

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2024

Bc. Jan Antoš



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## DOCHÁZKOVÝ TERMINÁL

ATTENDANCE TERMINAL

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Jan Antoš

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Macho, Ph.D.

BRNO 2024



# Diplomová práce

magisterský navazující studijní program **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

**Student:** Bc. Jan Antoš

**ID:** 220962

**Ročník:** 2

**Akademický rok:** 2023/24

**NÁZEV TÉMATU:**

## Docházkový terminál

**POKYNY PRO VYPRACOVÁNÍ:**

1. Seznamte se s problematikou docházkových terminálů a uveďte základní parametry docházkových terminálů dostupných na trhu.
2. Definujte požadavky zákazníka na fungování docházkového terminálu s dotykovým displejem a čtečkou RFID karet. Specifikujte komunikaci terminálu s nadřazeným serverem a případnou spolupráci s RFID zámkem.
3. Navrhněte koncepci docházkového terminálu, vyberte vhodné komponenty pro realizaci terminálu a nakreslete blokové schéma zařízení.
4. Nakreslete schéma zapojení terminálu, stanovte hodnoty jednotlivých součástek.
5. Navrhněte konstrukční uspořádání terminálu.
6. Terminál realizujte a oživte.
7. Vytvořte a odladte potřebný software pro terminál.
8. Ověřte fungování terminálu ve spolupráci s kamerou a nadřazeným serverem. Vyhodnoťte dosažené výsledky a navrhněte případná vylepšení terminálu.

**DOPORUČENÁ LITERATURA:**

Raspberry Pi hardware. Raspberry Pi [online]. [cit. 2022-12-24]. Dostupné z: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>

**Termín zadání:** 5.2.2024

**Termín odevzdání:** 15.5.2024

**Vedoucí práce:** Ing. Tomáš Macho, Ph.D.

**doc. Ing. Petr Fiedler, Ph.D.**  
předseda rady studijního programu

**UPOZORNĚNÍ:**

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **Abstrakt**

Tato diplomová práce má za úkol se seznámit s problematikou terminálů a uvést parametry docházkových terminálů dostupných na trhu. Definiovat požadavky zákazníka na fungování docházkového terminálu s dotykovým displejem a RFID čtečkou. Specifikovat komunikaci terminálu s nadřazeným serverem. Navrhnout koncepci docházkového terminálu, vybrat vhodné komponenty pro realizaci terminálu a nakreslit blokové schéma. Navrhnout konstrukční uspořádání terminálu. Terminál realizovat, oživit a následně odladit potřebný software.

## **Klíčová slova**

Raspberry, displej, RFID, kamera, docházkový terminál, fusion360, https

## **Abstract**

This master thesis aims to get acquainted with the issue of terminals and to list the parameters of attendance terminals available on the market. To define the customer requirements for the operation of the attendance terminal with touch screen and RFID reader. Specify the communication of the terminal with the parent server. Design the concept of the attendance terminal, select the appropriate components for the implementation of the terminal and draw the block diagram. Design the structural layout of the terminal. Implement the terminal, revive it and then debug the necessary software.

## **Keywords**

Raspberry, display, RFID, camera, attendance terminal, fusion360, https

## **Bibliografická citace**

ANTOŠ, Jan. *Docházkový terminál* [online]. Brno, 2024 [cit. 2024-04-17]. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/159885>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Tomáš Macho.

# Prohlášení autora o původnosti díla

<b>Jméno a příjmení studenta:</b>	<b>Jan Antoš</b>
<b>VUT ID studenta:</b>	<b>220962</b>
<b>Typ práce:</b>	<b>Diplomová práce</b>
<b>Akademický rok:</b>	<b>2023/24</b>
<b>Téma závěrečné práce:</b>	<b>Docházkový terminál</b>

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 13. května 2024

-----  
podpis autora

## Poděkování

Děkuji vedoucímu diplomové práce Ing. Tomáši Machovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: 13. května 2024

-----  
podpis autora

# Obsah

<b>1. TERMINÁLY DOSTUPNÉ NA TRHU .....</b>	<b>13</b>
1.1 DOCHÁZKOVÝ TERMINÁL IT-WATT R3 [1].....	13
1.2 DOCHÁZKOVÝ TERMINÁL ZPAD SYSDO BIOPAD [2] .....	14
1.3 DOCHÁZKOVÝ TERMINÁL QUBOS FINGER [3].....	15
1.4 DOCHÁZKOVÝ TERMINÁL GIRITON NA ZEĎ 7“ [5] .....	16
1.5 POROVNÁNÍ DOCHÁZKOVÝCH TERMINÁLŮ .....	16
<b>2. POŽADAVKY ZÁKAZNÍKA.....</b>	<b>19</b>
<b>3. POŽADAVKY NA DOCHÁZKOVÝ TERMINÁL .....</b>	<b>21</b>
<b>4. KOMUNIKACE TERMINÁLU SE SERVEREM.....</b>	<b>23</b>
<b>5. NÁVRH DOCHÁZKOVÉHO TERMINÁLU .....</b>	<b>28</b>
5.1 ŘÍDICÍ JEDNOTKA .....	28
5.1.1 <i>Raspberry Pi 4 Model B</i> [11].....	28
5.1.2 <i>Radxa ROCK Pi 4 C+</i> [15].....	29
5.1.3 <i>Výběr řídicí jednotky</i> .....	30
5.2 DOTYKOVÝ DISPLEJ.....	30
5.2.1 <i>Raspberry Pi 7“</i> [19].....	30
5.2.2 <i>LaskaKit 10.1“</i> [21].....	31
5.2.3 <i>Výběr displeje</i> .....	32
5.3 RFID ČTEČKA.....	32
5.3.1 <i>MF RC522</i> [23].....	32
5.3.2 <i>RDM6300</i> [25].....	33
5.3.3 <i>USB čtečka 125 kHz</i> [27].....	33
5.3.4 <i>Výběr RFID čtečky</i> .....	34
5.4 KAMERA.....	34
5.5 ZVUKOVÁ SIGNALIZACE .....	35
5.6 NAPÁJENÍ TERMINÁLU POMOCÍ POE.....	36
<b>6. NÁVRH MECHANICKÉ KONSTRUKCE.....</b>	<b>37</b>
6.1 PRVOTNÍ NÁVRHY.....	37
6.2 REALIZACE NÁVRHU.....	40
6.3 PROPOJENÍ JEDNOTLIVÝCH KOMPONENT .....	45
<b>7. REALIZACE TERMINÁLU .....</b>	<b>47</b>
7.1 3D TISK .....	47
7.2 MOŽNOSTI HW ÚPRAV TERMINÁLU.....	48
7.3 SESTAVENÍ .....	49
7.4 NAHRÁNÍ RASPBERRY PI OS .....	49
<b>8. GRAFICKÉ PROSTŘEDÍ.....</b>	<b>50</b>
8.1 NÁVRH GUI .....	50
8.2 REALIZACE GUI.....	53
<b>9. VÝVOJ PROGRAMU .....</b>	<b>55</b>
9.1 PROGRAMOVACÍ JAZYK A VÝVOJOVÉ PROSTŘEDÍ .....	55
9.2 VYTVOŘENÍ VZDÁLENÉHO VÝVOJOVÉHO PROSTŘEDÍ PRO RPI.....	55

9.3	STRUKTURA PROGRAMU .....	55
9.4	POPIS PROGRAMU .....	59
9.4.1	<i>Obrazovky</i> .....	59
9.4.2	<i>Připojené komponenty</i> .....	62
9.4.3	<i>API volání</i> .....	65
9.4.4	<i>Hlavní skript</i> .....	65
9.5	OVĚŘENÍ KOMUNIKACE SE SERVEREM.....	68
9.6	KONFIGURACE RPI.....	70
9.7	ZABEZPEČENÍ .....	72
<b>10.</b>	<b>VZDÁLENÁ SPRÁVA .....</b>	<b>73</b>
<b>11.</b>	<b>INTEGRACE DO INFORMAČNÍHO SYSTÉMU .....</b>	<b>75</b>
<b>12.</b>	<b>TESTOVÁNÍ A NAsAZENÍ TERMINÁLU .....</b>	<b>78</b>
<b>13.</b>	<b>VYHODNOCENÍ .....</b>	<b>80</b>
<b>14.</b>	<b>NÁVRHY NA VYLEPŠENÍ SYSTÉMU .....</b>	<b>82</b>
<b>15.</b>	<b>ZÁVĚR.....</b>	<b>86</b>

# SEZNAM OBRÁZKŮ

1.1	Docházkový terminál Ikos IT-WATT R3 [1].....	13
1.2	ZPad SYSDO BIOPAD [2].....	14
1.3	Docházkový terminál Qubos Finger [3].....	15
1.4	Docházkový terminál GIRITON Na zed' 7'' [5].....	16
2.1	Píchnací hodiny.....	19
2.2	Areál zemědělského družstva – vyznačení budov s terminály.....	20
3.1	Princip PoE [7].....	21
4.1	Vývojový diagram – komunikace terminál x server.....	26
4.2	Dostupné požadavky pro terminál.....	27
5.1	Blokové schéma terminálu a jeho připojení do rozvaděče.....	28
5.2	Raspberry Pi 4 Model B [14].....	29
5.3	Radxa ROCK Pi 4 C+ [15].....	29
5.4	Raspberry Pi 7'' [20].....	31
5.5	LaskaKit 10.1'' [22].....	31
5.6	MF RC522 [24].....	32
5.7	RDM6300 [26].....	33
5.8	USB čtečka 125 kHz [28].....	33
5.9	Raspberry Pi kamera V2 [32].....	35
5.10	Modul akustického bzučáku [34].....	35
5.11	PoE splitter Joy-It SBC [36].....	36
5.12	PoE injektor TP-Link TL-POE150S [48].....	36
6.1	Logo Techcrowd s.r.o. [37].....	37
6.2	První návrh konstrukce.....	37
6.3	Druhý návrh konstrukce.....	38
6.4	Třetí návrh konstrukce.....	39
6.5	Čtvrtý návrh konstrukce.....	39
6.6	Tělo, displej, vzpěry.....	40
6.7	Řez – tělo, vzpěra, displej.....	40
6.8	Přidání RDM6300, prostoru pro kameru a průchodek.....	41
6.9	Řez – kamera, PoE.....	41
6.10	Detail – průchodka kamera.....	42
6.11	Detail – výstup Ethernet.....	42
6.12	Detail – ventilátory.....	43
6.13	Zadní kryt.....	43
6.14	Řez – tělo, zadní kryt.....	44
6.15	Výsledné konstrukční uspořádání.....	44
6.16	Schéma zapojení terminálu.....	45
6.17	Detailní zapojení GPIO na RPi.....	46
7.1	Jednotlivé části pro 3D tisk.....	47
7.2	Příprava 3D tisku.....	48
7.3	Personalizovaný přední kryt terminálu.....	49
8.1	Inicializační obrazovka (init_window).....	50
8.2	Hlavní obrazovka (main_window) – příchod.....	51
8.3	Hlavní obrazovka (main_window) – odchod, lékař.....	51
8.4	Vyskakovací obrazovka (popup_window) – příchod.....	52
8.5	Vyskakovací obrazovka (popup_window) – odchod.....	52



8.6	Chybová obrazovka (status_window) .....	53
8.7	Nastavení barev u CTKButton [45].....	54
8.8	Příklad nastavení tlačítka příchod na hlavní obrazovce .....	54
9.1	Diagram stavového automatu .....	56
9.2	Atribut – fullscreen (False, True) .....	59
9.3	Povolení sériové komunikace na RPi .....	63
9.4	Startovací obrazovky .....	71
9.5	Zamezení výpisu při startu terminálu .....	71
9.6	Zakázání zobrazování kurzoru .....	72
10.1	Obrazovky aplikace TeamViewer .....	74
11.1	Use-case diagram pro uživatele IS .....	75
11.2	IS – Správa docházky pro uživatele Admin .....	76
11.3	IS – Pořízené fotografie při zvolení akce docházky .....	77
12.1	Časová osa testování a nasazení terminálu.....	78
12.2	Nainstalovaný terminál v budově „dílna“ .....	79
13.1	Časová osa vývoje terminálu.....	80
14.1	Informační widget na inicializační obrazovce .....	83
14.2	RFID zámek – blokové schéma.....	84

## SEZNAM TABULEK

1.1 Srovnání dostupných terminálů na trhu.....	18
9.1 Tabulka přechodů pro stavový automat .....	58

# ÚVOD

Tato diplomová práce je vytvořena ve spolupráci s firmou Techcrowd s.r.o., která se zabývá zakázkovým vývojem webových stránek, aplikací a chytrých systémů. Firma Techcrowd získala celkovou zakázku na dodání docházkového systému pro zemědělskou akciovou společnost Bečváry, kde součástí této zakázky je vývoj a realizace docházkových terminálů společně s kompletním informačním systémem pro správu docházky. Kolegové s firmy Techcrowd zodpovídají za dodání informační systému v podobě webové aplikace pro správu docházky a za serverovou část.

Tématem této diplomové práce je vývoj a realizace docházkových terminálů, které jsou určeny jako součást docházkových systémů v zemědělském družstvu ZAS Bečváry a.s. Cílem je nahradit v současné době používané píchací hodiny, které neodpovídají dnešním požadavkům na administrativní zpracování docházky zaměstnanců.

Docházkové systémy tvoří nedílnou součást mnoha firem a podniků, kteří chtějí mít efektivní přehled o svých zaměstnancích, kde tyto systémy umožňují jednoduše sledovat, evidovat a vyhodnocovat docházku zaměstnanců.

Docházkové systémy se skládají jednak z docházkových terminálů, ke kterým mají zaměstnanci přístup a jednak z informačního systému ke kterému má přístup už pouze odpovědný personál, jelikož zde může kontrolovat, upravovat a vyhodnocovat docházku jednotlivých zaměstnanců.

V dnešní době existuje mnoho komerčních řešení docházkových terminálů, kde je možné si vybrat na základě typu ověřování uživatelů. Jsou řešení, která používají k ověření biometrické údaje, jako je otisk prstu nebo ověření pomocí rozpoznávání obličeje. Další možností je použití NFC nebo RFID karet a čipů, kde je uživateli přiřazen daný čip nebo karta, která má unikátní identifikační číslo, pomocí kterého se uživatel ověří. Avšak tyto terminály pracují společně s informačním systémem daného výrobce, tudíž nepodporují informační systémy třetích stran. Z tohoto důvodu by na ně nešlo napojit řešení informačního systému, kteří vyvíjejí kolegové, a proto se rozhodlo, že se vytvoří vlastní řešení docházkového terminálu.

# 1. TERMINÁLY DOSTUPNÉ NA TRHU

Docházkové terminály se většinou skládají z mikropočítače, který řídí veškerý běh terminálu, informativního displeje a zařízení ověřující identitu zaměstnance. V této kapitole jsou popsány terminály dostupné na trhu.

## 1.1 Docházkový terminál IT-WATT R3 [1]

Terminál (1.1) od firmy Ikos nabízí deseti palcový kapacitní dotykový displej o rozlišení 1280 x 800 bodů. Je postaven na systému Windows 10 IoT Enterprise, který disponuje pamětí ROM typu SSD o velikosti 32 GB. Terminál je napájen pomocí 12 až 24 V DC zdroje ze sítě.

Pro ověření uživatele se používají technologie bezkontaktní nebo kontaktní. Mezi bezkontaktní technologie patří technologie RFID (Radio Frequency Identification), která může operovat na frekvencích 125 kHz nebo 13,56 MHz. Pro kontaktní ověření je použit kontaktní optický snímač otisků prstů. V případě potřeby je možné terminál osadit CCD kamerou pro kontrolu identifikace, která disponuje rozlišením 320 x 240 bodů.

Provozní teploty terminálu se pohybují v rozsahu 0 až +35 °C a stupeň krytí IP 40, což znamená, že je terminál chráněn před vniknutím cizích předmětů velmi drobných a zcela bez ochrany proti vniknutí vody. Z toho lze odvodit, že terminál je uzpůsoben pro používání ve vnitřních prostorech, kde je možno zajistit stabilní pracovní podmínky.

Komunikace probíhá přes rozhraní Ethernet TCP/IP, kde v případě výpadku sítě je možné uživatele ověřit i v off-line režimu. Disponuje akustickou i optickou signalizací, která usnadní orientaci v grafickém prostředí terminálu. Pomocí terminálu je možno ovládat například otevření dveří pomocí dvou relé uvnitř terminálu.



Obrázek 1.1 Docházkový terminál Ikos IT-WATT R3 [1]

## 1.2 Docházkový terminál ZPad SYSDO BIOPAD [2]

Tento terminál (1.2) je vybaven sedmi palcovým kapacitním dotykovým displejem o rozlišení 1024 x 600 bodů. Je postaven na systému Android 6.0, který disponuje pamětí ROM typu FLASH o velikosti 8 GB. Terminál je napájen pomocí 12 V DC / 3A zdroje ze sítě.

Pro ověření uživatele se používají stejné technologie jako v předešlém terminálu (1.1), tudíž bezdotykový RFID snímač a dotykový optický snímač otisků prstů. V terminálu je osazena 2Mpx kamera pro případnou pozdější identifikaci zaměstnance, nejedná se tedy o rozpoznávání obličeje terminálem, ale člověkem, který bude mít správu nad terminály v dané firmě.

Provozní teploty terminálu se pohybují v rozsahu 0 až +35 °C a stupeň krytí není uveden. Z toho lze odvodit, že terminál je uzpůsoben pro používání ve vnitřních prostorech, kde je možno zajistit stabilní pracovní podmínky.

Komunikace probíhá přes rozhraní Ethernet TCP/IP anebo přes WiFi. Disponuje pouze optickou signalizací, která usnadní orientaci v grafickém prostředí terminálu. Pomocí terminálu je možno ovládat například otevření dveří pomocí jednoho relé uvnitř terminálu.



Obrázek 1.2 ZPad SYSDO BIOPAD [2]

### 1.3 Docházkový terminál Qubos Finger [3]

Terminál (1.3) od firmy cominfo nabízí sedmi palcový kapacitní dotykový displej o rozlišení 1024 x 600 bodů. Terminál je napájen pomocí 12 až 36V DC zdroje ze sítě nebo je možnost napájet terminál pomocí PoE (Power over Ethernet).

Nabízí se v mnoha provedeních, kde je na výběr základní verze Qubos, která poskytuje pouze RFID snímač, ale díky tomu má vyšší odolnost vůči vnějším vlivům, takže je možné tento terminál umístit do venkovního prostředí. Druhou verzí je zde vybraná, tedy Qubos Finger, která disponuje navíc čtečkou otisků prstů. Třetí verzí je Qubos Face, která umožňuje identifikovat uživatele pomocí kamery s rozpoznáním obličeje. Poslední verzí je Qubos Finger&Face, což je kombinace Qubos Finger a Qubos Face a jedná se tak o nejpokročilejší model výrobce.

Pro ověření uživatele se používají stejné technologie jako v předešlých dvou terminálech (1.1, 1.2), tudíž bezdotykový RFID snímač a dotykový optický snímač otisků prstů. Navíc je zde možnost vybrat obdobný model terminálu, který je osazen kamerou pro rozpoznávání obličejů.

Provozní teploty terminálu se pohybují v rozsahu -15 až +60 °C a stupeň krytí není přesně specifikován, ale podle katalogu [4] výrobce je možné zjistit, že terminál je vhodný do venkovního prostředí, pokud je umístěn pod střechou.



Obrázek 1.3 Docházkový terminál Qubos Finger [3]

## 1.4 Docházkový terminál GIRITON Na zed' 7“ [5]

Terminál (1.4) od firmy GIRITON nabízí sedmi palcový kapacitní dotykový displej. Je postaven na systému Android, a je napájen pomocí 12V/1,5A zdroje ze sítě nebo je zde možnost napájet terminál pomocí PoE.

K ověření uživatele se rovněž používají bezdotykové a dotykové snímače. Mezi ty dotykové patří optický snímač otisků prstů nebo ověření pomocí hesla zadáním do terminálu na dotykové klávesnici. Mezi bezdotykové patří RFID snímač na frekvencích 125 kHz nebo 13,56 MHz, dále NFC snímač a poslední možností je ověření pomocí QR kódu, který terminál přečte přes integrovanou kameru. Tato kamera zde slouží pro případnou pozdější identifikaci zaměstnance, nejedná se tedy o rozpoznávání obličeje terminálem, ale člověkem, který bude mít správu nad terminály v dané firmě.

Stupeň krytí není výrobcem uveden, ale jedná se o terminál určený pro používání ve vnitřních prostorech. Dále není výrobcem specifikován rozsah provozních teplot, ve kterém může být terminál umístěn.

Komunikace probíhá buďto přes rozhraní Ethernet TCP/IP anebo přes WiFi. Pomocí terminálu je možno ovládat například otevření dveří pomocí jednoho relé, které se nachází uvnitř terminálu.



Obrázek 1.4 Docházkový terminál GIRITON Na zed' 7“ [5]

## 1.5 Porovnání docházkových terminálů

Všechny terminály výše popsané terminály mají podobnou funkcionalitu a liší se například v použitých snímačích, v rozlišení displeje, úhlopříčkou a grafickým prostředím. Podrobné srovnání vybraných terminálů je uvedeno v tabulce 1.1. RFID čtečku na frekvenci 125 kHz a 13,56 MHz používají všechny zmíněné terminály krom terminálu ZPad, který podporuje pouze frekvenci 125 kHz. Technologii NFC, využívá pouze terminál Qubos Finger.

Všechny možné terminály umožňují pořízení snímku osoby, která provedla akci na terminálu. Tuto fotku posléze uloží nebo odešle na server, kde ji manuálně ověří člověk zodpovědný za docházku v dané firmě. Na trhu je i varianta Qubos Finger&Face [6], která umožňuje identifikaci uživatele pomocí rozpoznání obličeje.

Co se týče kontaktních snímačů, tak všechny využívají snímač otisku prstů, kde Qubos a GIRITON tuto variantu nabízí za příplatek, kde v případě GIRITONu se jedná o příplatek 5 000 korun českých.

Všechny terminály jsou určeny do vnitřních prostor, jelikož nemají patřičnou ochranu proti vniknutí cizích těles a kapalin. Jedinou výjimkou je terminál Qubos, a to konkrétně Qubos (základní verze) a Qubos Face, kteří jsou schopni pracovat ve venkovním prostředí.

Všechny terminály jsou napájeny minimálně 12 V DC, kde terminály Qubos a GIRITON využívají technologii PoE (Power over Ethernet). Jedná se o napájení po ethernetovém kabelu podle specifikace IEEE 802.3af využívá standardní Category 5 (CAT-5) kabel. Přestože CAT-5 kabel obsahuje čtyři kroucené páry, pouze dva z nich se u sítí 10BaseT a 100BaseT používají pro přenos dat [7]. Tato varianta napájení je velice vhodná, jelikož přenos dat i napájení obstarává jeden kabel.

Displeje terminálů jsou nejčastěji sedmi palcové, avšak najdou se i výjimky, které mají větší displej, a to konkrétně deseti palcový. Ve srovnávaných terminálech se jedná o terminál IT-WATT R3. Větší displeje, jak deset palců se běžně na trhu nevyskytují, jelikož by byl terminál zbytečně rozměrný. Rovněž to platí pro displeje menší než sedm palců, a to tentokrát z důvodu nečitelnosti, kde grafické prostředí by bylo příliš malé, a tím neergonomické pro používání.

Komunikace probíhá u všech terminálů pomocí Ethernet rozhraní. Další možností komunikace je pomocí rozhraní RS-232. Tuto možnost komunikace využívají pouze terminály IT-WATT R3 a Qubos. ZPad a GIRITON komunikační rozhraní RS-232 nenabízejí, avšak nabízejí alternativu v podobě WiFi.

Qubos a GIRITON nabízejí za příplatek mobilní aplikaci, přes kterou si uživatel může zkontrolovat svoji docházku anebo se pomocí této aplikace identifikovat u terminálu, kde se mobilní aplikace v mobilním telefonu spáruje s terminálem pomocí Bluetooth nebo pomocí porovnání geolokace terminálu a mobilního telefonu. Pokud geolokace terminálu a mobilního telefonu bude shodná, tak uživateli dovolí zapsat a uložit docházku do systému.

Možnost záložního akumulátoru v případě výpadku elektrické sítě nabízí pouze terminál ZPad, a to jako volitelnou položku za příplatek, který na jejich webových stránkách [2] není uveden.

Posledním parametrem v tabulce 1.1 je pořizovací cena terminálu, kde je jsou známy pouze ceny terminálů GIRITON a ZPad SYSDO. Ceny terminálů začínají na 7 380,- bez DPH v případě ZPadu SYSDO a pokračuje na 8 900,- bez DPH u terminálu



GIRITON, Ostatní terminály neuvádí pořizovací cenu na svých webových stránkách [1],[3].

Tabulka 1.1 Srovnání dostupných terminálů na trhu

Název terminálu	IT-WATT R3	ZPad SYSDO	Qubos Finger	GIRITON
<b>Operační systém</b>	Windows 10 IoT	Android 6.0	-	Android x.x
<b>RFID</b>	125 kHz, 13,56 MHz	125 kHz	125 kHz, 13,56 MHz	125 kHz, 13,56 MHz
<b>NFC</b>	Ne	Ne	Ano	Ne
<b>Otisk prstu</b>	Ano	Ano	Ano*	Ano*
<b>Kamera</b>	Ano (QVGA)	Ano (2 Mpx)	Ne (Ano – jiný model Qubos Finger&Face)	Ano
<b>Rozpoznání obličeje</b>	Ne	Ne	Ne (Ano – jiný model Qubos Finger&Face)	Ne
<b>Pracovní prostředí</b>	Vnitřní prostory	Vnitřní prostory	Venkovní/venkovní prostory	Vnitřní prostory
<b>Napájení</b>	12–24 V DC	12 V DC	12–36 V DC	12 V DC
<b>PoE</b>	Ne	Ne	Ano	Ano
<b>Úhlopříčka displeje</b>	10.1“	7“	7“	7“
<b>Rozlišení displeje</b>	1280 x 800 px	1024 x 600 px	800 x 480 px	-
<b>Komunikace</b>	Ethernet TCP/IP, RS-232, RS-485	Ethernet TCP/IP, WiFi	Ethernet TCP/IP, RS-232	Ethernet TCP/IP, WiFi, Bluetooth
<b>Mobilní aplikace</b>	Ne	Ne	Ano*	Ano*
<b>Záložní akumulátor</b>	Ne	Ne*	Ne	Ne
<b>Rozměry (V x Š x H)</b>	224 x 251 x 52 mm	240 x 130 x 45 mm	190 × 219 × 40 mm	265 x 155 x 65 mm
<b>Cena bez DPH (Kč)</b>	Cena nezveřejněna	7 380	Cena nezveřejněna	8 900 bez 13 900 s otisky prstů

## 2. POŽADAVKY ZÁKAZNÍKA

Zákazníkem je zemědělské družstvo ZAS Bečváry a.s., které má následující požadavky. Zaměstnanci budou mít k dispozici čip nebo kartu pro provedení autorizace na terminálu. Dále je požadováno zachycení snímku zaměstnance při provedení akce docházky, aby se zamezilo podvodům. Jsou požadovány tři základní akce evidence docházky:

- Příchod
- Odchod
- Lékař

Je nutné zajistit univerzálnost terminálů pro případnou výměnu nefunkčního terminálu anebo rozšíření systému. Zákazník dále požaduje jednoduché uživatelské prostředí, které bude jednoduše ovládatelné. Následně je vyžadována zvuková signalizace o provedení či neprovedení akce. Zákazník nepožaduje zálohování napájení a dat terminálu pro případ výpadku elektrické sítě nebo internetového připojení.

Doposud sloužily k evidenci docházky v zemědělském družstvu píchací hodiny (obr. 2.1). Tato metoda je již zastaralá náchylná na vznik chyb, jelikož je potřeba zaměstnancem firmy ručně přepisovat hodiny u každého zaměstnance do tabulek v programu MS Excel pro výpočet mezd. Z důvodu zefektivnění práce a dohledem nad svými zaměstnanci se družstvo rozhodlo zmodernizovat evidenci docházky pomocí docházkového systému.



Obrázek 2.1 Píchací hodiny

Celkově bude po areálu rozmístěna trojice terminálů. Na obrázku 2.2 je zobrazena satelitní mapa areálu zemědělského družstva, kde jsou barevně vyznačené hlavní budovy, ve kterých jsou plánovány docházkové terminály. Červeně označená je administrativní budova, která se nachází na nejzápadnější části areálu. Druhou budovou je dílna, která je označená žlutou barvou a na mapě je na severozápadní části areálu.

Třetí a poslední budovou, která je umístěna ve východní části areálu, je budova pod názvem „živočišná“ a na mapě je označená modrou barvou.

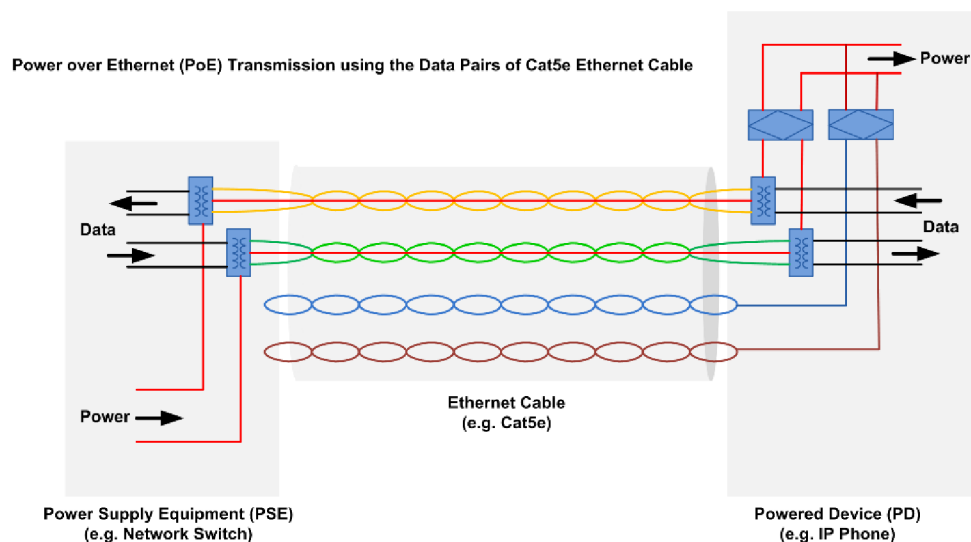


Obrázek 2.2 Areál zemědělského družstva – vyznačení budov s terminály

### 3. POŽADAVKY NA DOCHÁZKOVÝ TERMINÁL

Zákazníkem jsou požadovány celkem čtyři docházkové terminály, kde tři budou nainstalovány v zemědělském družstvu ZAS Bečváry a jeden bude sloužit jako rezervní, v případě poruchy jednoho ze tří nainstalovaných terminálů anebo jako pozdější rozšíření systému.

Terminály budou umístěny ve vnitřních prostorech budov zemědělského družstva (obr. 2.2), tudíž není nutné terminály navrhovat tak, aby splnily stupně krytí IP 54 a výše. Terminály budou umístěny u vchodů do jednotlivých budov ve výšce 150 cm nad zemí. Tato výška byla zvolena z důvodu kamery, která potřebuje být dostatečně vysoko, aby dokázala vyfotit obličej zaměstnance. Takovou výšku je možné použít, jelikož je zde nepočítá s bezbariérovým přístupem. U jednotlivých vchodů je pro terminály již připravena datová zásuvka pro připojení k internetu. Nenachází se zde však zásuvka na elektrickou síť, proto je nutné terminál napájet pomocí PoE (Power over Ethernet). Pro zajištění tohoto typu napájení je nutné přidat do switch zařízení (switche) PoE injektor, který pomocí volných vodičů v síťovém kabelu bude napájet terminál (obr. 3.1).



Obrázek 3.1 Princip PoE [7]

Použití akumulátoru se nevažuje z důvodu nepožadování záložního napájení zákazníkem. Jediným zdrojem napájení bude tedy PoE a případě výpadku elektrické sítě terminál nebude fungovat.

Terminál bude disponovat dotykovým displejem, RFID čtečkou, kamerou. Dále je potřeba použít vhodnou řídicí jednotku, která bude jednotlivé komponenty ovládat. Další použitou komponentou je bzučák nebo podobné zvukové zařízení pro zvukovou

signalizaci, aby zaměstnanec měl zpětnou vazbu o provedení akce nebo o případné chybě. Jednotlivé komponenty a řídicí jednotka jsou přesněji popsány a zdůvodněn jejich výběr v kapitole 5.

Dotykový displej osazený v terminálu by měl mít dostatečné rozlišení a úhlopříčku minimálně sedm palců, jelikož terminál bude umístěn na stěně, kde bude zaměstnancům vzdálen na dosah ruky (do 1 m). Na tomto displeji zákazník požaduje zobrazení aktuálního času a volbu jednotlivých akcí docházky. Dále je požadováno, aby jednotlivá tlačítka pro volbu akce byla dostatečně velká a byla barevně odlišená, pro lepší ovládání a zamezení nechtěným zápisům docházky.

Identifikace zaměstnance pomocí biometrických údajů jako je čtečka otisku prstů nebude použita z důvodu možného nerozpoznání otisku prstu zaměstnance. Jelikož zaměstnanci v zemědělském družstvu, kde budou terminály instalovány, pracují například i v dílnách, jezdí se stroji, tak by byla pro ně přítěž si pokaždé sundávat rukavice. Nebo může nastat situace, kdy zaměstnanec bude mít špinavé ruce od oleje nebo je může mít pořezané, tak by je čtečka otisku prstů nerozpoznala. Je ale možné použít čtečku otisku prstů v kombinaci s další metodou identifikace, například se čtečkou RFID. Ale tím by se zbytečně terminál prodražil. Vhodnější by bylo, použít již zmíněnou čtečku RFID. Kde by se zaměstnanci identifikovali pouze přiložením karty ke čtečce umístěné v terminálu. Tato čtečka není nikterak náchylná na případné zašpinění. Navíc je tato technologie levnější než čtečka otisku prstů. Vystává zde však problém s možným falešným identifikováním, kde by jeden zaměstnanec mohl vzít vícero karet a provést docházku za ostatní zaměstnance. Z tohoto důvodu je požadována v terminálu kamera, která by sloužila jen pro ověření daného uživatele. Nejednalo by se však o rozpoznávání obličeje, ale byla by pořízena fotografie zaměstnance při provedení akce docházky, ta by se následně uložila se všemi potřebnými daty do databáze. Tuto fotografii si může prohlédnout daná odpovědná osoba ve firmě a ověřit pravost docházky, pokud by nastaly pochybnosti o docházce daného zaměstnance.

## 4. KOMUNIKACE TERMINÁLU SE SERVEREM

Pro komunikaci terminálů se serverem je zvolen komunikační protokol HTTPS [9]. Protokol funguje na principu dotaz-odpověď. Pro komunikaci terminálu se serverem budou používány požadavky POST a DELETE.

Komunikace terminálu se serverem je popsána vývojovým diagramem na obrázku 4.1. Druhý detailněji popsán diagram je přiložen v příloze A.1. Uvažujme výchozí stav, kdy je terminál spuštěn a čeká až zaměstnanec přiloží kartu. Následně se zaměstnanec pokusí o přihlášení přiložením RFID karty. Pro přihlášení zaměstnance se odešle na server požadavek typu POST /api/terminal/authorization, kde se jako parametr vloží identifikační číslo terminálu a identifikační číslo přiložené karty. Obě tyto čísla musí být před odesláním přetypována do formátu řetězec.

Pokud obě identifikační čísla souhlasí s údaji uloženými v databázi, server vrátí odpověď se stavovým kódem 200 společně s následujícími údaji o uživateli:

- ID zaměstnance
- Jméno a příjmení zaměstnance
- Roli
- Dostupné akce
- Poslední přihlášení
- Přiřazené karty
  - ID karty, pod kterým je uložena v DB
  - Identifikační číslo
  - Datum vytvoření

Pokud některé ze zaslanych identifikačních čísel neexistuje, server vrátí odpověď se stavovým kódem 403. Pokud uživatel není autorizován, server vrátí stavový kód 401.

Pokud byl zaměstnanec korektně ověřen (vrácen stavový kód 200), terminál zobrazí na displeji nabídku s volbou akce (příchod, odchod, lékař). Server zajistí, že v případě, kdy zaměstnanec již zvolil akci příchod, tak při dalším přiložení karty mu bude nabídnuta akce „odchod“ a „lékař“. Pokud byl poslední akcí „odchod“ nebo „lékař“, zobrazí se pouze možnost příchod.

Na následující stránce je vyobrazena ukázka odpovědi serveru na požadavek typu POST /api/terminal/authorization {"0003","3338790"}:





Zaměstnanec potvrdí zvolenou akci (např. příchod) a zobrazí se mu nová vyskakovací obrazovka na terminálu, kde bude zobrazeno následující:

- Provedená akce
- Jméno a příjmení zaměstnance
- Identifikační číslo karty zaměstnance

Během zobrazování obrazovky, terminál pomocí metody POST /api/terminal/action odešle zprávu obsahující následující údaje:

- “actionId“ – identifikační číslo vybrané akce na terminálu
- “cardId“ – identifikační číslo karty zaměstnance
- “deviceId“ – identifikační číslo terminálu
- “image“ – pořízená fotografie zaměstnance ve formátu JPEG zakódovaná ve formátu base64 (\*not required)

Pokud je některé ze zaslanych identifikačních čísel chybné (neodeslání fotografie se nepovažuje za chybu), server vrátí odpověď se stavovým kódem 400. Pokud je zaměstnanec neautorizovaný, server vrátí stavový kód 401. V obou případech se zaměstnanci zobrazí chybová obrazovka s informací o neúspěšně zapsané docházce a následně je zaměstnanec odhlášen. Pakliže jsou odeslaná data korektní, je vrácen stavový kód 200, docházka je zaevidována v databázi a zaměstnanec může být odhlášen.

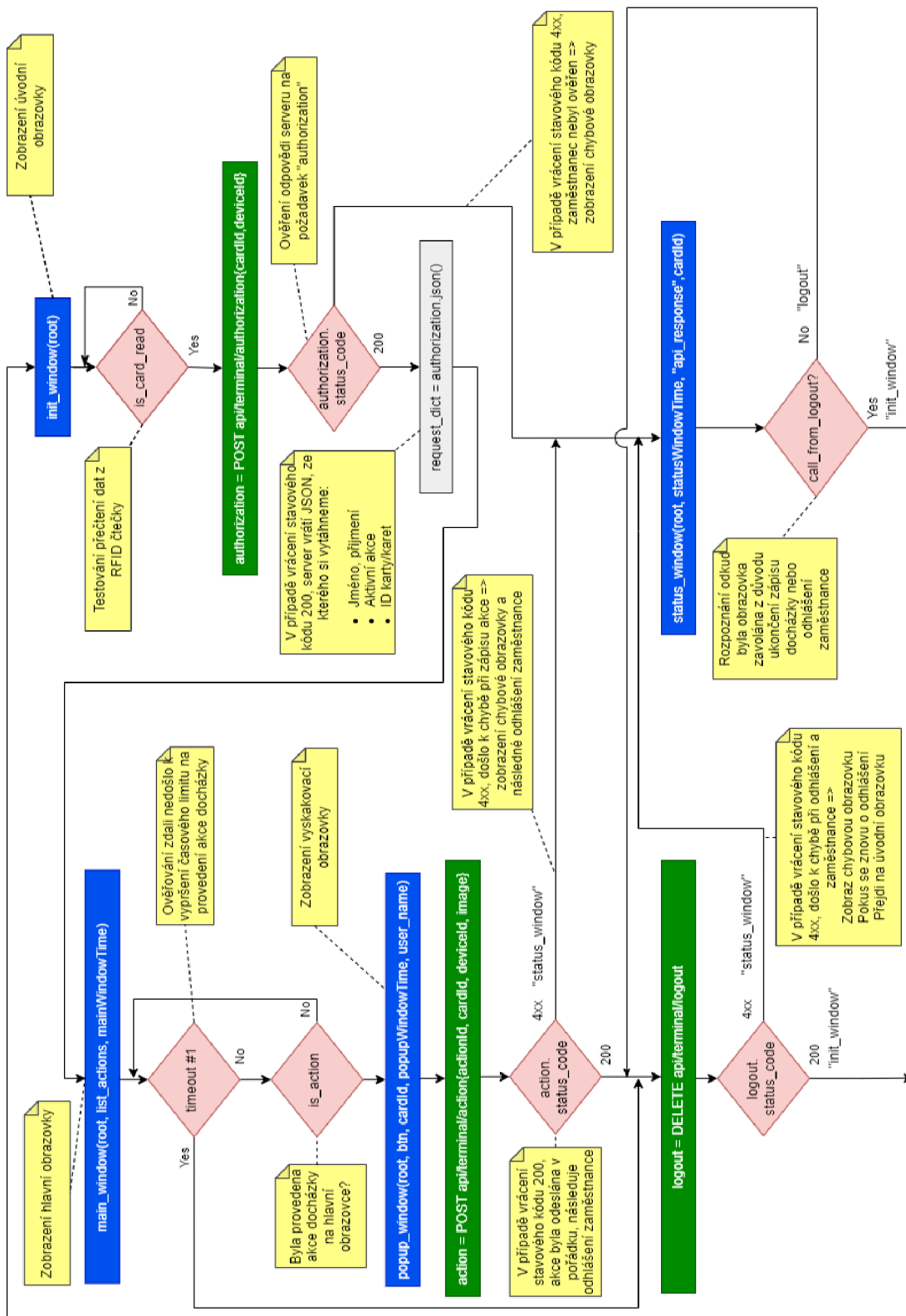
Ve všech případech je posledním krokem odhlášení zaměstnance, a to z důvodu, aby mohlo dojít k zápisu docházky u dalších zaměstnanců. Pro odhlášení se odešle požadavek DELETE /api/terminal/logout, kde se nic nedává do parametru. V případě špatného přístupu, server vrátí stavový kód 400. V případě úspěšného projití požadavku vrátí stavový kód 200, který značí úspěšné odhlášení zaměstnance a terminál je nyní připraven na příjem dat z RFID čtečky, tedy na zapsání docházky u dalšího zaměstnance.

Ukázka odpovědi na DELETE /api/terminal/logout:

```
{
  "message": "Successfully logged out",
  "code": 200
}
```

Sekvence jednotlivých akcí je znázorněna sekvenčním diagramem uvedeném v příloze A.2.



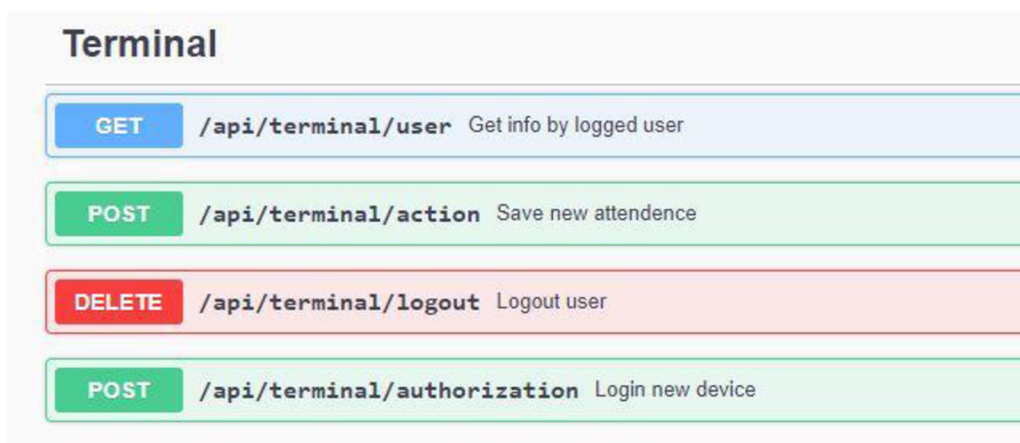


Obrázek

4.1 Vývojový diagram – komunikace terminál x server

Pro docházkový systém a rovněž pro docházkový terminál byla vytvořena API dokumentace od kolegů z firmy Techcrowd s.r.o., která je vytvořená v softwaru Swagger. Swagger je open source framework pro návrh, tvorbu, dokumentaci a konzumaci RESTful web API [8], kde REST znamená Representational State Transfer, což je termín, jak jednoduše vytvořit, číst, editovat nebo smazat informace ze serveru pomocí HTTPS požadavků [9]. API znamená Application Programming Interface neboli rozhraní pro programování aplikací [9].

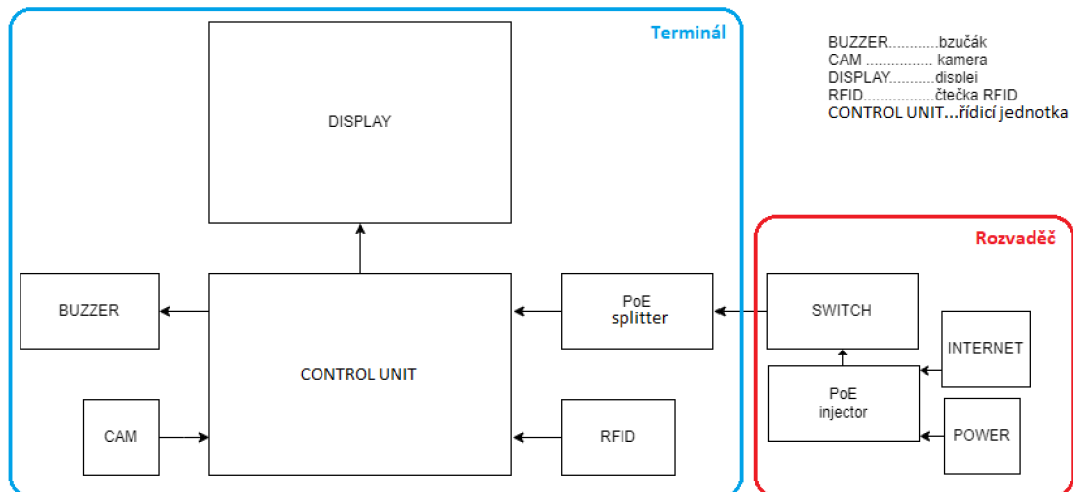
API dokumentace vytvořená kolegy z firmy Techcrowd s.r.o. je dostupná na adrese [38]. Na obrázku 4.2 jsou zobrazeny požadavky pro terminál v softwaru Swagger.



Obrázek 4.2 Dostupné požadavky pro terminál

## 5. NÁVRH DOCHÁZKOVÉHO TERMINÁLU

V této kapitole je popsán výběr jednotlivých komponent docházkového terminálu na základě požadavků v kapitole 2. Rovněž podle těchto požadavků vyplynulo následující blokové schéma docházkového terminálu (obr. 5.1).



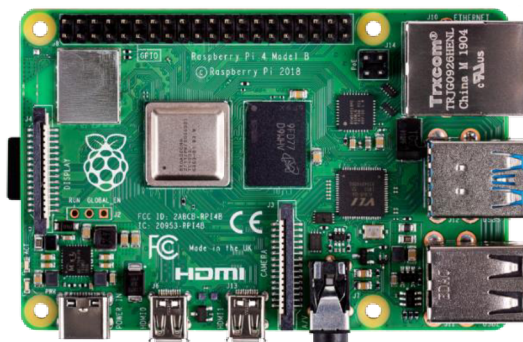
Obrázek 5.1 Blokové schéma terminálu a jeho připojení do rozvaděče

### 5.1 Řídící jednotka

Základem docházkového terminálu je řídicí jednotka. Řídící jednotku je možno navrhnout kompletně od začátku, ale toto řešení bylo zavrhnuto z důvodu finanční a časové náročnosti. Z tohoto důvodu byla vybrána řídicí jednotka, která je běžně dostupná na trhu.

#### 5.1.1 Raspberry Pi 4 Model B [11]

Raspberry Pi (dále jen RPi) je nejrozšířenější jednodeskový počítač na světě. Verze 4 Model B nabízí 1GB, 2GB, 4GB a 8GB RAM (Random Access Memory) typu LPDDR4. Má osazený čtyřjádrový 64 bitový procesor Broadcom BCM2711 na frekvenci 1,5 GHz. Pro zobrazovací výstup poslouží DSI port nebo dva microHDMI porty, které jsou schopny přenášet 4K obraz při 60 snímcích za sekundu. Navíc je zde dedikovaný CSI port pro kameru. Pro napájení slouží USB typu C, kde je doporučeno výrobcem napájení 5 V / 3 A. Deska je také vybavena Full Gigabit Ethernetem (až 1000 Mbit/s) a WiFi modulem. Operační systém je Raspberry Pi OS (dříve Raspbian). Jelikož je tento počítač mezi vývojáři velice oblíbený, tak existuje mnoho knihoven a vytvořených projektů, ze kterých se dají čerpat informace. Navíc existuje mnoho kompatibilních vstupně/výstupních periférií, pro které jsou na internetu volně dostupné návody. Cena počítače závisí na velikosti paměti RAM, kde cena začíná na 899 korunách za základní 1GB verzi [13] a končí na 2069 korunách za 8GB verzi [12].



Obrázek 5.2 Raspberry Pi 4 Model B [14]

### 5.1.2 Radxa ROCK Pi 4 C+ [15]

Jedná jednodeskový počítač je v některých ohledech lepší volbou než Raspberry Pi. Stejně jako RPi nabízí 1GB, 2GB, 4GB a 8GB RAM typu LPDDR4. Je osazen šesti jádrovým procesorem Rockchip RK3399-T na frekvenci 2,4 GHz. Pro zobrazovací výstup slouží DSI port a jeden HDMI port. Je zde rovněž osazen CSI port pro kameru. Pro napájení slouží USB typu C, nebo GLAN port s podporou PoE. Navíc je možné rozšířit ROM paměť pomocí volného M.2 slotu. Uživatel si může vybrat ze tří dostupných operačních systémů: Android 10 Tablet version, Ubuntu Server 20.04, Debian 9 Desktop. Tento počítač ale nepodporuje některé knihovny pro RPi, převážně ty, které komunikují se vstupně/výstupními piny. Navíc není ve světě tak rozšířený, tudíž neexistuje tolik návodů a projektů. Cena je stanovena na 2765 za 4GB verzi [16], ostatní verze nejsou v ČR dostupné.



Obrázek 5.3 Radxa ROCK Pi 4 C+ [15]

### 5.1.3 Výběr řídicí jednotky

Na trhu jsou dostupné i další obdobné počítače (např. BANANA Pi M5 [18]), které by byly vhodné pro implementaci do terminálu, ale do výběru nejsou zařazeny, jelikož mají hodně podobné parametry anebo se jedná o klony RPi. Tudíž ve výběru je RPi 4 Model B a Radxa ROCK 4 SE.

Výhodou Radxy je o 50 % více jader oproti RPi a také možnost rozšíření ROM paměti pomocí volného M.2 slotu. Další výhodou je možnost napájení pomocí GLAN portu s podporou PoE.

Rpi má výhodu v početné komunitě, která vytváří návody, projekty a knihovny, ze kterých mohou ostatní uživatelé čerpat. Další výhodou je velké množství vstupně/výstupních komponent jako jsou displeje, kamery, rozšiřovací GPIO shieldy a mnoho dalšího. A v neposlední řadě má RPi výhodu v nižší pořizovací ceně, která je 1539 korun za 4GB verzi oproti 2765 korunám na Radxu ve stejné paměťové variantě.

Podle těchto informací byl zvolen počítač RPi 4 Model B, a to z důvodů větší komunity, více vstupně/výstupních komponent, nižší pořizovací cena a rovněž díky zkušenostem s touto jednotkou z předchozích projektů.

## 5.2 Dotykový displej

Tato podkapitola bude projednávat popis jednotlivých dotykových displejů pro RPi 4 Model B, který byl vybrán v kapitole 5.1.3. Displej musí splňovat požadavky dané v kapitole 3.

### 5.2.1 Raspberry Pi 7“ [19]

Jedná se o oficiální dotykový displej pro RPi, který disponuje úhlopříčkou 7 palců o rozlišením 800 x 480 bodů a konektorem DSI, který je přítomen na RPi.

Typ panelu je LCD (Liquid Crystal Display) s dotykovou kapacitní vrstvou, která dokáže zaznamenat až 10 dotyků najednou.

Pro propojení displeje s RPi slouží DSI (Display Serial Interface) port a dva napájecí piny (5V a GND). Poté je možné upevnit RPi na zadní stranu displeje pomocí 4 distančních podložek, které jsou součástí balení. Pro tento displej existuje mnoho projektových řešení a různých doplňků, jako jsou například pouzdra. Pořizovací cena se pohybuje v rozmezí 1900 až 2200 korun českých [20].



Obrázek 5.4 Raspberry Pi 7“ [20]

### 5.2.2 LaskaKit 10.1“ [21]

Jedná se o dotykový displej od firmy LaskaKit, který disponuje úhlopříčkou 10 palců o rozlišením 1024 x 600 bodů a HDMI konektorem pro spojení s RPi.

Typ panelu je stejný jako v 5.2.1, tudíž LCD s dotykovou kapacitní vrstvou, která dokáže zaznamenat až 10 dotyků zároveň.

Pro propojení slouží proprietární redukce HDMI/miniHDMI. Dále displej nabízí dva 5W reproduktory a externí ovládání hlasitosti. RPi je zde možno připevnit na záda displeje pomocí distančních podložek, které jsou součástí balení. Pořizovací cena je 2988 korun.



Obrázek 5.5 LaskaKit 10.1“ [22]

### 5.2.3 Výběr displeje

Na trhu existuje mnoho dalších variant displejů, ale často disponují menší úhlopříčkou než 7 palců a tyto displeje nesplňují požadavky v kapitole 3. Další typy displejů disponují úhlopříčkou v rozmezí 7 až 10 palců, ale nemají možnost na své tělo připevnit RPi, tudíž nebyly uvažovány ve výběru. Ve výběru zůstal oficiální displej RPi a větší displej od firmy LaskaKit. Na trhu jsou podobné varianty těchto displejů, ale nebyly zařazeny do výběru, jelikož se jedná o velice podobné produkty.

Výhodou oficiálního RPi displeje je připojení pomocí DSI portu k RPi, nižší pořizovací cena, lepší dostupnost na trhu a možnost lepšího uchycení na tělo terminálu, kde jsou na zadní straně displeje připraveny čtyři závity. Nevýhodou může být nepřítomnost reproduktorů a náchylnost na poškození flex kabelů displeje, během manipulace.

Výhodou LaskaKit displeje je jeho větší úhlopříčka, dva 5W reproduktory, možnost nastavení hlasitosti analogově a stojánek pro prvotní fázi vývoje. Nevýhodou však je horší rozlišení na palec (119 dpi) oproti oficiálnímu RPi displeji (133 dpi), horší možnost uchycení displeje na tělo terminálu a vyšší pořizovací cena.

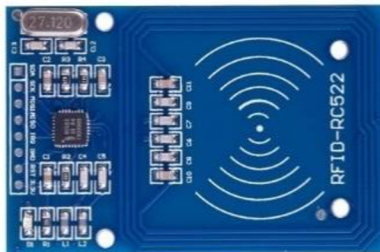
Podle výše zmíněných výhod a nevýhod byl zvolen oficiální dotykový displej Raspberry Pi 7". Byl vybrán z důvodu nižší pořizovací ceny, lepší možnosti uchycení na tělo terminálu, lepší dostupnosti a možnosti objednat tento displej i mimo tuzemský trh. Dalším důvodem byla i menší úhlopříčka, tím pádem tělo terminálu nebude muset být tak velké a ušetří se tak na materiálu.

## 5.3 RFID čtečka

Tato podkapitola bude projednávat popis a výběr jednotlivých RFID čteček dostupných na trhu.

### 5.3.1 MF RC522 [23]

Jedná se o RFID čtečku na frekvenci 13,56 MHz s integrovanou anténou. Komunikace probíhá skrze SPI sběrnici. Napájecí napětí je 3,3 V a provozní proud se pohybuje v rozmezí 13 až 26 mA. Rozměry činí 40 x 60 mm a pořizovací cena je 59 korun.

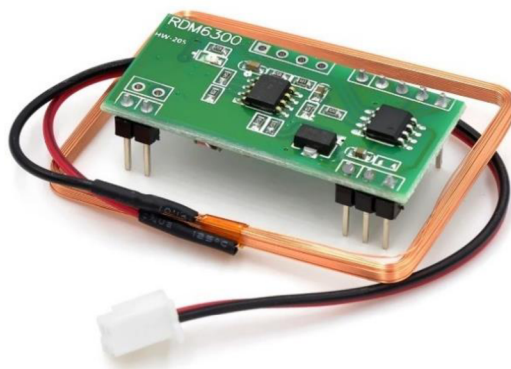


Obrázek 5.6 MF RC522 [24]



### 5.3.2 RDM6300 [25]

Jedná se o RFID čtečku pracující na frekvenci 125 kHz s externí anténou a je vhodná pro karty nebo čipy typu EM4100 a EM410X. Komunikace probíhá pomocí sériového rozhraní (TTL RS232) s rychlostí 9600 baudů a pro její napájení je nutné použít napětí o velikosti 5 V. Pracovní proud je menší jak 50 mA. Rozměry činí 38,5 x 19 mm a pořizovací cena je 117 korun.



Obrázek 5.7 RDM6300 [26]

### 5.3.3 USB čtečka 125 kHz [27]

Jedná se o USB čtečku, která pracuje na frekvenci 125 kHz a je vhodná pro karty nebo čipy typu EM4100 a EM410X. Na trhu existuje obdobná verze, která pracuje na frekvenci 13,56 MHz [35]. Obsahuje vestavěný bzučák a LED diody pro zjištění stavu napájení. Komunikace probíhá pomocí sériové komunikace (USB) a funguje na principu Plug and Play, kde je veškeré potřebné ovladače automaticky nainstalují. Rozměry jsou 95 x 60 mm a pořizovací cena je 265 korun.



Obrázek 5.8 USB čtečka 125 kHz [28]



### 5.3.4 Výběr RFID čtečky

Na trhu existuje mnoho dalších variant RFID čteček, které by byly vhodné pro implementaci do terminálu, ale jedná se převážně o podobné čtečky, které jsou již ve výběru.

Výhodou MF RC522 je nejnižší pořizovací cena, nejnižší napájecí napětí a malé rozměry. Nevýhodou však může být integrovaná anténa, jelikož může být výhodné, aby se anténa mohla umístit na kraj těla terminálu.

Výhodou RDM6300 je externí anténa, kterou je možno vyměnit a může být přidělena na kraj těla terminálu. Další výhodou jsou menší rozměry. Nevýhodou však může být vyšší pořizovací cena.

Výhodou USB čtečky je jednoduché připojení Plug and Play a vestavěný bzučák. Nevýhodou jsou větší rozměry, integrovaná anténa a vyšší pořizovací cena.

Na základě výše zmíněných výhod a nevýhod byla zvolena čtečka RDM6300 (5.3.2) z důvodů nejmenších rozměrů a použití externí antény. Externí anténa umožní větší variabilitu umístění na těle terminálu.

## 5.4 Kamera

Jako kamera byla zvolena oficiální Raspberry Pi Camera. Momentálně existují 3 verze této kamery.

Verze V1 [29], která disponuje 5Mpx čipem OmniVision OV5647 se světelností  $f/2,9$ . Pořizovací cena je 168 korun. Verze V2 (obr. 5.9) [30] disponuje 8Mpx čipem Sony IMX219 se světelností  $f/2,7$  s manuálním nastavením ostření. Pořizovací cena je 399 korun. Verze V3 [31] disponuje 12Mpx snímačem Sony IMX708 se světelností  $f/1,8$  a automatickým ostřením. Pořizovací cena je 695 korun.

Pro implementaci do terminálu je zvolena verze V2 z důvodu dostačující kvality obrazu, nízké ceně a také z důvodu delší doby na trhu, kdy vyšlo několik drobných vylepšení a oprav, tudíž by měla být nejspolehlivější. Verze V1 nebyla vybrána z důvodu horší dostupnosti na trhu a nejhoršího rozlišení. Verze V3 nebyla vybrána z důvodu vyšší pořizovací ceny a také z důvodu krátké doby na trhu, jelikož tato verze vyšla v roce 2023 a mohou se projevit jisté problémy z počátku životnosti verze.



Obrázek 5.9 Raspberry Pi kamera V2 [32]

## 5.5 Zvuková signalizace

Pro zvukovou signalizaci byl vybrán modul akustického bzučáku [33]. Tento modul má rozměry 33 x 13 mm a napájí se 3,3 V. Na desce modulu je otvor pro snadné připevnění modulu pomocí šroubu. Tento modul akustického bzučáku byl vybrán z důvodu, že tyto moduly jsou k dispozici ve skladu firmy.



Obrázek 5.10 Modul akustického bzučáku [34]

## 5.6 Napájení terminálu pomocí PoE

Jako PoE splitter byl vybrán Joy-It SBC-PoE-Power-C splitter (obr. 5.11). Tento splitter je vybaven USB-C konektorem, tudíž je kompatibilní s vybraným RPi. Výstupní napětí/proud je 5 V/2 A, dosah napájení je až 100 m. Pořizovací cena je 499 korun. Tento konkrétní model byl vybrán z důvodu vybavení USB-C konektorem a také pro jeho malé rozměry.



Obrázek 5.11 PoE splitter Joy-It SBC [36]

Protože ethernetové switche umístěné v rozvaděčích nejsou vybaveny porty s PoE, je třeba PoE pro napájení terminálů zajistit pomocí externího PoE injektoru. Byl vybrán injektor TP-Link TL-POE150 [48], který pracuje na PoE standardu 802.3af (PoE). Maximální výstupní výkon na port činí 15,4 W, což stačí pro napájení terminálu. Pořizovací cena činí 427 korun.



Obrázek 5.12 PoE injektor TP-Link TL-POE150S [48]

## 6. NÁVRH MECHANICKÉ KONSTRUKCE

V této kapitole je probrán návrh konstrukčního uspořádání docházkového terminálu od prvotních návrhů designu konstrukce až po finální verzi, která je detailně popsána.

### 6.1 Prvotní návrhy

Z počátku se vytvořilo několik různých konceptů designu konstrukce docházkového terminálu v programu Fusion360, aby se poté mohlo pracovat detailněji na zvoleném konceptu. Hlavním cílem bylo, aby design konstrukce byl v souladu s logem firmy Techcrowd s.r.o. (obr. 6.1).



Obrázek 6.1 Logo Techcrowd s.r.o. [37]

Celkem byly vytvořeny 4 koncepty designu. První návrh je na obrázku 6.2, kde dominantu tvoří sedmi palcový dotykový displej vybraný v kapitole 5.2.3. Celý displej je zasazen do konstrukce, která má stejnou tloušťkou po celém obvodě a to 5mm. Pod displejem je vytvořen prostor pro RFID anténu a kameru. Není zde však udělán žádný grafický prvek, který by se odlišil a který by byl na první pohled rozpoznatelný, že se jedná o výrobek od firmy Techcrowd s.r.o. Tudíž tento návrh byl zamítnut vedením firmy.



Obrázek 6.2 První návrh konstrukce

Druhý návrh (obr. 6.3) se nesl v podobném designovém směru, tedy stejná základní konstrukce, do které je zapuštěn displej rovnoměrně po celém okraji. Tentokrát byl vytvořen grafický prvek odkazující na logo firmy. Byl umístěn do levého rohu prostoru pod displejem, kde roh byl vytvarován tak, aby přesně seděl se zaoblením „t“ v logu. Navíc zde byly přidány prolisy, které byly souměrné s danou částí loga. Avšak i tento návrh byl zamítnut, kvůli nesymetrickému zakončení prolisu na pravé straně prostoru pod displejem a také kvůli ostrým hranám.



Obrázek 6.3 Druhý návrh konstrukce

Třetí návrh (obr. 6.4) upravil neduhy z druhého návrhu. Tedy prostor pod displejem byl udělán, aby byl symetrický a došlo také k zaoblení hran. Avšak prolisy kopírující linie loga se nepodařilo esteticky implementovat, takže vznikla nesymetrie prolisů v levé a pravé části, kde se prostor pro anténu RFID a kameru spojuje s tělem displeje. Tento návrh nebyl firmou zamítnut, avšak stále se nejednalo o nejlepší řešení, tak byl vytvořen čtvrtý návrh, kompletně odlišný od přechozích návrhů.



Obrázek 6.4 Třetí návrh konstrukce

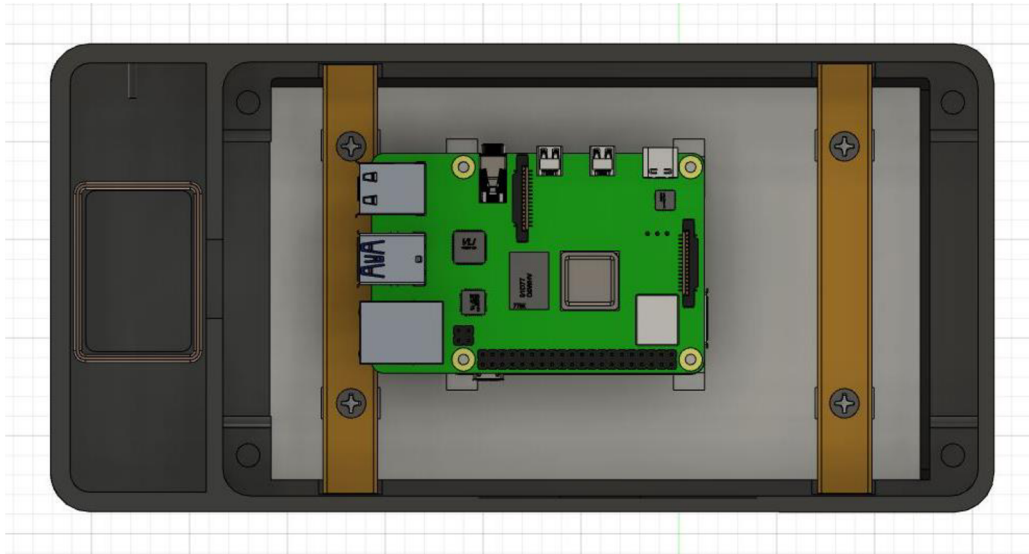
Ve čtvrtém návrhu (obr. 6.5) se přesunul prostor pod displejem, který byl určený pro kameru a anténu RFID čtečky, na pravou stranu. Tím se vytvořilo více prostoru pro již zmíněné komponenty, ale také se zde vytvořil prostor pro PoE splitter. Navíc kamera zde je umístěna výše a nebude tak pořizovat fotky zespod. Grafický prvek je tu v podobě loga a symetrických prolisů, které jsou až na nepatrný rozdíl ve spodní části symetrické. Toto řešení se ve firmě uchytilo a rozhodlo se, že se bude realizovat.



Obrázek 6.5 Čtvrtý návrh konstrukce

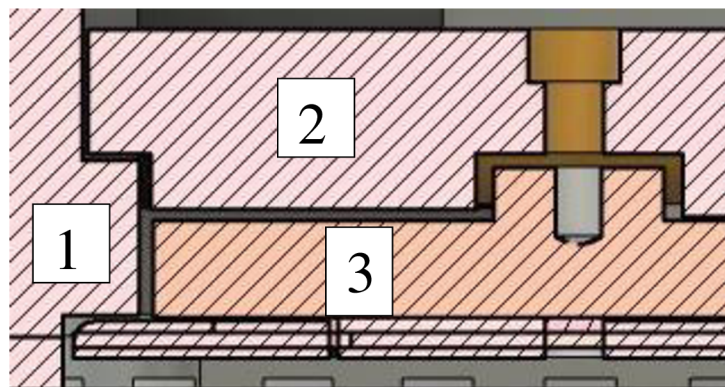
## 6.2 Realizace návrhu

Prvním krokem bylo namodelovat nebo stáhnout jednotlivé komponenty, které budou implementovány v terminálu. Následovalo umístění komponent podle čtvrtého návrhu z předchozí podkapitoly. Bylo vytvořeno tělo (obr. 6.6), do kterého byl umístěn displej (5.2.3) a RPi (5.1.3). Dále byly vytvořeny vzpěry, držící displej připevněný k tělu terminálu a poté byl vytvořen prostor pro RFID anténu.



Obrázek 6.6 Tělo, displej, vzpěry

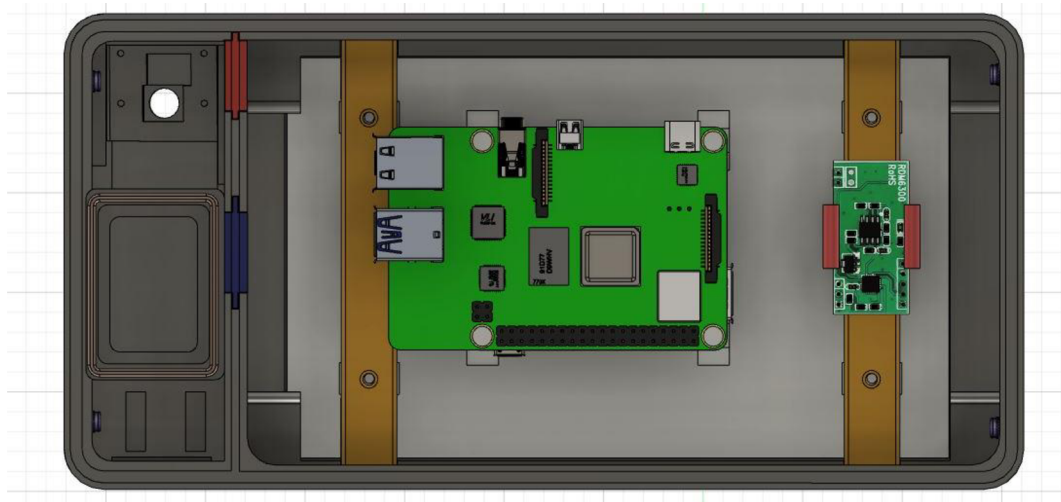
Na následujícím obrázku 6.7 je řez mezi tělem (1), vzpěrou (2) a displejem (3). Dohromady to drží šroub M3.0, který není vyobrazen na řezu. Mezery mezi jednotlivými částmi jsou max 0.15mm, a to z důvodu nepřesnosti 3D tisku jednotlivých částí.



Obrázek 6.7 Řez – tělo, vzpěra, displej

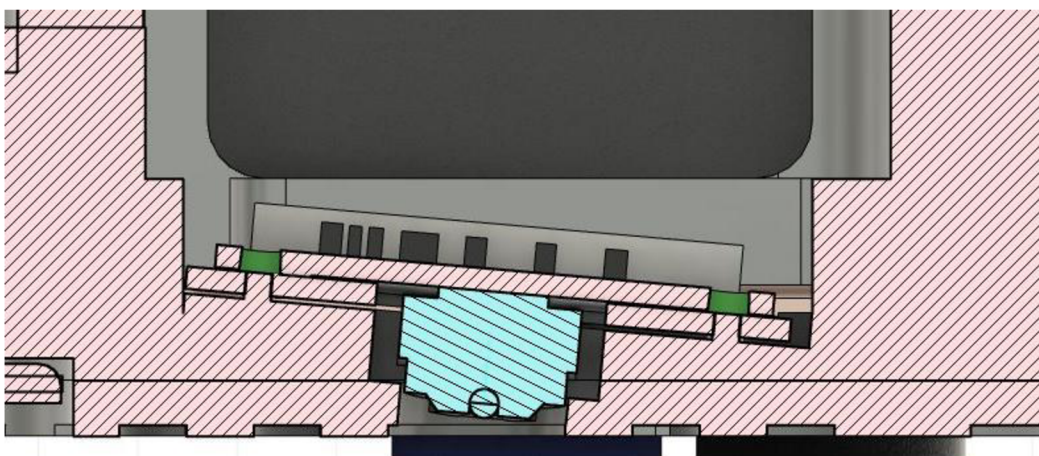


Následovalo napojení RPi na řídicí desku displeje, přidání RDM6300 (RFID čtečka) včetně držáku, vytvoření prostoru pro modul kamery a vytvoření průchodek, mezi jednotlivými komorami (obr. 6.8).



Obrázek 6.8 Přidání RDM6300, prostoru pro kameru a průchodek

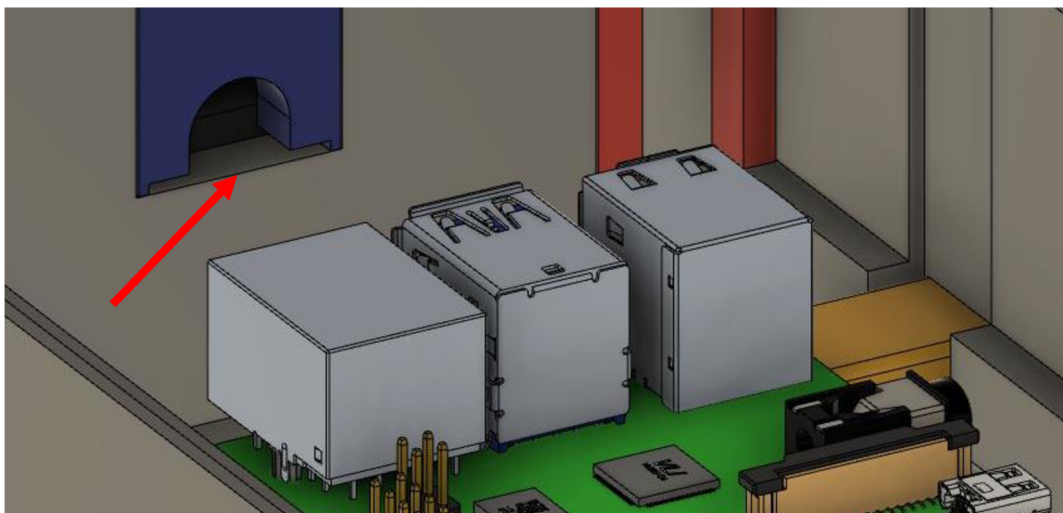
Poté byla přidána kamera (5.4). Z obrázku 6.9 je patrné natočení kamery o  $5^\circ$ , aby se obličej zaměstnance nacházel ve středu pořízené fotografie. Nad kamerou je umístěn PoE splitter.



Obrázek 6.9 Řez – kamera, PoE

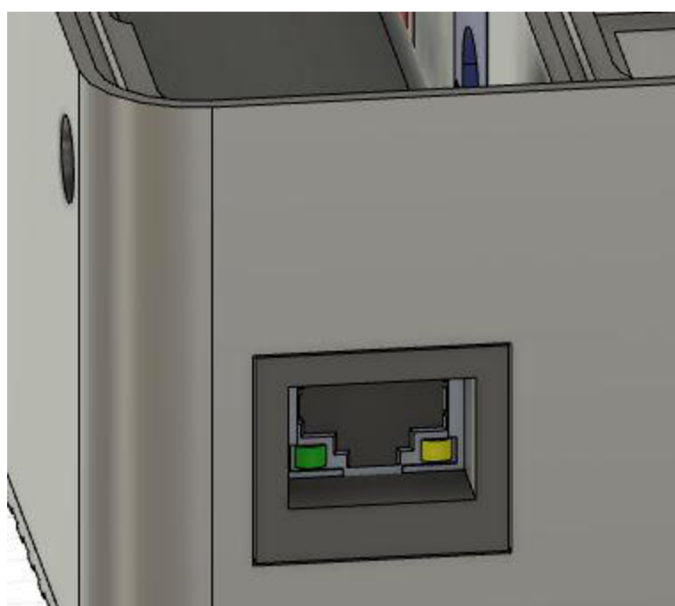
Kamera je k RPi připojena je pomocí CSI kabelu, který projde skrz vytvořenou průchodku (znázorněna modrou barvou) mezi jednotlivými komorami. Na obrázku 6.10 na prostor pro CSI kabel ukazuje červená šipka.





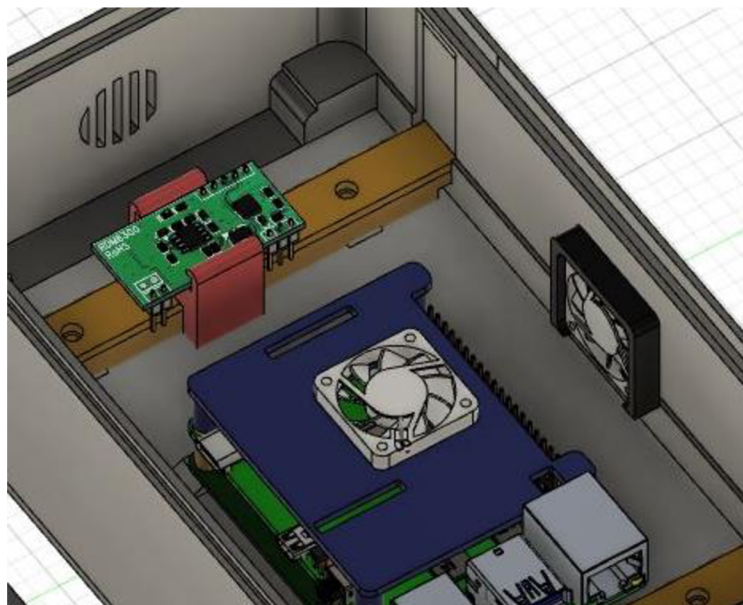
Obrázek 6.10 Detail – průchodka kamera

Následně byl vytvořen otvor pro ethernet port PoE splitteru a záslepka pro zmenšení mezery.



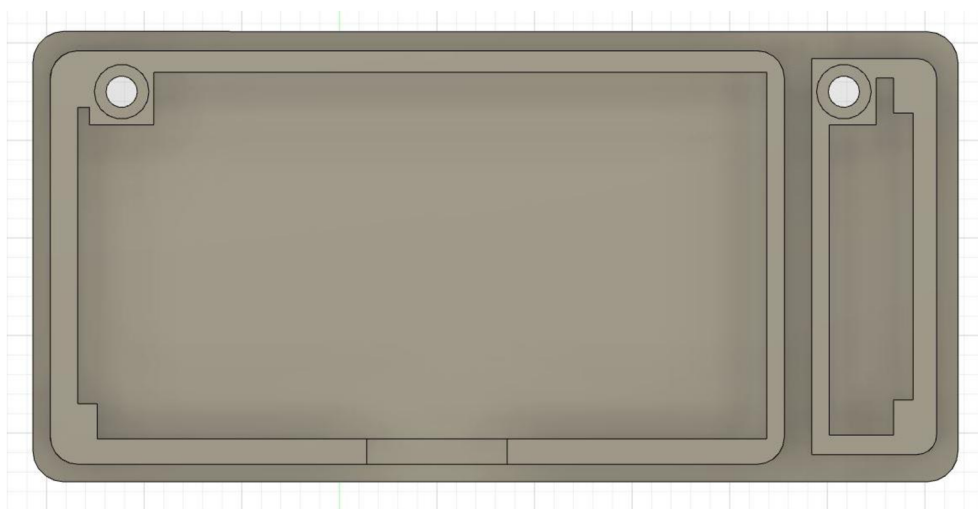
Obrázek 6.11 Detail – výstup Ethernet

V první testovací verzi terminálu jsou navíc přidány ventilátory 25x25 mm, kde jeden slouží k ochlazení procesoru RPi a druhý slouží pro cirkulaci vzduchu uvnitř terminálu (obr. 6.12). Pro lepší cirkulaci jsou vytvořeny dvě mřížky na těle terminálu. Toto řešení bylo zvoleno kvůli obavám o případné přehřátí, jelikož jeden terminál bude umístěn nad topením. Pokud problém nenastane, tak ve finální verzi budou jen mřížky bez ventilátorů.

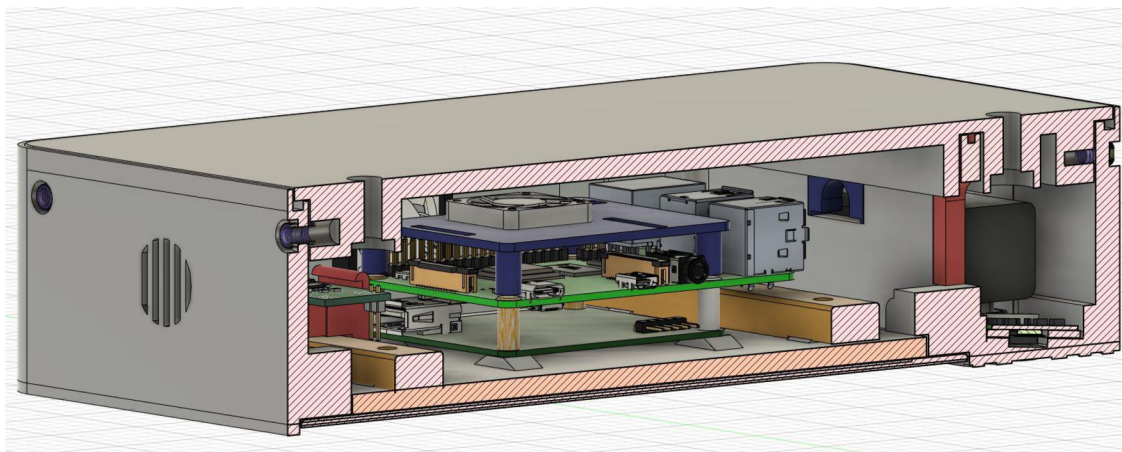


Obrázek 6.12 Detail – ventilátory

Poslední částí je zadní kryt těla terminálu (obr. 6.13). Pro nejjednodušší instalaci terminálu u zákazníka, byla navržena tato varianta. Jako první se přidělá zadní strana na zeď pomocí 2 vrtů a 2 hmoždinek. Následně se nasadí na zadní stranu osazené tělo terminálu, kde po bočních stranách jsou zataveny závitové vložky M4.0 (dvě na každé straně). Díky těmto vložkám a šroubům stejného průměru je možné tyto dvě části spojit (obr. 6.14).



Obrázek 6.13 Zadní kryt



Obrázek 6.14 Řez – tělo, zadní kryt

Na obrázku 6.15 je vyobrazena finální podoba konstrukce terminálu. 3D model je k dispozici na odkaze uvedeném ve zdroji [39].

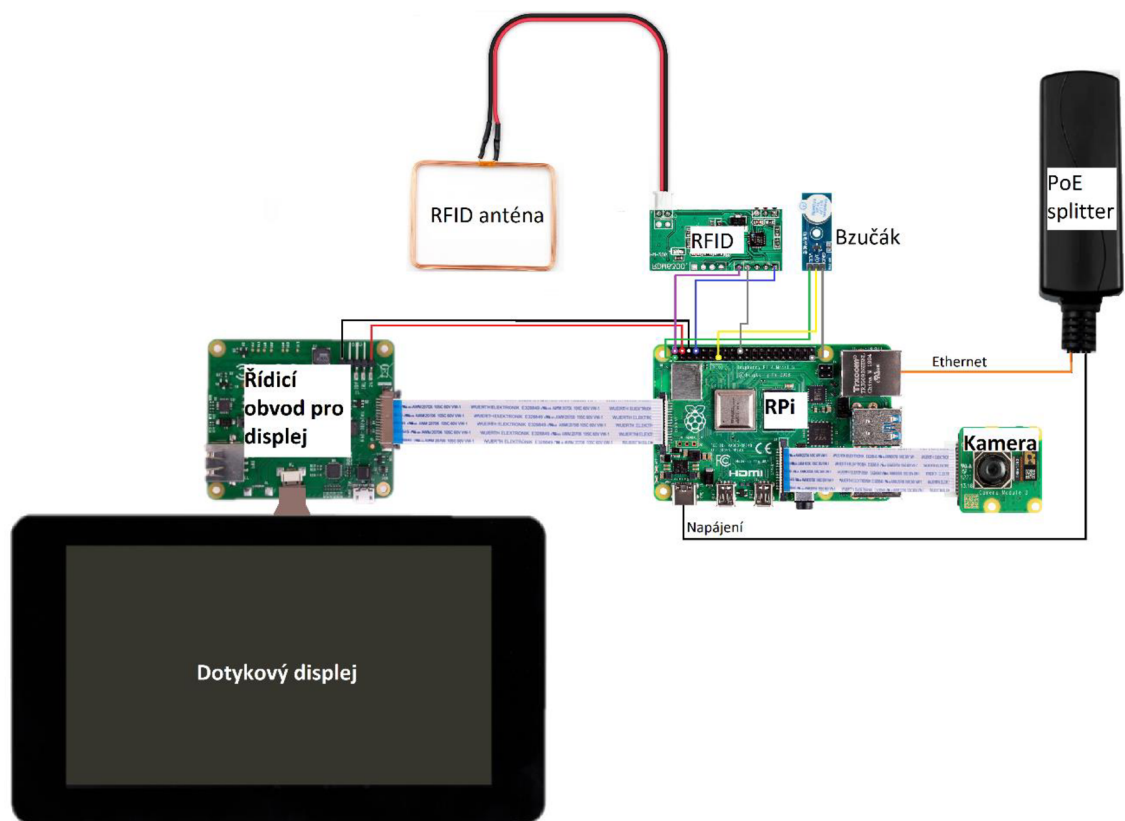


Obrázek 6.15 Výsledné konstrukční uspořádání

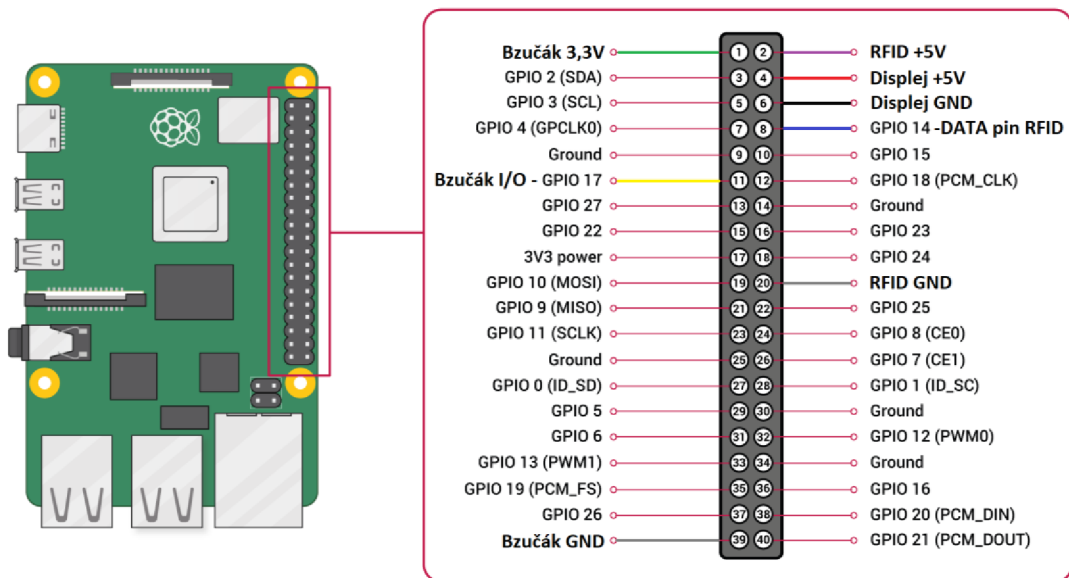
### 6.3 Propojení jednotlivých komponent

K Raspberry Pi 4 Model B je celkem připojeno 5 součástí, které jsou vyjmenovány v kapitole 5. Na obrázku 6.16 a v příloze C je znázorněno reálné zapojení jednotlivých komponent k RPi. Přes DSI port je připojen řídicí obvod pro dotykový displej (5.2.1) a pomocí vodičů 5V, GND je tento řídicí obvod napájen. Přes CSI port je připojena kamera (5.4), dále je do RPi připojen PoE splitter (0) pomocí USB-C kabelu, který slouží pro napájení RPi a pomocí RJ-45 (Ethernet) kabelu, sloužící pro připojení k síti. Další připojenou komponentou je bzučák neboli zvuková signalizace (5.5), ten je s RPi spojen pomocí vodičů 3,3V, I/O a GND, kde pin I/O je zapojen do GPIO 17 na RPi. Poslední komponentou je RFID čtečka (5.3.2), která je k RPi připojena pomocí vodičů 5V, GND a DATA. Pin DATA je připojen na GPIO 14 na RPi. Ke čtečce je připojena externí anténa pomocí dvou vodičů.

Na obrázku 6.17 je zobrazeno detailnější zapojení (pinout) komponent do RPi, pro lepší přehled. Barvy vodičů na obrázku 6.17 odpovídají barvám na zapojení na obrázku 6.16.



Obrázek 6.16 Schéma zapojení terminálu



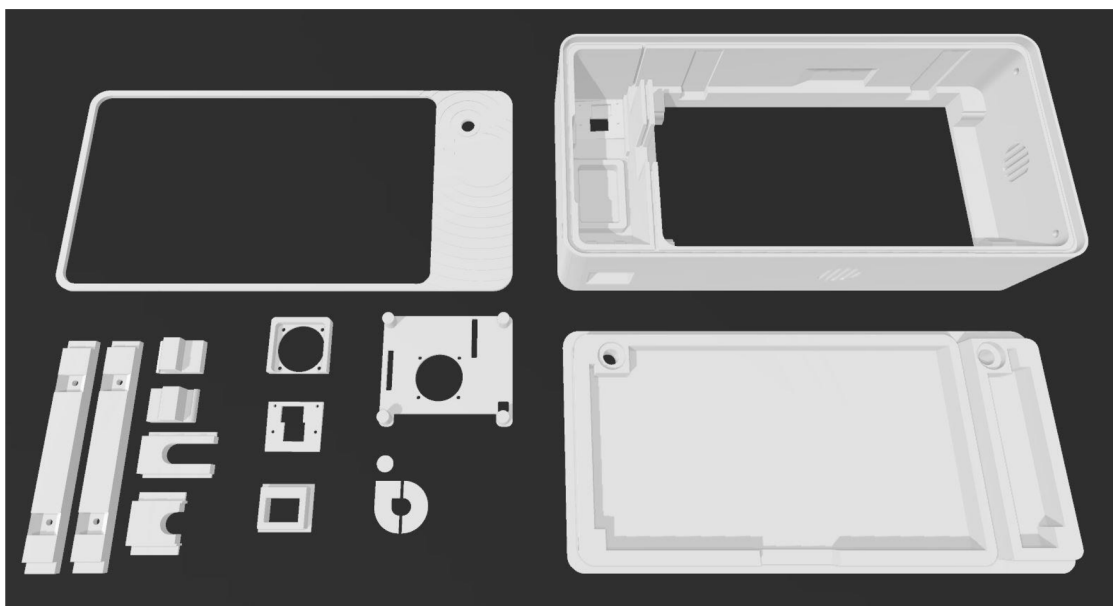
Obrázek 6.17 Detailní zapojení GPIO na RPi

## 7. REALIZACE TERMINÁLU

V této kapitole budou popsány detaily 3D tisku terminálu, jeho možná HW konfigurace, sestavení a nahrání operačního systému.

### 7.1 3D tisk

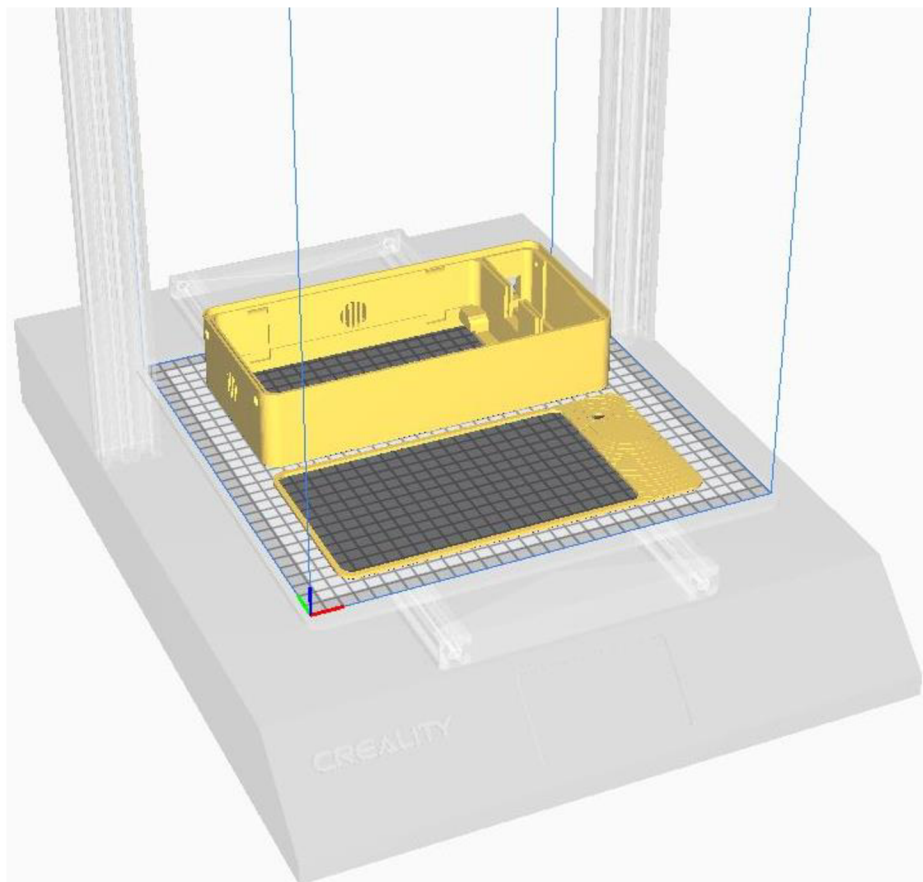
Výsledné konstrukční řešení (obr. 6.15) je zpětně rozebráno na jednotlivé části (obr. 7.1) a ty jsou připraveny pro tisk ve formátu STL. Pro lepší kvalitu výsledné vytištěné konstrukce jsou některé části rozděleny na dva díly. Například hlavní tělo terminálu je rozděleno na dvě části z důvodu správného vytištění vzoru na přední části těla. Navíc je tímto způsobem zajištěna jednoduchá personalizace výsledného terminálu pro potenciální další zákazníky, kteří by chtěli zaměnit logo Techcrowd s.r.o. za vlastní logo.



Obrázek 7.1 Jednotlivé části pro 3D tisk

Jednotlivá soubory typu STL jsou postupně vložena do programu Ultimaker Cura [40], který slouží pro nastavení parametrů tisku jako jsou kvalita tisku, procento a typ výplně, podpěry a mnoho dalšího. V programu je vybrána tiskárna Creality CR-10S Pro [41], která bude použita pro tisk. Tato tiskárna disponuje pracovní plochou pro tisk 243,3 x 247,1 x 53,7 mm. Na obrázku 7.2 je znázorněno tělo terminálu připravené pro tisk na zvolené tiskárně.





Obrázek 7.2 Příprava 3D tisku

Po zaplnění prostoru pro tisk jednotlivými částmi terminálu a zvolení parametrů tisku se provede „slicování“, které vypočítá dobu tisku a množství materiálu, který bude spotřebován, a vygeneruje G-code, který se vloží do 3D tiskárny pomocí microSD karty. Vytisknutí veškerých potřebných částí pro sestavení terminálu trvá 52 hodin.

## 7.2 Možnosti HW úprav terminálu

Jsou možné určité možnosti úprav podle specifických potřeb zákazníků. Je možné změnit jednoduše typ napájení, pokud zákazník nebude vyžadovat napájení přes PoE, tak bude terminál napájen přes adaptér s USB-C koncovkou z elektrické sítě.

Druhou možností personalizace je odstranění kamery v případě, kdy ji zákazník nebude požadovat. Otvor pro kameru v těle terminálu se zásepkou zakryje. Tudiž není nutné upravovat a znovu tisknout celé tělo terminálu a může se použít tělo terminálu, které již s použitím kamery počítají.

Dále je možné buď zcela odebrat zvukovou signalizaci anebo změnit typ zvukové signalizace.

Čtvrtou možností personalizace je možnost změny designu předního krytu těla terminálu podle představ budoucího zákazníka, například podle loga firmy zákazníka. Příklad takové personalizace je na obrázku 7.3, kde se vycházelo z loga VUT.



Obrázek 7.3 Personalizovaný přední kryt terminálu

### 7.3 Sestavení

Pro celkové sestavení terminálu z vytištěných částí (obr. 7.1), vybraných komponent (5) a spojovacích dílů byl vytvořen návod, který je přiložen v příloze D.

### 7.4 Nahrání Raspberry Pi OS

Po sestavení terminálu následuje nahrání image (Raspberry Pi OS) na RPi. Jedná se o nahrání vybraného operačního systému na microSD kartu, která se vloží do RPi. Nejprve je nutné si z oficiální webové stránky [42] stáhnout Raspberry Pi Imager, což je aplikace, pomocí které je nahrán image na microSD kartu. Následný popis a nahrání image je přiložen v příloze E.



## 8. GRAFICKÉ PROSTŘEDÍ

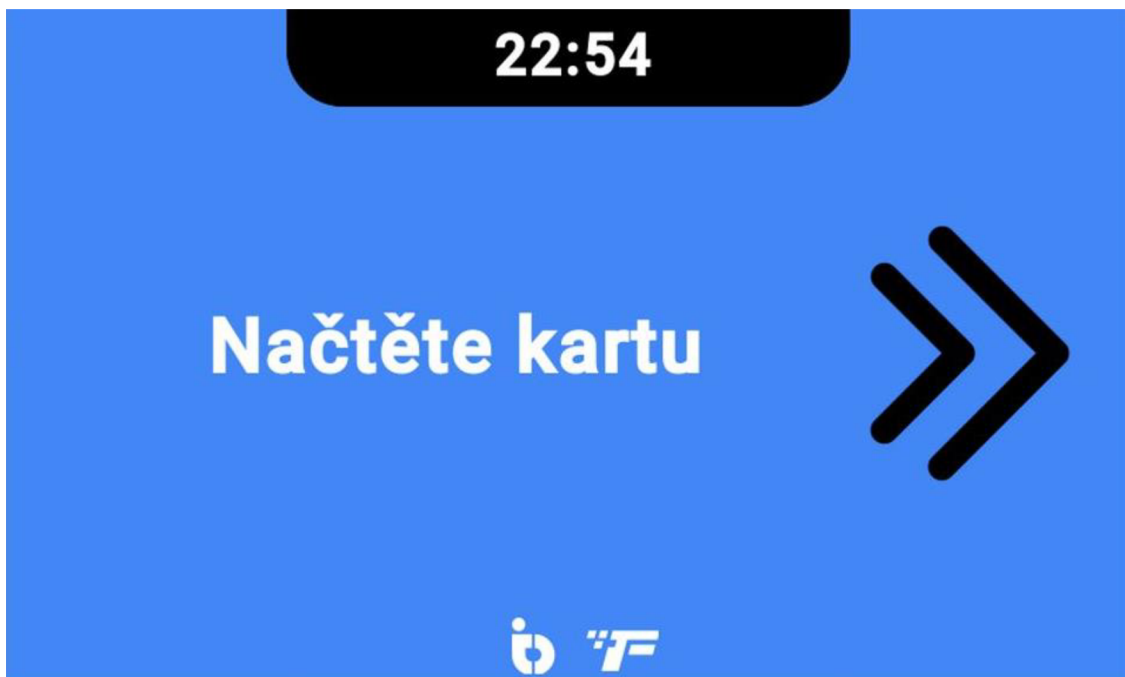
V této kapitole jsou popsán návrh grafického uživatelského prostředí neboli GUI (Graphical User Interface) pro docházkový terminál. Je zapotřebí navrhnout čtyři obrazovky o rozlišení 800x480 pixelů, kde toto rozlišení je dáno vybraným displejem v kapitole 5.2.3.

### 8.1 Návrh GUI

Celkově byly vytvořeny tři návrhy GUI, které jsou přiloženy v příloze F. V této kapitole bude podrobněji popsán realizovaný návrh GUI. Pro vývoj grafického prostředí byl použit program Figma [43]. Použití tohoto programu usnadnilo práci na vývoji GUI, kde grafické uživatelské prostředí je rozděleno do čtyřech hlavních obrazovek:

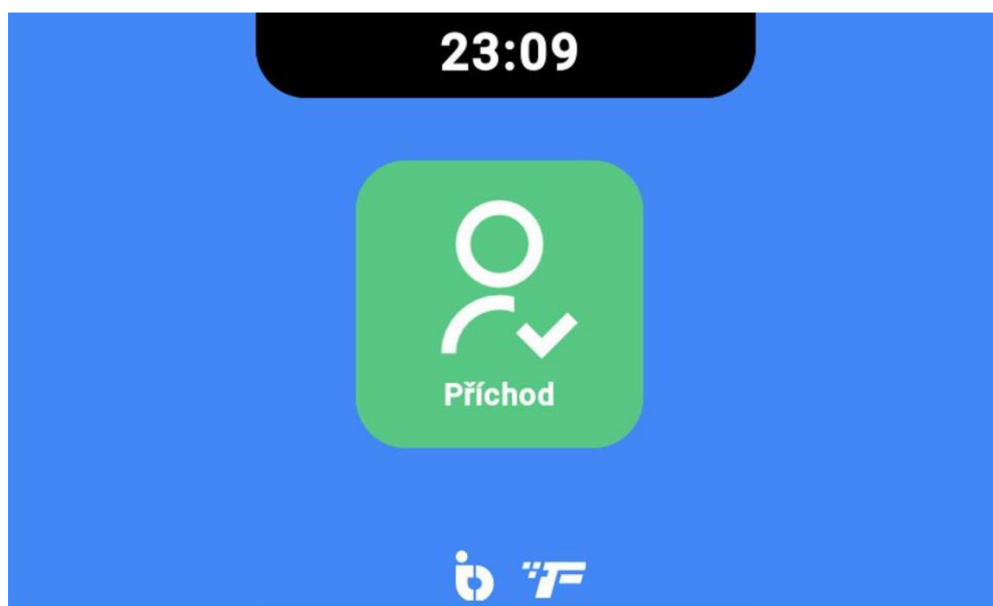
- **init\_window** – inicializační obrazovka
- **main\_window** – hlavní obrazovka s výběrem akcí
- **popup\_window** – vyskakovací obrazovka
- **status\_window** – chybová obrazovka

První obrazovka je inicializační (obr. 8.1), kde požádá uživatele o načtení karty ke čtečce terminálu. Dále je na této obrazovce zobrazen aktuální čas, šipka směřující na RFID čtečku a logo Techcrowd s.r.o.

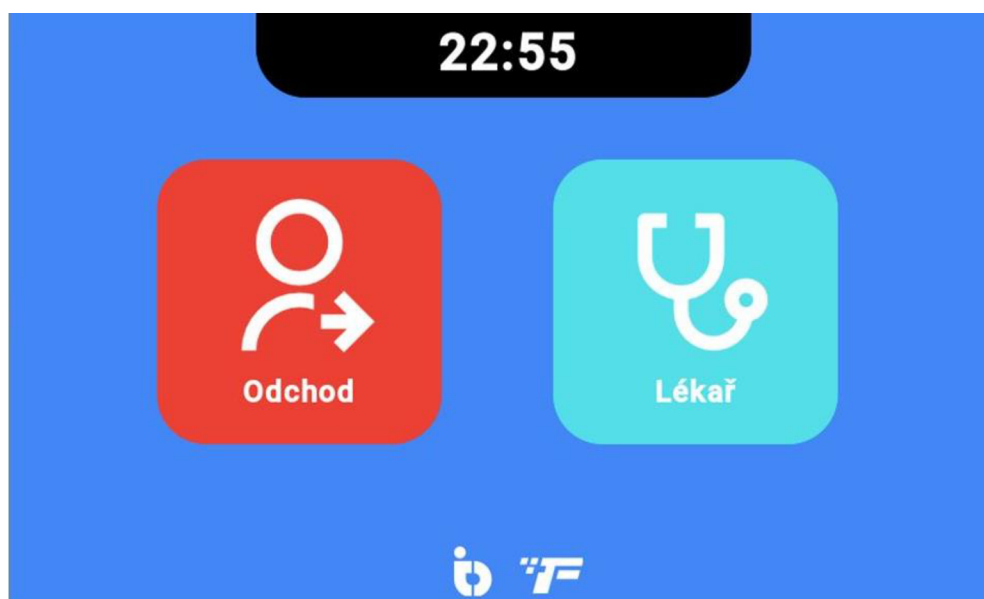


Obrázek 8.1 Inicializační obrazovka (init\_window)

Po přiložení karty je zaměstnanec ověřen a zobrazí se mu hlavní obrazovka (main\_window) s výběrem akcí, které může provést. Zde má uživatel na výběr z akce „Příchod“ (obr. 8.2) anebo z akcí „Odchod“ a „Lékař“ (obr. 8.3). Která z těchto obrazovek se zaměstnanci zobrazí závisí na jeho poslední provedené akci (4). Jednotlivé akce jsou barevně odlišeny, aby byly dobře zapamatovatelné, obzvlášť pro zaměstnance, kteří mohou mít zrakové obtíže.

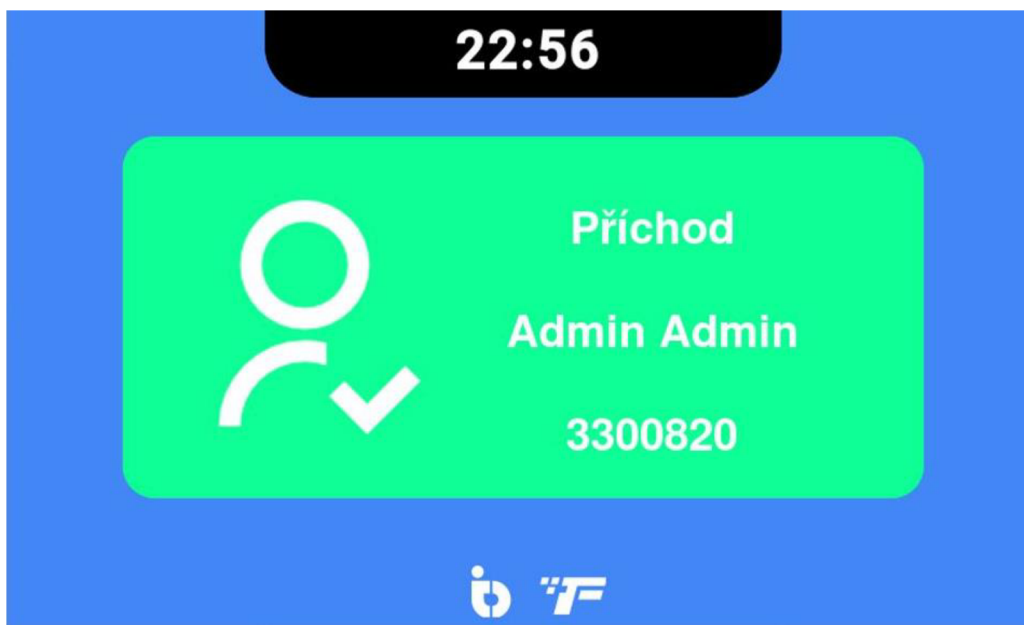


Obrázek 8.2 Hlavní obrazovka (main\_window) – příchod

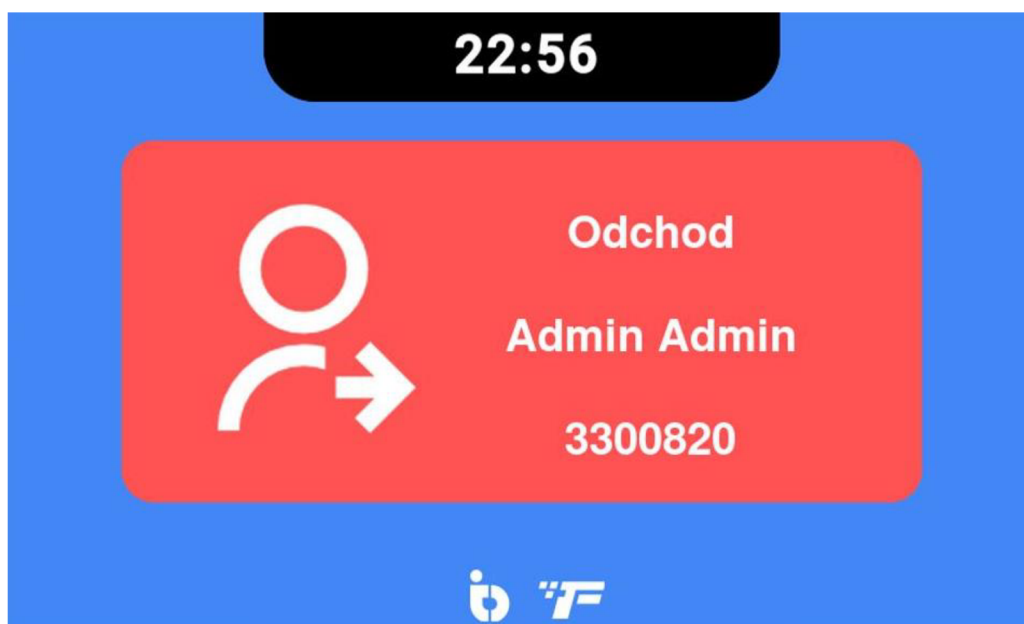


Obrázek 8.3 Hlavní obrazovka (main\_window) – odchod, lékař

Po stisknutí tlačítka akce například „příchod“ na hlavní obrazovce (obr. 8.2) se zobrazí vyskakovací obrazovka (popup\_window), která slouží pro rekapitulaci zvolené akce a pro zobrazení údajů o uživateli. Na této obrazovce (obr. 8.4) se uživateli zobrazí barevný rámeček, kde je barva souhlasná s vybranou akcí, a v tomto rámci se rovněž zobrazí typ vybrané akce, jméno uživatele a ID číslo karty. Na obrázku 8.5 je ukázka vyskakovacího okna pro akci odchod.



Obrázek 8.4 Vyskakovací obrazovka (popup\_window) – příchod



Obrázek 8.5 Vyskakovací obrazovka (popup\_window) – odchod

Poslední obrazovkou je chybová obrazovka (status\_window), která se zaměstnanci zobrazí v případě neznámého ID čísla terminálu anebo neznámé karty nebo nefunkčních serverů, kdy terminál nemá přístup k databázi uživatelů. Tato obrazovka (obr. 8.6) je červeně podbarvena pro zdůraznění chyby.

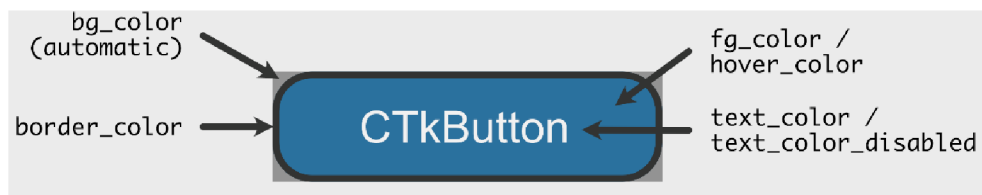


Obrázek 8.6 Chybová obrazovka (status\_window)

## 8.2 Realizace GUI

Pro vytvoření obrazovek byla využita knihovna CustomTkinter [44]. Většina widgetů jsou vytvořena jako tlačítka (CTkButton) z důvodu největší možnosti nastavení. U tlačítek knihovny CustomTkinter je možné nastavit:

- master – do které obrazovky nebo rámce tlačítko patří
- bg\_color, fg\_color, hover\_color, border\_color – viz. obr 8.7
- text, text\_color, text\_font – zobrazený text, barva a font textu
- state – stav tlačítka (disable, normal)
- width, height – šířka, výška tlačítka
- command – příkaz, který se provede po stisku tlačítka
- image – vložení obrázku
- další možnosti nastavení zde [45]

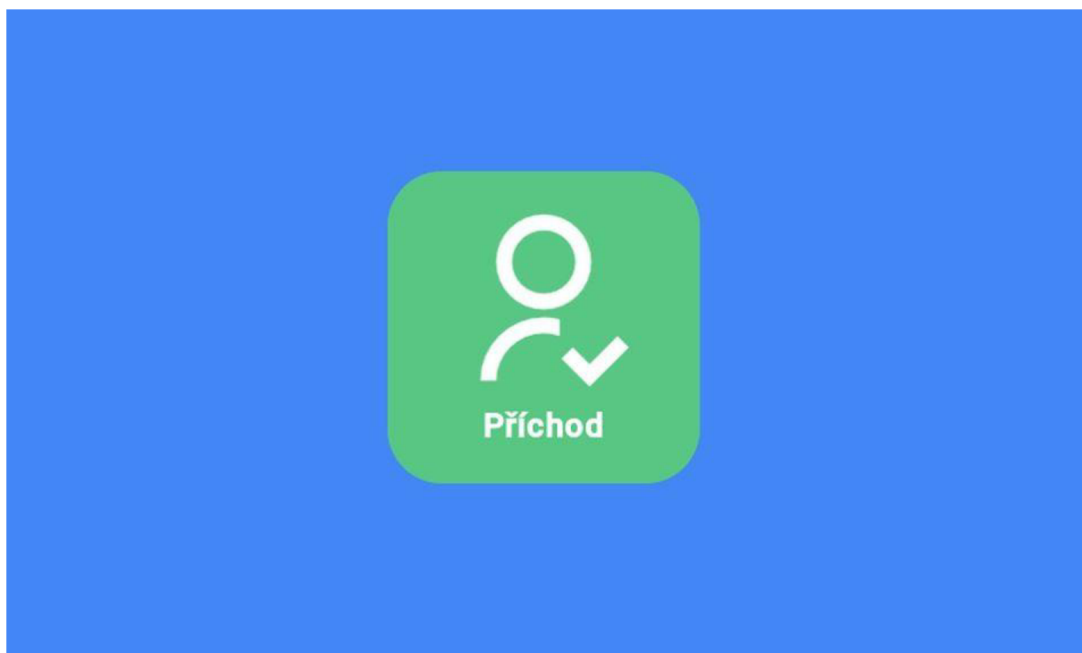


Obrázek 8.7 Nastavení barev u CtkButton [45]

Příklad nastavení tlačítka „příchod“ na hlavní obrazovce:

```
button_in=customtkinter.CTkButton(master=mainwindow,
                                   state='normal',
                                   command=mainWidorButtonOpen,
                                   fg_color="#58c784",
                                   image=prichod,
                                   hover_color="#439c66",
                                   bg_color=background_color,
                                   width=200, height=200,
                                   corner_radius=40,
                                   text="Příchod",
                                   compound="top",
                                   text_color='white',
                                   text_font=('Roboto',18,"bold"))
button_in.place(height='230', width='230', relx=0.35, rely=0.25)
```

Výsledek:



Obrázek 8.8 Příklad nastavení tlačítka příchod na hlavní obrazovce

## 9. VÝVOJ PROGRAMU

V této kapitole je nejprve popsána celková struktura programu za použití stavového automatu, následně je vytvořena tabulka přechodů pro jednotlivé stavy. Poté jsou zde popsány jednotlivé skripty představující stavy stavového automatu. Ke konci kapitoly jsou zmíněny nutné konfigurace RPi pro nasazení do provozu.

### 9.1 Programovací jazyk a vývojové prostředí

Pro realizaci programu byl zvolen programovací jazyk Python. Tento jazyk byl zvolen z důvodu používání tohoto jazyka ve firmě Techcrowd s.r.o. Navíc tento jazyk disponuje velkou uživatelskou základnou a existuje tak mnoho projektů a návodů, ze kterých je možné čerpat inspiraci.

Pro vývoj programu bylo vybráno vývojové prostředí PyCharm ve verzi 2021.2.1. Vybráno bylo z důvodu dobrých zkušeností z předchozích projektů a také z důvodu možnosti vytvoření vzdáleného vývojového prostředí.

### 9.2 Vytvoření vzdáleného vývojového prostředí pro RPi

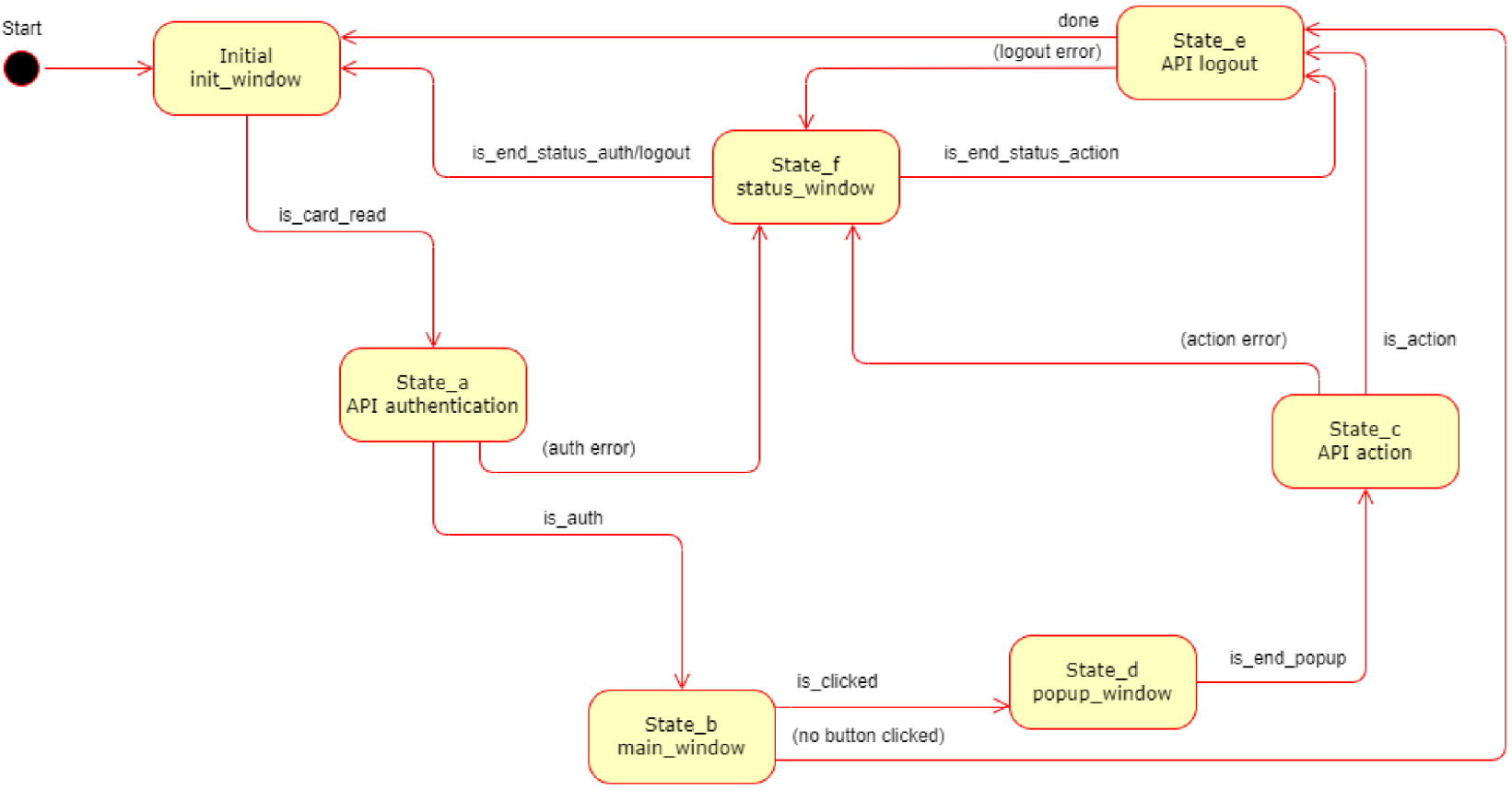
Vzdálené vývojové prostředí usnadní odladování kódu tím, že se budou kódy automaticky odesílat po provedení jakékoliv změny z vývojového prostředí PyCharm do RPi. Pro spojení vývojového prostředí na PC a RPi je nutné, aby PC a RPi byly připojeny na stejnou síť (WiFi). Dalším krokem je zjištění IP adresy RPi pomocí příkazu *ifconfig* v terminálu. Následné detailní nastavení v prostředí PyCharm je popsáno v příloze G.

### 9.3 Struktura programu

Program je rozdělen do následující částí:

- Grafické prostředí (jednotlivé obrazovky)
- API volání
- Připojené komponenty k RPi

Všechny tyto části ovládá hlavní skript `main_state_machine.py`. Jedná se o stavový automat, kde jednotlivé obrazovky a API volání představují stavy automatu. Stavového automatu zobrazuje následující diagram:



Obrázek 9.1 Diagram stavového automatu

Stavový automat má následující sedm stavů:

- **Initial** – zobrazení inicializační obrazovky
- **State\_a** – autorizace zaměstnance (POST api\_terminal/authorization)
- **State\_b** – zobrazení hlavní obrazovky
- **State\_c** – provedení akce docházky (POST api\_terminal/action)
- **State\_d** – zobrazení vyskakovací obrazovky
- **State\_e** – odhlášení zaměstnance (DELETE api\_terminal/logout)
- **State\_f** – zobrazení chybové obrazovky

Po spuštění programu se automaticky přepne do stavu Initial, kde se kontroluje, zdali nedošlo k přiložení karty ke čtečce. Pokud zaměstnanec přiloží kartu a její identifikační číslo (ID) bude nenulové, tak se přejde na stav State\_a. V tomto stavu „A“ se provede autentifikace zaměstnance, kde se odešle identifikační číslo karty a terminálu metodou POST api\_terminal/authorization. Pokud zaměstnanec nebude autorizován, server vrátí odpověď se stavovým kódem 400 a více, přejde následně do stavu „F“. State\_f slouží pro zobrazení chybového obrazovky, na které jsou zobrazeny informace o dané chybě (obr. 8.6). Po uplynutí definované doby se chybová obrazovka zavře a přejde se na počáteční stav Initial. Pokud však server vrátí odpověď se stavovým kódem 200, zaměstnanec je autorizován, a tak automat přejde do State\_b, kde zobrazí hlavní obrazovku s výběrem dostupných akcí docházky pro zaměstnance (4). Pokud si zaměstnanec nevybere z dostupných akcí v definovaném čase, je odhlášen ve stavu „E“. Pakliže si zaměstnanec vybere z nabídky dostupných akcí, stiskne tlačítko na terminálu a přejde do stavu „D“. Ve State\_d se zobrazí vyskakovací okno rekapitulující zvolenou akci a informace o zaměstnanci. Během zobrazení tohoto okna se přejde do stavu „C“, kde se vybraná akce docházky odešle na server pomocí metody POST api\_terminal/action. Zde je situace podobná jako u stavu „A“, tedy pokud akce docházky nebude zapsána, vrátí server stavový kód 400 a více, a následně přejde do stavu „F“. Následný stav bude odlišný, jelikož zaměstnanec je autorizován a je nutné ho odhlásit, proto se přejde do stavu „E“. Pokud však zápis docházky na server projde, server vrátí odpověď se stavovým kódem 200, přejde se do stavu „E“. Ve stavu „E“ dojde k odhlášení zaměstnance pomocí metody DELETE api\_terminal/logout. V případě, že server odpoví stavovým kódem 400 a více, tak se přejde do stavu „F“, kde zobrazí chybovou obrazovku a poté přejde do úvodního stavu Initial. Jestliže odhlášení zaměstnance proběhne v pořádku, server vrátí stavový kód 200, docházka je tímto úspěšně zaevidována a přejde se na úvodní stav Initial.

Pro zřehlednění funkčnosti stavového automatu byla vytvořena tabulka přechodů tab. 9.1.



Tabulka 9.1 Tabulka přechodů pro stavový automat

Počáteční stav	Cílový stav	Přechod	Podmínka přechodu
Initial	A	is_card_read	Přečtení ID karty zaměstnance
A	B	is_auth	Odpověď serveru číselným kódem 200 na POST api/terminal/authorization
A	F	(auth error)	Odpověď serveru číselným kódem 400 a více na POST api/terminal/authorization
B	D	is_clicked	Stisknutí tlačítka akce na hlavní obrazovce
B	E	(no button clicked)	Nestisknutí tlačítka akce na hlavní obrazovce v časovém intervalu
D	C	is_end_popup	Zobrazení vyskakovací obrazovky s rekapitulujícími informacemi
C	E	is_action	Odpověď serveru číselným kódem 200 na POST api/terminal/action
C	F	(action error)	Odpověď serveru číselným kódem 400 a více na POST api/terminal/action
E	Initial	done	Odpověď serveru číselným kódem 200 na DELETE api/terminal/logout
E	F	(logout error)	Odpověď serveru číselným kódem 400 a více na DELETE api/terminal/logout
F	E	is_end_status_action	Zobrazení chybové obrazovky o neprovedení zápisu akce docházky
F	Initial	is_end_status_auth	Zobrazení chybové obrazovky o neprovedení autorizace zaměstnance
F	Initial	is_end_status_logout	Zobrazení chybové obrazovky o neprovedení odhlášení zaměstnance

## 9.4 Popis programu

Celkem je program rozdělen do devíti skriptů. Všechny skripty jsou přiloženy v příloze H. Jednotlivé skripty se dělí následovně:

Obrazovky:

- init\_window.py
- main\_window.py
- popup\_window.py
- status\_window.py

Připojené komponenty:

- reader\_RFID.py
- capture\_image.py
- buzzer.py

API volání:

- API\_calls.py

Stavový automat:

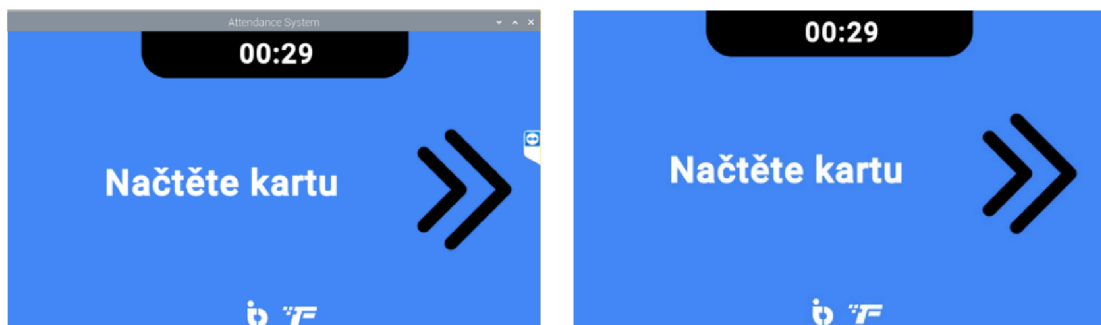
- main\_state\_machine.py (9.3).

### 9.4.1 Obrazovky

Pro vytváření obrazovek se vychází z kapitoly 8.2, tedy využívá se knihovny CustomTkinter [44]. Je vyžadováno, aby všechny obrazovky byly zobrazeny přes celý displej terminálu, aby jej zaměstnanec nemohl zavřít nebo minimalizovat. V hlavním skriptu (main\_state\_machine.py) proto nastavíme atributy CustomTkinter okna „root“:

```
root = customtkinter.CTk()
root.geometry("800x480")
root.title("Attendance System")
root.attributes('-fullscreen', True)
...
root.mainloop() # Main loop for GUI
```

Vytvoří se proměnná root jako hlavní okno programu, následně se nastaví rozměry podle zvoleného displeje (5.2.3), pojmenuje se okno, a nakonec se nastaví atribut „-fullscreen“ na hodnotu True. Tento atribut je v základu nastaven na hodnotu False. Na obrázku 9.2 je znázorněn rozdíl mezi True, False atributu „fullscreen“.



Obrázek 9.2 Atribut – fullscreen (False, True)

Tyto atributy se následně předávají při volání obrazovek z hlavního skriptu

```
init_window.init_window(root)
```

Inicializační obrazovka `init_window` (obr. 8.1) zobrazuje pouze aktuální čas ve formátu  `'%H:%M'`, text o načtení karty a obrázek šipky, která směřuje na RFID čtečku. Avšak jsou zde schovaná tlačítka, které zaměstnanec nevidí a slouží jen pro účely vzdálené správy terminálů. Na inicializační obrazovce jsou tři taková tlačítka. První je `hidden_button`, který nastaví atribut „fullscreen“ na `False` a umožní tak minimalizovat obrazovku a dostat se například k příkazovému řádku nebo do složky programu. Klávesová zkratka pro zobrazení příkazového řádku (`Ctrl+Alt+T`) nefunguje v případě nastavení atributu „fullscreen“ na hodnotu `True`. Po stisku tlačítka `hidden_button` se rovněž zobrazí nové tlačítko, ale to má již funkci zcela opačnou, tedy nastavit atribut „fullscreen“ na hodnotu `True`. Toto tlačítko se použije v případě, že chceme vrátit terminál zpět do provozního stavu.

```
hidden_button=customtkinter.CTkButton(master=root,
                                       state='normal',
                                       command=lambda:showPanel(),
                                       fg_color=background_color,
                                       bg_color=background_color,
                                       width=10, height=10,
                                       border_width=0,text="")
hidden_button.place(...)

def showPanel():
    root.attributes('-fullscreen', False)
    ...
```

Druhé je tlačítko `hidden_button_admin`, které simuluje přiložení karty ke čtečce. To slouží k ověření správné funkčnosti terminálu.

```
hidden_button_admin=customtkinter.CTkButton(master=root,
                                             state='normal',
                                             command=lambda:card_sim_insert(),
                                             ...)
hidden_button_admin.place(...)

def card_sim_insert():
    CardValue = 3300820
    ...
    if button_admin_pressed:
        return CardValue
    else:
        return 0
```

Třetí tlačítko slouží pro restart terminálu v případě neočekávané chyby programu.

```
Restart_btn=customtkinter.CTkButton(master=root,
                                     state='normal',command=lambda:restart(),
                                     ...)

def restart():
    os.system("sudo reboot")
```

Hlavní obrazovka `main_window` (obr. 8.2) zobrazuje zaměstnanci dostupné akce docházky podle odpovědi serveru (4). Z hlavního skriptu se obrazovka volá následovně:

```
main_window.mainWindow(root, auth_response[1], mainWindowTime)
```

, kde:

- **root** – nastavené atributy
- **auth\_response[1]** – dostupné akce
- **mainWindowTime** – čas pro kterou dobu má být obrazovka zobrazena

Ve skriptu `main_window.py` je na začátku vytvořena nová obrazovka „`mainwindow`“, která se zobrazí přes inicializační obrazovku, ale je pořád navázána na okno „`root`“. Po uplynutí nastavené doby `mainWindowTime` se zavolá funkce `mainwindowDestroy`, která zavře hlavní obrazovku.

```
mainwindow = Toplevel(root)
mainwindow.geometry("800x480")
mainwindow.overrideRedirect(True)
...
mainwindow.after(mainWindowTime, mainWindowDestroy)
```

Na této obrazovce je rovněž stejné skryté tlačítko jako na inicializační obrazovce, které nastaví atribut „`fullscreen`“ na `False` a umožní tak minimalizovat obrazovku.

Vyskakovací obrazovka `popup_window` (obr. 8.4) zobrazuje zaměstnanci rekapitulaci zvolené akce, jméno a příjmení zaměstnance a identifikační číslo karty zaměstnance. Z hlavního skriptu se volá následovně:

```
popup_window.popup(root, btn, CardValueReader, popWindowTime,
auth_response[2])
```

, kde:

- **root** – nastavené atributy
- **btn** – id stisknutého tlačítka na `main_window`
- **CardValueReader** – načtené číslo karty zaměstnance
- **popWindowTime** – čas pro kterou dobu má být obrazovka zobrazena
- **auth\_response[2]** – jméno a příjmení zaměstnance

Na začátku skriptu `popup_window.py` je vytvořena nová obrazovka „`pop`“ stejným způsobem jako předešlá obrazovka „`mainwindow`“. Rovněž tato obrazovka obsahuje skrytá tlačítka pro minimalizaci okna a restart terminálu.

Chybová obrazovka `status_window` (obr. 8.6) zobrazuje detail chyby, která nastala. Obsahuje rovněž skrytá tlačítka při minimalizaci okna a restart terminálu. Z hlavního skriptu se volá následovně při neúspěšné autorizaci:

```
status_window.statusCodeWindow(root, statusWindowTime,
auth_response[1], CardValueReader)
```

, kde:

- **root** – nastavené atributy
- **statusWindowTime** – čas pro kterou dobu má být obrazovka zobrazena
- **auth\_response[1]** – detailní popis chyby
- **CardValueReader** – načtené číslo karty zaměstnance

## 9.4.2 Připojené komponenty

Skript `reader_RFID.py` slouží pro čtení přiložených karet k terminálu. Byla použita python knihovna `rdm6300` [46]. Ve skriptu se využívají knihovní funkce `card_inserted` a `card_removed`, které zajistí přečtení ID karty, změni stav pomocné proměnné `CardInserted` na `True` nebo `False`, kde tato proměnná je důležitá u funkce `check_and_return_value`, která vrací nulu při nepřičtení karty anebo číslo přečtené karty do hlavního skriptu.

```
class Reader(rdm6300.BaseReader): # Read card ID
    def card_inserted(self, card):
        print(f"[{card.value}] ID přečtené karty")
        global CardValue
        global CardInserted
        CardInserted = True
        CardValue = card.value
        return CardValue

    def card_removed(self, card):
        global CardInserted
        print(f"card removed {card}")
        CardValue = card.value
        CardInserted = False
        return CardValue

# Function to return card value out of the class to main_state_machine
def check_and_return_value():
    global CardInserted
    new_value = CardValue
    if CardInserted:
        return new_value
    else:
        return 0
```

V hlavním skriptu `main_state_machine.py` je funkce `check_and_return_value` volána ve funkci `is_card_read` (funkce pro přechod ze stavu `Initial` do stavu `A` [tab. 9.1]). Funkce `is_card_read` se volá ve stavu `Initial` každých 100ms a pokud nedojde k přečtení karty (tj. funkce `check_and_return_value` vrátí hodnotu 0), tak se pokračuje ve volání funkce `is_card_read`. Při přečtení karty, stavový automat přejde do následujícího stavu (`State_a`).

```
# Checking if card is inserted and if it is read properly
def is_card_read():
    global CardValueReader
    CardValueReader = reader_RFID.check_and_return_value()
    CardValueSim = init_window.card_sim_insert() # Sim card btn

    if CardValueReader != 0 or CardValueSim != 0:
        if CardValueSim != 0:
            CardValueReader = CardValueSim
            buzzer.buzz(0.3) # Buzzer turn on
            init_window.false_btn() # Restart sim card btn
            next_state() # Transition to next state
        else:
            initial_state(1) # 1 for not show init_window again
```

```

# Initial state showing init_window
# Every 100ms checking if card is read
def initial_state(counter_init):
    global current_state

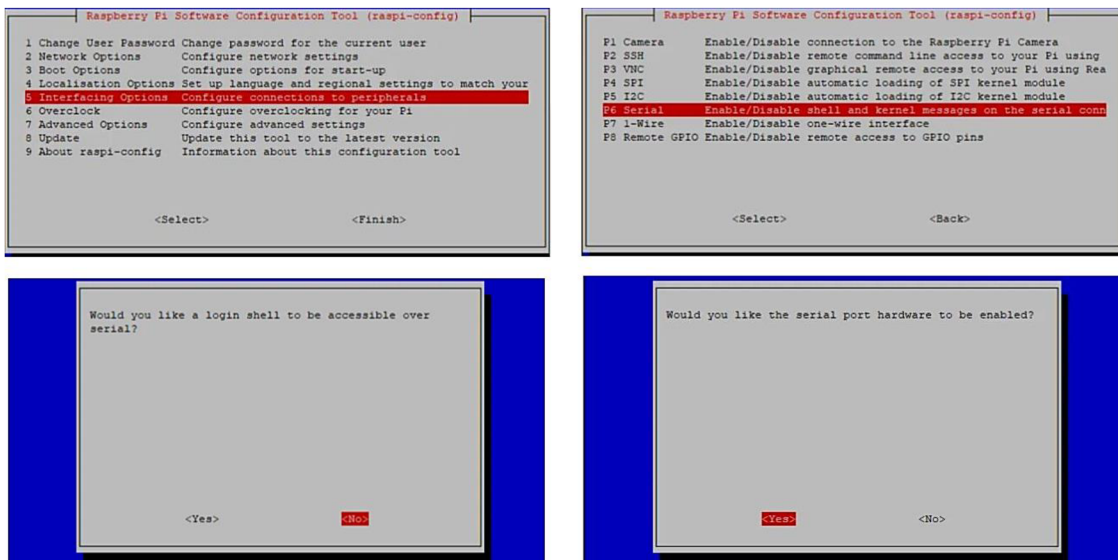
    if counter_init == 0:
        init_window.init_window(root)

# Update the current state
current_state = 'initial'

# Transition to the next state
root.after(100, is_card_read)

```

Pro správnou funkčnost RFID čtečky je nutné povolit komunikaci po sériové lince. Do terminálu (příkazového řádku) se zadá příkaz `sudo raspi-config` a zobrazí se nabídka softwarové konfigurace, zvolí se možnost číslo 5 Interfacing Options, následně se zvolí možnost P6 Serial, zablokuje se „login shell“ a povolí se „serial port hardware“ (obr. 9.3).



Obrázek 9.3 Povolení sériové komunikace na RPi

Dalším krokem pro zajištění správné funkčnosti RFID čtečky je spuštění event loopu (smyčky událostí) ve hlavním skriptu.

```

# Function to start RFID reader in main loop
def rfidreader():
    global r
    r = Reader('/dev/ttyS0') # Port
    r.start() # Start the event loop for reading RFID cards

```

Avšak hlavní skript už běží ve smyčce `root.mainloop()`, která zajišťuje grafické uživatelské rozhraní (9.4.1), tudíž není možné tyto dvě smyčky spustit zároveň, jelikož by vždy běžela jen jedna. Buďto by fungovala jen RFID čtečka bez grafického rozhraní, anebo by fungovalo pouze grafické rozhraní, bez RFID čtečky. Z tohoto důvodu byla

použita vlákna (threading) pro funkci rfidreader. Vytvoří se nové vlákno pomocí modulu „threading“, které spouští funkci „rfidreader“. Po spuštění vlákna se program zastaví na nastavenou dobu pomocí funkce „time.sleep()“. Tím se dá dostatek času pro inicializaci RFID čtečky před pokračováním. Program následně pokračuje spuštěním hlavním smyčky pro GUI.

```
# Function to start RFID reader in main loop
def rfidreader():
    global r
    r = Reader('/dev/ttyS0')
    r.start()

# Thread for RFID reader
t1 = threading.Thread(target=rfidreader)
time.sleep(1)
t1.start()

# Run the Tkinter event loop
root.mainloop()
```

Následující komponentou je kamera, pro kterou je vytvořen skript capture\_image.py. Kamera slouží k zachycování snímků zaměstnanců, kteří provedou akci na docházce. Jsou využity knihovny picamera, numpy, base64 pro zachycení, zpracování a přeformátování snímku. Ve skriptu capture\_image.py je vytvořena funkce capture\_image\_to\_base64, kde se nejprve nastaví rozlišení kamery. Následně se vytvoří pole image\_array, do kterého se uloží pořízený RGB snímek, který se poté převede do formátu base64 (4). Výsledkem vrátí funkce capture\_image\_to\_base64 řetězec představující pořízený snímek.

```
# Function for camera snapshot
def capture_image_to_base64():
    with picamera.PiCamera() as camera:
        # Configure camera resolution and capture format
        camera.resolution = (640, 480)
        camera.framerate = 30

        # Array to store the image
        image_array = np.empty((camera.resolution[1],
                                camera.resolution[0], 3), dtype=np.uint8)

        # Capture the image as an RGB array
        camera.capture(image_array, format='rgb')

        # Create a PIL Image from the RGB array
        pil_image = Image.fromarray(image_array)

        # Convert the image to base64 format
        buffered = io.BytesIO()
        pil_image.save(buffered, format="JPEG")
        encoded_bytes = base64.b64encode(buffered.getvalue())
        encoded_string = "data:image/jpeg;base64," +
            encoded_bytes.decode('utf-8')

    return encoded_string
```



Tato funkce je volána ze skriptu `API_calls.py` ve funkci `action`, která obstarává odeslání vybrané akce, informací o přihlášeném zaměstnanci a pořízeného snímku pomocí metody `POST api/terminal/action` (4). To znamená, že se snímek pořídí jen tehdy, kdy zaměstnanec provede akci docházky (příchod, odchod, lékař).

Poslední připojenou komponentou je zvuková signalizace neboli bzučák, pro který je vytvořen skript `buzzer.py`. Využívá se knihoven `RPi.GPIO` a `time` pro obsluhu bzučáku. Ve skriptu `buzzer.py` je vytvořena funkce `buzz(duration)`, kde je nejprve nastaven GPIO mód a pin, na který je bzučák připojen. Poté nastaví pin jako výstup a je nastaven na hodnotu `HIGH` a `LOW`, kde mezi těmito stavy je nastaveno zpoždění.

```
# Function to make the buzzer buzz
def buzz(duration):
    # Set the GPIO mode and buzzer pin number
    GPIO.setmode(GPIO.BCM)
    buzzer_pin = 27

    # Setup the buzzer pin as an output pin
    GPIO.setup(buzzer_pin, GPIO.OUT)
    GPIO.output(buzzer_pin, GPIO.HIGH)
    time.sleep(duration)
    GPIO.output(buzzer_pin, GPIO.LOW)

    GPIO.cleanup()
```

Tato funkce je volána z hlavního skriptu (`main_state_machine.py`) a hned několikrát. Poprvé je zavolána při přečtení karty (funkce `is_card_read`), poté při provedení akce (funkce `is_clicked`) a potřetí v případě zobrazení chybové obrazovky (`status_window`).

### 9.4.3 API volání

Pro API byl vytvořen skript `API_calls.py`, který využívá knihovnu `requests`. V tomto skriptu jsou vytvořeny jednotlivé funkce pro každý požadavek (4).

První funkcí je `authentication(CardValue, internalDeviceId)`, která obstarává požadavek `POST api/terminal/authorization` pro přihlášení uživatele a také zajišťuje výpis do textového souboru v případě špatných přístupů na server (9.5). Zápis špatných přístupů na server mají veškeré funkce, které zpracovávají požadavky.

Pro odeslání akce docházky, požadavek `POST api/terminal/action`, byla vytvořena funkce `action(actionId, internalDeviceId)`. Tato funkce obstarává odeslání vybrané akce docházky přihlášeným zaměstnancem společně s pořízenou fotografií zaměstnance, kde se v této funkci zavolá funkce `capture_image_to_base64` ze skriptu `capture_image.py`, která pořídí a vrátí fotografii ve formátu `base64` (9.4.2).

Pro odhlášení zaměstnance, požadavek `DELETE api/terminal/logout`, byla vytvořena funkce `logout()`. Tato funkce odhlásí přihlášeného zaměstnance.

### 9.4.4 Hlavní skript

Hlavní skript `main_state_machine.py` ovládá ostatní již zmíněné skripty pro obrazovky,



připojené komponenty a API volání (9.3). Na začátku programu se tyto skripty naimportují společně s knihovnami time, threading a customtkinter. Následně se nastaví atributy customtkinter okna „root“ (9.4.1), identifikační číslo terminálu a časy pro jednotlivé obrazovky. Poté jsou vytvořeny funkce pro přechody mezi stavy, které jsou popsány v tabulce 9.1. Pro přepínání mezi stavy je vytvořena funkce next\_state(). Jednotlivé stavy automatu mají rovněž své vlastní funkce initial\_state(), state\_a(),.. Na konci programu je zavolán stav initial, vytvořeno vlákno pro RFID čtečku a spuštěna hlavní smyčka pro obrazovky.

```
# Start the initial state
initial_state(0)

# Function to start RFID reader in main loop
def rfidreader():
    global r
    r = Reader('/dev/ttyS0')
    r.start()

# Thread for RFID reader
t1 = threading.Thread(target=rfidreader)
time.sleep(1)
t1.start()

# Run the Tkinter event loop
root.mainloop()
```

Mějme případ, kdy je zaměstnanec přijde k terminálu a přiloží kartu. Stavový automat je před přiložením karty k RFID čtečce v následujícím stavu Initial. Kde se zobrazí inicializační obrazovka (init\_window). Nastaví se proměnná current\_state na „initial“ a každých 100 ms je volána funkce is\_card\_read.

```
# Initial state showing init_window
# Every 100ms checking if card is read
def initial_state(counter_init):
    global current_state

    if counter_init == 0:
        init_window.init_window(root)

    # Update the current state
    current_state = 'initial'

    # Transition to the next state
    root.after(100, is_card_read)
```

Funkce is\_card\_read kontroluje, zdali nedošlo k přiložení reálné nebo simulované karty (9.4.2). Pokud karta nebyla přiložena, vrátí se zpět do stavu initial (initial\_state), kde do argumentu je dána 1, aby se zabránilo opětovnému zobrazení inicializační obrazovky. Pokud však bude karta přiložena, tzn. hodnota CardValueReader ze skriptu reader\_RFID.py nebude nulová, tak se spustí zvuková signalizace ze skriptu buzzer.py na 0.3 vteřiny. Následně se zavolá funkce next\_state().

```

# Checking if card is inserted and if it is read properly
def is_card_read():
    global CardValueReader
    CardValueReader = reader_RFID.check_and_return_value()
    CardValueSim = init_window.card_sim_insert()# Sim card btn

    if CardValueReader != 0 or CardValueSim !=0:
        if CardValueSim !=0:
            CardValueReader = CardValueSim
            buzzer.buzz(0.3) # Buzzer turn on
            init_window.false_btn()# Restart sim card btn
            next_state()# Transition to next state
        else:
            initial_state(1) # 1 for not show init_window again

```

Proměnná `current_state` byla nastavena na hodnotu „initial“ ve stavu initial, tedy přejde se na stav A (`state_a`), podle následujícího kódu.

```

# Switching states by current state variable
def next_state():
    global current_state
    if current_state == 'initial':
        state_a()
    elif current_state == 'A':
        state_b(0)
    elif current_state == 'B':
        state_d(0)
    elif current_state == 'C':
        state_e()
    elif current_state == 'D':
        state_c()
    elif current_state == 'E':
        state_f(0, "")

```

Ve stavu A se zavolá funkce `authentication()` ze skriptu `API_calls.py` (9.5). Do parametru funkce se vložila přečtená hodnota přiložené karty (`CardValueReader`) a identifikační číslo terminálu (`internalDeviceId`). Vrácená hodnota z funkce `authentication()` se uloží do proměnné `auth_response`. Následně se změní hodnota proměnné `current_state` na hodnotu „A“ a každých 100 ms je volána funkce `is_auth`.

```

# Function to authenticate user by read CardValue and InternalDevideID
def state_a():
    global current_state
    global auth_response

    auth_response = API_calls.authentication(CardValueReader,
                                             internalDeviceId)
    print("auth response: ", auth_response[0])

    # Update the current state
    current_state = 'A'

    # Transition to the next state
    root.after(100, is_auth)

```

Funkce `is_auth()` kontroluje stav přihlášení zaměstnance pomocí proměnné `auth_response` do které byla uložena vrácená hodnota z funkce `authentication()`. Pokud byla uložena první hodnota (`auth_response[0]`) představující stav přihlášení zaměstnance rovna 1, zaměstnanec byl přihlášen a stavový automat přejde do stavu B pomocí funkce `next_state()`. Následně by se ve stavu B zobrazila hlavní obrazovka (obr. 8.2, 8.3) a automat by pokračoval do dalšího stavu podle diagramu (obr. 9.1) na základě provedení či neprovedení akce docházky na terminálu. Pokud však hodnota byla rovna 2, zaměstnanec není přihlášen a stavový automat přejde do stavu F, kde se zobrazí chybová obrazovka (obr. 8.6) společně s textem dané chyby. To samé se stane v případě hodnoty rovné -1, kde tento stav představuje výjimku (exception) přihlášení.

```
# Transition function between stateA and stateB
# Checking if user is authenticated
def is_auth():
    if auth_response[0]==2:
        state_f(0,"auth")

    elif auth_response[0]==-1:
        state_f(0,"exception")

    elif auth_response[0]==1:
        next_state()
```

Zmíněná funkce `authentication()` ze skriptu `API_calls.py` je zobrazena v kapitole 9.5, kde je použita pro znázornění ověření komunikace se serverem. Tato funkce však není zobrazena kompletní, jelikož by zabrala několik stránek. Je zobrazeno pouze to hlavní pro pochopení funkčnosti. Celý kód a všechny ostatní skripty jsou přiloženy v příloze H.

## 9.5 Ověření komunikace se serverem

Pro zpětné ověření komunikace se serverem jsou veškeré chybové přístupy na server zapsány do textového souboru uloženém v paměti RPi. Toto zálohování do textového souboru bylo zvoleno z důvodu, že program je spuštěn přes příkazový řádek, kde v případě spadnutí programu a nutnosti následného restartu terminálu nebo výpadku elektrické sítě se veškerý výpis do konzole ztratí (včetně chybových přístupů na server), které by pomohli kolegům k rychlejšímu nasazení oprav.

Ukázka zjednodušeného kódu funkce `authentication()` pro autorizaci uživatele (`api/terminal/authorization`) pomocí přiložené karty včetně zápisu špatných přístupů na server do souboru `restart_log.txt`:

```

def authentication(CardValue,internalDeviceId):
    if CardValue !=0 or (CardValue !=0 and CardAgain):
        CardValueString = str(CardValue)
        payload={'internalDeviceId':internalDeviceId,
                'cardNumber':CardValueString}
        authorization = requests.post(
            "https://api-becvary.techcrowd.space/
            api/terminal/authorization",
            data=payload)

    try:
        if authorization.status_code == 200:
            # User info
            # Card info
            # Get active action
            ...
            statusCodeText.set("")
            state = 1
        else:
            error_message = f"Authentication Error
            {authorization.status_code} for card number
            {CardValueString} at {current_time}: "
            if authorization.status_code == 403:
                error_message += "Forbidden 403 - access to the
                requested resource is forbidden"
                statusCodeText.set("Zařizení není rozpoznáno " +
                str(authorization.status_code))
                state = 2

            elif authorization.status_code == 401:
                error_message += "Unauthorized 401- authentication
                required"
                statusCodeText.set("Karta není rozpoznána " +
                str(authorization.status_code))
                state = 2

            elif authorization.status_code == 400:
                error_message += "Bad Request 400 "
                statusCodeText.set("Karta nebo zařizení nebylo
                rozpoznáno ")
                state = 2

            error_log.append(error_message)

            # Specify the path to the error log file
            log_file_path = "/home/pi/Desktop/restart_log.txt"

            with open(log_file_path, "a") as log_file:
                for error_entry in error_log:
                    log_file.write(error_entry + "\n")

        except requests.exceptions.RequestException as e:
            state = -1
            statusCodeText.set("RequestException occurred in
            authentication")

        if state == 1:
            return state, list_actions, user_name
        else:
            return state, statusCodeText

```

Výsledný špatný přístup k serveru může vypadat následovně, kde:

- „Authentication Error“ – chyba při autorizaci/akci/odhlášení zaměstnance
- „400“ – stavový kód chyby
- „for card number 15086729“ – číslo karty zaměstnance, se kterou se přihlásil
- „at YYYY-MM-DD hh:mm:ss“ – časové razítko vzniku chyby
- „Bad Request 400“ – Detailnější popis chyby

```
Authentication Error 400 for card number 15086729 at 2023-10-01
22:48:17.875504: Bad Request 400
```

## 9.6 Konfigurace RPi

Aby se program terminálu spustil po připojení do elektrické sítě je nutné zajistit automatické spuštění programu (main\_state\_machine.py). Existují různé přístupy [47], jak lze automaticky spustit skript po zapnutí. Nakonec byla použita varianta s vytvořením souboru typu .desktop, který vytvoří zástupce aplikace ve složce autostart na adrese /home/pi/.config/autostart. Tato složka slouží k automatickému spuštění aplikací po spuštění RPi. Soubor Attendance.desktop vypadá následovně:

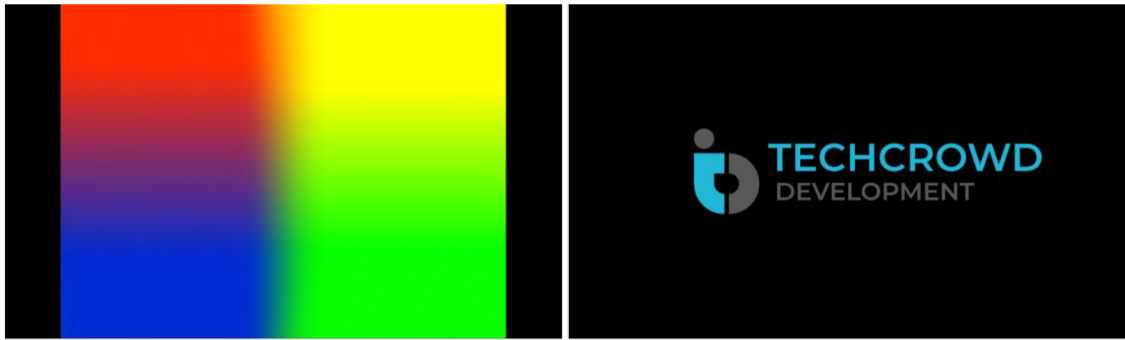
```
[Desktop Entry]
Name=Attendance
Type=Application
Exec=/usr/bin/python3
/home/pi/Desktop/attendance_terminal/main_state_machine.py
```

, kde:

- **Name:** název aplikace
- **Type:** specifikace typu položky
- **Exec:** spuštění interpretu Python3 a provedení hlavního skriptu

Dalším krokem je změna startovací obrazovky (splash screen) při spouštění RPi. Zakázala se původní obrazovka v souboru /boot/config.txt vložení řádku disable\_splash=1 na konec souboru. Následně se změnil původní obrázek splash.png uložený v /usr/share/plymouth/themes/pix/splash.png na požadované logo firmy Techcrowd. Pro překopírování byl použit následující příkaz:

```
sudo cp /path/to/my_image/logo.png
/usr/share/plymouth/themes/pix/splash.png
```



Obrázek 9.4 Startovací obrazovky

Následně se zamezí výpisu textu při startu RPi. V souboru `pix.script` umístěný v `/usr/share/plymouth/themes/pix` se zakomentují čtyři řádky kódu (obr. 9.5), které provádějí výpis. Dalším krokem bylo do souboru `/boot/cmdline.txt` dopsání na konec řádku `logo.nologo consoleblank=0 loglevel=1 quiet`.

```
GNU nano 5.4 /usr/share/plymouth/themes/pix/pix.script *
    image_y = (screen_height - image_height) / 2;
}
if (Plymouth.GetMode() != "shutdown")
{
    sprite = Sprite (resized_image);
    sprite.SetPosition (image_x, image_y, -100);
}
#message_sprite = Sprite();
#message_sprite.SetPosition(screen_width * 0.1, screen_height * 0.9, 10000);
fun message_callback (text) {
#   my_image = Image.Text(text, 1, 1, 1);
#   message_sprite.SetImage(my_image);
    sprite.SetImage (resized_image);
}
Plymouth.SetUpdateStatusFunction(message_callback);
```

Obrázek 9.5 Zamezení výpisu při startu terminálu

Pro nezobrazování kurzoru myši na displeji terminálu, kde by kurzor zůstal na místě posledního dotyku, se odkomentuje v souboru `/etc/lightdm/lightdm.conf` možnost `xserver-command=X -nocursor` (obr. 9.6).

```
GNU nano 5.4 /etc/lightdm/lightdm.conf
# autologin-guest = True to log in as guest by default
# autologin-user = User to log in with by default (overrides autologin-guest)
# autologin-user-timeout = Number of seconds to wait before loading default user
# autologin-session = Session to load for automatic login (overrides user-session)
# autologin-in-background = True if autologin session should not be immediately activated
# exit-on-failure = True if the daemon should exit if this seat fails
#
[Seat:*]
#type=local
#pam-service=lightdm
#pam-autologin-service=lightdm-autologin
#pam-greeter-service=lightdm-greeter
#xserver-backend=
xserver-command=X -nocursor
#xmir-command=Xmir
#xserver-config=
#xserver-layout=
#xserver-allow-tcp=false
#xserver-share=true
#xserver-hostname=
#xserver-display-number=
```

Obrázek 9.6 Zakázání zobrazování kurzoru

Pro zamezení vypínání displeje terminálu po určité době nečinnosti je nutné nastavit v konfigurátoru RPi (Raspberry Pi Configuration) v záložce Display možnost „Screen Blanking“ na vypnuto.

## 9.7 Zabezpečení

Zabezpečení terminálu proti cizímu vniknutí je klíčové pro ochranu citlivých dat a zachování integrity systému. Pro zajištění co nejvyšší bezpečnosti byly provedeny následující kroky na všech terminálech:

- **Změna výchozích hesel:** Provedena změna výchozího hesla („raspberry“) na unikátní hesla.
- **Omezení přístupu přes SSH:** Je zakázán přístup přes SSH (Secure Shell), jelikož tato metoda není využívána (10), představuje tak zbytečné bezpečnostní riziko.
- **Omezení WiFi:** Je zakázána možnost připojení přes tento standard.
- **Omezení Bluetooth:** Je zakázána možnost připojení přes tento protokol.
- **Zamezení přístupu mimo GUI:** Nastavením programu na celou obrazovku (9.4.1) je zamezen přístup zaměstnanci do systému terminálu.
- **HW zabezpečení:** Terminál je možno otevřít pouze za použití správných nástrojů a odšroubování 4 postranních šroubů. Nemůže tak dojít k případu, kde by zaměstnanec během pár sekund dostal dovnitř terminálu.



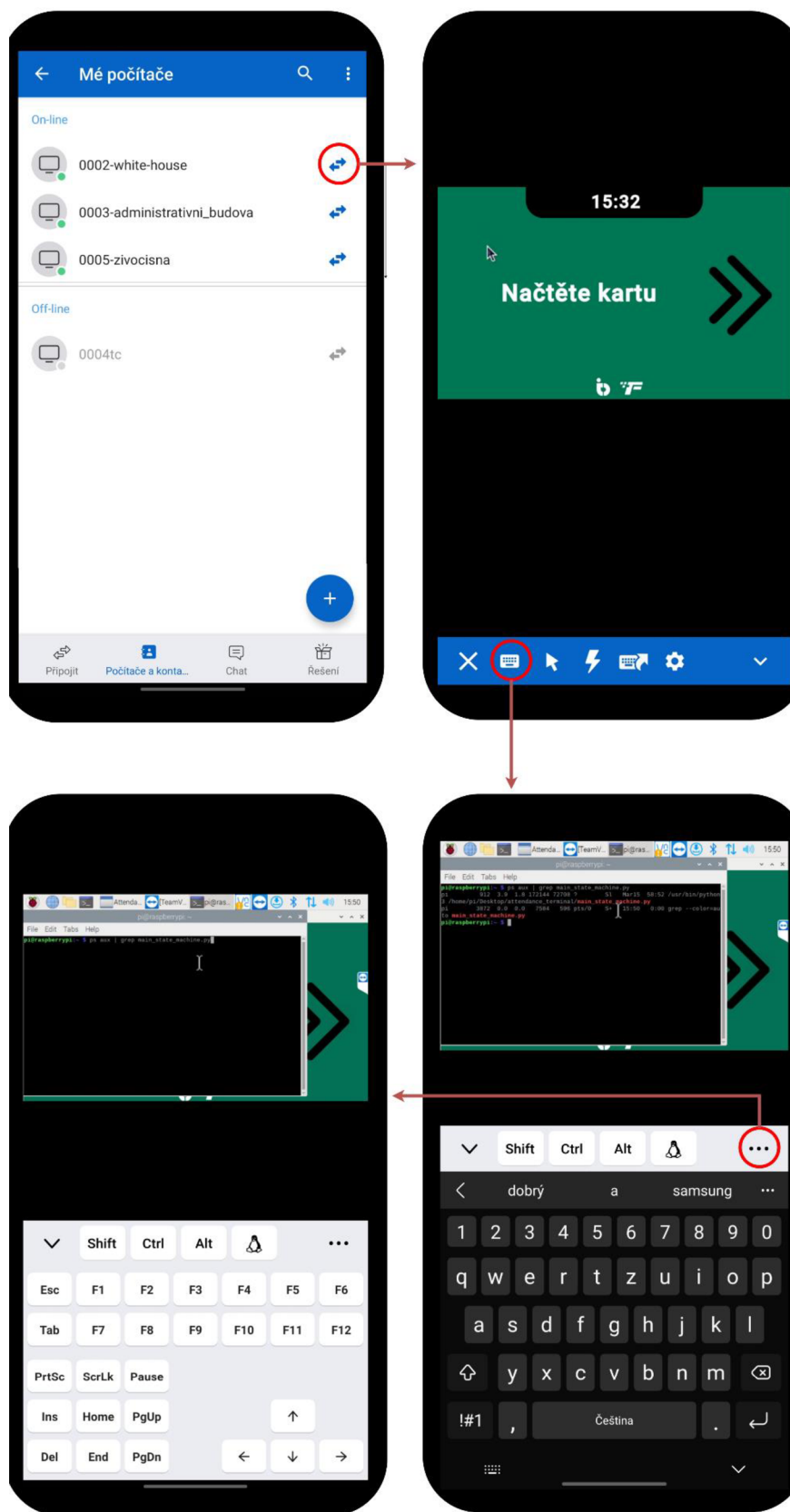
## 10. VZDÁLENÁ SPRÁVA

Vzdálená správa zařízení zajistí dohled nad nasazenými terminály v provozu, rovněž zajistí jednoduché aktualizace softwaru a tím se zamezí výjezdům k zákazníkovi, které nejsou nutné a dají se vyřešit na dálku v kratším čase. Jelikož je nutné mít správu i nad grafickým rozhraním odpadá možnost spravovat terminál pomocí běžně používaného SSH (Secure Shell), kde komunikace probíhá přes příkazový řádek.

Nástroje pro vzdálenou správu (včetně grafického prostředí) na trhu jsou RealVNC a TeamViewer. RealVNC je nainstalovaný v RPi nativně a TeamViewer se musí instalovat dodatečně. Z počátku byly používány oba programy, abychom došli ke zjištění, který nám více vyhovuje. Zůstalo se u používání TeamVieweru z důvodu přehlednější a lépe zabezpečené (biometricky) mobilní aplikace, jednoduššího přenosu dat a již používání tohoto nástroje na předchozích projektech.

Na obrázku 10.1 jsou zobrazeny obrazovky mobilní aplikace TeamViewer. Na první obrazovce vlevo nahoře je seznam aktuálně přihlášených zařízení na TeamViewer účet, včetně terminálů, které jsou popsány podle místa umístění (2). Zařízení jsou rozdělené podle toho, zdali jsou anebo nejsou aktivní. Přes vyznačené tlačítko se zahájí spojení a jeli zařízení aktivní, tak se zobrazí obrazovka terminálu (nahore vpravo). Zde je možné minimalizovat okno programu anebo prověřit fungování terminálu a serveru přes skrytá tlačítka (9.4.1). Po stisku vyznačeného tlačítka na obrazovce se zobrazí klávesnice, přes kterou je možné zadávat příkazy do terminálu (příkazového řádku). Na klávesnici jsou rovněž tlačítka Shift, Ctrl a Alt, díky nimž je možné používat klávesové zkratky, jako je například Ctrl+Alt+T pro zobrazení nového terminálu (příkazového řádku). Po stisku vyznačeného tlačítka se změní klávesnice, na které jsou další funkční klávesy.





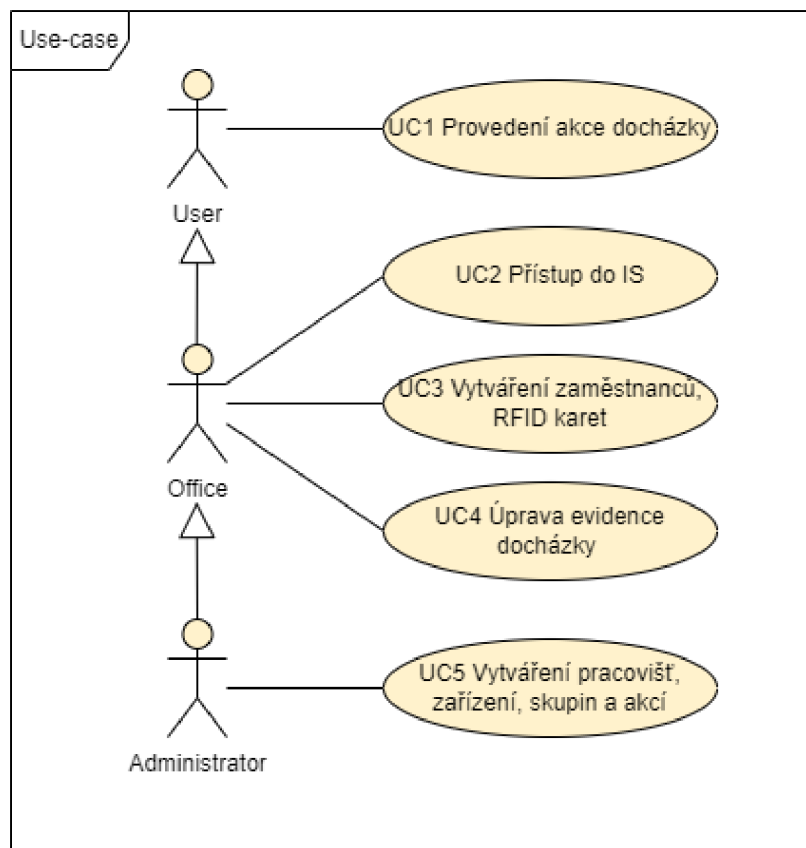
Obrázek 10.1 Obrazovky aplikace TeamViewer

# 11. INTEGRACE DO INFORMAČNÍHO SYSTÉMU

Kolegy z firmy Techcrowd s.r.o. byl vyvinut informační systém (IS) pro správu docházky. Přístup do systému je dán podle přiřazené role zaměstnance, při jeho vytváření. Role jsou následující:

- **User** (uživatel): uživatel, který provádí pouze akce docházky
- **Office** (personální oddělení): uživatel zodpovídající za správu docházky, který má přístup do IS, možnost úprav na docházce, přidávání zaměstnanců a RFID karet
- **Administrator**: uživatel s nejvyššími právy, možnost vytvářet nová pracoviště, pracovní skupiny, upravovat a přidávat zařízení a měnit akce docházky


Z výše uvedených rolí vzešel následující use-case diagram:



Obrázek 11.1 Use-case diagram pro uživatele IS

IS je rozdělen do sedmi sekcí, ke kterým má přístup administrátor. Jsou to kategorie Zaměstnanci, Zařízení, Akce, Karty, Střediska, Uživatelské skupiny a Docházka. Bude zde popsána pouze sekce Docházka, jelikož ostatní sekce nesouvisí s DP. Sekce Docházka je vyobrazena pro zaměstnance Admin za měsíc březen roku 2024 na obrázku 11.2.

Obrázek 11.2 IS – Správa docházky pro uživatele Admin



Admin Admin

---

## Správa docházky

31 záznamů

**Vybrat den**

1. 3. 2024 – 31. 3. 2024










**Vybrat zaměstnance**

Admin Admin



Export docházky

Export docházky (official)

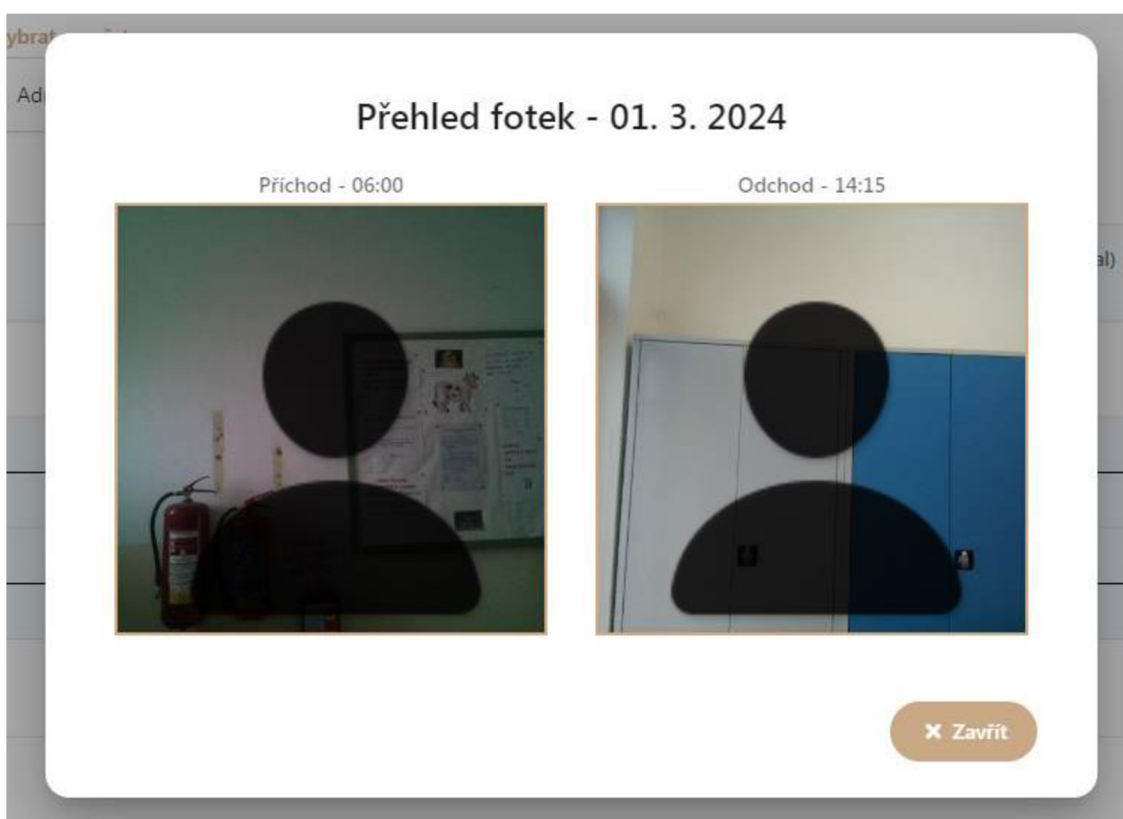
Celkový počet hodin: 42.73  
 Celkový počet hodin (official): 42.23

Den	Čas od	Čas do	Čas od (Official)	Čas do (Official)	Hodin	Hodin (Official)	Akce (Official)	
+ 01. 3. 2024	06:00	14:15	06:00	14:00	7.75	7.5	Příchod / Odchod	 
+ 02. 3. 2024 <small>Vikend</small>	Vikend							
+ 03. 3. 2024 <small>Vikend</small>	Vikend							
+ 04. 3. 2024	06:15	15:15	06:00	14:00	8	7.5	Příchod / Odchod	 
+ 05. 3. 2024	06:15	14:00	06:15	14:15	7.25	7.5	Příchod / Odchod	 
06. 3. 2024 <small>Dovolená</small>	Dovolená							
+ 07. 3. 2024	08:00	16:30	08:00	16:30	8	8	Příchod / Lékař	 

**APLIKACE ZAS**  
Vše co potřebujete vědět

-  Podpora
-  Changelog

V této sekci, se nejprve zvolí daný zaměstnanec, u kterého chceme zjistit docházku z výběru. V našem případě se jedná o vytvořeného testovacího zaměstnance a správce Admin. Automaticky je nastavena docházka na celý aktuálně probíhající měsíc, ale tuto volbu lze změnit výběrem kterýkoliv dvou dat z kalendáře. Pod výběrem data a zaměstnance je umístěna tabulka s celkovou docházkou. Každý řádek tabulky představuje jeden kalendářní den, který je specifikován v prvním sloupci. Ve druhém a třetím sloupci jsou časy od kdy do kdy byl zaměstnanec v práci. Ve čtvrtém a pátém sloupci je oficiální čas, kdy byl zaměstnanec v práci. Oficiální znamená, že zaměstnavatel má možnost upravit čas v práci podle nastavených pracovních pravidel (např. čas příchodu a odchodu ve smlouvě nebo zaokrouhlování času na půlhodiny). Šestý a sedmý sloupec zobrazuje reálný a oficiální odpracovaný čas. Osmý sloupec zobrazuje první a druhou akci docházky (např. příchod/odchod). V devátém sloupci je ikona fotoaparátu, kde po kliknutí na něj se zobrazí vyskakovací okno s pořízenými fotografiemi při provedení akce docházky (obr. 11.3). Kvůli ochraně soukromí zaměstnanců na obrázku 11.3 není zobrazena fotografie zaměstnance. Vedoucímu práce byla funkčnost pořizování fotografií předvedena. V posledním desátém sloupci je možnost smazat zápis docházky pomocí ikony koše.



Obrázek 11.3 IS – Pořízené fotografie při zvolení akce docházky

## 12. TESTOVÁNÍ A NASAZENÍ TERMINÁLU

Testování terminálu bylo rozděleno do následujících čtyř fází.



Obrázek 12.1 Časová osa testování a nasazení terminálu

První fáze spočívala v sestavení a oživení terminálu, otestování správnosti odesílaných dat a komunikace se serverem. Během této fáze byl sestaven první terminál a testování probíhalo po dobu 14 dní, kde byly upravovány detaily ohledně GUI, byli vytvořeni testovací zaměstnanci v informačním systému docházky a testovala se jejich docházka. Terminál nebyl v této fázi v nepřetržitém provozu, byl spuštěn jen během pracovní doby.

V druhé fázi byl terminál nainstalován a testován spolupracovníky z firmy Techcrowd s.r.o. v jejich kanceláři. Terminál již byl v nepřetržitém provozu po dobu jednoho měsíce, aby se odhalilo, zdali nedojde k závadě při dlouhodobém provozování. Během této doby byly sestaveny další tři terminály.

Ve třetí fázi byl zákazníkovi předán a umístěn v budově „dílna“ (2) první terminál na testování po dobu jednoho měsíce (obr. 12.2). Docházka řešena hybridní formou, tzn. zůstává v provozu dosavadní řešení docházky (píchnací hodiny) a společně s ním je do provozu nasazen docházkový terminál. Vedením ZAS Bečváry byli vybráni spolehliví zaměstnanci, kteří pracují na daném pracovišti, kde je terminál umístěn, kteří následně obdrželi RFID karty a byl jim vytvořen profil v informačním systému. Zaměstnancům bylo vysvětleno, jak se terminál obsluhuje a kam se mají obrátit v případě poruchy terminálu. Během této fáze byl zvýšený dohled na terminál přes vzdálenou správu (10) v pracovní době.



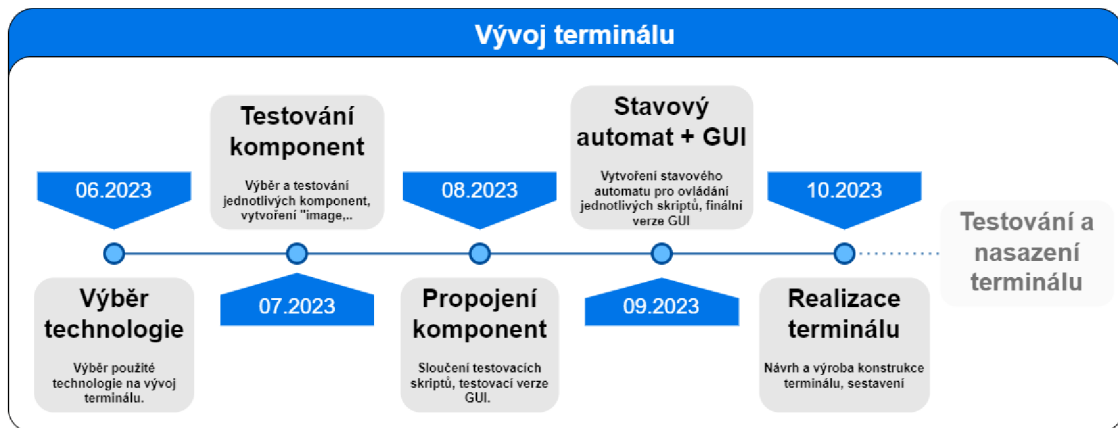
Obrázek 12.2 Nainstalovaný terminál v budově „dílna“

Ve čtvrté fázi byly zákazníkovi předány zbývající tři terminály a dva z nich byly umístěny na budovy „živočišná“ a „administrativní“. Třetí terminál má zákazník v záloze v případě rozbití terminálu nebo rozšíření systému. Docházka z počátku stále řešena hybridní formou z důvodu postupného zaregistrování zaměstnanců do docházkového systému a také z důvodu čekání na ukončení kalendářního měsíce pro zpracování docházky zaměstnanců a vypočítání mezd. Od začátku nového měsíce (dubna) byla docházka řešena pouze docházkovým systémem a byla ukončena čtvrtá fáze testování a přešlo se na poslední fázi „Správa“. V této fázi se na terminály přistupuje pouze v případě oprav a aktualizací přes vzdálenou správu.



## 13. VYHODNOCENÍ

Jednotlivé fáze při vývoji terminálu jsou znázorněny na časové ose uvedené na obrázku 13.1:



Obrázek 13.1 Časová osa vývoje terminálu

První a druhá fáze vývoje proběhla bez problémů v souladu se stanoveným časovým harmonogramem. Během třetí fáze se narazilo na problém, kde RFID čtečka a GUI musely mít vlastní smyčku událostí. Nebylo možné, aby ze smyčky obsluhující událost „přiložení karty“ byla přímo generována událost „zobrazení hlavního okna“ pro grafické uživatelské rozhraní. Nastalý problém byl dočasně vyřešen přidáním tlačítka na inicializační obrazovku, které zaměstnanec musel stisknout po přiložení karty, aby se mu zobrazila nová obrazovka s volbami akcí docházky. Definitivním řešením bylo použití více vláken, jak je popsáno v kapitole 9.4.2., což se povedlo realizovat až během čtvrté fáze vývoje.

V páté fázi vývoje došlo k problému při realizaci konstrukce terminálu, kdy stěny navržené konstrukce nebyly dostatečně pevné. Po připevnění zadního krytu (obr. 6.13) terminálu na zeď došlo k jeho prasknutí a stejná situace se stala při spojení zadního krytu a těla terminálu. Problém byl vyřešen zvětšením tloušťky těla terminálu na 5 mm, pro její větší pevnost.

První a druhá fáze testování (obr. 12.1) odhalila pár problémů, kde například došlo k nedorozumění s kolegy zajišťující IS. Problémem bylo špatně nastavené ID akce docházky, kde terminál odesílal akci „odchod“ s ID = 1, ale kolegové měli pod ID = 1 nastavenou akci „lékař“. Tento problém byl jednoduše opraven přenastavením ID akcí. Dalším problémem bylo opakované stisknutí tlačítka volby akce, které způsobilo zobrazení několika vyskakovacích obrazovek (obr. 8.4), což mělo za následek prodloužení doby potřebné pro zápis docházky. Tento problém byl vyřešen omezením počtu kliknutí na tlačítko akce, a to konkrétně na jedno kliknutí.

Po předání prvního terminálu zákazníkovi (3. fáze testování) bylo zákazníkem požadováno zvýšení doby zobrazení hlavního okna (obr. 8.2), kde si zaměstnanci stěžovali na krátký interval pro zvolení akce docházky. Vyřešeno zvýšením doby zobrazení hlavního okna z původních třech sekund na čtyři. Během testovacího provozu nedošlo k obávanému přehřátí terminálu, jelikož tento první terminál je umístěn nad topením. Nebylo tedy nutné terminály osazovat ventilátory (obr. 6.12), které tu byly pro případ přehřátí. Po předání a nasazení veškerých terminálů nedošlo k závažnějším problémům a zákazníkem nebyla vyžadována žádná úprava.

Od začátku vývoje až po finální předání zákazníkovi se nevyskytnul žádný zásadní problém, který by jakýmkoliv způsobem nedodržel stanovené termíny jednotlivých fází vývoje a testování. Zákazníkovi byl předán celkový docházkový systém v předem domluveném termínu ve formě, jakou požadoval.



## 14. NÁVRHY NA VYLEPŠENÍ SYSTÉMU

Prvním možným návrhem na změnu by mohla být detekce přítomnosti zaměstnance v záběru kamery. Stává se občas situace, kde zaměstnanec provede na terminálu akci docházky, ale je mimo záběr kamery, tudíž se pořídí fotografie bez něj. V ZAS Bečváry mají zaměstnanci umístěné pravidla používání terminálu nad každým terminálem a jedním z bodů je, že při zadávání akce docházky musejí být čelem k terminálu, aby je zachytila kamera. I přes tuto skutečnost se stane, že zaměstnanec není na pořizené fotografii vidět. Možný způsob realizace by byl za použití OpenCV knihovny [49], kde by se mohla použít funkce `cv2.CascadeClassifier.detectMultiScale()`. Tato funkce slouží pro detekci objektů na základě kaskádového klasifikátoru. Zde je uveden skript, jak by přibližně mohla detekce obličeje fungovat:

```
# Načtení kaskádového klasifikátoru pro detekci obličeje
face_cascade = cv2.CascadeClassifier(cv2.data.harcascades +
'haarcascade_frontalface_default.xml')

# Načtení obrazu
image = cv2.imread("image.jpg")
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Detekce obličeje
faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30))

# Kontrola, zda byl obličej nalezen
if len(faces) > 0:
    # Obličej byl nalezen
    # Pořízení snímku
    # Přejde se do dalšího stavu

    cv2.waitKey(0)
    cv2.destroyAllWindows()
else:
    # Obličej nebyl nalezen
    # Pokud nebude obličej nalezen do určité doby, odhlas uživatele
    print("Obličej nebyl nalezen.")
```

Zákazník tuto funkcionalitu navíc nevyžadoval a z tohoto důvodu nebyla realizována.

Druhým návrhem na změnu by byla implementace nouzového režimu. Hlavním účelem by bylo to, aby zaměstnanci mohli provést akci docházky i v případě nefunkčního internetového připojení. Jelikož zákazník nepožadoval žádnou formu zálohy (2), tento nouzový režim nebyl implementován. Pro provedení tohoto režimu by bylo nutné přidat novou obrazovku akcí, kde by byly veškeré akce docházky. Jelikož by nebylo možné kartu uživatele ověřit bez internetového připojení, terminál by nevěděl, jaké jsou dostupné akce a kterou hlavní obrazovku (obr. 8.4, 8.5) zobrazit. Následně by se musel vytvořit skript zajišťující zaznamenávání dat o přiložené kartě, provedené akci a čase provedení akce. Tyto data by se následně odeslala na server po opětovném připojení k internetu.

Třetím návrhem by bylo vytvoření skriptu pro testování internetového připojení terminálu, který by zobrazil v případě nefunkčního internetového připojení informační widget na inicializační obrazovce (obr. 8.5) s informací o nemožnosti se přihlásit na terminálu (obr. 14.1).



Obrázek 14.1 Informační widget na inicializační obrazovce

Funkce pro ověření internetového připojení a zobrazení informačního widgetu byly umístěny na konec skriptu `init_window.py` následovně:

```
# Functions for checking internet connection
def check_internet():
    try:
        # Try to ping to Google DNS server (8.8.8.8)
        subprocess.check_call(["ping", "-c", "1", "8.8.8.8"])
        return True
    except subprocess.CalledProcessError:
        return False

def check_and_display():
    notch_err = customtkinter.CTkButton(master=root,
                                        bg_color=background_color, fg_color="red",
                                        corner_radius=40, height=320, width=750,
                                        state='disable', text="", text_color='white',
                                        text_font=('Roboto', 40, "bold"))

    text_err = customtkinter.CTkButton(master=root, text="Nelze se
                                        přihlásit", state='disable', bg_color="red",
                                        fg_color="red", text_color='white',
                                        text_font=('Roboto', 40, "bold"))

    if not check_internet():
        notch_err.grid(row=0, column=0, padx=25, pady=90)
        text_err.grid(row=0, column=0, padx=25, pady=90)

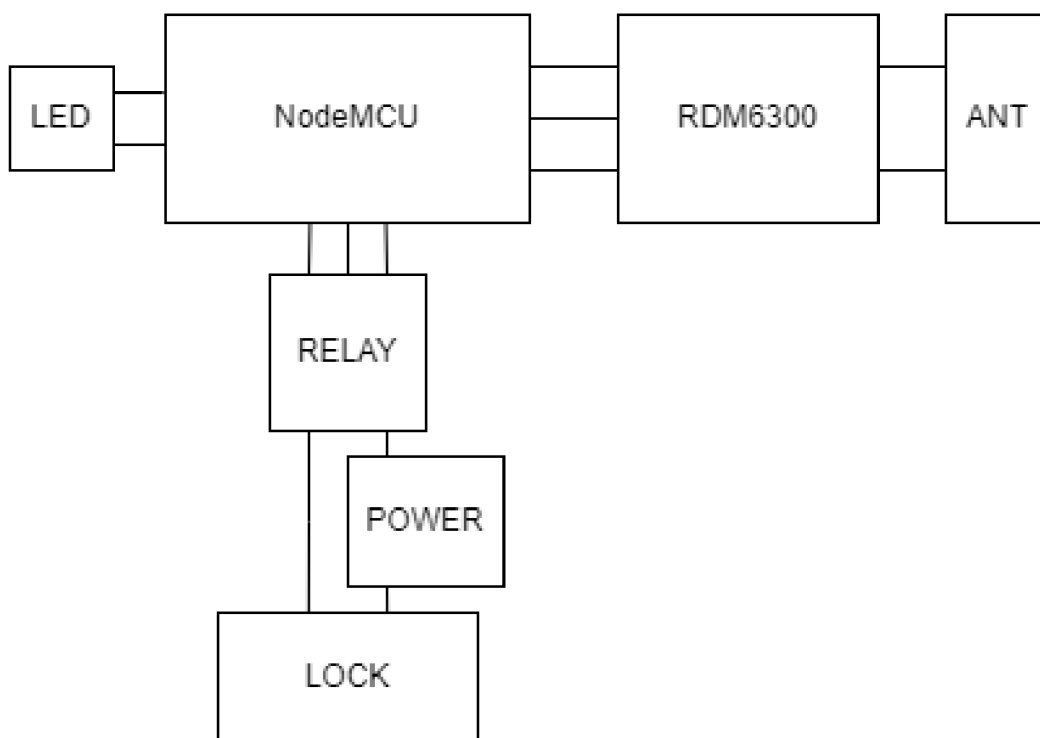
    if check_internet():
        notch_err.grid_forget()
        text_err.grid_forget()
```

Kde by se funkce `check_and_display` volala z `main_state_machine.py` a to konkrétně ze stavu `Initial`. Tento návrh je realizován, avšak ještě není na terminálech nasazen. Nasazen bude až po provedení testování při plánovaných aktualizacích terminálů.

Čtvrtým návrhem by bylo vytvoření RFID zámku, který je zmíněn v zadání v bodě číslo 2. Tyto zámky by byly umístěny do jednotlivých budov, kde jsou umístěny terminály. Blokové schéma systému pro otevírání dveří pomocí čtečky RFID karet je uvedeno na obrázku 14.2. Systém obsahuje následující komponenty:

- RFID čtečka – RDM6300 s anténou (ANT)
- Jednoduchá řídicí jednotka – NodeMCU ESP8266
- Elektrický zámek dveří (LOCK)
- Relé (RELAY)
- LED dioda
- Zdroj napájení (POWER)

Z výše zmíněných komponent vzešlo následující blokové schéma:



Obrázek 14.2 RFID zámek – blokové schéma

Otevírání dveří pomocí RFID karty by fungovalo následujícím způsobem. Zaměstnanec přiloží kartu, ID karty se odešle na server, jelikož NodeMCU disponuje WiFi modulem. Pokud server zaměstnance autorizuje, rozsvítí se LED dioda, sepne se relé a otevřou se dveře.

NodeMCU by bylo nutné připojit k internetové síti pro komunikaci se serverem. Jelikož v areálu ZAS Bečváry není k dispozici bezdrátové WiFi připojení, jako možná varianta se jeví využití RPi jako WiFi hotspotu [50], ke kterému by se NodeMCU připojilo. Vytvoření WiFi hotspotu je možné provést přes následující příkaz:

```
sudo nmcli device wifi hotspot ssid <example-network-name> password <example-password>
```

V příloze H je přiložen zdrojový kód `rfid_locker_idea.c` pro NodeMCU, který nastiňuje funkcionalitu RFID zámku.

Tento zámek měl být z počátku realizován společně s docházkovým terminálem. Avšak zákazník od tohoto řešení nakonec upustil. Z tohoto důvodu nebylo otevírání dveří pomocí RFID karet realizováno.

## 15. ZÁVĚR

Cílem této diplomové práce bylo seznámení se s problematikou docházkových terminálů a uvést parametry docházkových terminálů dostupných na trhu. Rešerše trhu byla provedena v kapitole 1, kde v tabulce 1.1 je porovnání docházkových terminálů dostupných na trhu. Všechny terminály používají podobnou strukturu, ale každý používá jiný operační systém, na kterém terminál běží. Informace uvedené v tabulce bohužel nemohou být kompletní, protože někteří výrobci je odmítli poskytnout. Byl zaslán dotaz o doplnění informací, ale nebyl žádný zodpovězen.

Ve druhé kapitole byly diskutovány požadavky zákazníka, kterým je zemědělské družstvo ZAS Bečváry a.s. Hlavním požadavkem byla identifikace zaměstnance a zajištění univerzálnosti terminálů, aby bylo možné terminál rychle nahradit v případě poruchy anebo v případě rozšíření systému. Dalším požadavkem byla kamera pro následnou identifikaci zaměstnance pomocí informačního systému.

Ve třetí kapitole byly definovány požadavky na fungování docházkového terminálu. V této kapitole byl definován způsob napájení terminálu pomocí PoE, následně byly stanoveny požadavky na dotykový displej a byl zvolen způsob identifikace uživatele pomocí RFID karet nebo čipů.

Čtvrtá kapitola se zabývá komunikací se serverem pomocí https protokolu. Byly zde popsány mechanismy komunikace terminálu se serverem (viz. příloha A) a rovněž zde byly uvedeny dotazy (POST, DELETE) posílané na server a odpovědi serveru.

V páté kapitole byl probírán výběr jednotlivých komponent. Jako řídicí jednotka byl vybrán jednodeskový počítač Raspberry Pi 4 Model B. Jako dotykový displej byl zvolen oficiální Raspberry 7“ displej. Pro evidenci zaměstnance byla vybrána RFID čtečka RDM6300. Pro následné ověření identifikace zaměstnance pomocí kamery byla vybrána Raspberry kamera V2. Dále byly vybrány komponenty pro zvukovou signalizaci a pro zajištění napájení pomocí PoE.

V šesté kapitole bylo navrženo konstrukční uspořádání terminálu, kde nejdříve bylo provedeno několik designových návrhů. Následně se vítězný návrh realizoval a lehce upravil. Výsledná podoba konstrukčního uspořádání je přiložena v příloze B. Rovněž v této kapitole bylo provedeno schéma zapojení jednotlivých vybraných komponent připojených na Raspberry Pi, které je zobrazeno v příloze C.

V sedmé kapitole byly nejprve popsány detaily 3D tisku terminálu s ohledem na možné HW konfigurace.

Osmá kapitola se zabývala návrhem grafického prostředí (GUI) pro docházkový terminál. Celkem byly vytvořeny tři návrhy, kde jednotlivé návrhy jsou popsány v příloze F. Ve všech návrzích byly vytvořeny čtyři základní typy obrazovek.

Devátá kapitola se zabývá vlastní realizací programového vybavení. Jako programovací jazyk byl vybrán Python (3.9.2) a programovací prostředí PyCharm.

Desátá kapitola se zabývá vzdálenou správou terminálů s využitím nástroje TeamViewer.

Jedenáctá kapitola popisuje integraci do informačního systému (IS), který byl vytvořen kolegy z firmy Techcrowd s.r.o. Nejprve byly popsány jednotlivé role uživatelů, které byly znázorněny v use-case digramu (obr. 11.1). Následně na obrázku 11.2 bylo zobrazení hlavní sekce IS, a to sekce Docházka. V této sekci jsou zobrazeny veškeré informace o docházce vybraného zaměstnance.

Dvanáctá kapitola je věnována testování a nasazování terminálů přímo u zákazníka. Dosažené výsledky diplomové práce jsou zhodnoceny v kapitole třináct a možná vylepšení terminálů jsou uvedena ve čtrnácté kapitole.

V rámci diplomové práce byly navrženy a vyrobeny docházkové terminály, které byly nasazeny do ostrého provozu u zákazníka ZAS Bečváry a.s. Terminály umožňují komunikaci přes rozhraní Ethernet a jsou napájeny pomocí technologie PoE. Komunikují pomocí https protokolu s nadřazeným serverem, na kterém běží docházkový systém. Docházkový systém nebyl součástí této práce a byl realizován kolegy z firmy. V rámci diplomové práce bylo navrženo konstrukční uspořádání a jednotlivé části byly vytisknuty na 3D tiskárně. Rovněž bylo vyvinuto a odladěno veškeré SW vybavení pro terminál. Po proběhnutí zkušebního provozu byl celý docházkový systém včetně terminálů předán zákazníkovi v měsíci dubnu.

## LITERATURA

- [1] *IT-WATT R3*. Online. In: [www.ikos.cz](http://www.ikos.cz). Dostupné z: <https://www.ikos.cz/produkt/56/it-watt-r3/22>. [cit. 2023-10-17].
- [2] *ZPad SYSDO BIOPAD*. Online. In: [Eshop.eurosat.cz](http://Eshop.eurosat.cz). Dostupné z: <https://eshop.eurosat.cz/product/99599/10646/ZPad%20SYSDO>. [cit. 2023-10-17].
- [3] *Qubos*. Online. In: [Cominfo-trade.com](http://Cominfo-trade.com). Dostupné z: <https://www.cominfo-trade.com/cz/identifikacni-systemy/dochazkovy-terminal-1>. [cit. 2023-10-17].
- [4] Katalog Qubos: Docházkový terminál. Online. In: . S. 1-4. Dostupné z: [https://www.cominfo-trade.com/uploads/central\\_storage\\_files/Qubos\\_brozura\\_COMINFO\\_CZ.pdf](https://www.cominfo-trade.com/uploads/central_storage_files/Qubos_brozura_COMINFO_CZ.pdf). [cit. 2023-10-17].
- [5] *GIRITON na zed' 7"*. Online. [Giriton.com](http://Giriton.com). Dostupné z: <https://giriton.com/cs/produkty/pichaci-hodiny-na-zed-7>. [cit. 2023-10-24].
- [6] *Qubos / Finger&Face: str. 4/4*. Online. In: [Cominfo-trade.com](http://Cominfo-trade.com). Dostupné z: [https://www.cominfo-trade.com/uploads/central\\_storage\\_files/Qubos\\_brozura\\_COMINFO\\_CZ.pdf](https://www.cominfo-trade.com/uploads/central_storage_files/Qubos_brozura_COMINFO_CZ.pdf). [cit. 2023-10-24].
- [7] *PoE princip*. Online. [Moxa.cz](http://Moxa.cz). Dostupné z: <https://www.moxa.cz/zpravodaj/2009/03/Vyuzijte-svou-sit-naplno-s-PoE.htm#:~:text=po%C4%8Det%20pot%C5%99ebn%C3%BDch%20za%C5%99%C3%ADzen%C3%AD,-,Jak%C3%BD%20je%20princip%20PoE,100BaseT%20pou%C5%BE%C3%ADvaj%C3%AD%20pro%20p%C5%99enos%20dat.,> [cit. 2023-10-25].
- [8] *REST*. Online. Wikipedia. Dostupné z: [https://cs.wikipedia.org/wiki/Representational\\_State\\_Transfer](https://cs.wikipedia.org/wiki/Representational_State_Transfer). [cit. 2023-12-27].
- [9] *HTTPS*. Online. [Rascasone](http://Rascasone.com). Dostupné z: <https://www.rascasone.com/cs/blog/co-je-https-http-ssl-tls>. [cit. 2024-04-17].
- [10] *API*. Online. Wikipedia. Dostupné z: <https://cs.wikipedia.org/wiki/API>. [cit. 2023-12-27].

- [11] *Raspberry Pi 4 Model B*. Online. Rpishop.cz. Dostupné z: <https://rpishop.cz/raspberry-pi-4/1598-raspberry-pi-4-model-b-4gb-ram-765756931182.html>. [cit. 2023-12-27].
- [12] *Raspberry Pi 4 Model B - 8GB*. Online. Rpishop.cz. Dostupné z: <https://rpishop.cz/raspberry-pi-4/2611-raspberry-pi-4-model-b-8gb-ram.html>. [cit. 2023-12-27].
- [13] *Raspberry Pi 4 Model B - 1GB*. Online. Rpishop.cz. Dostupné z: <https://rpishop.cz/raspberry-pi-4/1600-rpi401.html>. [cit. 2023-12-27].
- [14] *Raspberry Pi 4 Model B - 4GB*. Online. In: Rpishop.cz. 2023. Dostupné z: <https://rpishop.cz/wp-content/uploads/2019/06/6030-Raspberry-Pi-4-Model-B-4GB-RAM.jpg>. [cit. 2023-12-27].
- [15] *Radxa ROCK PI 4 C+*. Online. Conrad.cz. Dostupné z: <https://www.conrad.cz/cs/p/radxa-rs114cp-d4-rock-pi-4-c-4-gb-6-x-2-4-ghz-2627030.html?refresh=true>. [cit. 2023-12-27].
- [16] *Radxa ROCK PI 4 C+*. Online. Laskakit.cz. Dostupné z: <https://blog.laskakit.cz/porovnani-rock-pi-a-raspberry-pi/>. [cit. 2023-12-27].
- [17] *Radxa ROCK PI 4 C+*. Online. In: Conrad.com. Dostupné z: <https://asset.conrad.com/media10/isa/160267/c1/-/cs/002627030PI11/image.jpg?x=1000&y=1000&format=jpg&ex=1000&ey=1000&align=center>. [cit. 2023-12-27].
- [18] *Banana Pi M5*. Online. Alza.cz. Dostupné z: <https://www.alza.cz/banana-pi-m5-d7740173.htm>. [cit. 2023-12-27].
- [19] *Raspberry Pi 7"*. Online. Rpishop.cz. 2023. Dostupné z: <https://rpishop.cz/lcd-oled-displeje/243-7-oficialni-kapacitni-lcd-displej-800x480.html>. [cit. 2023-12-27].
- [20] *Raspberry Pi 7"*. Online. In: Alza.cz. 2023. Dostupné z: <https://www.alza.cz/raspberry-pii-touch-display-7-d4268133.htm>. [cit. 2023-12-27].
- [21] *LaskaKit 10.1"*. Online. Laskakit.cz. Dostupné z: <https://www.laskakit.cz/10-1--1024x600-tft-displej-pro-raspberry-pi--hdmi--dotykovy/>. [cit. 2023-12-27].
- [22] *LaskaKit 10.1"*. Online. In: Laskakit.cz. 2023. Dostupné z: [https://cdn.myshoptet.com/usr/www.laskakit.cz/user/shop/big/5342-3\\_10-1-h----3.jpg?634968f4](https://cdn.myshoptet.com/usr/www.laskakit.cz/user/shop/big/5342-3_10-1-h----3.jpg?634968f4). [cit. 2023-12-27].



- [23] *MF RC522*. Online. Dratek.cz. 2023. Dostupné z: <https://dratek.cz/arduino/833-rfid-ctecka-s-vestavenou-antenou.html>. [cit. 2023-12-27].
- [24] *MF RC522*. Online. In: Dratek.cz. 2023. Dostupné z: [https://dratek.cz/photos/produkty\\_gal/f/48/48416.jpg?m=1644908125](https://dratek.cz/photos/produkty_gal/f/48/48416.jpg?m=1644908125). [cit. 2023-12-27].
- [25] *RDM6300*. Online. Dratek.cz. 2023. Dostupné z: [https://dratek.cz/arduino/1297-rfid-ctecka-125k-em4100-modul-rdm6300-uart-vystup.html?gad\\_source=1&gclid=Cj0KCQiA7aSsBhCiARIsALFvovzLJ-zO3bUVHyq0y-aDbvs6tB-8UzHlupSPaLoIDS1pKlFRIFitW0aAjAmEALw\\_wcB](https://dratek.cz/arduino/1297-rfid-ctecka-125k-em4100-modul-rdm6300-uart-vystup.html?gad_source=1&gclid=Cj0KCQiA7aSsBhCiARIsALFvovzLJ-zO3bUVHyq0y-aDbvs6tB-8UzHlupSPaLoIDS1pKlFRIFitW0aAjAmEALw_wcB). [cit. 2023-12-27].
- [26] *RDM6300*. Online. In: Dratek.cz. 2023. Dostupné z: <https://dratek.cz/photos/produkty/f/1/1297.jpg?m=1702990682>. [cit. 2023-12-27].
- [27] *USB čtečka*. Online. Dratek.cz. 2023. Dostupné z: <https://dratek.cz/arduino/954-usb-rfid-ctecka-125khz.html>. [cit. 2023-12-27].
- [28] *USB čtečka*. Online. In: Dratek.cz. 2023. Dostupné z: <https://dratek.cz/photos/produkty/f/0/954.jpg?m=1646035557>. [cit. 2023-12-27].
- [29] *Raspberry Pi kamera V1*. Online. Laskakit.cz. 2023. Dostupné z: [https://www.laskakit.cz/raspberry-pi-kamera-5mp-v1-3/?gad\\_source=1&gclid=CjwKCAiAs6-sBhBmEiwA1Nl8s9TifWzWl5DLekjyEYAAmLh2ahUZS87su\\_RQd537QYfbEVhnsWXR0CHwMQAvD\\_BwE](https://www.laskakit.cz/raspberry-pi-kamera-5mp-v1-3/?gad_source=1&gclid=CjwKCAiAs6-sBhBmEiwA1Nl8s9TifWzWl5DLekjyEYAAmLh2ahUZS87su_RQd537QYfbEVhnsWXR0CHwMQAvD_BwE). [cit. 2023-12-27].
- [30] *Raspberry Pi kamera V2*. Online. Rpishop.cz. 2023. Dostupné z: <https://rpishop.cz/mipi-kamerove-moduly/329-raspberry-pi-kamera-modul-v2.html>. [cit. 2023-12-27].
- [31] *Raspberry Pi kamera V3*. Online. Rpishop.cz. 2023. Dostupné z: <https://rpishop.cz/mipi-kamerove-moduly/5599-raspberry-pi-camera-3.html>. [cit. 2023-12-27].
- [32] *Raspberry Pi kamera V2*. Online. In: Rpishop.cz. 2023. Dostupné z: <https://rpishop.cz/wp-content/uploads/2016/04/10440-Raspberry-Pi-kamera-V2.jpg>. [cit. 2023-12-27].

- [33] *Akustický bzučák*. Online. Laskakit.cz. 2023. Dostupné z: [https://www.laskakit.cz/akusticky-bzucak--modul/?gad\\_source=1&gclid=CjwKCAiAs6-sBhBmEiwA1Nl8sx9TicDrOsCd63Q57N\\_4Gt0B2Gzfi8lZsOqIEDW7hNeu5vu7MlzF1hoCjWQQAvD\\_BwE#productDiscussion](https://www.laskakit.cz/akusticky-bzucak--modul/?gad_source=1&gclid=CjwKCAiAs6-sBhBmEiwA1Nl8sx9TicDrOsCd63Q57N_4Gt0B2Gzfi8lZsOqIEDW7hNeu5vu7MlzF1hoCjWQQAvD_BwE#productDiscussion). [cit. 2023-12-27].
- [34] *Akustický bzučák*. Online. In: Laskakit.cz. 2023. Dostupné z: [https://cdn.myshoptet.com/usr/www.laskakit.cz/user/shop/big/6174-1\\_6174-1-akusticky-bzucak-modul.jpg?6137b46c](https://cdn.myshoptet.com/usr/www.laskakit.cz/user/shop/big/6174-1_6174-1-akusticky-bzucak-modul.jpg?6137b46c). [cit. 2023-12-27].
- [35] *USB čtečka 13,56MHz*. Online. Dratek.cz. 2023. Dostupné z: <https://dratek.cz/arduino/1924-usb-rfid-ctecka-13.56mhz.html>. [cit. 2023-12-27].
- [36] PoE splitter. Czc.cz [online]. 2023 [cit. 2023-12-28]. Dostupné z: <https://www.czc.cz/poe-splitter-pro-raspberry-pi-2b-3b-3b-a-a-b-b-banana-pi-cubieboard-pcduino-joy-it-usb-c/343612/produkt>
- [37] *Techcrowd logo*. Online. In: Techcrowd.tech. 2023. Dostupné z: [https://techcrowd.tech/wp-content/themes/techcrowd/assets/img/logo\\_techcrowd.png](https://techcrowd.tech/wp-content/themes/techcrowd/assets/img/logo_techcrowd.png). [cit. 2023-12-30].
- [38] *API dokumentace*. Online. Dostupné z: <https://api-becvary.techcrowd.space/api/documentation>. [cit. 2024-03-29].
- [39] FUSION360. *3D model konstrukce terminálu*. Online. 2024. Dostupné z: <https://a360.co/3RjlgKa>. [cit. 2024-03-31].
- [40] *Program – Ultimaker Cura*. Online. Dostupné z: <https://ultimaker.com/software/ultimaker-cura/>. [cit. 2024-03-31].
- [41] *Creality CR-10S Pro*. Online. 2020. Dostupné z: <https://www.creality.com/products/cr-10s-pro-v2-3d-printer>. [cit. 2024-03-31].
- [42] *Raspberry Pi Imager*. Online. Dostupné z: <https://www.raspberrypi.com/software/>. [cit. 2024-03-31].
- [43] *Program – Figma*. Online. Dostupné z: <https://www.figma.com/design/>. [cit. 2024-03-31].
- [44] *CustomTkinter*. Online. 2023. Dostupné z: <https://customtkinter.tomschimansky.com/>. [cit. 2024-03-31].
- [45] *CTkButton CustomTkinter*. Online. 2023. Dostupné z: <https://customtkinter.tomschimansky.com/documentation/color>. [cit. 2024-03-31].

- [46] *RDM6300 knihovna*. Online. Dostupné z: <https://github.com/mad-tinkerer/python-rdm6300>. [cit. 2024-04-01].
- [47] *Autorun skript Raspberry*. Online. Dostupné z: <https://www.dexterindustries.com/howto/run-a-program-on-your-raspberry-pi-at-startup/>. [cit. 2024-04-02].
- [48] *TP-Link TL-POE150s*. Online. Alza.cz. Dostupné z: <https://www.alza.cz/tp-link-tl-poe150s-d185018.htm>. [cit. 2024-04-26].
- [49] *OpenCV - CascadeClassifier*. Online. Opencv.org. Dostupné z: [https://docs.opencv.org/3.4/d1/de5/classcv\\_1\\_1CascadeClassifier.html](https://docs.opencv.org/3.4/d1/de5/classcv_1_1CascadeClassifier.html). [cit. 2024-04-27].
- [50] *Povolení hotspotu na Rapsberry Pi*. Online. Www.raspberrypi.com. Dostupné z: <https://www.raspberrypi.com/documentation/computers/configuration.html#enable-hotspot>. [cit. 2024-05-10].

# **SEZNAM PŘÍLOH**

**PŘÍLOHA A – KOMUNIKACE TERMINÁL – SERVER**

**PŘÍLOHA B – 3D MODEL TERMINÁLU (STL) NA PŘILOŽENÉ SD KARTĚ**

**PŘÍLOHA C – SCHÉMA ZAPOJENÍ**

**PŘÍLOHA D – SESTAVENÍ TERMINÁLU NA PŘILOŽENÉ SD KARTĚ**

**PŘÍLOHA E – NAHRÁNÍ RPIOS NA PŘILOŽENÉ SD KARTĚ**

**PŘÍLOHA F – GRAFICKÉ UŽIVATELSKÉ PROSTŘEDÍ (GUI) NA PŘILOŽENÉ SD KARTĚ**

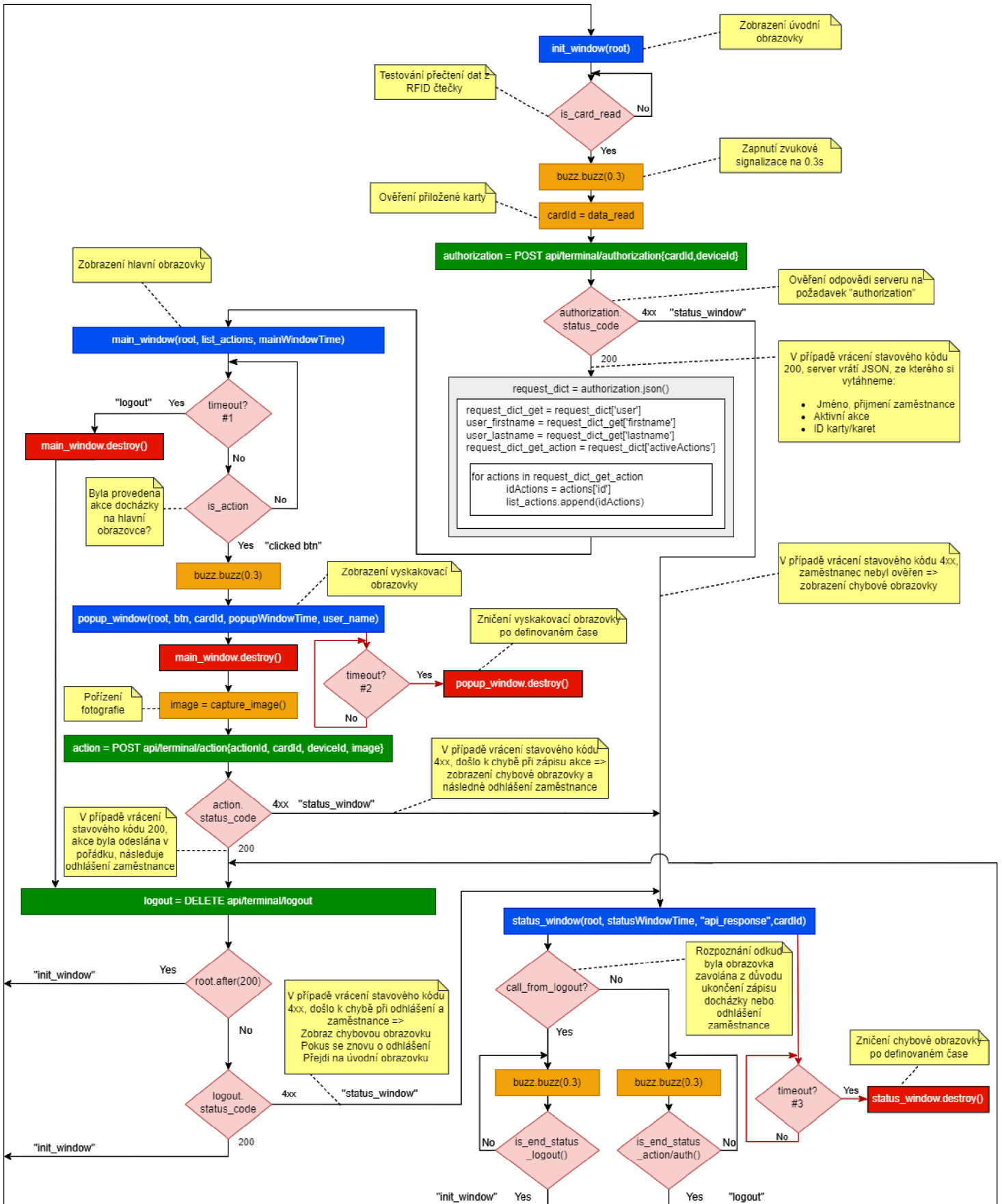
**PŘÍLOHA G – VZDÁLENÉ VÝVOJOVÉ PROSTŘEDÍ NA PŘILOŽENÉ SD KARTĚ**

**PŘÍLOHA H – SKRIPTY PROGRAMU NA PŘILOŽENÉM SD KARTĚ**

**PŘÍLOHA I – FOTOGRAFIE NASAZENÝCH TERMINÁLŮ NA PŘILOŽENÉ SD KARTĚ**

# Příloha A - Diagramy

## A.1 Vývojový diagram – detailní





# Příloha C - Schéma zapojení

