TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Online systém pro bezkontaktní dobíjení konta

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 1802T007 – Informační technologie

Autor práce: **Bc. Lukáš Hanuš**
Vedoucí práce: Ing. Lenka Kosková Třísková, Ph.D.

Liberec 2018

TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies

Online system for contactless payments into your account

Master thesis

Study programme: N2612 – Electrical Engineering and Informatics
Study branch: 1802T007 – Information Technology

Author: **Bc. Lukáš Hanuš**
Supervisor: Ing. Lenka Kosková Třísková, Ph.D.

Liberec 2018

Technická univerzita v Liberci
Fakulta mechatroniky, informatiky a mezioborových studií
Akademický rok: **2017/2018**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Lukáš Hanuš**
Osobní číslo:	**M15000165**
Studijní program:	**N2612 Elektrotechnika a informatika**
Studijní obor:	**Informační technologie**
Název tématu:	**Online systém pro bezkontaktní dobíjení konta**
Zadávající katedra:	**Ústav nových technologií a aplikované informatiky**

Zásady pro vypracování:

1. Seznamte se s konceptem zařízení pro autentizaci uživatelů v síti LIANE s využitím identifikačních karet, navrženým v BP Bc. Romana Beldy.
2. Prototyp zařízení rozšiřte o hardware nutný pro práci s platební kartou.
3. Navrhněte vhodné uživatelské rozhraní a obslužný software.

Rozsah grafických prací: **dle potřeby**

Rozsah pracovní zprávy: **40 - 60 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

[1] DOSEDĚL, T.: Počítačová bezpečnost a ochrana dat, Computer Press 2004, ISBN: 80-251-0106-1
[2] LUDVÍK, M: Teorie bezpečnosti počítačových sítí, Computer Media 2008, ISBN: 8086686353
[3] BELDA, R.: Autentizace uživatelů pro vzdálený přístup do sítě LIANE při ztrátě hesla, Bakalářská práce TUL, obhájeno 2017

Vedoucí diplomové práce: **Ing. Lenka Kosková - Třísková**
Ústav nových technologií a aplikované informatiky

Datum zadání diplomové práce: **19. října 2017**
Termín odevzdání diplomové práce: **14. května 2018**

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

Ing. Josef Novák, Ph.D.
vedoucí ústavu

V Liberci dne 19. října 2017

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 30. 8. 2018

Podpis: 

Poděkování

Mé poděkování patří především Ing. Lence Koskové Třískové za odborné vedení, trpělivost a ochotu, kterou mi v průběhu zpracování diplomové práce věnovala.

Dále děkuji Petru Martínkovi z technického úseku banky ČSOB za ochotu a cenné rady ohledně platebního terminálu.

Poděkování patří též Ing. Zdeňku Kračmarovi za poskytnutí souhlasu k vytvoření dvou testovacích účtů v univerzitní databázi. Též bych rád poděkoval Radku Melzerovi, který pro mě na základě tohoto souhlasu, fiktivní účty vytvořil a řešil se mnou veškerou komunikaci se systémem menz na TUL.

Na závěr bych rád poděkoval své přítelkyni za její psychickou oporu a trpělivost.

Abstrakt

Práce se zabývá tvorbou prototypu zařízení, které bude umožňovat studentům a zaměstnancům TUL bezhotovostní a bezkontaktní dobití stravovacího konta v menzách. Práce vznikla proto, že stávající systém dobíjení nepostačuje plně potřebám studentů.

První část práce obsahuje diskusi k výběru a zapojení potřebného hardwaru. Druhá část popisuje softwarové vybavení včetně implementace řídící aplikace, která následně celý hardware uvede do provozu a zajistí jeho plnou funkčnost.

Klíčová slova: platební terminál, platební transakce, Raspberry PI 3

Abstract

The thesis inquires into designing a prototype of equipment that will allow students and employees of TUL cashless and contactless top-up of their credit on university canteen account. The need for this paper has arisen from the insufficiency of the current charging system. The first part of the paper contains a discourse on how to select and connect the necessary hardware. The second part describes the software, as well as the implementation of the control application that puts the hardware into operation and maintains its fluent functioning.

Keywords: payment terminal, payment transaction, Raspberry PI 3

Obsah

1 Úvod....14
2 Popis zařízení....15
2.1 Požadavky na zařízení....15
2.2 Funkce zařízení....15
3 Hardwarové vybavení....17
3.1 Řídící jednotka....17
3.2 Zobrazovací zařízení....19
3.3 Autentizační zařízení....20
3.4 Platební terminál....21
3.5 Paměťová karta....22
3.6 Oživení hardwaru....22
4 Síťová topologie....24
5 Softwarové vybavení....25
5.1 Operační systém....25
5.1.1 Instalace OS....25
5.2 Vzdálená komunikace s Raspberry PI 3....26
5.3 Node JS 26
5.4 NPM....27
5.5 Knihovna pro čtečku identifikačních karet....28
5.6 Nastavení systému....29
6 Komunikace s menzou a její bezpečnost....30
7 Komunikace s platebním terminálem....33
7.1 Možnosti komunikační linky....33
7.2 Výběr komunikační linky....35
7.3 UDP Protokol a UDP Datagram....35
7.4 Samotná komunikace s platebním terminálem....36
7.4.1 Hlavička....37
7.4.2 Datová část....38
7.4.3 Servisní příkazy FIDu T....42
7.4.4 Povinnost FIDů v příkazech....44
7.4.5 Průběh platební transakce....44
8 Řídící aplikace....46
8.1 Vývojové prostředí....46
8.1.1 WebStorm....46
8.2 Základní obrazovky (pohledy)....47
8.3 Průběh platby....50
8.4 Watchdog....53
9 Závěr....54

Seznam obrázků

Obrázek 1: Use Case diagram....16
Obrázek 2: Technický popis Raspberry PI 3 - pohled zpředu....18
Obrázek 3: Technický popis Raspberry PI 3 - pohled zezadu....18
Obrázek 4: Fyzické spojení Raspberry PI 3 a displeje Raspberry PI Touch....20
Obrázek 5: Čtečka karet ACR122U....20
Obrázek 6: Platební terminál VeriFone VX 805....21
Obrázek 7: Blokové schéma zapojení HW....23
Obrázek 8: První reálné zapojení HW....23
Obrázek 9: Blokové schéma síťové topologie....24
Obrázek 10: Struktura UDP datagramu....36
Obrázek 11: Požadavek FIDu T80....43
Obrázek 12: Odpověď FIDu T80....43
Obrázek 13: Průběh finanční transakce....45
Obrázek 14: Řídící aplikace - přihlašovací obrazovka....47
Obrázek 15: Řídící aplikace – obrazovka s profilem strávníka....48
Obrázek 16: Řídící aplikace - dobíjecí obrazovka....49
Obrázek 17: Vývojový diagram základních obrazovek řídící aplikace....50
Obrázek 18: Vývojový diagram průběhu platby....52
Obrázek 19: Vývojový diagram watchdog....53

Seznam tabulek

Tabulka 1: Technické údaje Raspberry PI 3.....17
Tabulka 2: Techné údaje displaje Raspberry PI Touch.....19
Tabulka 3: Technické údaje čtečky karet.....21
Tabulka 4: Struktura hlavičky protokolu B.....37
Tabulka 5: Definice hodnot pravého bytu.....37
Tabulka 6: Definice hodnot pole flag.....38
Tabulka 7: Návratové kódy serveru a platebního terminálu.....41
Tabulka 8: Kódy transakce.....42
Tabulka 9: FIDy u normální transakce (T00).....44

Seznam zkratek a cizích slov

CPU	Central Processing Unit (centrální procesorová jednotka)
CRC	Cyclic Redundancy Check (cyklický redundantní součet)
CSS	Cascading Style Sheets
DSI	Display Serial Interface
FAT	Fiel Allocation Table
GB	Gigabyte
GPIO	General-Purpose Input/ Output
GPU	Graphic Procesing Unit (grafická procesorová jendotka)
HDMI	High-Definition Multimedia Interface
HTML	HyperText Markup Language
HW	Hardware
IDE	Integrated Development Environment
IP	Internet Protocol
ISIC	International Student Identity Card
ITIC	International Teacher Identity Card
JSON	JavaScript Object Notation
LAN	Local Area Network
LED	Light-Emitting Diode
MB	Megabyte
MB/ s	Megabyte za sekundu
OS	Operating System (operační systém)
REST	Representational State Transfer
SDHC	Secure Digital High Capacity
SDRAM	Synchronous Dynamic Random Access Memory

SoC	System On Chip
SSH	Secure Shell
SSL	Secure Sockets Layer
SW	Software
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
URL	Uniform Resource Locator (jednotná adresa zdroje)
USB	Universal Serial Bus
.NET	Framwork společnosti Miscrosoft

1 Úvod

Hlavní motivací celé práce byly nedostačující možnosti dobíjení stravovacího konta na TUL. Aktuálně univerzita nabízí pouze čtyři varianty navýšení konta. První tři jsou pouze prostřednictvím vkladů v hotovosti a to na kolejích Harcov, na přepážkách menz nebo na kterékoli pobočce ČSOB. Poslední možností je bezhotovostní převod z osobního bankovního účtu na účet menzy. Všechny varianty s sebou nesou jisté omezení, jak v podobě časové náročnosti, tak v rychlosti připsané částky. V neposlední řadě mnoho studentů z vyšších ročníků již není ubytováno nakolejích, někteří už pouze dojíždí. Z toho důvodu je potřeba brát v úvahu i dostupnost dobíjení přímo v místě menz.

Celá práce navazuje na semestrální projekt, ve kterém jsem zkoumal, jestli by o nový systém měli studenti zájem a zda by měl nějaký přínos. Po vyhodnocení všech dotazníků a po konzultacích s vedením kolejí a menz se ukázalo, že by byl nový systém velice žádaný a většina studentů by ho uvítala. Zároveň by se tak zvýšila návštěvnost univerzitních stravovacích zařízení.

Nové zařízení, které se bude nacházet ve všech menzách umožní studentům a zaměstnancům TUL dobíjení stravovacího konta pomocí kontaktní i bezkontaktní debetní karty. Jedná se o bezhotovostní transakci, která se uskuteční během několika sekund.

2 Popis zařízení

V prvním kroku bych rád definoval kritéria celého systému, aby byla dosažena jeho maximální efektivita. Bude se jednat o malý automat, který bude zcela samoobslužně umožňovat bezhotovostní dobíjení konta do stravovacího systému. Tento „kiosek“ bude umístěn ve všech menzách TUL.

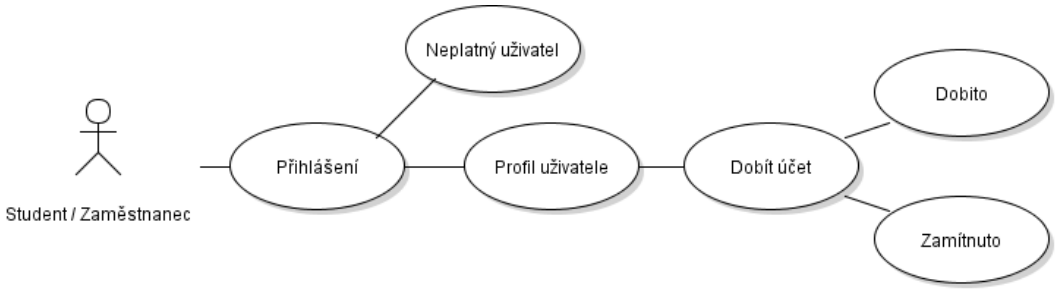
2.1 Požadavky na zařízení

- možnost dobíjení studentského konta kdykoliv
- okamžité připsání dobité částky na konto studenta/ zaměstnance
- bezhotovostní transakce
- bezkontaktní transakce
- samoobslužný přístup
- okamžité a komplexní informace o uživateli
- přívětivé uživatelské rozhraní
- bezpečnost
- nízké pořizovací náklady
- malé rozměry a malá hmotnost

2.2 Funkce zařízení

Jak bude celý systém fungovat? Jak jsem se již zmínil v úvodu kapitoly budou dobíjecí terminály umístěny ve všech menzách aby byla zajištěna jeho plná dostupnost. V první fázi k němu přistoupí samotný uživatel, který se přihlásí a zároveň ověří pomocí své identifikační karty ISIC/ ITIC. V této fázi identifikace může dojít automatem k dvojímu vyhodnocení dat. V případě platného identifikačního průkazu přesměruje uživatele na novou obrazovku, na které najde své základní osobní údaje jako je jméno, příjmení, fakulta a jeho aktuální stav účtu. V případě neplatného průkazu nebo identifikátoru,

který nemá zastoupení v univerzitní databázi uživatelů, vypíše terminál chybovou hlášku. Druhá fáze pak řeší samotné dobíjení stravovacího konta, kde si uživatel navolí požadovanou částku a potvrdí. Dále nastává pověstný scénář, který známe z obchodů. Platební terminál vyzve uživatele k vložení nebo přiložení jeho debetní karty. Pokud se bude jednat o částku přesahující pětset korun bude dále vyzván k zadání PIN kódu. Platební terminál informace o platbě zpracuje a v případě, že transakce proběhne úspěšně, řídící aplikace danou částku připíše na konto uživatele.



Obrázek 1: Use Case diagram

3 Hardwarové vybavení

Tato kapitola popisuje veškeré hardwarové komponenty, které jsou zcela nezbytné pro funkčnost celého dobíjecího terminálu. Dále zde popisuji jejich fyzické zapojení a následné oživení systému po přivedení elektrické energie do celého zařízení.

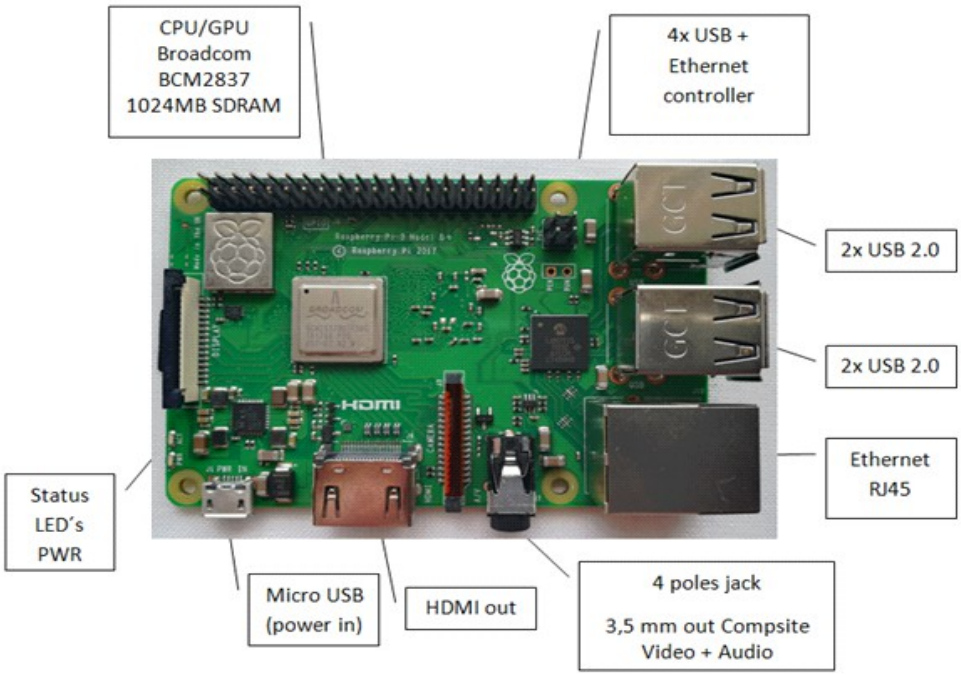
3.1 Řídící jednotka

V projektové semestrální práci jsem se zabýval možnostmi hardwarové realizace řídící jednotky. Usoudil jsem, že nejvhodnější pro realizaci celého zařízení bude použit Raspberry PI. Tento jednodeskový počítač nabízí několik modelů.. Zvolil jsem nejmodernější a dnes nejrozšířenější model, a to model B+ Raspberry PI 3. Tento model disponuje otvory, jimiž ho lze za pomocí šroubků snadno přidělat k zobrazovacímu zařízení. Základní specifikace jsou zobrazeny v tabulce č. 1.

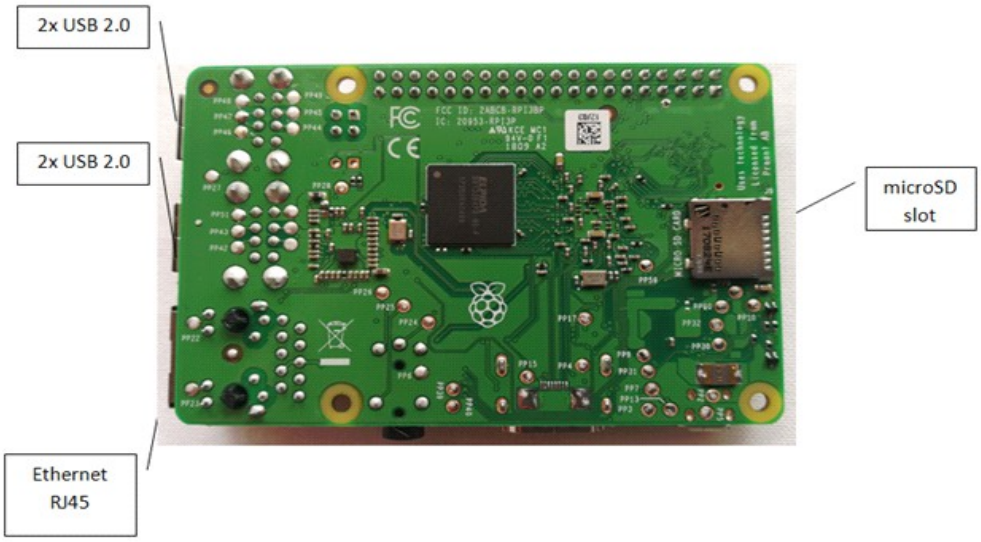
Raspberry PI 3			
Hmotnost (g)	45	**Architektura**	ARMv8 (64/32 - bit)
Rozměry (mm)	85,6 x 56,5 x 17	**SoC**	Broadcom BCM2837
Příkon (W)	1	**CPU**	1,4 GHz 64-bit quad-core ARM cortex-A53
USB 2.0 port	4x	**GPU**	Broadcom VideoCore IV @ 250 MHz
		Paměť (SDRAM)	1 GiB (sdílené s GPU)
		Video - výstupy	HDMI (revize REF1.3, kompozitní video, displej rozhraní DSI)
		Úložný prostor	MicroSDHC slot
		Síť	Gbit Ethernet, 802.11.b/g/n/ac, bluetooth 4.2
		Zdroj energie	5 V přes microUSB nebo GPIO

Tabulka 1: Technické údaje Raspberry PI 3

Obrázek č. 1 a obrázek č. 2 zobrazují technický popis jednotlivých dílčích komponent a vstupně výstupních portů Raspberry PI 3. [1]



Obrázek 2: Technický popis Raspberry PI 3 - pohled zpředu



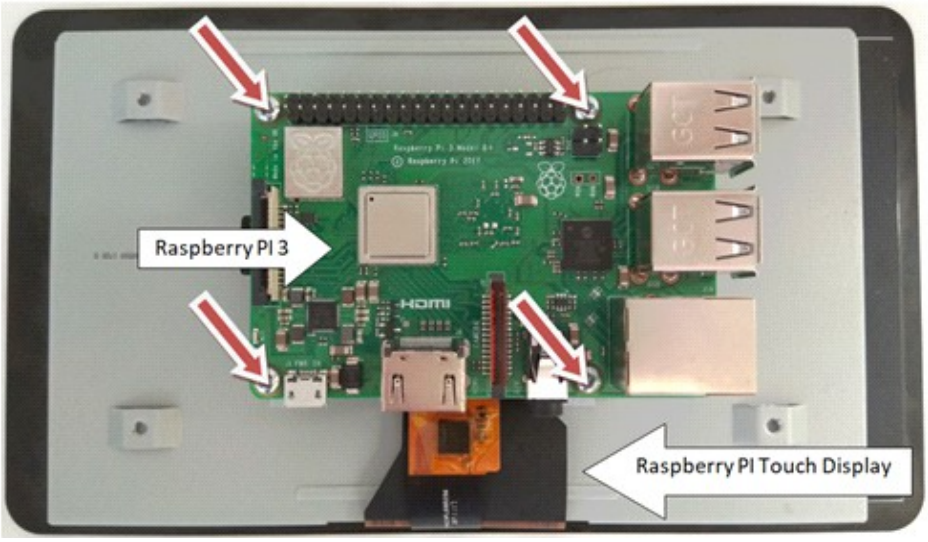
Obrázek 3: Technický popis Raspberry PI 3 - pohled zezadu

3.2 Zobrazovací zařízení

Na základě mého výzkumu v semestrálním projektu jsem se rozhodl jako zobrazovací zařízení použít Raspberry PI Touch display. Jedná se o sedmi palcový dotykový displej, který umožňuje tvorbu vlastních all-in-one systémů (např. tablet) apod. Tento displej je oficiálně vyráběný pro Raspberry PI, tudíž je plně kompatibilní s OS Raspbian. Raspberry PI Touch display podporuje ovládání až deseti prsty najednou a nabízí on- creen klávesnici v systému Raspbian. Funkce on-screen klávesnice s sebou nese výhodu ve snadném ovládání bez potřeby připojení periferních ovládacích prvků, jako je klávesnice a myš. Samotné připojení k Raspberry je realizováno přes dva kabely, kabel napájecí do GPIO portu a ribbon kabel do DSI portu. Výhodou je možnost zkompletování Raspberry PI 3 a Raspberry PI Touch displeje do jednoho celistvého zařízení. [2]

Raspberry PI Touch displej	
Rozlišeni	800 x 480 pixelů
Rozměr displeje	194 mm x 110 mm x 20 mm
Zobrazitelná velikost displeje	155 mm x 86 mm
Kontrastní poměr	500:1
Jas	250 cd/ m^2
Barevná hloubka	24 bit
Cena	cca 2 300 Kč

Tabulka 2: Techné údaje displaje Raspberry PI Touch



Obrázek 4: Fyzické spojení Raspberry PI 3 a displeje Raspberry PI Touch

3.3 Autentizační zařízení

Pro identifikaci studentů a zaměstnanců univerzity musí zařízení obsahovat čtečku identifikačních karet ISIC/ ITIC. Čtečka vyčte ID z karty každého studenta/ zaměstnance ve školní databázi.

Od mé vedoucího diplomové práce mi byla zapůjčena univerzitní čtečka NFC karet typu ACR122U od amerického výrobce ACS. Jedná se o velmi rychlou bezkontaktní čtečku, která splňuje nejen normu ISO 14443 A i B, ale také normu ISO 18092. Toto čtecí zařízení disponuje bezkontaktní technologií operující na frekvenci 13,56 Mhz. Dále podporuje rychlost komunikace až 424 kbps pro přístup k NFC tagu a umožňuje využít maximální rychlost USB rozhraní, tedy až 12 Mbps. Dosah čtení čipových karet je až do vzdálenosti 50 mm. [3]

Obrázek 5: Čtečka karet ACR122U

Čtečka karet ACR122U	
Hmotnost	70 g
Rozměry	(98 x 65 x 12,8) mm
Datové rozhraní	USB 2.0
Čtecí vzdálenost	až 50 mm
Frekvence	13,56 MHz

Tabulka 3: Technické údaje čtečky karet

3.4 Platební terminál

Aby bylo možné zrealizovat bezhotovostní platbu, bylo nutné opatřit platební zařízení, které tento typ plateb podporuje. Jak jsem již popisoval v semestrálním projektu, nebylo zcela jednoduché platební terminál obstarat. Nakonec mi byl k testovacím účelům bankou ČSOB (jejíž bankovních služeb využívá právě TUL) zapůjčen platební terminál VeriFone, model VX 805. Toto platební zařízení podporuje kontaktní i bezkontaktní transakce, má k dispozici ethernetový port a je napájeno 12V. Spolu s platebním terminálem mi byly zapůjčeny dvě testovací platební karty, Maestro a Visa. Tyto karty se od sebe liší možnostmi placení, z nichž rozšířenější u nás jsou karty typu Maestro. Detailní popis je nahlédnutí v příloze na CD. [4]

Obrázek 6: Platební terminál VeriFone VX 805

3.5 Paměťová karta

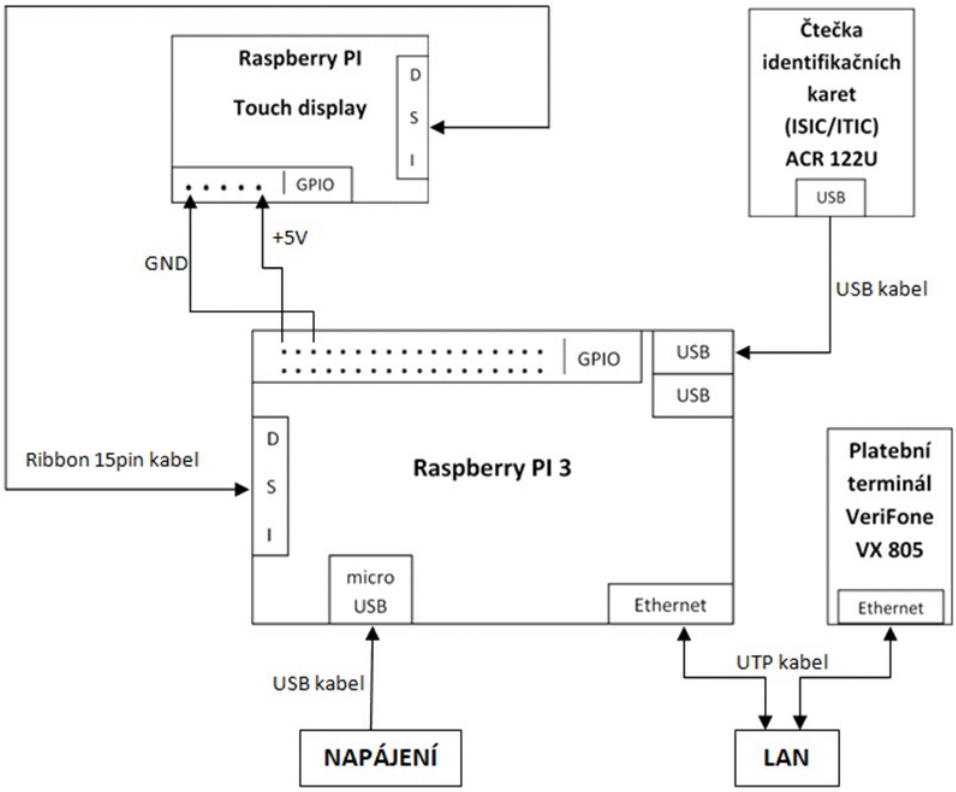
Jelikož Raspberry PI nemá integrované datové úložiště, na které se instaluje OS a jsou uložena data, ale zároveň disponuje slotem na microSD, bylo nutné zakoupit paměťovou kartu.

Jedná se o paměťovou kartu typu microSDHC class 10 UHS-I od amerického výrobce Kingston s kapacitou 16 GB. Tato karta má rychlost čtení 80 MB/ s a rychlost zápisu 10 MB/ s. S těmito parametry je tato paměťová karta pro můj projekt zcela dostačující.

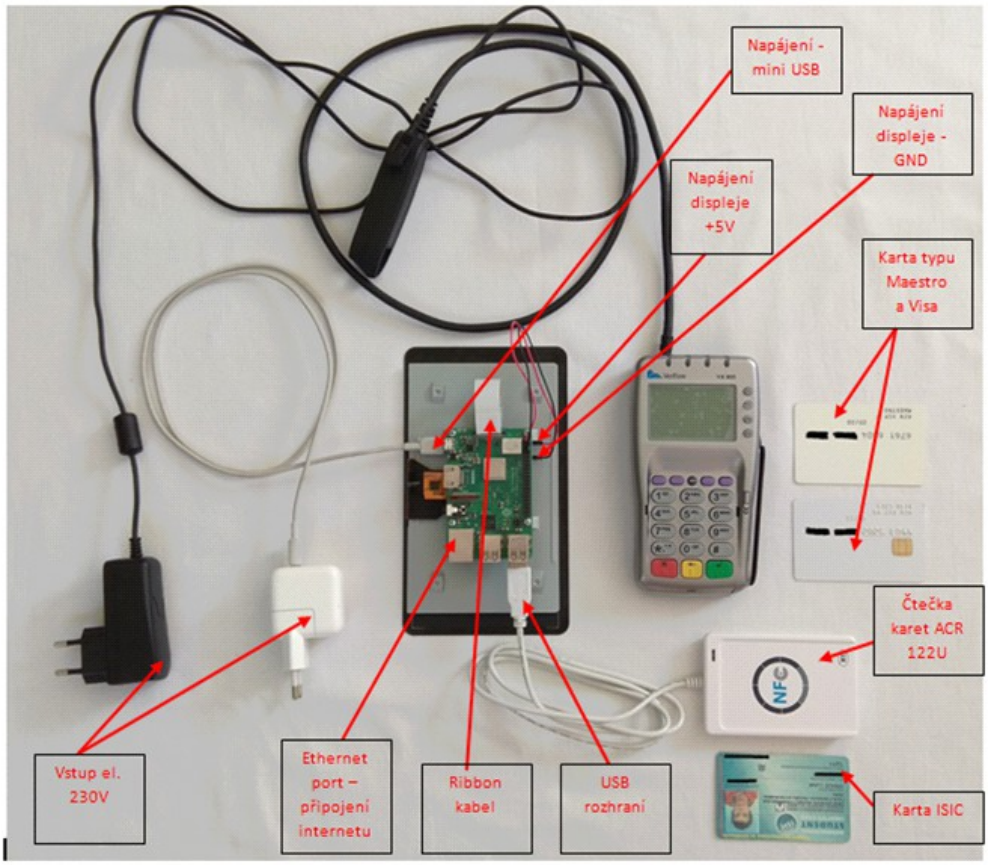
3.6 Oživení hardwaru

Po vhodném zvolení a získání všech hardwarových komponent, jsem je patřičnými kabely připojil do příslušných portů a oživil je přivedením elektrické energie do řídící jednotky, tj. Raspberry PI 3. Jako elektrický zdroj jsem použil 800mA napájecí adaptér, který mi zapůjčila má vedoucí diplomové práce. Jedná se o originální adaptér od Raspberry PI pro tento minipočítač. Při přivedení elektřiny do řídící jednotky, jsem narazil na první problém – LED diody na Raspberry sice blikaly, ale displej se vůbec nerozsvítil. Při zapojení Raspberry přes HDMI do televize vše fungovalo. Začal jsem tedy řešit možné příčiny nefunkčnosti displeje. Nakonec jsem zjistil, že vše bylo způsobené nedostatečně silným napájecím zdrojem. Po zakoupení nového 2A adaptéru, již vše fungovalo jak mělo, a tak jsem se mohl věnovat softwarové části.

Na obrázku č. 5 je vyobrazeno blokové schéma zapojení. Dále pak na obrízku č. 6 první reálné zapojení.



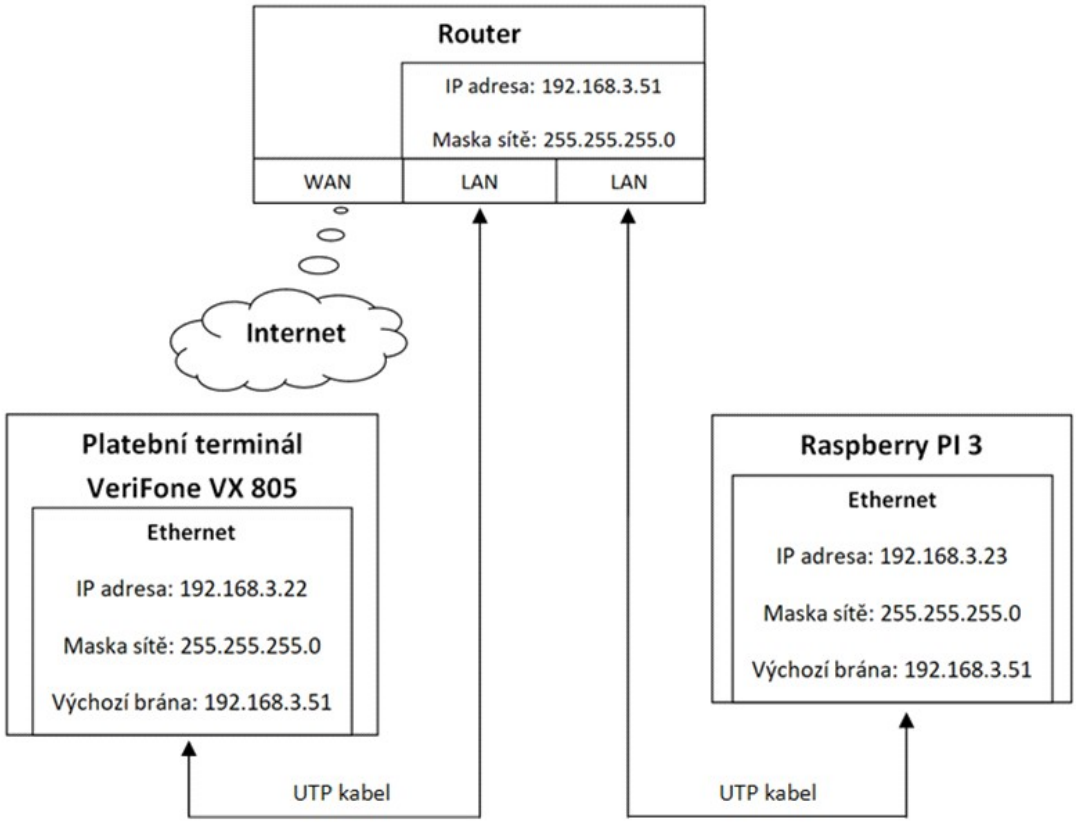
Obrázek 7: Blokové schéma zapojení HW



Obrázek 8: První reálné zapojení HW

4 Síťová topologie

Jelikož se jedná o dobíjecí terminál, který bude umožňovat bezhotovostní platební transakce, hraje zde klíčovou roli síťové zapojení. Platební terminál musí mít přístup do internetu, aby mohl komunikovat s autorizačním serverem banky ČSOB. Cílová IP adresa tohoto serveru se nastavuje přímo v platebním terminále pod servisní nabídkou ke které tester nemá přístup. Dále je zapotřebí, aby platební terminál znal IP adresu řídící jednotky, se kterou bude navazovat spojení při vykonávání platební transakce. Toto nastavení rovněž nemůže učinit uživatel, jako jsem já. Z tohoto důvodu jsem nastavení musel dobře promyslet, a to z pohledu konfigurace své domácí sítě. Následně jsem požadavek na nastavení zaslal technikům v ČSOB, a to dříve než mi samotný testovací terminál byli ochotni poslat. Nynější síťová konfigurace pro mé testovací účely vypadá takto:



Obrázek 9: Blokové schéma síťové topologie

5 Softwarové vybavení

Po zapojení jednotlivých komponent a portů bylo nutné nejprve nainstalovat OS, který tvoří komunikační rozhraní mezi uživatelem a HW.

Aby má aplikace správně fungovala, bylo nutné do OS doinstalovat veškeré nezbytné programy, které jsou pro implementaci nezubytné. Jedná se např. o NodeJS, NPM, ovladače pro čtečku identifikačních karet ISIC/ ITIC.

5.1 Operační systém

Oficiálním operačním systémem pro Rapsberry PI 3 je Raspbian. Výrobce Raspberry PI 3 nabízí dva způsoby instalace tohoto operačního systému: s využitím instalátoru NOOBS nebo zkopírováním obrazu operačního systému na SD kartu. Zvolil jsem cestu pomocí instalátoru. Varianta NOOBS je dostupná ve dvou verzích – NOOBS a NOOBS Lite. Plná verze NOOBS obsahuje obraz distribuce Raspbian pro offline instalaci. Verze Lite má 20 MB, a zbytek balíčků si stahuje při samotné instalaci přímo ze sítě. Vzhledem k vyšší bezpečnosti jsem se rozhodl použít verzi Lite. Dále také z důvodu, aby systém neobsahoval zbytečné balíčky, které by zpomalovaly chod celého operačního systému.

5.1.1 Instalace OS

Samotná instalace OS je poměrně jednoduchá. Staženou zabalenou verzi NOOBS Lite stačilo rozbalit na paměťovou kartu, která musí být zformátovaná na souborový systém typu FAT 32. Následně jsem kartu zasunul do samotného Raspberry PI 3. Po zapnutí Raspberry PI 3 se automaticky z paměťové karty spustil instalátor OS. Lze volit mezi instalací systémů Raspbian, OpenElec, OSMC, Risc-OS atd. Pro mé účely postačí pouze čistý Raspbian.

Jedná se o distribuci OS Linux, která je založena na Debianu Wheezy (tzn. stable) a je upravená pro specifický HW Raspberry PI 3. OS Raspbian má podporu v grafickém prostředí, což je uživatelsky přívětivější.

Vzhledem k tomu, že pro moji práci jsem potřeboval grafický webový prohlížeč, zvolil jsem OS Raspbian s grafickým prostředím. Tento OS je velký 4 125 MB. Následně sám instalátor rozdělí SD kartu na jednotlivé oddíly podle potřeb zvoleného systému. Většina systémů vyžaduje dva oddíly – boot a root. V mém případě je karta rozdělena na čtyři oddíly. První oddíl je boot o velikosti 69 MB. Oddíl druhý tzv. root disponuje velikostí 32 MB a může se dynamicky zvětšovat. Dále pak oddíl recovery o velikosti 63 MB. Zbytek prostoru je pro samotný OS Raspbian. Po úspěšné instalaci se spustí grafické prostředí OS a systém je plně připraven pro okamžité používání. [5]

5.2 Vzdálená komunikace s Raspberry PI 3

Vzdálené příhlášení vede k rychlejší práci se samotným Rapsberry PI 3. Využil jsem možnosti SSH. Secure Shell vytvoří zabezpečený tunel mezi dvěma počítači. Využití má zejména v případě, že chceme ovládat počítač na dálku. SSH server standardně naslouchá na TCP portu 22. Tento komunikační protokol vznikl na podkladě dřívější verze Telnet, která však nebyla řádně zabezpečena. Počítač je pomocí SSH ovládán pouze pomocí příkazové řádky. Dále najde SSH široké uplatnění v administraci veškerých dobíjecích terminálů. Veškeré opravy a aktualizace bude moci správce těchto terminálů provádět ze všech míst, kde je k dispozici připojení k internetu.

Tímto je Raspberry PI 3 připraveno na vzdálený přístup k jinému počítači.

5.3 Node JS

Celá aplikace musí umět reagovat na vstupy a výstupy v reálném čase, např. čtečka identifikačních karet ISIC/ ITIC, platební terminál, ověřování dostupnosti internetu. Z tohoto důvodu jsem nainstaloval Node.js určený pro ARM procesory. Jedná se o software vý systém, který byl navržen pro psaní celé řady internetových aplikací, zvlá elké části webových serverů. Dalo by se říci, že jde o serverový JavaScript, proto jsou i samotné aplikace spouštěné pomocí Node.js psané v JavaScriptu.

5.4 NPM

Abych mohl stahovat a využívat různé knihovny a balíčky, nainstaloval jsem manažer balíčků pro Node.js, označovaný jako npm. Balíčky jsou zcela nezbytné pro realizaci celé řídící aplikace. Následně uvedu několik základních balíčků a jejich význam.

- **Express**

  Jedná se framework, který zásadně zjednodušuje základní možnosti Node.js.

- **Pcsclite**

  Tato knihovna umožňuje komunikaci s čtečkou identifikačních karet.

- **Crypto**

  Pomocí tohoto balíčku jsou podepisována data, která jsou odesílána webové službě za pomocí privátního klíče.

- **Dgram**

  Díky této knihovně může aplikace odesílat udp datagram, na kterém je založena komunikace s platebním terminálem.**Websocket**

  Aby bylo možné navázat spojení v reálném čase, bylo zapotřebí doinstalovat balíček socket.io. Ten zajistí komunikaci mezi klientem a serverem v reálném čase, tzn. obě strany čekají dokud některá z nich nepošle nějaký požadavek.

  Jedná se o nezávislý protokol, který je založený na základech TCP. Tento protokol nám umožňuje vzájemné působení mezi webovým serverem a prohlížečem. Celá komunikace probíhá prostřednictvím TCP portu s výchozí hodnotou 80. Lze použít i šifrované spojení (výchozí port 443). Web Socket dnes podporují již téměř všechny webové prohlížeče (tj. nejen Google Chrome, Microsoft Edge, Firefox, Opera, Safari, ale také Microsoft Internet Explorer). [6]

5.5 Knihovna pro čtečku identifikačních karet

Čtečku indentifikačních karet ISIC/ ITIC jsem měl zapůjčenou od mé vedoucí diplomové práce. Zapůjčná čtečka měla označení ACR122u. Aby fungovala komunikace mezi řídící aplikací a čtečkou karet, bylo zapotřebí doinstalovat do operačního systému ovladače pro daný typ čtečky. K těmto účelům jsem použil knihovnu libnfc verze 1.8.0. Bohužel už v této první fázi jsem narazil na značný problém. Po vyzkoušení celé řady návodů z internetu operační systém čtečku nedetekoval. Rozhodl jsem se oslovit pana Romana Beldu, který se zabýval ve své bakalářské práci ověřováním studentů a zaměstnanců TUL pomocí NFC čtečky. Po několika telefonátech jsme se dohodli na osobní schůzce. Zjistil jsem, že jeho ověřování bylo značně jednodušší, protože používal dražší čtecí zařízení. Použitá čtečka v bakalářské prácí Romana Beldy emulovala v podstatě stisk tlačítka na klávesnici. Tzn., že jeho aplikace čekala na stisk daného tlačítka, což se stalo po přiložení identifikační karty na čtecí plochu čtečky.

Nejdříve jsem chtěl využít stejného způsobu jako Roman Belda, ale po konzultaci s panem Radkem Melzerem mi byla tato varianta z bezpečnostních důvodů rozmluvena a pan Melzer trval na tom, aby byla komunikace mezi čtečkou identifikačních karet a řídící aplikací kompletně mnou naprogramována. Dalším aspektem byl vysoký nárok na pořízovací cenu stejného druhu čtečky jako použil pan Belda.

Vrátil jsem se tedy opět k typu ACR122u a dále jsem hledal důvod, proč tato čtečka odmítá komunikovat s operačním systémem. V dokumentaci k příslušné knihovně je tato čtečka ACR122u zastoupená v listu podporovaných typů. Nakonec jsem zjistil, že se ve skutečnosti nejedná o typ ACR122u, který je uveden na plastovém pouzdře ale, že se uvnitř ukrývá čip označený jako ACR38. Tento typ bohužel s knihovnou libnfc nekomunikuje. Po opatření nové čtečky neměl systém problém tuto čtečku korektně detekovat.

5.6 Nastavení systému

Proto, aby bylo zajištěno automatické spuštění celé aplikace, bylo nutné provést několik nastavení systému. V první řadě bylo třeba zajistit, aby se po nastartování OS sám spustil webový prohlížeč, v němž se aplikace zobrazí. Je potřebné, aby se webový prohlížeč zároveň spustil v celém okně. Aby bylo toto zajištěno, musel jsem editovat soubor „autostart“, který se nachází v domovském adresáři výchozího uživatele pod touto cestou : *config/lxsession/LXDE-pi.* V tomto souboru pak stačilo přidat jeden řádek:

/usr/bin/chromium-browser --kiosk --disable-restore-session-state "http://localhost:3000.

Dále bylo nutné zajistit, aby se webový server spouštěl ihned po startu systému. Abych tohoto docílil, vytvořil jsem jednoduchý script, který obsahuje příkaz na spuštění aplikace. Dále jsem tomuto scriptu musel nastavit patřičná oprávnění, aby ho systém mohl po nastartovnání spustit. Obdobným postupem jako spouštění webového prohlížeče jsem nastavil spouštění tohoto scriptu.

Díky tomuto závěrečnému nastavení je aplikace plně připravena k provozu.

6 Komunikace s menzou a její bezpečnost

Ověření uživatele je založeno na přístupových identikačních kartách ISIC/ ITIC, k čemuž je bezpodmínečně nutná komunikace se školní databází uživatelů. Každá karta má svůj unikátní identifikační klíč. Získávání dat ze školní databáze probíhá pomocí REST služby, která běží na univerzitním serveru. Komunikace mezi řídící jednotkou a službou probíhá ve fomrátu JSON. Tento formát je zcela nezávislý na počítačové architektuře a je určený pro přenos dat.

Abych získal přístup k datům pro jednotlivé uživatele, navštívil jsem správce systému pro menzu TUL, pana Radka Melzera, a požádal ho o poskytnutí přístupu k webové službě. V případě tohoto systému by se jednalo o dvě metody: první metoda by vracela základní údaje o uživateli a metoda druhá by uživateli dobila příslušnou částku na jeho stravovací konto. Pan Melzer s tímto souhlasil, ale nemohl tak učinit bez souhlasu ředitele kolejí a menz. Následně jsem se tedy obrátil na pana Ing. Zdeňka Kračmara a požádal jej o vydání souhlasu, díky němuž mi bude moci pan Radek Melzer poskytnout potřebná data. Po zasvěcení ředitele kolejí a menz do celé problematiky a po nastínění finálního zařízení, se mu celý nápad líbil, byl ke všemu zcela otevřený a s vydáním souhlasu neměl žádný problém.

Aby nebyl narušen dosavadní chod celého systému a neohrozili jsme stávající konta uživatelů menz TUL, dohodli jsme s panem Ing. Kračmarem na vytvoření dvou fiktivních studentských účtů, ke kterým mi následně bude umožněn přístup a já na nich budu moci celý projekt testovat. Abychom zamezili potížím s účetnictvím, budou tato fiktivní konta těsně před uzávěrkou na konci měsíce vynulována (tzn. nulový zůstatek na kontě).

Při práci s citlivými daty je nezbytné dbát na jejich zabezpečení. Aby si univerzita byla jistá, že se jedná o výměnu dat s jejich dobíjecím terminálem, bylo nutné vytvořit sadu klíčů pro podpis odesílaných dat, které se budou posílat webové službě. Klíče jsem vygeneroval přímo na Raspberry PI 3 (RSA o délce 2048). Vygenerovaný veřejný klíč spolu s identifikačním číslem terminálu jsem osobně donesl panu Radku Melzerovi, aby mohl na straně webové služby vyhodnotit podpis a ověřit zda se jedná o příslušný platební terminál.

Metoda pro získání údajů konkretního uživatele má tento tvar:

https://menza.tul.cz/api/rest/getUserByCard

POST{

[terminal] = číslo terminálu

[card] = UID karty

[signature] = podpis dat (terminal+'|'+card)

}

Metoda pro připsání konkretní částky konkrétnímu uživateli má potom tvar takový:

https://menza.tul.cz/api/rest/addDeposit

POST{

[terminal] = číslo terminálu

[card] = UID karty

[deposit] = částka, oddělovač je tečka

[transaction_id] = číslo stvrzenky z terminálu

[note] = volitelná poznámka

[signature] = podpis dat

(terminal+'|'+card+'|'+deposit+'|'+transaction_id+'|'+note)

}

První metoda slouží k získaní údajů pro konkretního uživatele. Volání této metody proběhne vždy po přiložení identifikační karty na čtečku karet. Obsahuje tři vstupní parametry. Prvním parametrem je identifikační číslo terminálu, v mém případě se jedná o tento tvar: **M1TEST1262.** Dalším parametrem je unikátní číslo identifikační karty uživatele, které vyčte čtečka karet. Toto číslo se skládá z osmi hexadecimálních znaků. Konkrétně unikátní identifikační číslo mého ISICu vypadá takto: **6BFA69DB.** Poslední parametr je podpis, který bude dost podobný pro obě metody, proto bude popsán níže.

Druhá metoda již realizuje samotné připsání dobíjené částky uživatelem na jeho stravovací konto a volá se poté, co z platebního terminálu příjde zpráva ,že daná transakce proběhla v pořádku. Zde je vstupních parametrů celkem šest.

První dva parametry jsou stejné jako v předchozí metodě. Třetí parametr obsahuje informace o výši dobíjené částky, kterou si uživatel stanoví pomocí dotykového displeje. Dalším parametrem je číslo stvrzenky, která příjde z platebního terminálu a

slouží k napárování konkretního uživatele ke konkrétní transakci. Předposlední parametr obsahuje volitelnou poznámku, kdyby bylo zapotřebí zasílat do univerzitní databáze ještě nějaké doplňující údaje. Posledním parametrem je podpis, který je stejný jako u metody první a bude vysvětlen v následujícím odstavci.

Jak jsem již zmínil výše, obě metody obsahují podpis. Ten se skládá z řetězce složeného z ostatních vstupních parametrů, které jsou odděleny svislicí. Příklad podpisu u první metody: **M1TEST1262|6BFA69DB**. Tento řetězec je následně za pomocí privátního klíče podepsán a odeslán jako samostatný parametr do webové služby.

Samotný přenos dat probíhá prostřednictvím zabezpečeného protokolu HTTPS.

7 Komunikace s platebním terminálem

Aby celý systém umožňoval platbu prostřednictvím debetních karet, bylo zapotřebí do celé struktury zakomponovat platební terminál. V předchozí kapitole jsem uvedl způsob fyzického zapojení platebního terminálu k řídící jednotce. Tato kapitola již popisuje detailnější komunikaci s platebním terminálem.

7.1 Možnosti komunikační linky

Komunikace je realizována za pomoci komunikačního protokolu B. Tento protokol nabízí celkem čytři režimy komunikace s pokladnou (tj. řídící jednotkou):

A) Sériový port

V případě propojení pokladny a terminálu použitím sériové linky je nutné nastavit linku 8N1 (tj. 8 datových bitů, bez parity, 1 stopbit). Je nutné, aby rychlost sériového portu pokladny byla nastavena na stejnou hodnotu jako rychlost sériového rozhraní platebního terminálu. U sériového portu terminálu je rychlost konfigurovatelná a jsou podporovány rychlosti: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bauds.

B) UDP/ IP/ Ethernet

Komunikuje-li terminál a pokladna prostřednictvím UDP, musí být v terminálu nastavena IP adresa pokladny a pokladna musí znát adresu terminálu.

Prostřednictvím IP protokolu na UDP portu 33333 probíhá komunikace. Při použití UDP se zasílá najednou kompletní zpráva protokolu B zabalená do jednoho UDP datagramu.

C) TCP/ IP/ Ethernet – terminál režim klient

Komunikuje-li pokladna prostřednictvím TCP, musí být v terminálu nastavena IP adresa pokladny. Při komunikaci tímto způsobem je pokladna server, který poslouchá na nastaveném portu.

Terminál představuje klienta, který se periodicky snaží připojit k pokladně. Po spojení terminál čeká na přijetí příkazů komunikačního protokolu B. Jelikož

při takto zvoleném způsobu spojení je terminál klientem, nemůže provádět udržování spojení. Odpovědnost za udržení spojení je tedy na pokladně (serveru) a to za použití TCP keep-alive paketů. Druhou variantou je zajištění pomocí aktivních prvků sítě mezi pokladnou a platebním terminálem, tím je také zajištěna i nepřerušitelnost tohoto TCP spojení.

Stejným způsobem jako na sériové lince jsou prostřednictvím TCP přenášeny zprávy B protokolu.

- **Víceportový pokladní server**

  Pokladní server naslouchá na specifikované skupině TCP portů z nichž každý port je pro jeden terminál. Na serveru je vazba PORT <___> TERMINAL držena v konfiguraci.

- **Jednoportový pokladní server**

  Pokladna naslouchá pouze na jednom TCP portu. Pro připojení na tento port jsou nakonfigurovány všechny terminály. Po vzniku TCP spojení, může pokladna identifikovat terminál, a to pomocí dvou způsobů. Buď můžeme použít přikaz GetAppInfo nebo speciálně upravené B protokolové žádosti. Na základě dat odpovědí získá identifikaci platebního terminálu, konkrétně v hlavičce.

D) TCP/ IP/ Ethernet – terminál režim server

Komunikuje-li pokladna prostřednictvím TCP v režimu terminál TCP server, musí být v terminále nastavena IP adresa pokladny. Při takto zvoleném způsobu kumunikace je terminál serverem poslouchajícím na nastaveném portu.

Pokladna se tedy chová jako klient snažící se periodicky připojit k terminálu. Po ustavení spojení čeká terminál na přijetí B protokolových příkazů.

Terminál přijme pouze spojení z nastavené IP adresy pokladny. Každý pokus z jiné IP adresy je zamítnut. Spojení z pokladny může být pouze jedno a při pokusu o navázání nového spojení z konkrétní adresy pokladny terminál stávající spojení ukončí a naváže nové. Tím je pokryt pro případ restartu či výpadku pokladny.

Je nutné v pokladně podporovat aplikační keep-alive, a to z důvodu pokrytí případných výpadků spojení či restartu terminálu (např. výpadek napájení či potíže s kabely).

Je zde podporován aplikační režim keep-alive stejně jako v režimu klient, protože implementace TCP v OS terminálu nezná linkový keep-alive. Dojde-li např. k výpadku napájení terminálu, pokladna za pomoci tohoto mechanismu problém detekuje, naváže nové spojení, a je schopna konkrétně provést transakci.

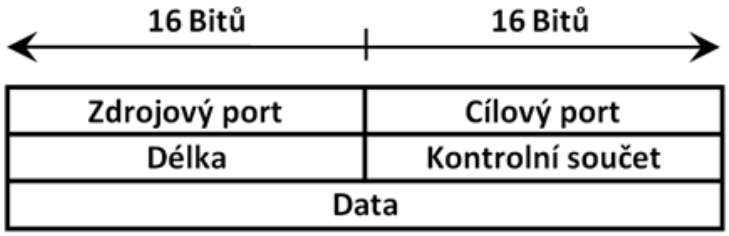
7.2 Výběr komunikační linky

Prvním krokem bylo vybrat vhodnou komunikační linku, kterou platební terminály od ČSOB podporují. Nastavení komunikační linky na platebním terminálu musí nakonfigurovat specializovaný pracovník banky ČSOB. Tester nemá oprávnění měnit typ komunikační linky. Z toho důvodu bylo zapotřebí vybrat typ linky, po které bude probíhat komunikace ještě před doručením platebního terminálu na moji adresu. Po výměně několika e-mailů a telefonních konzultacích s technikem banky, jsem se rozhodl pro nastavení terminálu do režimu UDP. Hlavním důvodem byla daleko menší režie z hlediska řídící aplikace. Režimy TCP jsou určeny na rozsáhlejší infrastrukturu platebních terminálů. Jedná se o případ, kde je jedna pokladna a desítky, či stovky platebních terminálů. Navíc by bylo zapotřebí ošetřit veškeré kolize na úrovni platebního softwaru. Jednalo by se např. o situaci, kdy by zároveň přišlo více požadavků z platebních terminálů a došlo by tak k rozporu.

7.3 UDP Protokol a UDP Datagram

Jedná se o základní protokol transportní vrstvy v architektuře TCP/ IP. Je to alternativa k protokolu TCP. Liší v tom, že nenavazuje přímé spojení mezi komunikujícími počítači. Strana, která odesílá zprávu odešle pouze paket a ztrácí nad ním veškerou kontrolu. To znamená, že už se nestará o to, jestli byl daný paket doručený nebo ne. O takové věci se stará protokol aplikační. Výhoda tohoto protokolu spočívá v malé náročnosti na síť. Navíc adresát nemusí být pouze jednoznačná IP adresa, ale skupina cílových počítačů, tím pádem můžeme posílat broadcast. Největší uplatnění UDP

v dnešní době nachází v překladu doménových jmen na IP adresy, znamená to, že jej využívají DNS servery. Aby bylo zajištěno správné doručení dat pro různé aplikace, UDP používá porty. Číslo portu představuje 16 bitová hodnota v rozsahu 0-65535. Každý UDP datagram se skládá z hlavičky a samotných dat. Hlavička obsahuje informaci o zdrojovém portu, cílovém portu, dálky UDP paketu a kontrolní součet. Vzhledem k tomu, že se jedná o protokol bezestavový, je zdrojový port volitelný. Jak zdrojový, tak cílový port je šestnáctibitový. Délka paketu je povinná a uvádí se v bytech včetně dat. Její minimální délka je 8 bytů. Šestnáctibitový kontrolní součet pokrývá hlavičku i data. [7]



Obrázek 10: Struktura UDP datagramu

7.4 Samotná komunikace s platebním terminálem

Samotná komunikace s platebním terminálem je založena na posílání a přijímání UDP datagramu. Každý tento datagram obsahuje přikaz (zprávu). Přikazy jsou složeny opět ze dvou částí, jako je to v přpadě UDP datagramu, a to z hlavičky a datové části. Každý přikaz je uvozen znakem STX (02h) a ukončen znakem ETX (03h). Hlavička i datová část jsou kódovány v ASCII. Struktura příkazu vypadá takto:

<STX> [Hlavička] [Datová část] <ETX>

7.4.1 Hlavička

Hlavička, která tvoří první část příkazu nese informace o protokolu, Pos ID, datumu a času transakce, délce datové části a další dodatečné informace v poli Flag. Struktura hlavičky je zobrazena v následující tabulce.

Pole	Formát	Délka v bytech	Význam pole	Hodnota
PT	alfanumerický	2	Typ protkolu	B0 – BF
PV	alfanumerický	2	Verze protokolu	01 - FF
TID	alfanumerický	8	Pos ID	Vysvětleno později
DT	numerický	12	Datum a čas	Akt. datum a čas
FLG	alfanumerický	4	Flag	Vysvětleno později
DLN	alfanumerický	4	Délka datové části	0000h - 01FFh
CRC	alfanumerický	4	Standard CRC16	Součtová hodnota

Tabulka 4: Struktura hlavičky protokolu B

Typ protokolu obsahuje hodnoty v rozsahu "B0" – "BF". B je protokol pro pokladnu a platební terminál. Levý byte B odpovídá identifikaci protokolu (protocol ID) a pravý byte 0 je vyhrazen pro ActivityInfoMessage. Z hodnot 1 - F vyplývají čísla zpráv.

Hodnota	Akce	Směr komunikace
1	Transaction Request	pokladna → terminál
2	Transaction Response	terminál → pokladna
3	Ticket Request	pokladna → terminál
4	Ticket Response	terminál → pokladna

Tabulka 5: Definice hodnot pravého bytu

Verze protokolu je číslo v rozmezí "01" – "FF". Terminal ID je jedinečný 8 bytů dlouhý identifikátor POS. Datum a čas transakce je uveden ve formátu YYMMDDHHMMSS (tj. rok|měsíc|den|hodina|minuta|sekunda). Flag je pole pro přenos dodatečných informací a může obsahovat rozsah hodnot: "0000" – "FFFF". Existují předdefinované hodnoty tohoto pole, které jsou uvedeny v tabulce na další stránce.

Bit	Nastaveno na „0“	Nastaveno na „1“
0	podpis není vyžadován	podpis je vyžadován
1	není lísteček	pokladna musí vytisknout lísteček
2-10	nejsou využity	nejsou využity
11		offline transakce
12		výše placené částky vyžaduje kontrolu totožnosti zákazníka
13		pokladna podporuje Keep-alive mechanismus na TCP spojení
14		při platbě je provedena kontrola whitelistu
15		je požadováno potvrzení přijetí zprávy od pokladny

Tabulka 6: Definice hodnot pole flag

Délka datové části určuje délku datové části příkazu: "00FF" = 255B. Poslední částí hlavičky je CRC, který se aktuálně nevyužívá a vyplňuje se konstantou "A5A5".

Příklad zápisu hlavičky:

[B1 | 01 | A123XX01 | 18 08 13 08 23 14 | 00 00 | 00 00 | A5 A5]

7.4.2 Datová část

V druhé části příkazu jsou přenášena data, která jsou uložena do jednotlivých polí. Každé z těchto datových polí začíná znakem – Field Separator **FS** (1Ch). Struktura datových polí má tento tvar:

<FS> [pole_ID | pole_DATA] <FS> [pole_ID | pole_DATA] …...

Do této části lze zahrnout i soubor dodatečných datových polí, která tvoří zbývající část příkazu. Tato datová pole mohou být zahrnuta v požadavcích od řídící jednotky, a v odpovědích od platebního terminálu. Jednotlivá pole jsou pak vymezená systémem identifikátorů polí (tj. Field Indentifiers → FIDs). Jejich pořadí v příkazu není vymezené, tzn. že jejich vzájemné pořadí může být různé v různých požadavcích

i odpovědích. Též není zaručena jejich neměnnost v následných aktualizacích protokolu.

Z toho předpokladu je zjevné, že parsování odpovědí nemůže být založeno na pořadí ani absolutní pozici jednotlivých FIDů.

FIDů je několik typů, kdy každý z nich má jiný formát, má vymezený rozsah délky a může se vyskytovat v odpovědi platebního terminálu, v požadavku řídící aplikace , popř. v obouch. Následující stránky popisují jednotlivé FIDy.

- **FID B | MNOŽSTVÍ**

  Tento FID obsahuje částku, kterou bude zákazník platit. Toto pole, které má délku jeden až osmnáct bytů, je povinné pro finanční transkace. V odpovědi je tento FID nepovinný.

- **FID F | AUTORIZAČNÍ KÓD**

  Jedná se o pole pevné délky osm bytů obsahující autorizační kód vygenerovaný při autorizaci této transakce. Daný kód je reprezentován alfanumerickým řetězcem. V požadavku je nepovinný v odpovědi taktéž. Ovšem není-li zaslán v požadavku je v odpovědi automaticky vygenerován při autorizaci. Je-li pole použito v požadavku příkazu je zopakováno i v odpovědi. V případě, že příkaz tento požadavek neobsahuje může být vygenerován při autorizaci. Poslední možností je, že příkaz neobsahuje tento požadavek, pak může být vygenerován při autorizaci.

- **FID I/ E | MĚNOVÝ KÓD**

  Toto pole obsahuje kód měny použité během platební transakce. Dnes se doporučuje používát FID E kvůli zpětné kompatibilitě. Je zde zachován i starší FID I. V požadavku je toto pole s pevnou délkou tři bytů nepovinné a v odpovědi se nepoužívá. Podpora jednotlivých měn je závislá na druhu a nastavení terminálu. Měny jsou ve formátu dle normy ISO 4217. Každá tato měna je reprezentována tříciferným číslem. např. Koruna česká (CZK) - **203,** Euro (EUR) - **978**.

- **FID L | ZŮSTATEK**

  Pokud tuto funkci podporuje příslušný autorizační server je prostřednictvím tohoto pole vracen zůstatek na kartě. V požadavku se nepoužívá. Tento FID nabývá různé délky od jednoho po osmnáct bytů.

- **FID g | ZPRÁVA SERVERU**

  Pole tohoto FIDu obsahuje dodatečný text v případě zamítnutí transakce autorizačním serverem. U požadavku se nepoužívá. V odpovědi pak obsahuje znaky pouze bez diakaritiky o proměnlivé délce.

- **FID S | VARIABILNÍ SYMBOL**

  Pomocí tohoto pole lze platebnímu terminálu zaslat hodnotu variabilního symbolu. Jedná se o volitelný FID o proměnlivé délce jednoho až deseti bytů. Variabilní symbol se používá pro jednoznačné označení transakce. Následně lze toto číslo najít ve výpisu transakcí od banky.

- **FID R | NÁVRATOVÝ KÓD**

  Toto pole je velmi důležité, protože obsahuje návratové kódy z platebního terminálu a z autorizačního serveru při transakci. Návratový kód nese informace o výsledku provedené transakce na platebním terminále. Pokud FID R obsahuje tři číslice, jedná se o návratový kód, který přišel ze strany autorizačního serveru. V případě, že návratový kód začíná znakem "-" za kterým následují dvě číslice, jedná se o návratový kód platebního terminálu. V požadavku se nepoužívá a v odpovědi je povinný o statické délce tří bytů.

NÁVRATOVÉ KÓDY SERVERU			
Potvrzovací kódy			
000	Autorizováno online	001 - 010	Autorizováno
Zamítací kódy			
050	Obecné zamítnutí	063	Timeout
051	Chyba spojení	100	Nepovolená transakce
052	Překročen počet limitů při zadávání PIN	100	Nepodporovaná transakce
053	Připojeno – chyba požadavku	101	Neplatná karta
060	Zrušeno obchodníkem	102	Prošlá platnost karty
061	Zrušeno zákazníkem	103	Chyba formátu dat
062	Zákazník nepotvrdil částku		
NÁVRATOVÉ KÓDY PLATEBNÍHO TERMINÁLU			
-01	Normální ukončení	-12	Nedostatek peněz
-02	Nemohu provést	-13	Příliš vysoká částka
-03	Nejsou definovány parametry karty	-15	Chyba servisního kódu
-04	Chyba v parametrech karty	-16	Chyba klávesnice
-05	Obecná chyba	-17	Chyba transportního klíče
-06	Nemohu se dovolat	-18	Timeout
-07	Neznámý typ karty	-19	Chyba autorizace
-08	Nemám oprávnění	-21	Interní chyba
-10	Prošlá platnost karty	-22	Transakce zamítnuta
-11	Karta zatím nevešla v platnost	-29	Karta blokována

Tabulka 7: Návratové kódy serveru a platebního terminálu

- **FID T | TYP TRANSAKCE**

  Toto pole obsahuje kód typu transakce. Jednotlivé kódy jsem sepsal do tabulky níže. Tento FID je povinný v požadavku o pevné délce dvou bytů a v odpovědi se toto pole opakuje.

Transakce	Hodnota	Odpověď musí obsahovat FIDy
Normální transakce	**00**	F, R, P, T
Handshake	**95**	R, T
Get App info	**80**	R, T
Passivate	**81**	R, T
Vrať poslední transakci	**82**	R, T

Tabulka 8: Kódy transakce

  V tabulce nejsou uvedeny všechny existující kódy typů transakce, jelikož nejsou pro realizaci této práce potřebné.

- **FID i | SÉRIOVÉ ČÍSLO TRANSAKCE**

  Toto pole lze použít pro identifikaci transakce. Pole s sebou nese jedinečné sériové číslo platební transakce, které je generováno platebním terminálem. V požadavku se nepoužívá. V odpovědi má pak pevnou délku devět bytů. Těchto devět bytů se skládá ze tří polí o třech bytech. První z nich je tzv. „Shift číslo“, které se zvyšuje po každé tisícáté uzávěrce. Další pole zvané „Batch číslo“ se zvyšuje po každé uzávěrce. A pole poslední tzv. „ Sequence číslo“ se zvyšuje po každé transakci a při uzávěrce je nulováno.

Kromě těchto zmíněných FIDů obsahuje komunikační protokol B ještě další FIDy. Ty, ale nejsou pro moje použití významné.

Dále zde zmíním některé servisní příkazy FIDu T.

7.4.3 Servisní příkazy FIDu T

- **Get Application Info (T80)**

  Jak už sám název napovídá, tento kód FIDu T slouží k identifikaci aplikace

v platebním terminále. Dále také obsahuje identifikační číslo platebního terminálu (Pos ID). Žádné další užitečné informace neobsahuje, a proto je vhodné jej použít před každou provedenou transakcí, aby se otestovala konektivita s platebním terminálem.

Požadavek:

```
0x0000:  02 42 31 30 31 20 20 20 20 20 20 20 20 31 38 30  =  '.B101          180'
0x0010:  37 31 33 31 35 31 30 32 31 30 30 30 30 30 30 30  =  '7131510210000000'
0x0020:  34 41 35 41 35 1c 54 38 30 03                    =  '4A5A5.T80.       '
```

Obrázek 11: Požadavek FIDu T80

Odpověď:

```
0x0000:  02 42 32 30 31 54 45 53 54 31 32 36 32 31 38 30  =  '.B201TEST1262180'
0x0010:  37 31 33 31 35 31 30 32 35 30 30 30 30 30 30 30  =  '7131510250000000'
0x0020:  46 41 35 41 35 1C 52 30 30 30 1C 67 56 3A 34 2E  =  'FA5A5.R000.gV:4.'
0x0030:  34 2E 31 34 03                                   =  '4.14.            '
```

Obrázek 12: Odpověď FIDu T80

- **Passivate (T81)**

  Tento kód umožňuje platební terminál uvést do pasivního režimu a to ve chvíli, kdy už byla zahájena platební transakce a platební terminál vyčkává na vložení platební karty. Tímto příkazem je pokladna schopna zjistit např. že se platební terminál zacyklil.

- **Vrať poslední transakci (T82)**

  Pomocí toho kódu FIDu T je platební terminál schopen opakovaně získat transakční data z poslední provedené transakce, která na něm proběhla. Platební terminál odpoví stejně jako při minulé uskutečněné platbě. Tzn., že může vrátit odpověď s nějakou chybou nebo návratový kód, který znamená, že transakce proběhla v pořádku.

  Pro případ, že by došlo ke ztrátě spojení mezi řídící jednotkou a platebním terminálem, je možné uplatnit právě tento příkaz. Řídící jednotka si může

vytáhnout data z poslední provedené transakce z platebního terminálu a porovnat navrácená datová pole s částkou, autorizačním kódem, jedinečným sériovým číslem transakce, popř. variabilním symbolem se stávající transakcí.

- **Handshake (T95)**

  Díky tomuto příkazu lze ověřit spojení mezi platebním terminálem a autorizačním serverem. Jedná v podstatě o vyvolání stejné akce, jako když obchodník fyzicky na platebním terminále vyvolá v nastavení akci "Test spojení". Své uplatnění nachází tento příkaz v každodenním spuštění, čímž si můžeme ověřit dostupnost autorizačního serveru.

7.4.4 Povinnost FIDů v příkazech

Každý příkaz, který je odesílán z řídící aplikace nebo přijímán z platebního terminálu, musí obsahovat FID T (typ transakce).

Dle typu zvolené transakce musí tento příkaz FID v požadavku či odpovědi obsahovat ještě další FIDy, a to buď povinně či volitelně. Příklad pro normální transakci:

Normální transakce (T00)	
Význam	Příkaz k provodení platby
Povinné FIDy v požadavku B1	T, B
Volitelné FIDy v požadavku B1	S
FIDy v odpovědi B2	T, P, F, R, g

Tabulka 9: FIDy u normální transakce (T00)

7.4.5 Průběh platební transakce

V prvním kroku je řídící aplikací zaslán požadavek B1, který obsahuje finanční transakci do platebního terminálu. Po dobu pěti sekund pak řídící aplikace vyčkává na přijetí potvrzení od platebního terminálu, že tuto zprávu obdržel.

Ihned po příjmu požadavku B1 od řídící aplikace, platební terminál potvrdí příjem tohoto požadavku a odešle zpět do řídící palikace zprávu B0.

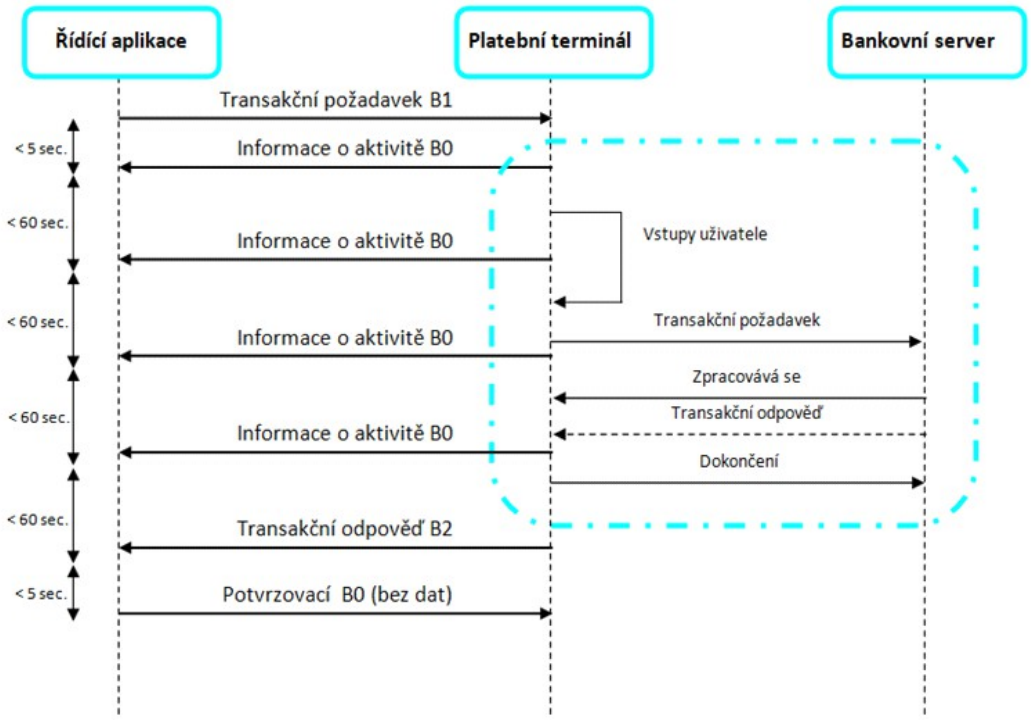
Po tom, co řídící aplikace obdržela zprávu B0, vyčkává na přijetí další zprávy

z platebního terminálu B0 nebo B2. Zároveň v tuto chvíli probíhá na platebním teminále čtení platební karty, zadávání PIN kódu a je zahájena komunikace s autorizačním serverem. Během této činnosti je řídící aplikací zasílána zpráva B0 v časové periodě šedesáti sekund. Tzn., že spojení je neustále aktivní a jsou zpracovávány vstupy ze stran uživatele nebo nadále probíhá komunikace s autorizačním serverem. Ve chvíli, kdy je činnost (autorizace) na platebním terminál ukončena, je řídící aplikaci zaslána zpráva B2.

V okamžiku, kdy řídící aplikace obdrží zprávu B2, musí příjem této zprávy potvrdit zasláním prázdné zprávy B0 do platebního terminálu, a to v intervalu do pěti sekund.

Poslední krok lze ovlivnit mechanismem vynuceného potvrzení. Tento mechanismus funguje následovně: pokud řídící aplikace obdrží od platebního terminálu zprávu B2 s výsledkem provedené transakce, a následně toto přijetí nepotvrdí zprávou B0 do pěti sekund od obdržení zprávy B2, platební terminál provede tzv. reversal. Znamená to, že celou tuto transakci anuluje. Celý mechanismus se nastavuje v hlavičce požadavku B1 v poli Flag na patnáctém bitu. Pokud je bit nastavený na hodnotu "1", je následně toto nastavení zopakováno v odpovědi B2 od platebního terminálu. [8]

Celý postup je znázorněn na obrázku č. 13.



Obrázek 13: Průběh finanční transakce

8 Řídící aplikace

Tato kapitola popisuje výběr vývojového prostředí, základní pohledy celé aplikace, průběh platební transakce. Poslední část této kapitoly popisuje watchdog.

8.1 Vývojové prostředí

Jako programovací jazyk jsem zvolil Node.js, proto bylo potřeba vybrat vhodné vývojové prostředí pro snadné a jednoduché psaní celé aplikace. Zde se nabízí celá řada editorů jako jsou např. PSPad, Visual Studio Code, Atom, Sublime Text, WebStorm, PHPStorm atd. Vzhledem ke svým zkušenostem s IDE WebStorm a možností využívat plné licence, kterou TUL nabízí svým studentům užívat zdarma, jsem se rozhodl použít právě tento SW.

8.1.1 WebStorm

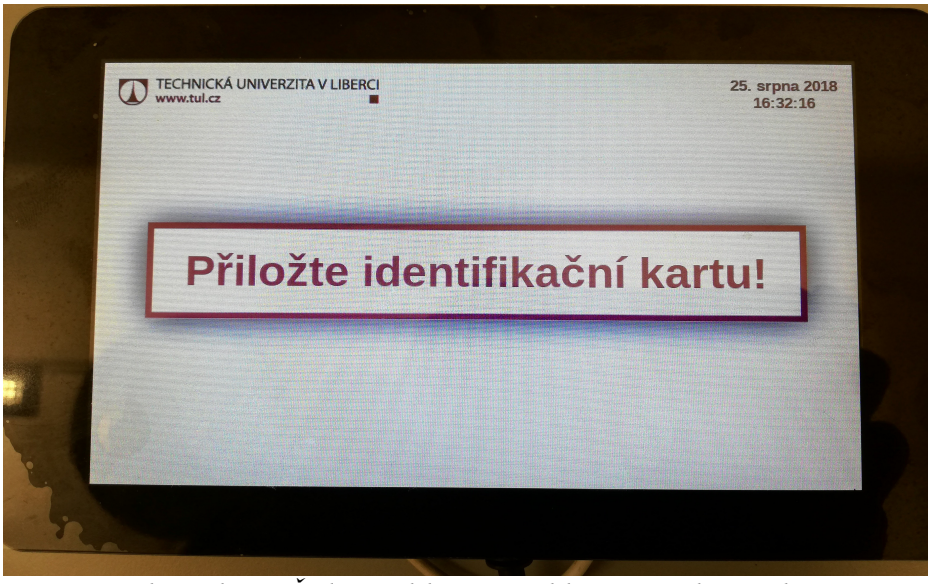
WebStorm je pokročilý editor pro velké množství programovacích a značkovacích jazyků (HTML, CSS, JavaScript apod.) od společnosti JetBrains. Díky tomu, že tento editor rozumí syntaxi kódu, je schopen tento kód automaticky doplňovat. Poradí si též i s opravou a doplňováním při kombinování jazyků (např. HTML kód uvnitř JavaScript řetězce). K dispozici má také různé integrované nástroje jako terminál, REST klient nebo npm a nepostrádá ani inteligentní opravu špatně napsaného kódu. Podporuje spoustu jazyků včetně HTML5, Node.js, TypeScript, CoffeeScript, Dart, EJS, Handlebars, Mustache, Web Components, Stylus, LESS, Sass, Jade či JSLint/ JSHint. [9]

8.2 Základní obrazovky (pohledy)

Celá aplikace se skládá z několika základních pohledů, které se v důsledku činnosti mění. Veškeré obrazovky designově korespondují se stránkami TUL, tzn. fialová barva, logo atd.

- **přihlašovací obrazovka**

  Tato obrazovka se automaticky objeví při spuštění a naběhnutí celého dobíjecího terminálu. V levém horním rohu obsahuje logo TUL, dále zde má uživatel přehled o aktuálním datu a času. Hlavní funkcí tohoto pohledu je informovat uživatele o přiložení identifikační karty. V tomto okamžiku řídící aplikace čeká na tomto pohledu dokud není přiložena identifikační karta ke čtečce.

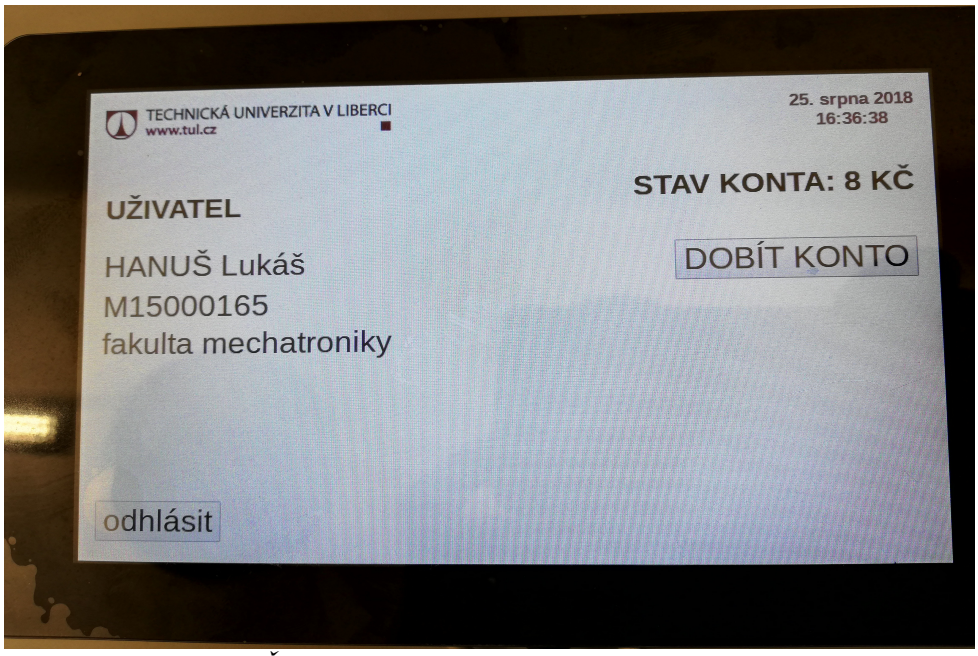


Obrázek 14: Řídící aplikace - přihlašovací obrazovka

- **obrazovka s profilem strávníka**

  Po přiložení platné identifikační karty se na terminále objeví obrazovka, která obsahuje základní údaje o strávníkovi jako jméno, příjmení, fakulta, osobní číslo, aktuální stav konta. Tato obrazovka je hlavní, a tvoří zároveň pozadí pro další pohledy u nichž se pouze mění informace v popředí. Dále má na této obrazovce uživatel možnost stisknout tlačítko **„DOBÍT“** a tím se tak posunout na další pohled. V případě, že by se uživatel rozhodl pouze zkontrolovat svůj

aktuální stav konta, a dále ho nijak nenavyšovat, obsahuje tato obrazovka i tlačítko **„ODHLÁSIT UŽIVATELE“**.



Obrázek 15: Řídící aplikace – obrazovka s profilem strávníka

- **dobíjecí obrazovka**

  Tato obrazovka je stejná jako předchozí s tím rozdílem, že v popředí se zobrazí pohled s číselnou klávesnicí od 0 do 9, tlačítkem **„SMAŽ“, „DOBIJ“** a **„ZPĚT“** Na tomto pohledu si uživatel může za pomoci dotykového displeje zvolit částku o kterou chce své konto navýšit.

Obrázek 16: Řídící aplikace - dobíjecí obrazovka

- **obrazovka s posíláním požadavku na platební terminál**

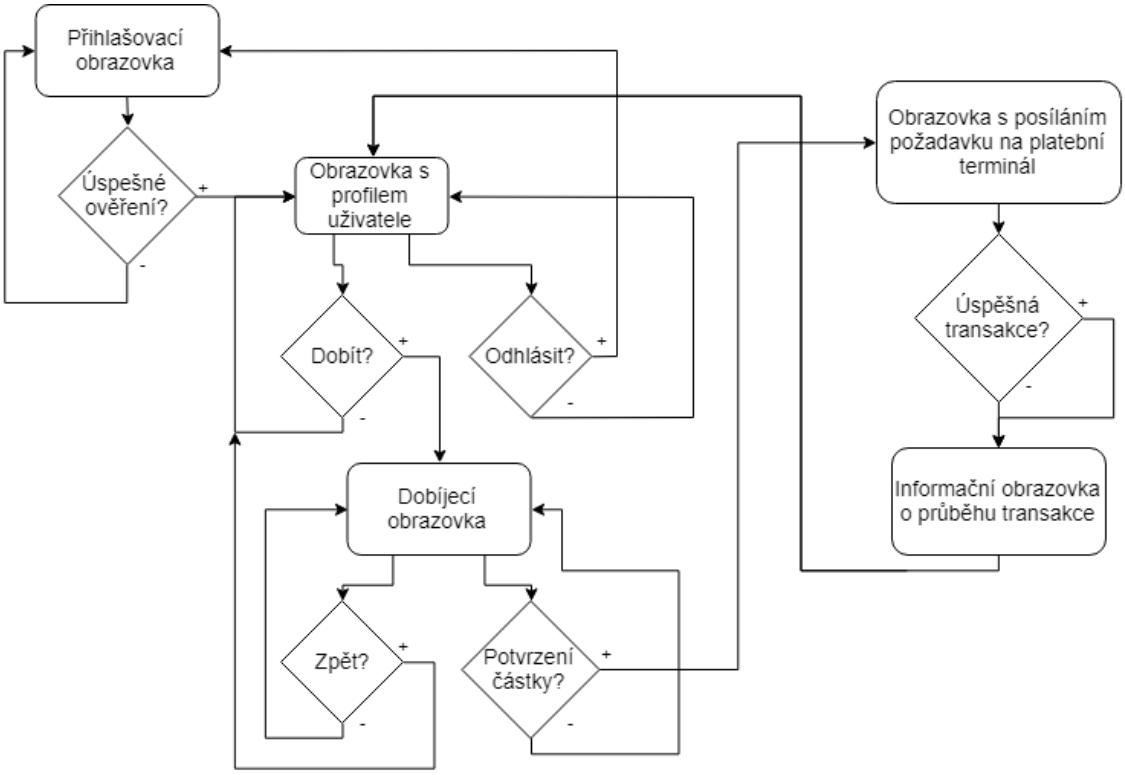
  Po stisku tlačítka **„DOBIJ“** na dobíjecí obrazovce se zobrazí právě tento pohled, který uživatele informuje o tom, že jeho požadavek byl odeslán na platební terminál a ať dále postupuje dle instrukcí na platebním terminále.

- **informační obrazovka o průběhu transakce**

  Tato obrazovka se zobrazí po tom, co z platebního terminálu přijde informace op rovedení či neprovedení platby.

Z důvodu bezpečnosti a ochrany osobních údajů je aktuálně přihlášený uživatel po době nečinnosti 30s automaticky odhlášen. V tu chvíli se opět zobrazí úvodní přihlašovací obrazovka.

Všechny obrazovky byly vytvořeny pomocí HTML a nastylované za pomoci kaskádových stylů. Na obrázku č. 17 je zobrazený vývojový diagram jednotlivých obrazovek.



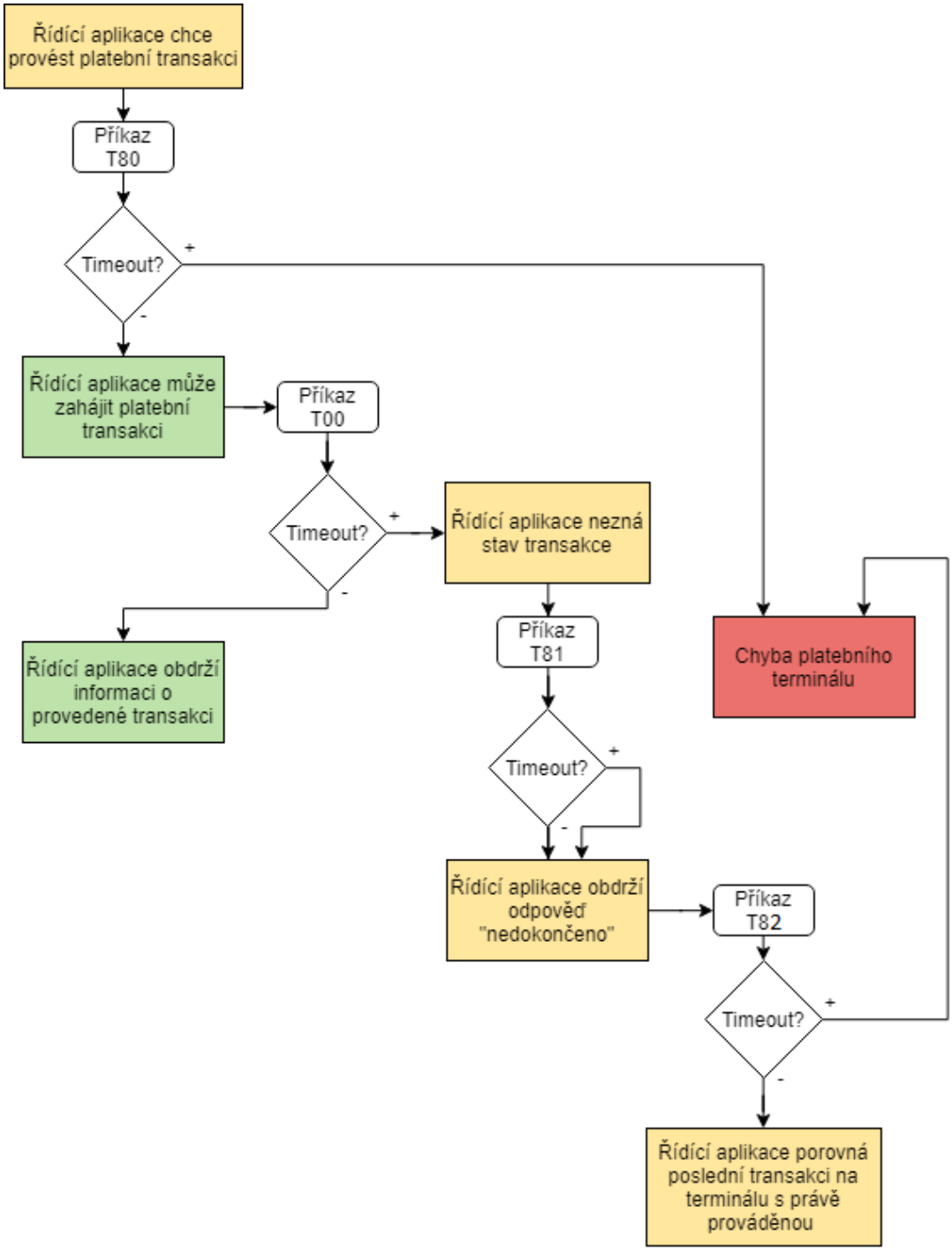
Obrázek 17: Vývojový diagram základních obrazovek řídící aplikace

8.3 Průběh platby

Ve chvíli, kdy má řídící aplikace připravenou platbu, zjistí stav platebního terminálu tím, že mu zašle příkaz T80. Tento příkaz slouží k ověřování dostupnosti autorizačního serveru. V případě, že tento příkaz selže, dostává se platební terminál do nekorektního stavu. Řídící aplikace potom informuje uživatele, že je platební terminál mimo provoz a platbu nelze uskutečnit. V opačném případě řídící aplikace připraví danou platbu a zašle ji do platebního terminálu příkazem T00. Nedostane-li během provádění platby řídící jednotka korektní odpověď znamená to, že řídící aplikací nezná stav platby. Příčinou může být ztráta příkazu požadavku na platbu, nebo mohla být platba uskutečněna a došlo ke ztrátě odpovědi od platebního terminálu. V tuto chvíli je nutné zjistit aktuální

stav platebního terminálu a jeho poslední uskutečněnou transakci. Je-li tomu však naopak a řídící jednotka obdrží korektní dopověď s hodnotou návratového kódu, je transkce dokončena a řídící jednotka se podle návratového kódu zachová.

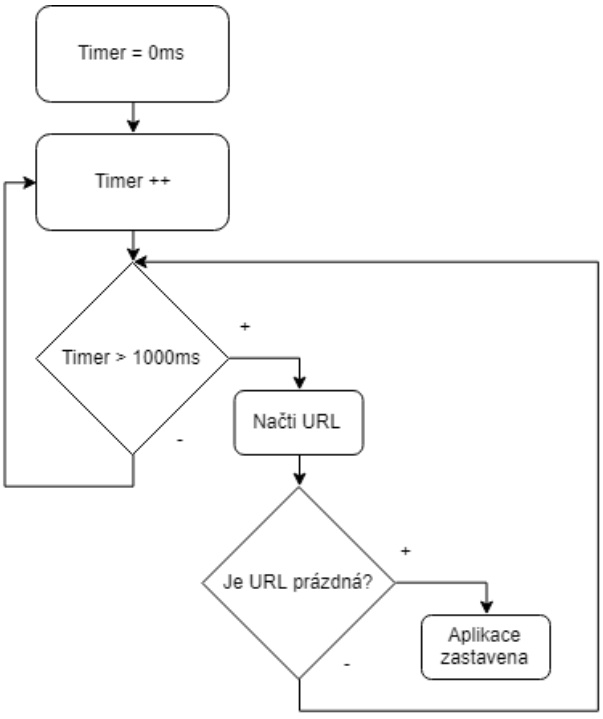
Pro zjištění stavu probíhající transakce na platebním terminále, zašle řídící jednotka příkaz T81, kterým řídící aplikace zjistí, zda-li se platební terminál nezacyklil ve smyčce při čekání na protažení platební karty nebo zákaznický vstup. Následně je zavolám příkaz T82 (Vrať poslední transakci), pomocí něhož lze zjistit informace o poslední prováděné transakci a jejím výsledku na platebním termínále. Pokud v tento okamžik nastane timeout, jedná se o poruchu na komunikační lince mezi řídící aplikací a platebním terminálem. Uživatel obdrží od řídící jednotky informaci o poruše platebního terminálu a transakci tak nelze uskutečnit. V případě, že timeout nenastane a data poslední transakce jsou k dispozici, řídící aplikace porovná obdržená data s aktuálně řešenou transakcí. Výsledkem mohou být dvě situace. Shodují-li se data obou transakcí a výsledek provedené transakce je uveden v návratovém kódu operace T82. Z tohoto vyplývá, že řídící aplikace se zachová jako při klasické platební transakci. Jestliže se data neshodují znamená to, že na platební terminál požadavek z řídící aplikace vůbec nedorazil a platba nebyla provedena. Nyní se řídící jednotka pokusí o znovunavázání spojení a o uskutečnění platby. Celý tento proces je znázorněn na následujícím vývojovém diagramu.



Obrázek 18: Vývojový diagram průběhu platby

8.4 Watchdog

Pro funkčnost celé aplikace je nezbytný internet, a proto bylo zapotřebí toto připojení nějakým způsobem hlídat. Za tímto účelem jsem použil timer, který je nastaven na jednu sekundu. Principem testování připojení k internetu je vytvoření objektu obrázku do něhož se každou vteřinu vkládá URL. Uniform Resource Locator se skládá z adresy obrázku (zde jsem zvolil logo z www.google.com). Tento webový server jsem zvolil jako výchozí z toho důvodu, že dle mého názoru patří mezi ty nejstabilnější, a tudíž šance, že by byl server nedostupný je minimální. Jako druhý parametr jsem použil aktuální čas, který je tam z důvodu toho, aby si prohlížeč nemohl adresu obrázku nakešovat, tzn. každou sekundu je adresa unikátní. Nastane-li případ, že řídící software neobdrží adresu obrázku, upozorní uživatele chybovou hláškou, kterou zašle do webového prohlížeče a následně celý chod aplikace zastaví. Průběh testování připojení internetu je znázorněn na následujícím vývojovém diagramu.



Obrázek 19: Vývojový diagram watchdog

9 Závěr

Cílem této diplomové práce byla realizace prototypu samoobslužného zařízení umožňujícího bezkonkontaktně a bezhotovostně dobíjet stravovací konto v univerzitních menzách.

Celá práce navazovala na můj semestrální projekt, který se zaměřoval na teoretický návrh a průzkum stávajících možností dobíjení stravovacího konta. Dalším aspektem vzniku této diplomové práce byly pozitivní ohlasy studentů v průzkumném šetření. V tomto šetření jsem zjišťoval, zda-li by byl o tento nový způsob dobíjení zájem.

Na začátku celé práce jsem si stanovil patřičná kritéria samotného zařízení. Na základě mého semestrálního projektu jsem zvolil vhodné hardwarové vybavení, aby byl prototyp finančně nenáročný, rozměrově přívětivý a pohodlně obsluhovatelný.

Celá práce z počátku vypadala poměrně jednoduše, avšak první komplikace se dostavily bohužel již u samotného oživení celého hardwaru, kde byl problém ve slabém zdroji elektrické energie.

V prvním části jsem se věnoval autentizací zaměstnanců a studentů TUL do stravovacího systému. Při tomto ověřování byl kladen velký důraz na bezpečnost, proto jsem se musel seznámit s generováním privátního a veřejného klíče a vhodně je uplatnit tak, aby byla zajištěna vysoká bezpečnost přenášených dat. I zde nastaly potíže. Čtečka identifikačních karet ISIC/ ITIC nekomunikovala s řídící aplikací, protože měla obsahovat čip typu ACR122U, ale obsahovala čip typu ACR38. Potíže jsem vyřešil výměnou čtečky se správným čipem.

Druhá část práce byla zaměřena na samotnou komunikaci řídící aplikace s platebním terminálem. Tento krok s sebou nesl velkou režii v nastudování a pochopení komunikačního protokolu B, který byl nezbytný pro samotnou komunikaci. Bohužel pouhé studování nestačilo a probíhala častá telefonická komunikace s technickým odborníkem z ČSOB. Nebylo jednoduché přinutit platební terminál, aby začal komunikovat s řídící aplikací. Ve výsledku se ukázalo, že veškeré chyby nebyly pouze na mé straně, ale také na straně ČSOB. Nakonec se mi podařilo všechny nedostatky odstranit.

Před nasazením tohoto nového systému je nezbytné podrobit řídící aplikaci certifikačnímu procesu, kde třetí strana prozkoumá bezpečnost celého systému a následně k němu vydá certifikát nebo ho vrátí k úpravě.

Výsledkem práce je malý prototyp za nízké pořizovací náklady, který je připravený na provoz v menzách. Dle mých informací z poslední schůzky s panem Melzerem jsem se dozvěděl, že menza chystá další novou možnost, kterou lze stravovací konto dobíjet. Jedná se o online platební bránu. Tato metoda je také velice účinná a okamžitá, nese s sebou ale opět jednu nevýhodu, a to v podobě povolení online plateb u jednotlivých karet koncových zákazníků.

Nyní je na univerzitě jestli bude tento prototyp pro menzu přínosný a rozhodne se pro jeho nasazení do ostrého provozu. V takovém případě budou klíčové první dva měsíce provozu, dle kterých bude mít univerzita možnost shromáždit data o tom, kolik se uskuteční jednotlivých transakcí, a v jakých částkách. Na základě těchto informací se pak s bankou ČSOB stanoví podmínky nutné pro provoz platebního terminálu. Dále se zjistí zda-li by bylo výhodnější si platební terminál pronajímat či si zakoupit svůj vlastní, a v neposlední řadě se také dohodne výše procent, které bude nutné odvádět za každou uskutečněnou transakci.

Seznam použité literatury

[1] *Introducing the Raspberry Pi 3 B+ Single Board Computer* [online]. [cit.2018-08-20]. Dostupné z:

https://www.raspberrypi-spy.co.uk/2018/03/introducing-raspberry-pi-3-b-plus-computer/

[2] *RASPBERRY Pi Touch display 7"* [online]. [cit. 2018-08-20]. Dostupné z: https://www.alza.cz/raspberry-pii-touch-display-7-d4268133.htm?o=1

[3] Čtečka *NFC čipů ACR122U NFC* [online]. [cit. 2018-08-20]. Dostupné z: https://cardhouse.cz/cs/eshop/ctecky-snimace-karet/ctecka-nfc-cipu-acr122u-nfc

[4] *Bezkontaktní platební brána pro menzu TUL*. Liberec, 2018. Semestrální projekt. Technická univerzita v Liberci. Vedoucí práce Ing. Lenka Kosková Třísková Ph.D.

[5] *NOOBS* [online]. [cit. 2018-08-20]. Dostupné z: https://www.raspberrypi.org/downloads/noobs/

[6] *What Socket.IO is* [online]. [cit. 2018-08-20]. Dostupné z: https://socket.io/docs/

[7] *Protokol UDP* [online]. [cit. 2018-08-20]. Dostupné z: http://www.earchiv.cz/a93/a303c110.php3

[8] *Interní dokument: Specifikace komunikačního protokolu B* [online]. 2017 [cit. 2018-08-20].

[9] *WebStorm IDE* [online]. [cit. 2018-08-20]. Dostupné z: https://confluence.jetbrains.com/display/WI/WebStorm+IDE