:\sw\PSWindowsUpdateLog.txt`, jenž se nachází na spravovaném zařízení.

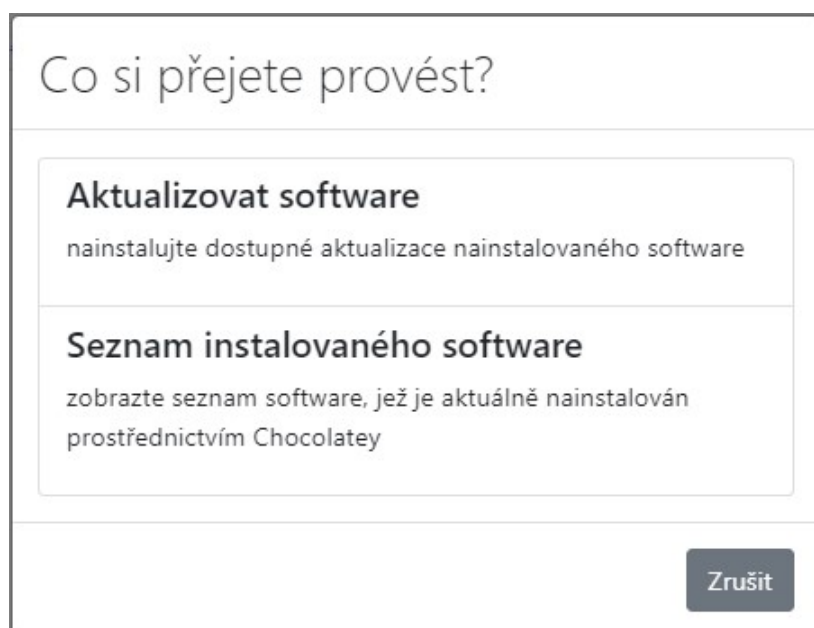
Detailní popis postupu, který vede k aktualizování operačního systému Windows, obsahuje testovací scénář 8.4.2.

## C.2.4 Správa balíčků Chocolatey

Proces správy balíčků Chocolatey zahajte stisknutím tlačítka s logem reprezentující operační systém Windows, jenž se nachází na úrovni seznamu či jednotlivých zařízení v seznamu. Následně z vyobrazené roletkové nabídky zvolte možnost Správa software. Zobrazí se modální okno obsahující seznam dostupných možností, jež jsou spojeny se správou Chocolatey balíčků (viz obrázek C.9). Proveďte volbu některé z vyobrazených možností správy pomocí klepnutí na odpovídající kartu.



Obrázek C.8: Volba položky Windows Update



Obrázek C.9: Dostupné možnosti správy Chocolatey balíčků

## C.2.5 Správa uživatelů Windows

Možnosti správy uživatelů systému Windows zobrazíte klepnutím na tlačítko s logem systému Windows a následně zvolením položky Správa uživatelů, jež se

nachází v rámci vyobrazené roletkové nabídky. Tlačítko s logem systému Windows je možno nalézt v rámci každé karty reprezentující seznam zařízení i na úrovni jednotlivých jednotlivých zařízení v seznamu. Funkcionality související se správou uživatelů Windows je tedy možno vyvolat na úrovni celého seznamu strojů i na úrovni jednotlivých zařízení.

Po zobrazení nabídky dostupných možností správy uživatelů systému Windows zvolte některou z vyobrazených možností a počkejte na zobrazení stránky s nadpisem Výsledek operace. Vyobrazená stránka obsahuje SSH log provedené akce pro všechna zařízení, na kterých byla operace vykonána. Získání seznamu lokálních účtů na dvou univerzitních strojích současně znázorňuje obrázek C.10. Na uvedeném obrázku lze pozorovat, že oba stroje mají shodné lokální účty, což je v daném případě žádoucí.

```
Výsledek operace

Níže jsou dostupné výsledky operace na jednotlivých strojích.

[1] 14:41:34 [SUCCESS] Administrator@147.228.63.101

Name                Caption                Domain
----                -
Administrator       UC333P01-KIV\Administrator UC333P01-KIV
DefaultAccount      UC333P01-KIV\DefaultAccount UC333P01-KIV
dod                  UC333P01-KIV\dod        UC333P01-KIV
Guest                UC333P01-KIV\Guest      UC333P01-KIV
vyuka                UC333P01-KIV\vyuka      UC333P01-KIV
WDAGUtilityAccount UC333P01-KIV\WDAGUtilityAccount UC333P01-KIV

[2] 14:41:34 [SUCCESS] Administrator@147.228.63.102

Name                Caption                Domain
----                -
Administrator       UC333P02-KIV\Administrator UC333P02-KIV
DefaultAccount      UC333P02-KIV\DefaultAccount UC333P02-KIV
dod                  UC333P02-KIV\dod        UC333P02-KIV
Guest                UC333P02-KIV\Guest      UC333P02-KIV
vyuka                UC333P02-KIV\vyuka      UC333P02-KIV
WDAGUtilityAccount UC333P02-KIV\WDAGUtilityAccount UC333P02-KIV
```

Obrázek C.10: Získání seznamu lokálních účtů na více zařízeních současně

## C.2.6 Vypnutí / restart zařízení

Vypnutí či restart spravovaného zařízení je možno provést pomocí volby „Vypnout“, resp. „Restartovat“, jež se nachází v nabídce poskytovaných služeb souvisejících se systémem Windows. Uvedenou nabídku vyvoláte stiskem tlačítka s logem Windows, jež se nachází u každého spravovaného zařízení, resp. seznamu zařízení. Spravované zařízení musí být pro úspěšné dokončení akce zapnuté v operačním systému Windows.

## C.3 Správa uživatelů

Následující podkapitoly obsahují postup při vykonávání typických činností, které může chtít uživatel programu vykonat v souvislosti se správou uživatelů řešení pro správu strojů.

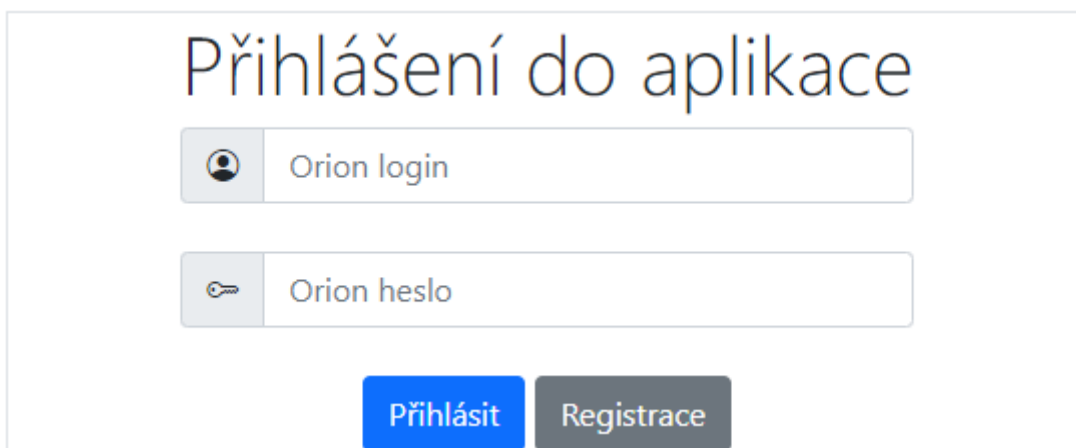
### C.3.1 Přihlášení uživatele

Přihlášení uživatele je možné po přechodu na domovskou stránku aplikace, kterou je `http://localhost:8080/login`. Uvedená adresa je platná za předpokladu, že webová aplikace je provozována na lokálním stroji. Pokud je aplikace hostována na jiném počítači, je potřeba změnit řetězec `localhost` v uvedené adrese na IP adresu či hostname stroje, jenž hostuje webový program.

Po přechodu na uvedenou stránku by měla být v okně prohlížeče viditelná stránka s přihlašovacím oknem (viz snímek obrazovky C.11). Proveďte vyplnění požadovaných údajů (Orion login + odpovídající heslo) a proveďte přihlášení stiskem tlačítka „Přihlásit“.

Konkrétní kroky, jež je potřeba vykonat pro přihlášení uživatele, jsou obsaženy v rámci testovacího scénáře 8.2.1 (běžný uživatel) a 8.2.2 (administrátor).

Přihlášení bude fungovat pouze s Orion loginem, jenž je v systému zaregistrován. Každá podaná žádost o registraci uživatele podléhá schválení některým z administrátorů.



Přihlášení do aplikace

Orion login

Orion heslo

Přihlásit Registrace

Obrázek C.11: Přihlašovací obrazovka webové aplikace

### C.3.2 Podání žádosti o registraci

Pro registraci nového uživatele přejděte na domovskou stránku aplikace a stiskněte tlačítko „Registrace“. Následně ve vyobrazeném modálním okně vyplňte veškeré požadované údaje, mezi které patří Orion login a jméno uživatele. Potvrďte odeslání žádosti o registraci stiskem tlačítka „OK“. Po odeslání registrační žádosti budete přesměrováni na hlavní stránku aplikace a musíte vyčkat na schválení registrace některým z administrátorů řešení pro správu strojů.

Kompletní seznam kroků, jež je potřeba vykonat pro podání žádosti o registraci uživatele, obsahuje testovací scénář 8.2.3.

### C.3.3 Rozhodnutí žádosti o registraci uživatele

Administrátoři řešení pro správu strojů mohou schválit či zamítnout čekající registrační žádosti uživatelů. Čekající žádosti jsou dostupné pod odkazem „Schválení registrací“, jež se nachází v sekci „Žádosti“ horního navigačního panelu aplikace. Po přechodu na uvedený odkaz uživatel uvidí veškeré registrační žádosti, jež doposud nebyly rozhodnuty. Schválení či zamítnutí každé z žádostí je možné provést klepnutím na odpovídající tlačítka, jež se nachází ve spodní části každé karty, která reprezentuje uživatelskou žádost o registraci (viz obrázek C.12).

Veškeré kroky, jež je nutné provést pro schválení registrace uživatele jsou uvedeny v testovacím scénáři 8.2.4.

## Nevyřízené žádosti

<b>Ondřej Drtina</b>
Orion login: drtinao Datum registrace: 03.05.23 15:55
<input type="button" value="Zamítnout"/> <input type="button" value="Schválit"/>

Obrázek C.12: Tlačítka umožňující schválení, resp. zamítnutí registrační žádosti





# Uživatelská příručka mobilní aplikace

## D

Níže jsou uvedeny postupy pro vykonání typických činností, jež může chtít uživatel provést pomocí mobilního klienta, jenž byl v rámci práce vytvořen. Následující text je rozdělen do dvou sekcí. První kategorie obsahuje postupy pro provedení činností, k jejichž úspěšnému vykonání musí být klient připojen k aplikačnímu serveru. Předmětem druhé sekce textu jsou postupy, které se týkají funkcionalit, jež aplikace poskytuje ve spojitosti se správou lokální sítě a které nejsou podmíněny připojením k serveru aplikace.

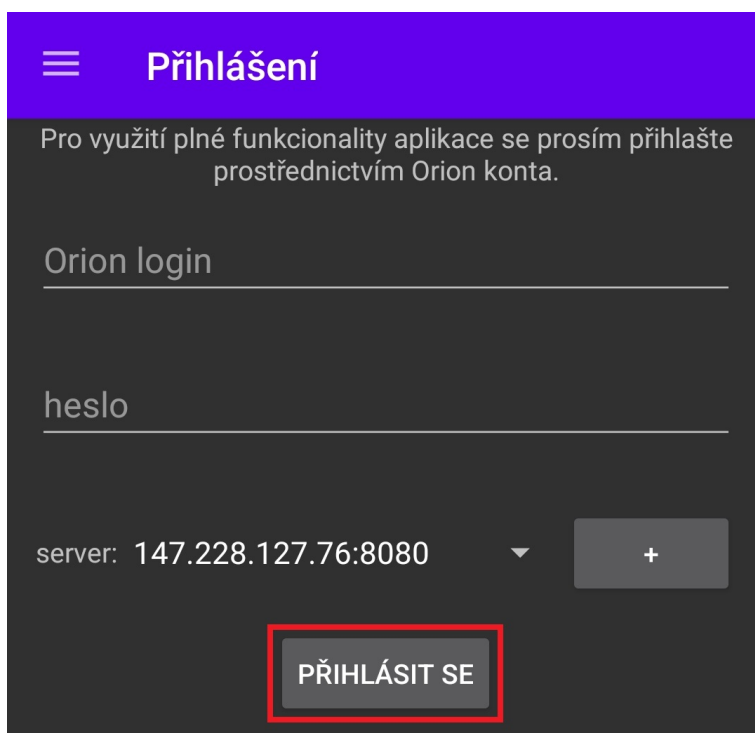
## D.1 Služby poskytované aplikačním serverem

Tato sekce obsahuje obvyklé postupy, kterými lze docílit vykonání typických akcí, jež jsou podmíněny připojením ke vzdálenému aplikačnímu serveru.

### D.1.1 Přihlášení uživatele

Přihlášení uživatele k serveru aplikace zahajte přechodem na kartu „Přihlášení“ nacházející se v sekci „Další možnosti“ postranního navigačního panelu. Po přechodu na uvedenou kartu vyplňte požadované informace, kterými jsou Orion login registrovaného uživatele produktu a odpovídající heslo. Následně zvolte z roletkové nabídky aplikační server, ke kterému se chcete připojit. V případě, že zatím nejsou žádné servery aplikace přidány, stiskněte tlačítko se symbolem „+“ a přidejte nový aplikační server. Po rekapitulaci zadaných informací proveďte přihlášení stiskem tlačítka „Přihlásit se“, jež je vyznačeno na obrázku D.1.

Při úspěšném přihlášení se zobrazí dialogové okno s hláškou, jež uživatele informuje o roli, kterou má v rámci systému přidělenou. V případě, že jsou zadány neplatné přihlašovací údaje, je o této skutečnosti uživatel rovněž informován prostřednictvím dialogového okna.



Obrázek D.1: Tlačítko pro odeslání přihlašovacích dat

Detailní popis kroků, jež je potřeba vykonat pro přihlášení uživatele obsahuje testovací scénář 8.2.2 (administrátor), resp. 8.2.1 (běžný uživatel).

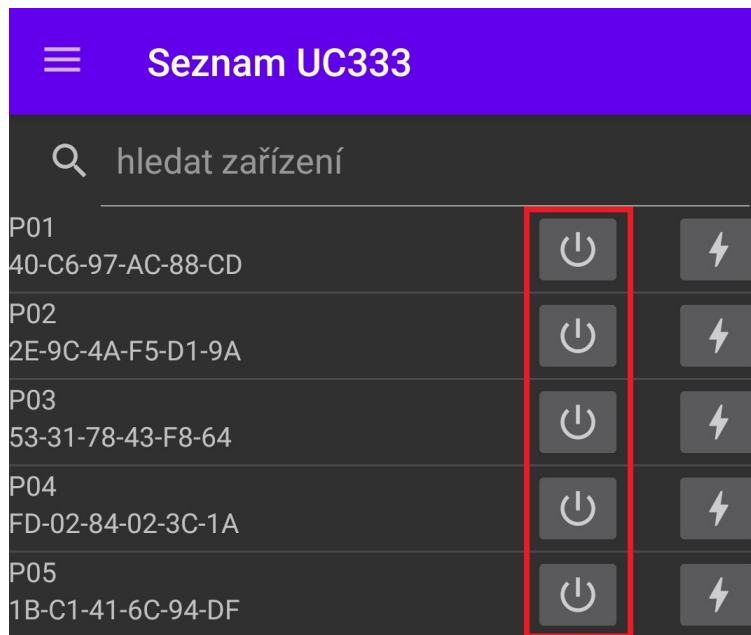
Přihlášení je možno pouze s Orion loginem, jež je v systému zaregistrován. Veškeré žádosti o registraci musí být schváleny administrátorem řešení pro správu strojů.

## D.1.2 Zapnutí zařízení

Pro zapnutí vzdáleného zařízení nejprve proveďte přechod na kartu „Moje zařízení“, která se nachází v sekci „Univerzita“ navigačního panelu aplikace. Klepněte na kartu seznamu, jež obsahuje počítač, který má být zapnut. Po zobrazení seznamu strojů ve zvoleném seznamu klepněte na ikonu zapínacího tlačítka nacházejícího se u stroje, jež má být zapnut. Zapínací tlačítka jsou vyznačena na obrázku č. D.2.

Aplikace umožňuje i zapnutí všech zařízení, jež se nacházejí v rámci seznamu strojů. Pro zapnutí více strojů stiskněte ikonu zapínacího tlačítka, jež se nachází na úrovni seznamu zařízení.

Všechny kroky, jež je potřeba vykonat pro zapnutí zařízení, jsou uvedeny v testovacím scénáři 8.4.1.



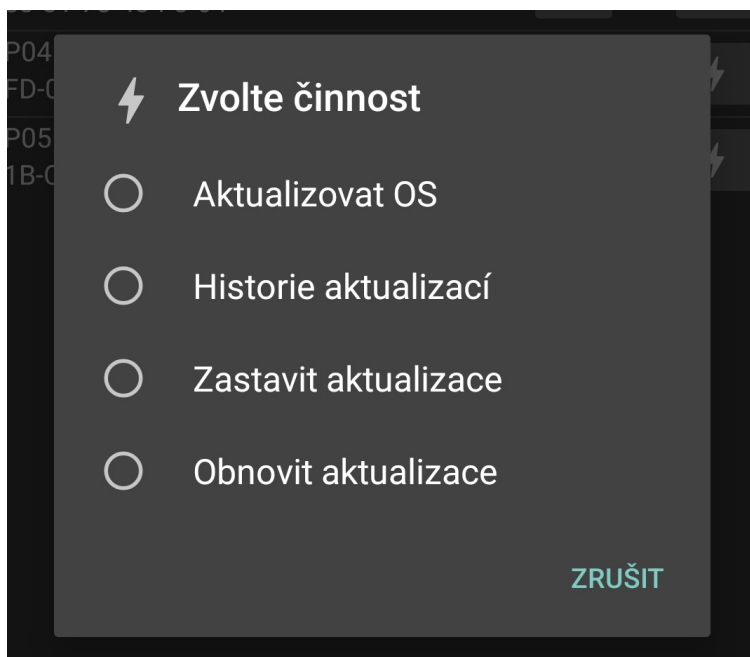
Obrázek D.2: Tlačítka umožňující zapnutí zařízení

### D.1.3 Správa aktualizací Windows

Zobrazení nabídky všech akcí, jež řešení nabízí ve spojitosti se správou aktualizací operačního systému Windows, zahajte přechodem na kartu „Moje zařízení“ nacházející se v kategorii „Univerzita“ navigačního panelu programu. Pokračujte v procesu klepnutím na kartu, jež reprezentuje seznam obsahující vybrané zařízení. Následně stiskněte tlačítko s ikonou blesku, které se nachází v rámci karty reprezentující požadovaný stroj. Po vyobrazení nabídky operačního systému zvolte položku „Windows“ a pokračujte volbou kategorie „Windows Update“. Zvolte požadovanou akci z vyobrazeného seznamu činností. Veškeré nabízené akce, jež řešení umožňuje vykonat ve spojitosti se správou aktualizací Windows, jsou vyobrazeny na snímku obrazovky D.3. Mezi nabízené možnosti patří aktualizování systému, zobrazení historie aktualizací, zastavení aktualizací či jejich opětovné povolení.

Po kontaktování aplikačního serveru bude vyobrazen SSH log informující o výsledku procesu. Aplikace samozřejmě umožňuje vyvolat akce související se správou aktualizací Windows i na úrovni celého seznamu počítačů. Pro provedení akce nad celým seznamem klepněte na ikonu blesku u požadovaného seznamu a dále postupujte analogicky k již uvedenému postupu.

Dokumentace veškerých kroků, jež je potřeba vykonat pro aktualizaci operačního systému Windows, je předmětem testovacího scénáře 8.4.2.



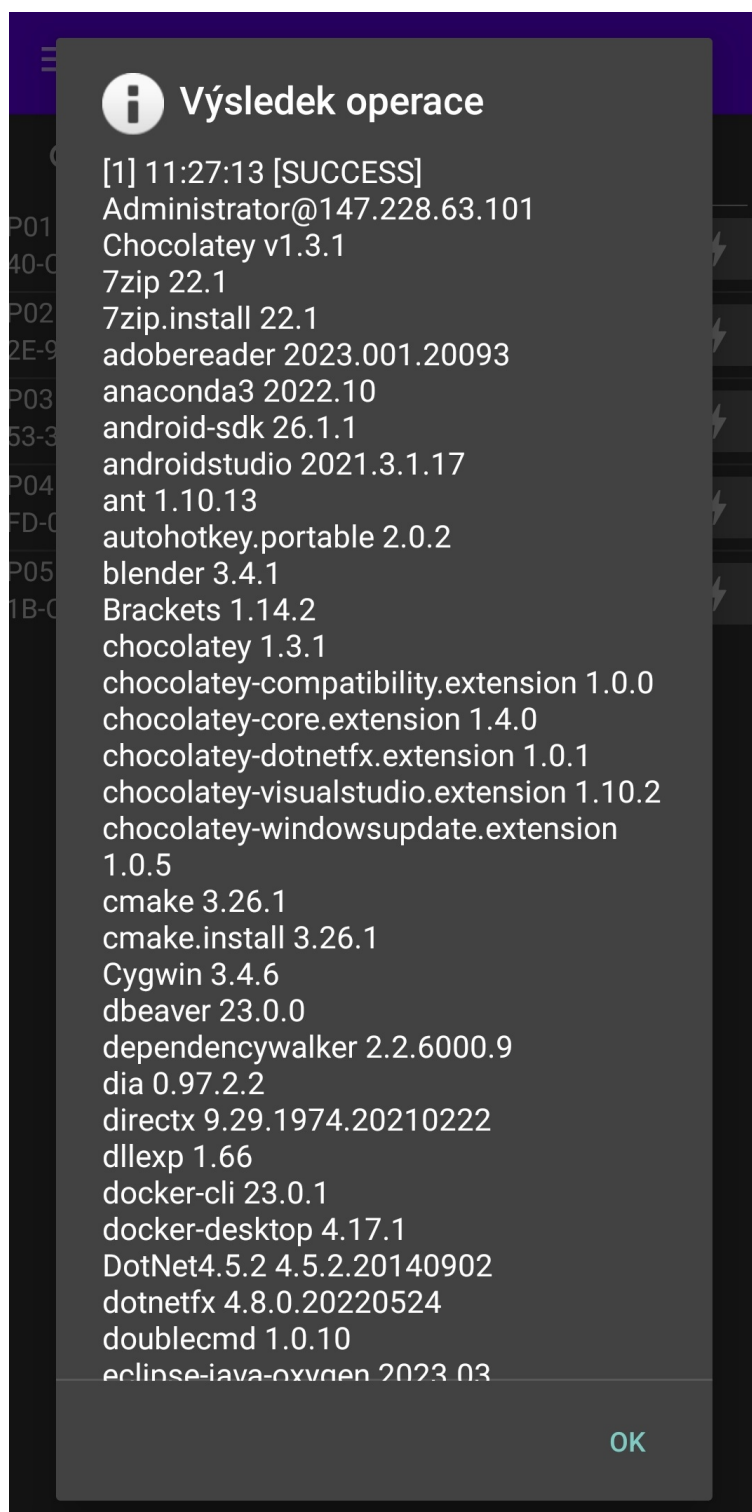
Obrázek D.3: Dostupné akce Windows Update

#### D.1.4 Správa balíčků Chocolatey

Zobrazení seznamu činností souvisejících s balíčkovacím systémem Chocolatey, které aplikace umožňuje vykonat, zahajte přechodem na záložku „Moje zařízení“, která se nachází v kategorii „Univerzita“ uvnitř navigačního panelu aplikace. Pokračujte volbou seznamu strojů, jež obsahuje požadované zařízení. Následně proveďte klepnutí na tlačítko se symbolem blesku, jež se nachází u vybraného zařízení. Z nabídky operačních systémů zvolte „Windows“ a v dalším kroku proveďte volbu možnosti „Správa software“. Závěrem z vyobrazeného seznamu činností zvolte požadovanou akci.

Po volbě akce dojde ke kontaktování aplikačního serveru, jež zajistí provedení požadované činnosti a následně je na mobilním zařízení vyobrazen SSH log obsahující informace o výsledku akce. Na obrázku D.4 je možno vidět příklad SSH logu, jež obsahuje seznam nainstalovaných Chocolatey balíčků na určitém stroji.

Veškeré kroky, jež je nutno vykonat pro aktualizaci Chocolatey balíčků na určitém zařízení, jsou uvedeny v testovacím scénáři 8.4.3.



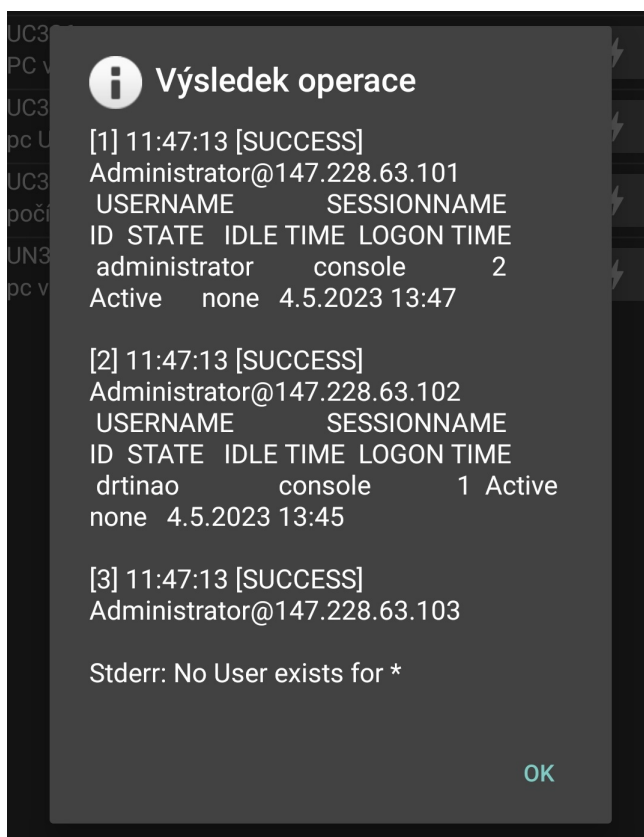
Obrázek D.4: Získání seznamu nainstalovaných Chocolatey balíčků

## D.1.5 Správa uživatelů Windows

Pro zobrazení seznamu aktivit souvisejících se správou uživatelů systému Windows, jež vytvořené řešení pro správu strojů poskytuje, nejprve proveďte přechod na kartu „Moje zařízení“, jež se nachází v segmentu „Univerzita“ navigačního panelu aplikace. Následně zvolte seznam strojů, ve kterém se nachází hledané zařízení. Pokračujte klepnutím na tlačítko s ikonou blesku přítomného v oblasti karty reprezentující vybraný počítač. Poté z nabídky operačních systémů zvolte položku „Windows“ a v dalším kroku proveďte volbu možnosti „Správa uživatelů“. Proces dokončíte výběrem akce ze seznamu vyobrazených činností.

Veškeré činnosti související se správou uživatelů je možno provádět i na úrovni celého seznamu zařízení. Například obrázek D.5 ukazuje seznam přihlášených uživatelů na třech strojích. Lze pozorovat, že na jednom zařízení je přihlášen Administrátor, na druhém doménový uživatel se jménem „drtinao“ a třetí stroj nikdo nevyužívá. U každého stroje je rovněž vidět datum a čas, kdy začal být počítač vyobrazeným uživatelem využíván.

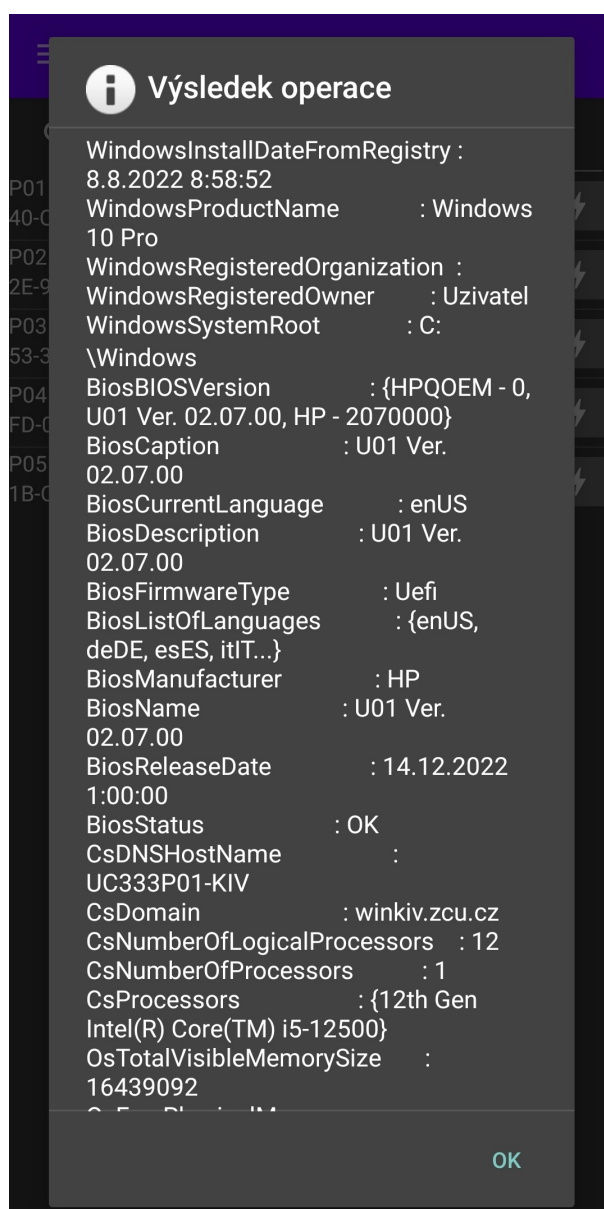
Kompletní dokumentace postupu, jenž je potřeba provést pro získání seznamu lokálních uživatelů stroje, je předmětem testovacího scénáře 8.4.4.



Obrázek D.5: Získání informací o přihlášených uživateli

## D.1.6 Zobrazení informací o stroji

U seznamu zařízení či konkrétního zařízení klepněte na ikonu blesku a ze zobrazené nabídky zvolte systém „Windows“. Následně stačí provést klepnutí na volbu „Informace o stroji“. Program po zvolení možnosti provede kontaktování aplikačního serveru a po získání odpovědi zobrazí získaný SSH log obsahující detaily vybavení stroje či strojů. Možný výsledek operace znázorňuje obrázek D.6, na kterém je možno vidět například verzi OS stanice, datum instalace OS, verzi BIOSu, model procesoru a další informace související se strojem.



Obrázek D.6: Zobrazení informací o stroji v mobilní aplikaci

## D.1.7 Vypnutí / restart zařízení

Pro vypnutí či restartování zařízení stačí stisknout tlačítko s odpovídajícím popisem, které se nachází v nabídce služeb souvisejících s operačním systémem Windows. Popis vyvolání nabídky služeb operačního systému Windows je uveden u popisu předchozích činností a rovněž je součástí testovacího scénáře 8.4.6, který ověřuje funkcionální vypnutí zařízení.

## D.2 Služby lokální síť

Předmětem následujícího textu jsou postupy k vykonání typických činností, jež může chtít uživatel mobilního klienta vykonat v souvislosti se správou lokální sítě. Mobilní aplikace umožňuje bez připojení k aplikačnímu serveru vytvářet seznamy lokálních strojů, jejichž údaje jsou uchovávány v rámci SQLite databáze, která je uložena v rámci mobilního zařízení. Mobilní aplikace kromě správy lokálních strojů umožňuje i zapnutí strojů, jež se nacházejí na stejném segmentu sítě, k němuž je mobilní zařízení připojeno.

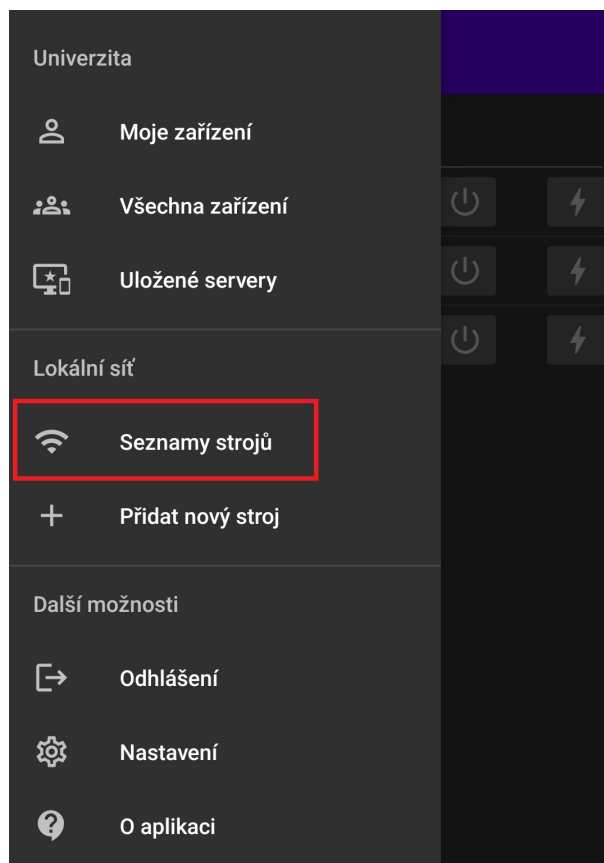
### D.2.1 Vytvoření seznamu zařízení

Vytvoření nového seznamu strojů zahajte přechodem na kartu „Seznamy strojů“, jež se nachází v sekci „Lokální síť“ postranního navigačního panelu aplikace (viz snímek D.7). Následně klepněte na tlačítko s ikonou symbolu „+“, které se nachází v pravém horním rohu aplikace. Po zobrazení modálního okna (dialogu) vyplňte informace o novém seznamu. Povinnou položkou, jež je třeba vyplnit, je název seznamu. Popis seznamu nemusí být uveden. Po kontrole zadaných dat proveďte přidání nového seznamu stiskem tlačítka „Přidat seznam“.

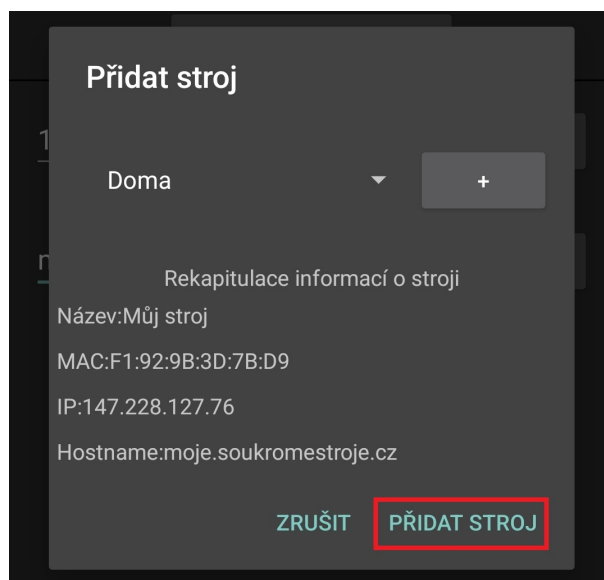
### D.2.2 Přidání zařízení do seznamu

Pro zahájení procesu přidání nového počítače do některého z lokálních seznamů proveďte přechod na kartu „Přidat nový stroj“, jež se nachází v kategorii „Lokální síť“ navigace programu. Po přechodu na danou kartu vyplňte požadované informace o novém zařízení a následně pokračujte stiskem tlačítka „Přidat do seznamu strojů“, jež je vyznačeno na obrázku. Následně zvolte seznam počítačů, do kterého chcete zařízení přidat, a proveďte rekapitulaci zadaných informací. Pokud jsou uvedené informace správné, potvrďte přidání stroje klepnutím na tlačítko „Přidat stroj“, jež je znázorněno na obrázku D.8.





Obrázek D.7: Navigační panel mobilní aplikace



Obrázek D.8: Tlačítko stvrzující přidání zařízení

### **D.2.3 Zapnutí zařízení**

Pro zapnutí zařízení nacházejícího se na stejném segmentu sítě jako mobilní zařízení nejprve přejděte na záložku „Seznamy strojů“ postranního navigačního panelu. Následně vyhledejte seznam obsahující požadované zařízení a klepněte na něj. Zapnutí konkrétního stroje proveďte klepnutím na ikonu zapínacího tlačítka nacházejícího se v rámci karty reprezentující stroj.

# Obsah přiloženého archivu



Součástí odevzdané diplomové práce je i zip archiv, jehož kořen obsahuje následující položky:

- `Text_prace` - text práce ve formátu pdf a zdrojové soubory typografického systému  $\LaTeX$ ,
- `Poster` - vytvořený poster ve formátu pdf a pub,
- `Aplikace_a_knihovny` - složka obsahující vytvořené řešení pro správu strojů,
- `Readme.txt` - popis adresářové struktury odevzdaného zip archivu.

Složka `Aplikace_a_knihovny` obsahuje veškeré artefakty, jež byly vytvořeny při vývoji řešení pro správu strojů a má následující strukturu:

- `Android Javadoc` - dokumentace Android aplikace vygenerovaná prostřednictvím nástroje Javadoc
- `Android Studio projekt` - projekt s mobilní aplikací, jež byl vytvořen ve vývojovém prostředí Android Studio
- `Docker Compose` - obsahuje veškeré položky, jež jsou potřebné pro nasazení aplikačního serveru, zejména se jedná o:
  - SSH klíč (privátní i veřejný),
  - aplikaci ve formátu `war`,
  - soubor `01.sql` zajišťující inicializaci databáze,
  - soubor `docker-compose.yml` zajišťující nasazení řešení.
- `IntelliJ IDEA projekt` - projekt prostředí IntelliJ IDEA, jež obsahuje kód webové aplikace a aplikačního serveru
- `Web Javadoc` - Javadoc dokumentace webové aplikace a aplikačního serveru

### *E. Obsah přiloženého archivu*

---

- `magicwol.apk` - instalační soubor vytvořené Android aplikace
- `model.mwb` - databázový model vytvořený v nástroji MySQL Workbench

# Bibliografie

- [21] *Kerberos V5 Exchanges*. Microsoft Corporation, 2021. Dostupné také z: [https://learn.microsoft.com/en-us/openspecs/windows\\_protocols/ms-kile/b4af186e-b2ff-43f9-b18e-eedb366abf13](https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-kile/b4af186e-b2ff-43f9-b18e-eedb366abf13). [MS-KILE]: Kerberos Network Authentication Service (V5) Synopsis [citováno 12.3.2023].
- [Kri18] KRILL, Paul. *15 Java frameworks that give developers a boost*. IDG Communications, Inc., 2018. Dostupné také z: <https://www.infoworld.com/article/3268724/15-java-frameworks-that-give-developers-a-boost.html>. 15 Java frameworks that give developers a boost | InfoWorld [citováno 6.4.2023].
- [23a] *About SourceForge*. Slashdot Media, 2023. Dostupné také z: <https://sourceforge.net/about>. About SourceForge [citováno 20.1.2023].
- [22a] *Approved Verbs for PowerShell Commands*. Microsoft Corporation, 2022. Dostupné také z: <https://learn.microsoft.com/en-us/powershell/scripting/developer/cmdlet/approved-verbs-for-windows-powershell-commands?view=powershell-7.3>. Approved Verbs for PowerShell Commands - PowerShell | Microsoft Learn [citováno 4.3.2023].
- [23b] *Best Remote Monitoring Management (RMM) Software*. G2, 2023. Dostupné také z: <https://www.g2.com/categories/remote-monitoring-management-rmm>. Best Remote Monitoring Management (RMM) Software 2023 [citováno 21.1.2023].
- [23c] *RMM Software*. Slashdot Media, 2023. Dostupné také z: [https://sourceforge.net/software/rmm/?sort=rating\\_avg](https://sourceforge.net/software/rmm/?sort=rating_avg). Best RMM Software - 2023 Reviews [citováno 20.1.2023].
- [22b] *Can a Linux computer control a Central computer?* LogMeIn, Inc., 2022. Dostupné také z: <https://support.logmeininc.com/central/help/can-a-linux-computer-control-a-logmein-computer>. Can a Li-

- nux computer control a Central computer? - Central Support [citováno 21.1.2023].
- [23d] *Can I monitor Windows, Mac and Linux?* Atera Networks Ltd., 2023. Dostupné také z: <https://support.atera.com/hc/en-us/articles/205809077>. Can I monitor Windows, Mac and Linux? - Atera Support [citováno 21.1.2023].
- [22c] *Hardware Inventory*. LogMeIn, Inc., 2022. Dostupné také z: <https://support.logmeininc.com/central/help/logmein-central-report-types-central-c-central-reporttypes>. Central Report Types - Central Support [citováno 22.1.2023].
- [23e] *Co je to TCO*. Radek Kučera, 2023. Dostupné také z: <https://www.salesman.cz/co-je-to-tco/>. Co je to TCO - salesman.cz [citováno 15.3.2023].
- [23f] *Difference between Adaptive and Non-Adaptive Routing algorithms*. GeeksforGeeks, 2023. Dostupné také z: <https://www.geeksforgeeks.org/difference-between-adaptive-and-non-adaptive-routing-algorithms/>. Difference between Adaptive and Non-Adaptive Routing algorithms - GeeksforGeeks [citováno 6.2.2023].
- [23g] *How does NinjaOne work?* NinjaOne, 2023. Dostupné také z: <https://www.ninjaone.com/faq>. FAQ: Common NinjaOne Questions Answered - NinjaOne [citováno 21.1.2023].
- [For00] FOROUZAN, Behrouz A. *Data Communications and Networking*. New York: McGraw Hill, 2000. ISBN 978-0073376226.
- [Tay23] TAYLOR, Petroc. *Market share of mobile operating systems worldwide 2009-2022*. Statista GmbH, 2023. Dostupné také z: <https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/>. Global mobile OS market share 2022 | Statista [citováno 15.4.2023].
- [22d] *Heslo ...* Západočeská univerzita v Plzni - CIV, 2022. Dostupné také z: [https://support.zcu.cz/index.php/Hesla\\_v\\_syst%C3%A9mu\\_Orion](https://support.zcu.cz/index.php/Hesla_v_syst%C3%A9mu_Orion). Hesla v systému Orion – Support [citováno 5.12.2022].
- [23h] *Is SSO Secure?* OneLogin, Inc., 2023. Dostupné také z: <https://www.onelogin.com/learn/how-single-sign-on-works>. How Does Single Sign-On (SSO) Work? | OneLogin [citováno 9.3.2023].

- [22e] *How to Wake a Computer in Sleep Mode or Powered Off Using Wake On LAN*. LogMeIn, Inc., 2022. Dostupné také z: <https://support.logmeininc.com/pro/help/how-to-wake-a-computer-in-sleep-mode-or-powered-off-using-wake-on-lan-logmein-t-host-preferences-wakeonlan>. How to Wake a Computer in Sleep Mode or Powered Off Using Wake On LAN - Pro Support [citováno 23.4.2023].
- [23i] *Unique Packages / Total Packages*. Chocolatey Software, Inc., 2023. Dostupné také z: <https://community.chocolatey.org/>. Chocolatey Software | Community [citováno 6.3.2023].
- [23j] *Description*. Chocolatey Software, Inc., 2023. Dostupné také z: <https://community.chocolatey.org/packages/ChocolateyGUI>. Chocolatey Software Docs | Chocolatey GUI [citováno 6.3.2023].
- [10] *isReachable*. Oracle, 2010. Dostupné také z: <https://docs.oracle.com/javase/1.5.0/docs/api/java/net/InetAddress.html>. InetAddress (Java 2 Platform SE 5.0) [citováno 23.1.2023].
- [23k] *Install an agent*. Atera Networks Ltd., 2023. Dostupné také z: <https://support.atera.com/hc/en-us/articles/360015643914-Install-an-agent>. Install an agent - Atera Support [citováno 21.1.2023].
- [23l] *Install Linux on Windows with WSL*. Microsoft Corporation, 2023. Dostupné také z: <https://learn.microsoft.com/en-us/windows/wsl/install>. Install WSL | Microsoft Learn [citováno 18.4.2023].
- [22f] *Installing the Host Software (Add a computer)*. LogMeIn, Inc., 2022. Dostupné také z: <https://support.logmeininc.com/pro/help/installing-logmein-host-software-add-a-computer-logmein-t-lmi-installing>. Installing the Host Software (Add a computer) - Pro Support [citováno 21.1.2023].
- [23m] *SSO Integrations*. NinjaOne, 2023. Dostupné také z: <https://www.ninjaone.com/integrations/>. Integrations for RMM, Endpoint Security, more - NinjaOne [citováno 21.1.2023].
- [23n] *How does SLAAC work?* NetworkAcademy.io, 2023. Dostupné také z: <https://www.networkacademy.io/ccna/ipv6/stateless-address-autoconfiguration-slaac>. IPv6 Stateless Address Auto-configuration (SLAAC) [citováno 16.2.2023].
- [95] *ISO/IEC 15802-1:1995(en)*. ISO/IEC, 1995. Dostupné také z: <https://www.iso.org/obp/ui#iso:std:iso-iec:15802:-1:ed-1:v1:en>. ISO/IEC 15802-1:1995(en) Information technology [citováno 6.2.2023].

- [13] *KDC servery*. Západočeská univerzita v Plzni - CIV, 2013. Dostupné také z: <https://support.zcu.cz/index.php/Kerberos>. Kerberos - Support [citováno 13.3.2023].
- [19] *Kerberos Fundamentals*. QOMPLX, 2019. Dostupné také z: <https://www.qomplx.com/blog/about-kerberos/>. Kerberos Fundamentals - How It Works - QOMPLX [citováno 12.3.2023].
- [22g] *What is Kerberos?* Massachusetts Institute of Technology, 2022. Dostupné také z: <https://web.mit.edu/kerberos/>. Kerberos: The Network Authentication Protocol [citováno 7.12.2022].
- [22h] *Podporované služby a zdroje*. Západočeská univerzita v Plzni - CIV, 2022. Dostupné také z: [https://support.zcu.cz/index.php/Konto\\_Orion](https://support.zcu.cz/index.php/Konto_Orion). Konto Orion – Support [citováno 5.12.2022].
- [Stř23a] STŘECHA, Tomáš. *Spring Boot*. ITnetwork s.r.o., 2023. Dostupné také z: <https://www.itnetwork.cz/java/spring-boot/zaklady/uvod-do-spring-boot-frameworku-pro-javu/>. Lekce 1 - Úvod do Spring Boot frameworku v Javě [citováno 6.4.2023].
- [Stř23b] STŘECHA, Tomáš. *Lekce 4 - Úvod do MVC architektury ve Spring Boot*. ITnetwork s.r.o., 2023. Dostupné také z: <https://www.itnetwork.cz/java/spring-boot/zaklady/uvod-do-mvc-architektury-ve-spring-bootu>. Lekce 4 - Úvod do MVC architektury ve Spring Boot [citováno 17.4.2023].
- [Pet23] PETERKA, Jiří. *možné přístupy ke směrování*. Univerzita Karlova, 2023. Dostupné také z: <https://www.ksi.mff.cuni.cz/~svoboda/courses/182-NSWI090/lectures/P%99edn%C3%A1%C5%A1ka-05-S%C3%AD%C5%A5ov%C3%A1-vrstva.pdf>. Lekce 8: Síťová vrstva a směrování [citováno 6.2.2023].
- [23o] *Select Number of Computers*. LogMeIn, Inc., 2023. Dostupné také z: <https://www.logmein.com/central/pricing>. LogMeIn Central Pricing – Remote Management Software Plan Options [citováno 22.1.2023].
- [Luc18] LUCAS, Michael W. *SSH Mastery: OpenSSH, PuTTY, Tunnels and Keys (IT Mastery), 2nd Edition*. Grosse Pointe: Tilted Windmill Press, 2018. ISBN 978-1642350029.
- [23p] *macOS Ventura is compatible with these computers*. Apple Inc., 2023. Dostupné také z: <https://support.apple.com/en-us/HT213264>. macOS Ventura is compatible with these computers - Apple Support [citováno 15.4.2023].



- [98] *Magic Packet Frame Detection*. Advanced Micro Devices, Inc., 1998. Dostupné také z: <https://www.amd.com/system/files/TechDocs/20213.pdf>. Magic Packet Technology [citováno 7.12.2022].
- [23q] *Spring Boot Starter Web*. MvnRepository, 2023. Dostupné také z: <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-web>. Maven Repository: [org.springframework.boot](https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-web) » [spring-boot-starter-web](https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-web) [citováno 6.4.2023].
- [23r] *Meet Android Studio*. Google LLC, 2023. Dostupné také z: <https://developer.android.com/studio/intro>. Meet Android Studio | Android Developers [citováno 16.4.2023].
- [23s] *Modely iPhonu kompatibilní se systémem iOS 16*. Apple Inc., 2023. Dostupné také z: <https://support.apple.com/cs-cz/guide/iphone/iphe3fa5df43/16.0/ios/16.0>. Modely iPhonu kompatibilní se systémem iOS 16 - Podpora Apple (CZ) [citováno 16.4.2023].
- [23t] *MVC Framework Introduction*. GeeksforGeeks, 2023. Dostupné také z: <https://www.geeksforgeeks.org/mvc-framework-introduction/>. MVC Framework Introduction - GeeksForGeeks [citováno 17.4.2023].
- [23u] *What is MySQL?* Docker Inc., 2023. Dostupné také z: [https://hub.docker.com/\\_/mysql](https://hub.docker.com/_/mysql). mysql - Official Image | Docker Hub [citováno 1.4.2023].
- [22i] *Nástroj - Rozvrh*. Západočeská univerzita v Plzni - CIV, 2022. Dostupné také z: [https://is-stag.zcu.cz/napoveda/rozvrh-editor/rozvrheditor2\\_nastroj\\_rozvrh.html](https://is-stag.zcu.cz/napoveda/rozvrh-editor/rozvrheditor2_nastroj_rozvrh.html). Nástroj - Rozvrh [citováno 4.12.2022].
- [23v] *Package management*. Canonical Ltd., 2023. Dostupné také z: <https://ubuntu.com/server/docs/package-management>. Package management | Ubuntu [citováno 6.3.2023].
- [23w] *Vrstevnatý model počítačové sítě*. CZ.NIC, 2023. Dostupné také z: <https://moodle.nic.cz/mod/book/view.php?id=672%5C&chapterid=162>. Packetový přenos, routing: ISO/OSI model vs TCP/IP [citováno 3.2.2023].
- [PS17] PAYETTE, Bruce; SIDDAWAY, Richard. *Windows PowerShell in Action, Third Edition*. New York: Manning Publications, 2017. ISBN 978-1633430297.

- [Gaj22] GAJDA, Michal. *PSWindowsUpdate*. Microsoft Corporation, 2022. Dostupné také z: <https://www.powershellgallery.com/packages/PSWindowsUpdate/2.2.0.3>. PowerShell Gallery | PSWindowsUpdate 2.2.0.3 [citováno 4.3.2023].
- [Far22] FARRIER, Ellie. *Public vs. Private IP Addresses: What's the Difference?* Avast Software s.r.o., 2022. Dostupné také z: <https://www.avast.com/c-ip-address-public-vs-private>. Private vs. Public IP Addresses | Differences Explained | Avast [citováno 16.2.2023].
- [23x] *Save data using SQLite*. Google LLC, 2023. Dostupné také z: <https://developer.android.com/training/data-storage/sqlite>. Save data using SQLite | Android Developers [citováno 18.4.2023].
- [23y] *Single sign-on with Microsoft Azure AD*. Atera Networks Ltd., 2023. Dostupné také z: <https://support.atera.com/hc/en-us/articles/6343284683548-Single-sign-on-with-Microsoft-Azure-AD>. Single sign-on with Microsoft Azure AD - Atera Support [citováno 21.1.2023].
- [SZ14] SPURGEON, Charles E.; ZIMMERMAN, Joann. *Ethernet: The Definitive Guide, 2nd Edition*. Sebastopol: O'Reilly Media, Inc., 2014. ISBN 978-1449361846.
- [22j] *How do you calculate TCO?* Manutan, 2022. Dostupné také z: <https://www.manutan.com/blog/en/glossary/understanding-tco-total-cost-of-ownership-origins-definition-calculation-advantages-and-so-on>. TCO: What are the components? [citováno 15.3.2023].
- [Bou08a] BOUŠKA, Petr. *Podsítě - Subnets a výpočty adres*. Petr Bouška aka Samuraj, 2008. Dostupné také z: <https://www.samuraj-cz.com/clanek/tcpip-adresy-masky-subnety-a-vypocty/>. TCP/IP - adresy, masky, subnety a výpočty [citováno 23.2.2023].
- [Pra23] PRAMBERGER, Roman. *OSI-Model*. Roman Pramberger, 2023. Dostupné také z: <https://osi-model.com/>. The OSI-Model in a simple way [citováno 4.2.2023].
- [23z] *About Tickets*. Massachusetts Institute of Technology, 2023. Dostupné také z: <https://web.mit.edu/kerberos/kfw-4.1/kfw-4.1/kfw-4.1-help/html/tickets.htm>. Tickets [citováno 12.3.2023].
- [23aa] *What is Tomcat?* Docker Inc., 2023. Dostupné také z: [https://hub.docker.com/\\_/tomcat](https://hub.docker.com/_/tomcat). tomcat - Official Image | Docker Hub [citováno 26.3.2023].

- [Cla23] CLARK, Jessica. *Top 10 Java Frameworks*. Back4App, 2023. Dostupné také z: [https://blog.back4app.com/java-frameworks/#What\\_is\\_a\\_Framework](https://blog.back4app.com/java-frameworks/#What_is_a_Framework). Top 10 Java Frameworks [citováno 2.4.2023].
- [Tur14] TURNBULL, James. *The Docker Book*. New York: James Turnbull, 2014. ISBN 978-0988820234.
- [Mah17] MAHESHWARI, Mudit. *Authentication of the client*. Medium Corporation, 2017. Dostupné také z: <https://medium.com/@hellomudit/understanding-ssh-workflow-66a0e8d4bf65>. Understanding SSH workflow [citováno 26.2.2023].
- [22k] *Using Azure Active Directory with Central*. LogMeIn, Inc., 2022. Dostupné také z: <https://support.logmeininc.com/central/help/using-azure-active-directory-with-logmein-central>. Using Azure Active Directory with Central - Pro Support [citováno 21.1.2023].
- [23ab] *Wake on LAN*. Atera Networks Ltd., 2023. Dostupné také z: <https://support.atera.com/hc/en-us/articles/115000560228-Wake-on-LAN>. Wake on LAN - Atera Support [citováno 23.4.2023].
- [Bou08b] BOUŠKA, Petr. *Wake on LAN - lokální i vzdálený subnet*. Petr Bouška aka Samuraj, 2008. Dostupné také z: <https://www.samuraj-cz.com/clanek/wake-on-lan-lokalni-i-vzdaleny-subnet/>. Wake on LAN - lokální i vzdálený subnet [citováno 7.12.2022].
- [Pri21] PRIYA, Bhanu. *Sub layers*. Tutorials Point (India) Ltd., 2021. Dostupné také z: <https://www.tutorialspoint.com/what-are-logical-link-control-llc-and-medium-access-control-mac>. What are Logical Link Control (LLC) and Medium Access Control(MAC) [citováno 3.2.2023].
- [Ibe23] IBEAKANMA, Chioma. *How Does MAC Spoofing Work?* MakeUseOf, 2023. Dostupné také z: <https://www.makeuseof.com/what-is-mac-spoofing/>. What Is a MAC Spoofing Attack and How Can You Prevent It? [citováno 7.2.2023].
- [22l] *What is a cmdlet?* Microsoft Corporation, 2022. Dostupné také z: <https://learn.microsoft.com/en-us/powershell/scripting/powershell-commands?view=powershell-7.3>. What is a PowerShell command? - PowerShell | Microsoft Learn [citováno 4.3.2023].
- [22m] *Automation platform*. Microsoft Corporation, 2022. Dostupné také z: <https://learn.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7.3>. What is PowerShell? - PowerShell | Microsoft Learn [citováno 1.3.2023].



# Seznam zkratek

AFP - protokol pro přenos souborů a sdílení tiskáren mezi počítači s macOS (Apple Filing Protocol)

API - soubor definovaných pravidel, jež umožňují komunikaci dvou či více počítačových systémů (Application Programming Interface)

AS - komponenta zajišťující ověření uživatele v rámci protokolu Kerberos (Authentication Server)

ATM - protokol umožňující vysokorychlostní přenos dat (Asynchronous Transfer Mode)

AVD - program umožňující správu virtuálních Android zařízení (Android Virtual Device)

BGP - dynamický směrovací protokol, který je využíván k výměně směrovacích informací mezi autonomními systémy v internetu (Border Gateway Protocol)

DHCP - protokol zajišťující dynamické přidělování IP adres zařízením, jež se připojí do počítačové sítě (Dynamic Host Configuration Protocol)

ICA - protokol pro přenos dat ze serveru ke klientovi, jež vyvinula firma Citrix Systems (Independent Computing Architecture)

ICMP - protokol pro zasílání stavových a chybových informací o stavu sítě (Internet Control Message Protocol)

IEEE - mezinárodní nezisková organizace usilující o rozšíření technologií v oblasti elektrotechniky a informatiky (Institute of Electrical and Electronics Engineers)

IPv4 - čtvrtá verze protokolu umožňujícího komunikaci v počítač. sítích (Internet Protocol version 4)

IPv6 - čtvrtá verze protokolu pro komunikaci v počítač. sítích (Internet Protocol version 6)

ISO - mezinárodní organizace, která sídlí ve Švýcarsku a zabývá se tvorbou norem v různých oblastech (International Organization for Standardization)

KDC - centrální server pro autentizaci v systému Kerberos skládající se z komponent pro ověřování uživatelů a vydávání pověření klientům (Key Distribution Center)

NetBIOS - protokol umožňující komunikaci zařízení v lokální síti (Network Basic Input/Output System)

NFS - protokol pro sdílení dat přes počítačovou síť (Network File System)

OS - operační systém (Operating System)

OSPF - směrovací protokol, který je využíván v počítačových sítích pro nalezení optimální cesty dat v síti (Open Shortest Path First)

PPP - protokol umožňující přímé propojení dvou síťových uzlů (Point-to-Point Protocol)

RDP - protokol pro vzdálený přístup k počítačové stanici, jež vytvořila společnost Microsoft (Remote Desktop Protocol)

REST - architektura rozhraní, která je datově orientovaná a navržena pro distribuované prostředí (REpresentational State Transfer)

RIP - dynamický směrovací protokol, jež používá počet mezilehlých uzlů jako metriku pro nalezení nejlepší možné cesty mezi dvěma uzly v síti (Routing Information Protocol)

RPC - protokol umožňující výkon kódu na vzdáleném síťovém zařízení (Remote Procedure Call)

SCP - protokol založený na SSH, jež slouží pro bezpečný přenos souborů po síti

(Secure Copy Protocol)

SDK - sada nástrojů, knihoven či dalšího softwarového vybavení, jež vývojářům umožňuje vytvářet aplikace pro určitou platformu (Software Development Kit)

SSH - síťový protokol umožňující bezpečné vzdálené přihlášení a vykonávání příkazů přes nezabezpečenou síť (Secure Shell)

TCP - protokol zajišťující spolehlivý přenos dat v počítačových sítích (Transmission Control Protocol)

TGS - komponenta zajišťující přidělení pověření klientům v rámci protokolu Kerberos (Ticket Granting Server)

TGT - token, jenž se používá pro komunikaci s TGS a v rámci protokolu Kerberos je poskytován ověřeným uživatelům (Ticket Granting Ticket)

UDP - protokol umožňuje přenos dat prostřednictvím počítačové sítě, přičemž ne-garantuje spolehlivost spojení ani pořadí doručení dat (User Datagram Protocol)

XDR - standard definující kódování dat, jež jsou odesílána přes počítačovou síť (eXternal Data Representation)





# Seznam obrázků

2.1	Změna MAC adresy v operačním systému Windows 11 Pro . . . . .	13
2.2	Princip funkce protokolu Kerberos [21] . . . . .	23
2.3	Ověření uživatele v rámci ZČU . . . . .	25
3.1	Ukázka uživatelského rozhraní Atera . . . . .	36
7.1	Schéma vytvořeného systému . . . . .	57
7.2	Popis komunikace v rámci architektury MVC [Stř23b] . . . . .	59
7.3	Databázový model aplikace . . . . .	66
7.4	Využití paměti RAM kontejnery aplikace . . . . .	66
7.5	Model použité SQLite databáze . . . . .	69
8.1	Přihlášení běžného uživatele do webové aplikace . . . . .	74
8.2	Přihlášení běžného uživatele do mobilní aplikace . . . . .	75
8.3	Zobrazení registrační žádosti v rámci webové aplikace . . . . .	77
8.4	Seznam importovaných informací o strojích . . . . .	84
8.5	Naplánování úlohy zajišťující aktualizace OS Windows . . . . .	86
8.6	Spuštění aktualizace OS Windows pomocí mobilního klienta . . . . .	87
8.7	Zobrazení lokálních uživatelských účtů v aplikaci Správa počítače . . . . .	90
8.8	Získání lokálních účtů v mobilní aplikaci . . . . .	90
C.1	Přidání seznamu zařízení . . . . .	108
C.2	Přidání zařízení do seznamu . . . . .	108
C.3	Export údajů zařízení . . . . .	109
C.4	Import údajů zařízení . . . . .	109
C.5	Karta reprezentující žádost o registraci stroje . . . . .	111
C.6	Tlačítko umožňující zapnutí stroje . . . . .	111
C.7	Tlačítko pro kontrolu dostupnosti stroje . . . . .	112
C.8	Volba položky Windows Update . . . . .	113
C.9	Dostupné možnosti správy Chocolatey balíčků . . . . .	113
C.10	Získání seznamu lokálních účtů na více zařízeních současně . . . . .	114
C.11	Přihlašovací obrazovka webové aplikace . . . . .	116

C.12	Tlačítka umožňující schválení, resp. zamítnutí registrační žádosti . . .	117
D.1	Tlačítko pro odeslání přihlašovacích dat . . . . .	120
D.2	Tlačítka umožňující zapnutí zařízení . . . . .	121
D.3	Dostupné akce Windows Update . . . . .	122
D.4	Získání seznamu nainstalovaných Chocolatey balíčků . . . . .	123
D.5	Získání informací o přihlášených uživateli . . . . .	124
D.6	Zobrazení informací o stroji v mobilní aplikaci . . . . .	125
D.7	Navigační panel mobilní aplikace . . . . .	127
D.8	Tlačítko stvrzující přidání zařízení . . . . .	127

# Seznam tabulek

2.1	Vrstvy ISO/OSI modelu . . . . .	10
2.2	Příklady SSH klientů . . . . .	17
2.3	Příklady Powershell cmdletů . . . . .	19
2.4	Alternativní nástroje pro správu Windows software . . . . .	21
2.5	Popis zpráv používaných při komunikaci prostřednictvím Kerberos . .	23
2.6	KDC servery na ZČU . . . . .	24
3.1	Zhodnocení funkcionality testovaných produktů . . . . .	41
7.1	Popis parametrů metody <code>execCmdDev</code> . . . . .	63
7.2	Popis parametrů metody <code>execAsEventGroup</code> . . . . .	64
7.3	Význam databázových tabulek (server aplikace) . . . . .	65
7.4	Význam databázových tabulek (mobilní klient) . . . . .	70
8.1	Testované konfigurace - webová aplikace . . . . .	72
8.2	Testované konfigurace virtuálních zařízení . . . . .	72
8.3	Testované konfigurace - aplikační server . . . . .	73



# Seznam výpisů

6.1	Ukázka anotací knihovny Retrofit . . . . .	55
7.1	Ukázka skriptu zajišťujícího získání informací o stroji . . . . .	64

101011000011100010 1100001  
1010110001 10001 10001

110100011101101001 1010101  
01100001 1010101  
11100010101110101