



Střední průmyslová škola,
Česká Lípa, Havlíčkova 426, příspěvková organizace

tel.: 487 833 123
fax: 487 833 101
email: sps@sps-cl.cz
web: www.sps-cl.cz

MATURITNÍ PRÁCE

EVIDENČNÍ SYSTÉMY

Studijní obor: **18-20-M/01 INFORMAČNÍ TECHNOLOGIE**

Autor:

Jakub Sirový

Podpis:

Vedoucí práce:

Mgr. Jan Fojtík

Třída: **4.D**

Školní rok: **2021/2022**



Zadání práce – vložený list (první výtisk práce originál, druhý výtisk práce kopie)
se zadáním podepsaný vedoucím práce a ředitelem školy.

Osnova maturitní práce

Jméno Příjmení: *Jakub Sirový*

Třída: 4.D

Téma: Evidence přístupů

Osnova:

1. Úvod
2. Teoretická část
 - a. Serverové technologie
 - b. Hardware
 - c. Detekce signálu
3. Praktická část
 - a. Návrh zařízení
 - b. Realizace
 - c. Ověření funkčnosti
 - d. Spolehlivost a bezpečnost
4. Závěr – zhodnocení

Podpis žák:

Vedoucí práce: Mgr. Jan Fojtík

Podpis VP:

V České Lípě 3.12.2021

Licenční ujednání

Ve smyslu zákona č. 121/2000 Sb., O právu autorském, o právech souvisejících s právem autorským, ve znění pozdějších předpisů (dále jen autorský zákon) jsou práva k maturitním nebo ročníkovým pracím následující:

Zadavatel má výhradní práva k využití práce, a to včetně komerčních účelů.

Autor práce bez svolení zadavatele nesmí využít práci ke komerčním účelům.

Škola má právo využít práci k nekomerčním a výukovým účelům i bez svolení zadavatele a autora práce.

Prohlášení

Prohlašuji, že jsem svou maturitní práci vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze práce jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu s autorským zákonem.

V České Lípě dne 21.4.2022

.....

Jméno a příjmení autora

Poděkování

Rád bych poděkoval vedoucímu práce Mgr. Janovi Fojtíkovi za pomoc a vstřícné rady a panu řediteli Ing. Petrovi Veselému za pomoc a nápady.

Anotace

Tato práce se zabývá evidenčními systémy, internetem věcí a jejich vzájemným využitím. Práce shrnuje návrh a výrobu vlastního zařízení. Cílem této práce je přiblížit využití a implementaci internetu věcí spolu se systémem evidence širokému spektru lidí.

Klíčová slova

Evidenční systémy; Internet věcí; Programování

Annotation

Evidence system; Internet of Things; Programming

Keywords

This work deals with evidence systems, the Internet of Things and their mutual use. The work summarizes the design and manufacture of its own equipment. The aim of this work is to bring the use and implementation of the Internet of Things together with the evidence system to a wide range of people.

Obsah

Úvod	8
1 Evidenční systémy	9
1.1 Internet věcí	9
2 Serverové technologie.....	10
2.1 Apache	10
2.2 Nginx	11
2.3 SSH Server.....	12
2.4 SFTP Server	13
2.5 Sematext monitoring	14
2.6 ServerCat	14
2.7 Zálohovací software	14
3 Vjezdová brána.....	15
3.1 Aktuální stav	15
3.2 Požadavek úpravy brány.....	15
3.3 Návrh řešení projektu	16
3.4 Sít'ové řešení	17
3.5 Hardwarové řešení ovládání brány.....	18
4 Ovládací zařízení	21
4.1 Relé.....	21
4.2 NodeMCU V3	23
5 Webová aplikace	24
5.1 PHP.....	24
5.2 MySQL	27
5.3 HTML a CSS	29
6 Závěr.....	34
7 Použitá literatura.....	35
8 Seznam obrázků a tabulek.....	36

Použité zkratky, značky a symboly

U – elektrické napětí,	[U] = V
Modulační rychlost	Bd = Baud rate
Hodinová linka	SCL = Serial clock
Datová linka	SDA = Serial data
Internet věcí	IoT
Řídící jednotka	ŘJ
Jednotný lokátor zdroje	URL = Uniform Resource Locator
Adresa Internetového protokolu	IP = Internet Protocol
Přenosový protokol	TCP = Transmission Protocol
Zabezpečený komunikační protokol	SSH = Secure Shell Protocol
Protokol přenosu souborů	FTP = File Transfer Protocol
Protokol zabezpečeného přenosu souborů	SFTP = Secure File Transfer Protocol
Protokol zabezpečeného přenosu souborů	SCP = Secure Copy Protocol
Protokol přenosu hypertextových souborů	HTTP = Hypertext Transfer Protocol
Protokol zabezpečeného HTTP	HTTPS = Secured Hypertext Transfer Protocol

ÚVOD

Evidenční systémy spolu s Internetem věcí jsou záležitosti, které nás denně obklopují a usnadňují nám život. S příchodem internetu věcí se každodenní věci stávají „chytrými“. Díky těmto systémům ulehčujeme nejen práci sobě, ale i lidem okolo nás a zároveň jsou to systémy, které dohlíží na naši bezpečnost.

Využití internetu věcí je velice rozsáhlé a často je spojováno právě s evidenčními systémy. Díky této vymoženosti můžeme spojovat téměř cokoli s online světem a ovládat různé věci například ze svého telefonu. Díky těmto systémům si například ze své kanceláře můžete zkontrolovat, zdali není otevřená garáž firmy a kdo ji zapomněl zavřít.

V této práci vás seznámím s evidenčními systémy a internetem věcí, které společně tvoří „chytrou“ bránu.

1 EVIDENČNÍ SYSTÉMY

Evidenční systém neboli docházkový systém je systém sloužící k záznamu příchodu nebo odchodu lidí, či jejich aktivit, které můžeme ukládat do databází a následně pak tato data zpracovávat. Tato práce volně navazuje na ročníkovou práci z minulého roku, ve které jsem se věnoval internetu věcí (IoT). V mé práci je tento systém využit s dalšími systémy, které dohromady představují „chytrou“ bránu. Díky evidenčnímu systému může správce objektu přehledně vidět, kdo a kdy tuto bránu otevřel nebo zavřel. Zásluhou evidenčních systémů jsou naše životy jednodušší a tyto systémy nám usnadňují každodenní práci. Evidenční systémy nám pomáhají zabezpečovat objekty, ale dokonce i hlídat zaměstnance ve výrobní hale.

Evidenční systém je komplexní pojem, pod kterým se skrývá mnoho dalších podpůrných systémů. Jako příklad můžeme uvést umělou inteligenci, kamerové systémy, mikrokontrolery, které tyto systémy ovládají, ale například i již zmíněný internet věcí.

Své uplatnění si evidenční systémy najdou zejména ve firmách. Podstatné jsou pro firmy docházkové systémy, díky kterým mají firmy přehled o docházce jejich zaměstnanců. Každý zaměstnanec musí mít svoji zaměstnaneckou kartu, díky které prokáže svůj příchod do práce.

1.1 Internet věcí

Internet věcí umožňuje zařízením, aby byla zjištěna či vzdáleně kontrolována v rámci sítě, ve které se toto zařízení nachází (internet, intranet, síť budovy apod.). Většinou se jedná o skupinu velmi malých zařízení například se senzorem a jednoho serveru, který se těchto zařízení dotazuje na data. Všechna data například o zůstatku toaletního papíru na všech veřejných toaletách ve městě přichází do jednoho zařízení, které je ukládá do databáze a postupně vyhodnocuje. Do tohoto zařízení mají klienti přístup a vidí zde vše přehledně pohromadě. Tento systém se využívá u sofistikovanějších uplatnění. Například u chytrých domů nebo chytrých měst. Další možností je použití pouze senzoru a zařízení, do kterého jsou data posílána (to bývají většinou chytré telefony s připojením k internetu), toto řešení se využívá zejména u nositelné elektroniky.

2 SERVEROVÉ TECHNOLOGIE

Velmi podstatnou roli zde hrají serverové technologie, které obstarávají komunikaci mezi zařízeními, přístup k ovládacím prvkům daného projektu, databázi a mnoho dalšího. Server jako takový je zařízení, které poskytuje své služby a výkon ostatním zařízením. Pokud bychom se na server chtěli podívat z fyzické stránky, tak zjistíme, že server představuje počítač obvykle s vysokým výkonem, který je velmi stabilní a běží nepřetržitě. Na tomto počítači najdeme programy, které server dělají serverem. Podíváme se spolu na pár základních, nejdůležitějších programů / služeb.

- Webové servery:
 - Apache
 - Nginx
- Komunikační a souborové servery:
 - SSH server
 - SFTP server
- Monitoring serveru
 - Sematext monitoring
 - ServerCat
- Zálohování serveru

2.1 Apache

Webový server (software)

Celým názvem Apache HTTP Server je softwarový webový server s otevřeným kódem pro GNU/Linux, BSD, Solaris, macOS, Microsoft Windows a další platformy. V současné době dodává prohlížečům na celém světě většinu internetových stránek.

Apache podporuje velké množství funkcí, mnoho z nich je implementováno jako kompilované moduly rozšiřující jádro. Mohou to být funkce podpory programovacích jazyků na straně serveru nebo různá autentizační schémata. Příkladem podporovaných programovacích jazyků je Perl, Python, Tcl nebo PHP. Autentizační schémata jako například `mod_access`, `mod_auth`, `mod_digest`, a `mod_auth_digest`. Příkladem dalších funkcí je podpora SSL, TLS (`mod_ssl`), proxy modul (`mod_proxy`), URL rewriter známý jako rewrite engine z modulu `mod_rewrite`, konfigurace souborů logu (`mod_log_config`) a filtrace (`mod_include` a `mod_ext_filter`).

Apache dále obsahuje externí modul pro kompresi dat webových stránek posílaných protokolem HTTP (mod_gzip), open source modul pro ochranu a prevenci webových aplikací před napadením (mod_security). Logy z Apache mohou být analyzovány pomocí browseru a skriptů jako AWStats/W3Perl nebo Visitors. Nastavit se dají i formy chybových zpráv, DBMS autentizační databáze a nechybí ani podpora mnoha grafických prostředí (GUI). [1]

Virtuální hosting je funkce dovolující jedné instalaci Apache na jednom fyzickém počítači obsluhovat více webových stránek.

Rád bych zmínil bezpečnostní zranitelnost, na kterou je náchylný právě Apache HTTP Server. Zranitelnosti využívá nástroj Slowloris. Typem útoku je odepření služby (DoS Attack), ale ne takové, jaké ho známe z útoků typu HOIC / LOIC (High Orbit Ion Cannon / Low Orbit Ion Cannon), které pracují na způsobu zahlcení serveru mnoha požadavky. Slowloris umožňuje pomocí jediného počítače vyřadit z provozu odmítnutím služby http server běžící na jiném počítači, přičemž mu k tomu stačí zcela minimální síťový provoz a útok neovlivní jiné služby.

Program je původně napsaný v jazyce Perl a jeho strategií je otevřít mnoho síťových spojení a udržet je otevřená co nejdéle. Toho dosáhne tak, že po otevření spojení pošle neukončený HTTP požadavek a pak postupně posílá další HTTP hlavičky tak, aby mezi požadavky nevypršel časový limit. Server bude držet spojení otevřená, ale může jich obvykle držet otevřený jen omezený počet – pak začne další požadavky na spojení odmítat a pro zbytek světa tak bude nedostupný. [2] Program byl záměrně pojmenován po outloních, skupině primátů, kteří jsou známí svým pomalým pohybem.

Na rozdíl od Apache serveru je Nginx server proti tomuto typu útoku chráněný.

V současné době je Apache druhým nejpoužívanějším webovým serverem, jeho zastoupení tvoří 31,3 % všech webů podle stránky W3Techs.com ke dni 4. dubna 2022.

2.2 Nginx

Webový server (software)

Nginx je softwarový webový server s load managment a reverzní proxy s otevřeným zdrojovým kódem. Pracuje s protokoly HTTP (i HTTPS), SMTP, POP3, IMAP a SSL. Zaměřuje se především na vysoký výkon a nízké nároky na paměť. Nginx je dostupný na Unixu, Linuxu a dalších Unix-like systémech pod BSD; existují varianty pro Solaris, macOS i MS Windows.

Základním cílem je rychlá distribuce statického obsahu a možnost rozložení zátěže na další servery dle nastavené priority. Systém dále umožňuje definovat záložní server, na který Nginx požadavek předá, pokud primární server neodpoví do stanoveného limitu. Příchozí požadavky Nginx asynchronně zpracovává a vyřizuje (na rozdíl třeba od Apache, který využívá vlákna nebo procesy). Velice častý způsob použití funguje na principu, že příchozí HTTP (nebo HTTPS) požadavek se nejprve pokusí vyhledat ve své cache (má konfigurovatelnou velikost a dobu uchovávání), pokud jej najde, rovnou odpoví. V opačném případě se obrátí na jeden z definované sady serverů (každý server má definovanou prioritu). Pokud mu server do definovaného času stihne odpovědět, předá odpověď; v opačném případě se obrátí na záložní server (samozřejmě je-li definován). Odpověď, pokud může, uloží do své cache a následující dotazy do vypršení časového limitu životnosti cache vyřizuje právě z cache. Jedním z nastavení je možnost nastavit limit počtu připojení z jedné IP adresy (jedna z nejjednodušších obran proti DOS útoku). Nginx je modulární systém a lze přidávat moduly (mnohdy je nutná rekompile). Jeden z modulů je například GEO lokace, která umožňuje například dle země předávat požadavky na definované servery nebo naopak zakázat přístup na stránky z některých zemí. Za zmínku určitě stojí také moduly pro přesměrování dle definovaných pravidel, zabezpečení stránky heslem, podpora komprese gzip, streaming (FLV, MP4) a mnoho dalších funkcí. [3]

V současné době je Nginx prvním nejpoužívanějším webovým serverem, jeho zastoupení tvoří 33,1 % všech webů podle stránky W3Techs.com ke dni 4. dubna 2022.

2.3 SSH Server

Server pro vzdálenou správu (software)

SSH (Secure Shell) je v informatice označení pro program a zároveň pro zabezpečený komunikační protokol v počítačových sítích, které používají TCP/IP. SSH byl navržen jako náhrada za telnet a další nezabezpečené vzdálené shelly (rlogin, rsh apod.), které posílají heslo v nezabezpečené formě a umožňují tak jeho odposlechnutí při přenosu pomocí počítačové sítě. Šifrování přenášených dat, které SSH poskytuje, slouží k zabezpečení dat při přenosu přes nedůvěryhodnou síť, jako je například Internet.

SSH umožňuje bezpečnou komunikaci mezi dvěma počítači, která se využívá pro zprostředkování přístupu k příkazovému řádku, kopírování souborů a též jakýkoliv obecný přenos dat (s využitím síťového tunelování).

Zabezpečuje autentizaci obou účastníků komunikace, transparentní šifrování přenášených dat, zajištění jejich integrity a volitelnou bezztrátovou kompresi. Server standardně naslouchá na portu TCP/22.

Označení „Secure Shell“ je mírně zavádějící, protože nejde ve skutečnosti o náhradu shellu ve smyslu interpret příkazů. Název byl odvozen z existujícího programu rsh, který má podobné funkce, ale není zabezpečený. [4]

Mezi software zprostředkující komunikaci pomocí SSH protokolu patří například OpenSSH server nebo TinySSH. Mezi klienty patří například software: PuTTY, WinSCP a Termius.

2.4 SFTP Server

Souborový server (software)

SSH File Transfer Protocol (zkratka SFTP) označuje v informatice protokol a zároveň i program pro bezpečný přenos souborů pomocí počítačové sítě. Jeho možnosti jsou obdobné jako u FTP a také je využíván jako náhrada za velmi jednoduchý protokol SCP. Pro vlastní přenos dat využívá SFTP (obvykle SSH-2), ale je navržen tak, aby ho bylo možné používat i nad jiným protokolem.

Protokol SFTP byl navržen pracovní skupinou IETF jako multiplatformní, takže například expanzi žolíkových znaků na rozdíl od protokolu SCP neponechává na implementaci v serveru. Protokol SFTP sám o sobě nezajišťuje ani autentizaci ani zabezpečení přenášených dat a obvykle k zajištění těchto služeb využívá protokol SSH-2. Je však navržen tak, aby mohl být použit i jakýkoliv jiný protokol. Například protokol SILC pro peer-to-peer sítě definuje SFTP jako svůj implicitní protokol pro přenos dat, avšak data nejsou zabezpečena protokolem SSH, ale je použit vlastní zabezpečený paketový SILC protokol.

Protokol SFTP je možné použít i nad protokolem SSH-1, avšak v tomto případě je narušena nezávislost na architektuře počítače. Protože protokol SSH-1 nepodporuje subsystémy, musí připojující se klient znát plnou cestu k programu SFTP serveru, aby si mohl sám spustit serverovou část.

Na rozdíl od jednoduchého protokolu SCP nabízí SFTP protokol široké možnosti pro doplňující operace se soubory (podobně jako FTP), takže bychom ho mohli označit i za jednoduchý vzdálený souborový systém. Umožňuje pokračovat v přerušených přenosech, vypisovat adresáře i odstraňovat soubory na vzdáleném počítači. [5]

2.5 Sematext monitoring

Monitoring serveru (software)

Každý stabilně fungující server by měl být aktivně monitorován, aby jeho vývojový tým byl schopný co nejdříve reagovat na jeho problémy. Jedna ze služeb nabízející monitoring serveru se nazývá Sematext (www.sematext.com) a nabízí kompletní monitoring serveru. Tato služba nabízí detekci anomálních jevů, oznámení o chybách, záznam chyb a spoustu dalších, služba je rozdělena na balíčky. Základní balíček je zdarma, pokud bychom chtěli monitorovat více serverů současně a zároveň by byla potřeba využít pokročilé funkce, je potřeba si koupit placený balíček.

2.6 ServerCat

Monitoring serveru (software)

ServerCat je aplikace, kterou bych v této práci rád zmínil a vyzdvihl. Tato aplikace je pouze pro mobilní telefony s operačním systémem iOS. Tato aplikace je ke stažení zdarma, ale obsahuje placené funkce. V aplikaci je velmi přehledně vidět souhrn serverů, které vlastníte, vytížení procesoru a další užitečné informace. V případě zakoupení plné verze pomocí této aplikace je možné ovládat váš server přes SSH, spravovat docker a spoustu dalšího.

Její hlavní výhodou je fakt, že na vašem serveru nemusí běžet žádná jiná aplikace kromě SSH serveru. Tuto aplikaci je možné použít pouze pro servery s operačním systémem Linux.

2.7 Zálohovací software

V neposlední řadě je podstatné zajistit zálohovací službu. Zálohování je proces, který vytvoří kopii důležitých souborů a v případě ztráty dat jsme díky tomuto procesu schopni ztracená data obnovit. Každý projekt a každá firma si žádá konkrétní řešení. Pojdme si představit nejznámější zálohovací služby: Acronis Cyber Protect Cloud, Acronis Cyber Backup, Veeam Backup & Replication, NAKIVO Backup & Replication, Barracuda Backup, Aiomei Backupper, IvyBackup, Ashampoo Backup, EaseUS Todo Backup a mnoho dalších...

3 VJEZDOVÁ BRÁNA

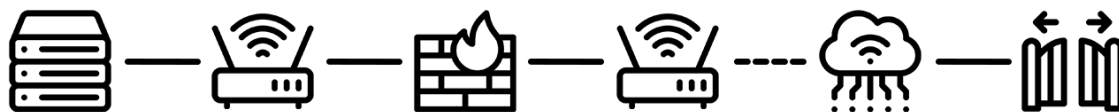
3.1 Aktuální stav

Aktuálně se brána určená pro vjezd aut do areálu jídelny používá tím způsobem, že zvolení zaměstnanci mají přidělený ovladač, kterým bránu stisknutím tlačítka otevřou. Tento stav byl nevyhovující vzhledem k tomu, že nebylo možné zjistit, kdo kdy bránu otevřel a zavřel. Pokud někdo ze zaměstnanců bránu zapomněl zavřít, tak nebylo možné zjistit, kdo přesně bránu zapomněl zavřít.

3.2 Požadavek úpravy brány

Požadavkem od ředitele školy bylo upravit bránu tak, aby se otvírala pomocí požadavku z mobilního telefonu / webové aplikace. Tato aplikace musí umět bránu nejen otevřít a zavřít, ale i zaznamenat kdo a kdy bránu zavřel či otevřel. V této aplikaci by každý uživatel měl mít roli (správce nebo uživatel). Díky těmto rolím bude určeno, co smí daný člověk v tomto systému provádět za operace. Mimo to musí být také možné upravit zvlášť přístup k jednotlivým branám pro konkrétního uživatele.

3.3 Návrh řešení projektu



Obrázek 1 – Schéma projektu

Základem pochopení tohoto projektu je představení všech zařízení, které jsou součástí tohoto projektu.

- Server
- Router
- Firewall
- IoT zařízení (NodeMCU V3)
- Vjezdová brána

Vzhledem rozsáhlosti oblastí, kterých se tato práce týká se vám budu co nejvíce snažit přiblížit všechny oblasti této práce pro představu komplexnosti řešení tohoto projektu. Postupně se v této práci dozvíte o řešení na úrovni sítě, poté si vysvětlíme řídicí jednotku, která se stará o ovládání brány, poté vám vysvětlím funkci mnou zkompletovaného zařízení, starajícího se o vysílání signálů pro řídicí jednotku vjezdové brány, a nakonec serverovou webovou aplikaci, která posílá požadavky na otevření brány včetně evidenčního systému, který monitoruje uživatele manipulující s bránou.

Vysvětlivky

	Server
	Wifi router
	Firewall
	IoT zařízení
	Vjezdová brána
	Vodič
	Bezdrátová komunikace

Obrázek 2 – Vysvětlivky

Pokusím se co nejstručněji přiblížit komunikaci mezi jednotlivými zařízeními a jejich funkcí.

Server → Vytvoří síťový požadavek a zaznamená akci

Router → Přijme požadavek ze serveru (paket)

Firewall → Tento síťový prvek povolí komunikaci na portu 80

Wifi router → Odešle paket (požadavek ze serveru) pomocí bezdrátové sítě

IoT zařízení → Přijme požadavek ze serveru a ověří jej, vygeneruje signál a sepne relé

ŘJ brány → Přijme signál a otevře nebo zavře bránu

Toto jsou chronologicky uspořádané procesy probíhající v pozadí mého projektu. S mnohými úkoly jsem si dlouhou dobu lámal hlavu, ale všechny problémy jsem v domácích podmínkách úspěšně vyřešil. Projekt je již funkční při připojení v rámci domácí sítě. O problémech, se kterými jsem se potýkal v rámci podnikové / školní sítě se dozvíte v následující podkapitole.

3.4 Síťové řešení

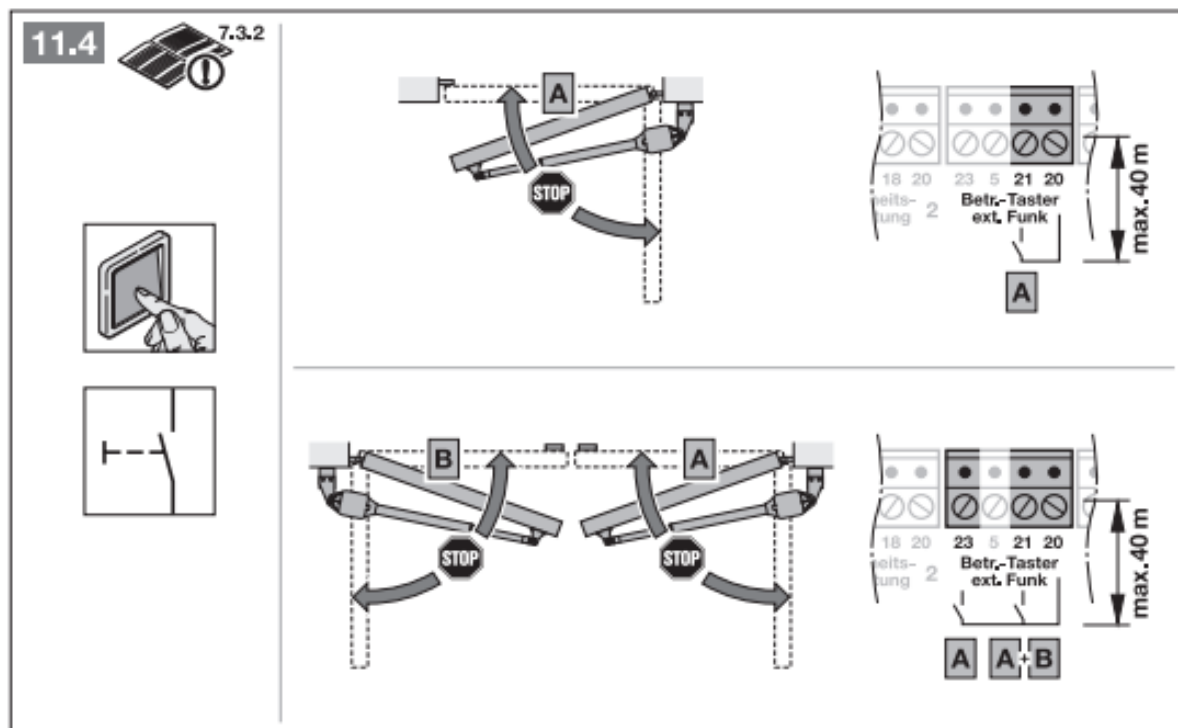
Další podstatnou částí této práce je řešení sítě. V části návrhu této práce bylo potřeba zjistit, jak k tomuto zařízení bude přistupováno. Po promyšlení a zohlednění bezpečnostních rizik jsem vyhodnotil, že bude nejlepším řešením umožnit přístup k zařízení pouze z lokální sítě školy (intranet). Bylo potřeba zjistit, jak je navržena infrastruktura sítě a poprosit vedoucího práce o pomoc s nastavením sítě za pomoci externí firmy starající se o školní síť.

V domácích testovacích podmínkách nebyly se sítí žádné problémy a pakety chodily přesně tak, jak jsem potřeboval. Všechny pakety, které komunikují mezi serverem a ovládacím zařízením řídící jednotky vjezdové brány (dále jen „NodeMCU V3“) chodily rychle a spolehlivě přes port 80 (HTTP). Vzhledem k velmi rozdílné topologii sítí (domácí a školní síť) je zapotřebí nakonfigurovat školní síť tak, aby všechny pakety chodily tam, kam mají. Aktuálním problémem je firewall, který blokuje a zahazuje pakety, které mají chodit do NodeMCU V3.

3.5 Hardwarové řešení ovládání brány

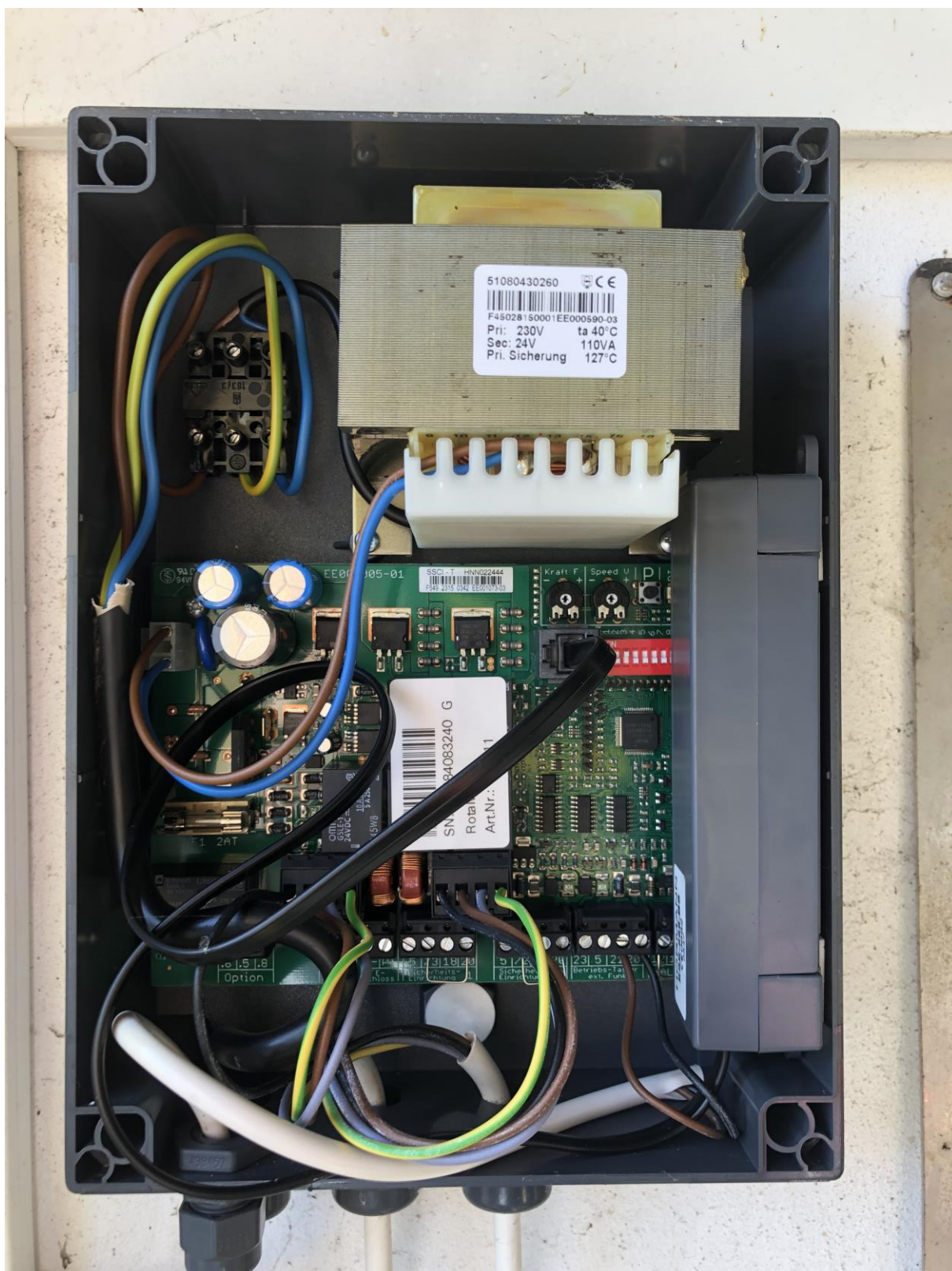
Jednokřídlá brána u jídelny školy je ovládána řídicí elektronikou Hörmann RotaMatic, který disponuje piny pro připojení externích ovladačů brány. Tyto piny je možné dohledat v manuálu výrobce.

[6]

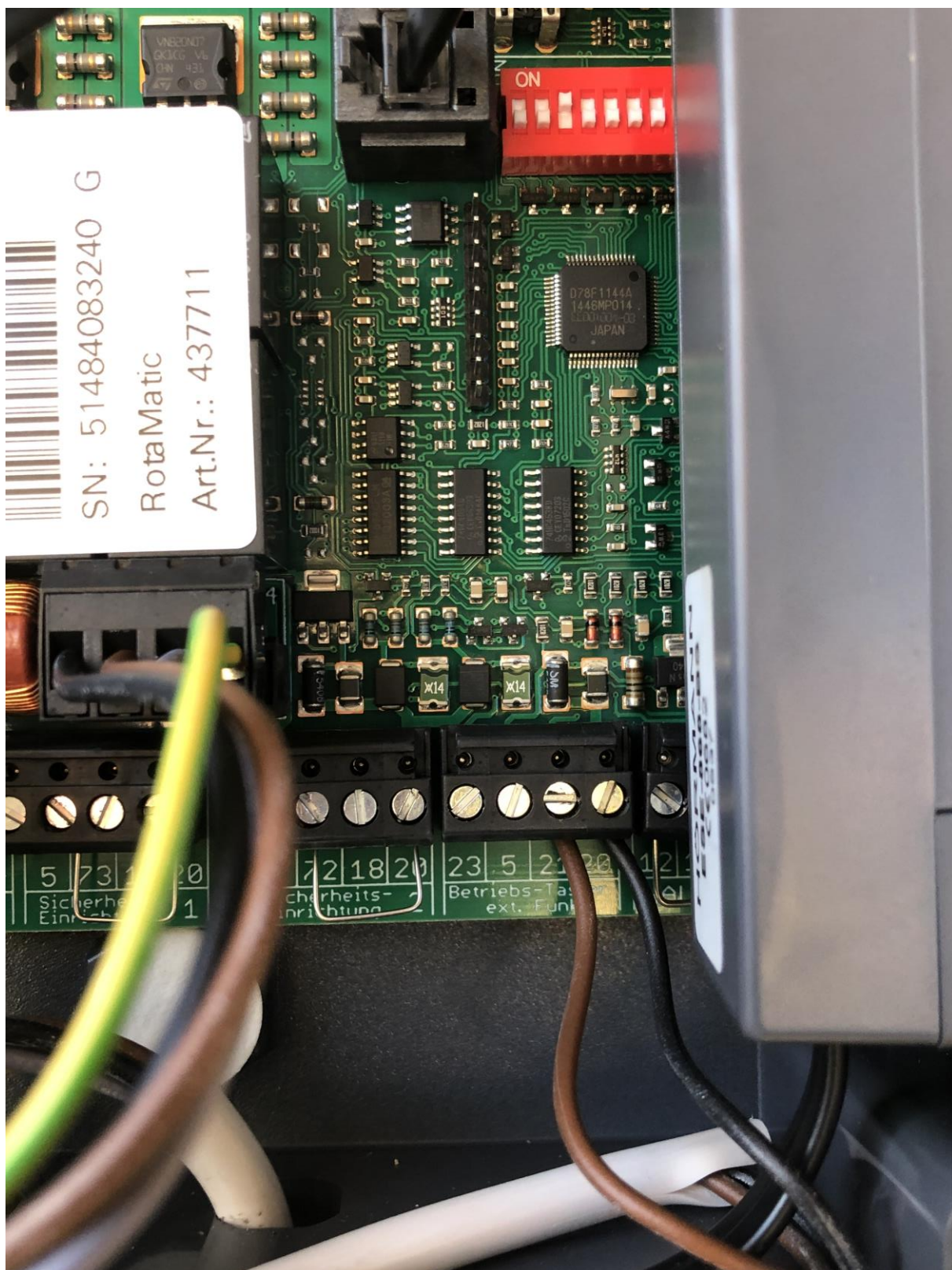


Obrázek 3 – Výstřižek z manuálu výrobce

Z manuálu je patrně čitelné, že ovládacími piny jsou piny označené čísla 20 a 21. K těmto pinům připojíme zařízení, o kterém se dozvíte více v následující kapitole. Tyto zařízení se prodávají jako set společně s motory, ovladači a dalšími podpůrnými prvky, které jsou součástí těchto bran. Mimo jiné se firma Hörmann zabývá i garážovými vraty a mnoho dalšími zařízeními. Svůj projekt jsem vymyslel tak, aby jej bylo možné připojit jak k bráně, tak i ke garážovým vratům. Na dalších dvou stránkách můžete vidět fotodokumentaci řídicí elektroniky Hörmann. Na druhé ze dvou fotek je možné vidět detail pinů řídicí jednotky. Povšimněte si dvou vodičů (hnědý a černý) zapojených do svorkovnice. Tyto dva vodiče jsou již připojené k pinům 20 a 21, je k nim totiž připojené externí uzamykatelný spínač pro otevření brány bez použití dálkového ovladače.



Obrázek 4 – Řídicí jednotka vjezdové brány



Obrázek 5 – Detail pinů řídicí jednotky

4 OVLÁDACÍ ZAŘÍZENÍ

V předchozí kapitole jsem se věnoval řídicí jednotce brány a způsobem jejího ovládání. Nyní vám představím zařízení, kterým budeme pro řídicí jednotku brány vytvářet elektronické signály, které laicky řečeno budou bráně říkat, kdy se má otevřít, a kdy zavřít.

Pokusím se vám popsat, jak toto zařízení bude fungovat. Základním prvkem tohoto zařízení je NodeMCU V3, které obsahuje radič s označením ESP8266. Tento radič disponuje integrovaným wifi modulem a zejména nízkou cenou. Ta se pohybuje přibližně okolo 90 Kč. K tomuto zařízení se ještě později vrátíme. Dalším prvkem tohoto zařízení je relé, kterým spínáme již zmiňované piny 20 a 21 na řídicí jednotce brány. Tento relé modul je spojen s NodeMCU V3 pomocí tří vodičů. Tyto vodiče jsou připojeny k pinům: VCC – tento pin je určen k přivedení napětí pro relé (3 V), GND – uzemnění a v neposlední řadě na desce relé najdeme i pin označený IN – tento pin slouží jako vstup, plní funkci ovládacího pinu relé, kterým řídíme jeho stav.



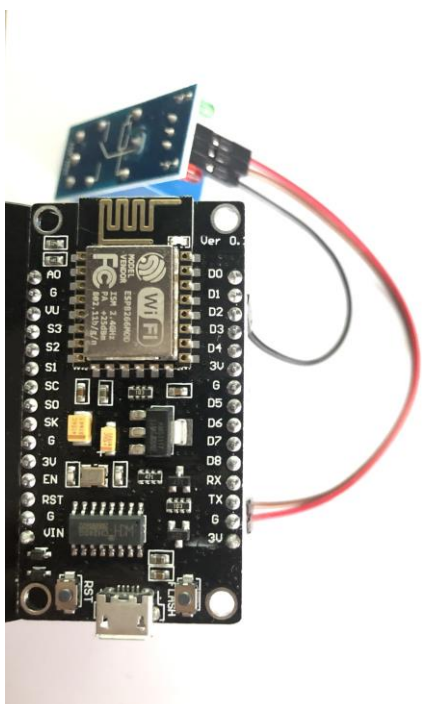
Obrázek 6 – Detail relé

4.1 Relé

Na obrázku 4 můžete vidět detail elektromagnetického relé. Takové relé se cenově pohybuje okolo 50 Kč. Na vrchní straně relé jsou připojené vodiče k pinům ovládacím relé. Zároveň jsou tyto vodiče připojené k NodeMCU V3, kterým je relé ovládáno.

Na spodní straně je k vidění modrá svorkovnice, která slouží jako výstup. V tomto projektu budeme tento výstup využívat zapojený tím způsobem, že bude buď rozpojený, nebo ve zkratu. Tím docílíme, že se na řídicí jednotce piny 20 a 21 spojí nebo rozpojí. Tímto signálem otevřeme nebo zavřeme bránu. Výstupní piny (zleva doprava) jsou označeny: NO (Normally Open, za normálních okolností otevřený), COM (Common, společný), NC (Normally Closed, za normálních okolností sepnutý / zavřený).

Celé zapojení NodeMCU V3 a relé můžete vidět na následujících dvou fotkách.

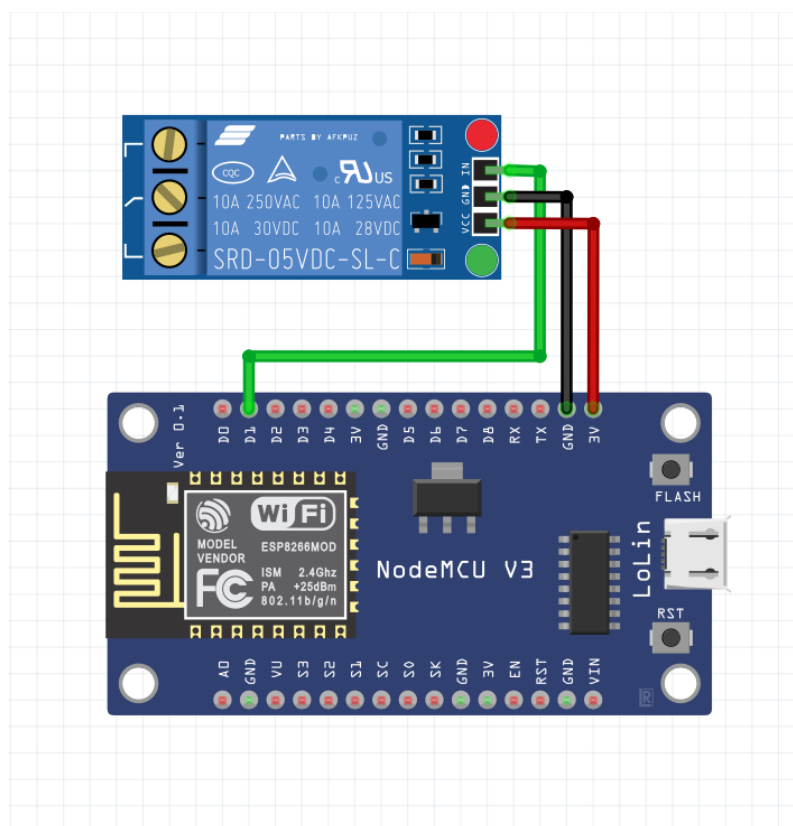


Obrázek 7 – NodeMCU V3 (ESP8266)



Obrázek 8 – Relé FL-3FF-S-Z

Na dalším obrázku je zobrazeno schéma zapojení těchto zařízení.



Obrázek 9 – Schéma zapojení

4.2 NodeMCU V3

Toto zařízení slouží jako mikrořadič s integrovaným wifi modulem a programátorem na jedné miniaturní desce. Jako programátor je na této desce čip CH340, který je mezi mnoha lidmi silně neoblíbený, díky jeho nestabilnímu chování. Já mám tento čip rád kvůli jeho nízké ceně. Tento programátor se připojí k PC pomocí USB a v počítači s ním komunikujeme přes virtuální sériový port. Důležité je stáhnout si správný driver, který ke komunikaci s CH340 potřebujeme a následně doinstalovat potřebný doplněk do vývojového prostředí. Poté stačí vybrat správný sériový port (COM1, COM2...) a nastavit rychlost sériové komunikace. Rychlost sériové komunikace se uvádí v jednotkách Bd (Baud rate). Tyto jednotky ovlivňují modulační rychlost a při špatném nastavení by nebylo možné navázat se zařízením komunikaci. Obvykle se tato rychlost nastavuje na hodnoty 9600 Bd, 115200 Bd.

K jeho funkčnosti nestačí jen jeho zapojení, ale je potřeba jej i naprogramovat. Kód pro tento čip je napsaný v jazyce Wiring, což je jazyk podobný jazyku C++. Tento program zabírá 118 řádků kódu, což je přibližně 7,7 % kódu z kódu celého projektu. Největší část tohoto projektu tvoří serverová část, což je PHP, HTML, CSS a pár SQL příkazů. Na tuto část se podíváme v následující kapitole.

Abych shrnul, jaká je role tohoto zařízení v mém projektu, tak si pojďme vysvětlit podrobnosti funkce tohoto zařízení. Toto zařízení napájí již zmíněný relé modul, kterému posílá požadavek na sepnutí. Tento požadavek přichází z pinu D1, který je pro tuto funkci vzhledem k vlastnostem nejideálnějším pinem. Pokud bychom chtěli použít pin D0, tak by se relé spínalo po každém připojení NodeMCU V3 do elektrické sítě, protože tento pin má logickou hodnotu 1 (HIGH) při sepnutí, tento pin se také využívá jako budič z hlubokého spánku. Pin D1 naštěstí tuto vlastnost nemá, a tak jsem tento pin vybral pro tento projekt. Dalšími ideálními piny pro připojení relé jsou: D2, D3, D4, D5, D6 a D7. Piny D1 a D2 lze využít pro I²C komunikaci – pin D1 slouží jako SCL a pin D2 slouží jako SDA.

5 WEBOVÁ APLIKACE

Jednou ze dvou hlavních částí této práce je webová aplikace, protože bez ní by můj hardware neplnil svůj účel. Webová aplikace běží na webovém serveru, který jsem již zmínil v kapitole Serverové technologie. Tato aplikace běží ne serveru Apache (software), protože byl již nainstalovaný na školním serveru (hardware) a já nemám oprávnění na serveru instalovat programy a měnit jeho nastavení.

Webová aplikace umí:

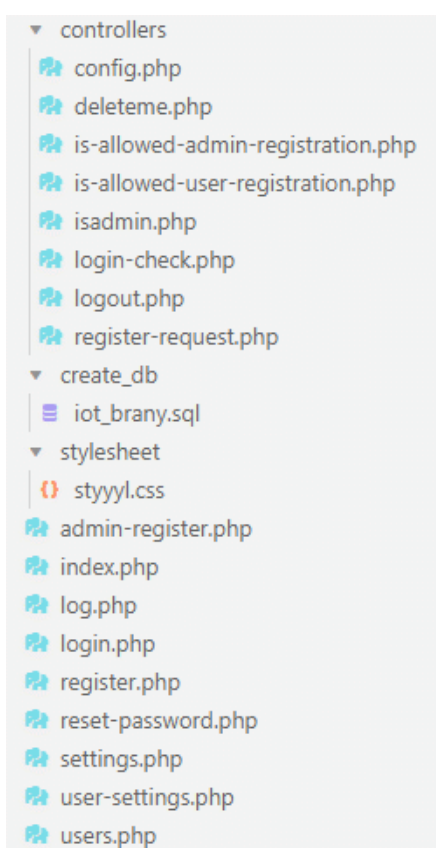
- Zaznamenávat akce vykonané jednotlivými uživateli
- Odesílat požadavky na NodeMCU V3
- Role správce a uživatel
- Správa jednotlivých uživatelů
- Registrace a přihlášení uživatelů
- Změna hesla uživatele
- Správce má možnost odstranit uživatele
- Povolení a zákaz dalších registrací
- Oprávnění ovládání více vjezdových bran

Webová aplikace je napsána v jazyce PHP. Mimo PHP tato aplikace využívá MySQL databáze, do které se zaznamenávají důležité informace. Mezi takové informace patří například: jména a hesla uživatelů, spravované brány, či záznamy akcí Jakub otevřel bránu v pátek v 7:35:42).

5.1 PHP

PHP je nejoblíbenější skriptovací jazyk na straně serveru v oblasti webového vývoje. Pohání téměř 78,9 % všech webových stránek. Vytvořil ho Rasmus Lerdorf v roce 1995. PHP byla původně zkratka pro „Personal Home Page (Tools)“ (což bychom mohli přeložit jako „Nástroje pro osobní domovské stránky“), ale nyní je spíše rekurzivní zkratkou pro „PHP: Hypertext Preprocessor“ (v překladu „Hypertextový preprocesor PHP“). Tento jazyk spravuje, sleduje a vyvíjí skupina vývojářů s názvem The PHP Group, která ho stále distribuuje jako volně dostupný skriptovací jazyk. Jazyk PHP je k dispozici ke stažení přímo na oficiálních webových stránkách. Jazyk PHP bývá většinou interpretován, zpracováván a jeho výsledky zobrazovány webovým serverem s nainstalovaným modulem PHP. Kód jazyka PHP můžeme vkládat do značkování kódu jazyka HTML v souborech s příponou .php. Jazyk PHP je možné

zprovoznit v téměř všech operačních systémech a na všech platformách, a to zdarma. Nejčastější volbou jsou osobní počítače s operačním systémem Linux. V dnešní době vývojáři používají jazyk PHP pro řešení úloh na straně serveru, a ne jako obecný skriptovací jazyk. Tento jazyk je konkurentem pro jazyk ASP.NET společnosti Microsoft, modul mod_perl od společnosti Apache Software Foundation a platformu Node.js od společnosti Joyent. Hlavním cílem jazyka PHP je zpracovávat data tak, aby je bylo možné dynamicky zobrazovat do webových stránek – provádí matematické výpočty, převádí formáty a spolupracuje s databázemi. Umožňuje vývojářům vylepšit statické stránky o reakce na uživatelské požadavky, nebo dokonce ukládat načtená data do databáze a později je uživatelům vypisovat. [7]



Obrázek 10 – Adresář webové aplikace

Pro představu velikosti projektu zde můžete vidět obrázek adresáře se všemi PHP soubory a podadresáři. V tomto adresáři můžeme vidět základní soubory jako například index.php, což je hlavní stránka projektu. Na této stránce jsou tlačítka, které ovládají NodeMCU V3.

Dále jsou v adresáři soubory login.php a register.php, které obstarávají přihlášení a registraci uživatelů do aplikace. Soubor settings.php je přístupný pouze správci – nastavují se díky tomuto souboru možnosti registrací. Soubor users.php zobrazí kompletní tabulku uživatelů. Soubor user-settings.php zobrazí tabulku nastavení možností jednotlivých uživatelů.

V podadresáři controllers najdeme krátké skripty, které se využívají na více místech v celé aplikaci. Jako příklad

můžeme uvést soubor config.php, ve kterém bych našli přihlašovací údaje k databázi, název databáze, způsob připojení a cílovou adresu SQL serveru.

Na další stránce jsou zobrazeny ukázky kódu. Na prvním obrázku je vidět funkce, do které přichází proměnná pro spojení s databází, uživatelské jméno a prvek, se kterým uživatel vyvolal akci. Všechny proměnné se ošetřeným způsobem kvůli SQL injekci dosadí do SQL příkazu a ten se následně vykoná. Na druhém obrázku je ukázka kódu kontrolující, jestli je právě přihlášený uživatel správce – porovnává se uživatelské jméno v databázi, ve které je uložena informace o roli uživatele (standartní uživatel / správce).

```

17 function branaLog($pdo, $username, $element) {
18     // Zkontrolovat chyby před vložením do databáze
19     if(!empty($element)){
20
21         // Proměnná s příkazem pro databázi
22         $sql = "INSERT INTO brana_log (user, element) VALUES (:user, :element)";
23
24         if($stmt = $pdo->prepare($sql)){
25             // Proměnné
26             $param_user = $username;
27             $param_element = $element;
28             // Přiřadí proměnné jako parametry pro předem připravený databázový příkaz
29             $stmt->bindParam(":user", $param_user, PDO::PARAM_STR);
30             $stmt->bindParam(":element", $param_element, PDO::PARAM_STR);
31
32             // Vykoná připravený databázový příkaz
33             $stmt->execute();
34         }
35     }
36 }

```

Obrázek 11 – Ukázka části kódu – index.php (funkce zaznamenávající uživatelské akce do databáze)

```

1 <?php
2 // Zjistí, jestli je uživatel Admin a jestli uživatel vůbec existuje. Pokud uživatel neexistuje, je odhlášen.
3 try {
4     // Nastaví parametry
5     $username = $_SESSION["username"];
6
7     $stmt = $pdo->prepare("SELECT admin FROM users WHERE username = :username");
8     $stmt->bindParam(":username", $username, PDO::PARAM_STR); // :username = proměnná
9
10    // Vykoná SQL příkaz
11    $stmt->execute();
12
13    // Pokud SQL vrátí řádek, tak existuje uživatel a je možné vyzvednout data
14    if($stmt->rowCount() == 1) {
15        $admin = $stmt->fetch();
16    }
17    // Pokud neexistuje uživatel -> odhlásit
18    else {
19        include "logout.php";
20    }
21 }
22 catch(PDOException $e) {
23     echo "Error: " . $e->getMessage();
24 }
25 ?>

```

Obrázek 12 – Ukázka části kódu – isadmin.php (zjišťuje, jestli je uživatel správce)

5.2 MySQL

MySQL je velmi rychlý a robustní systém pro správu relačních databází (RDBMS). Databáze umožňuje efektivně ukládat, vyhledávat, řadit a načítat data. MySQL server řídí přístup k vašim datům. Zajišťuje, aby k datům mohlo přistupovat více uživatelů současně, nabízí rychlý přístup a dohlíží na to, aby přístup získali pouze oprávnění uživatelé. MySQL je tudíž server s podporou více uživatelů a více vláken. Používá jazyk SQL (Structured Query Language) – standardní databázový dotazovací jazyk. MySQL je veřejně dostupný od roku 1996, ale počátek jeho vývoje sahá až do roku 1979. Jedná se o celosvětově nej-oblíbenější databázi s otevřeným zdrojovým kódem, která mnohokrát vyhrála cenu Volba čtenářů časopisu Linux Journal. [8]

SQL databáze pro webovou aplikaci, o které pojednává tato práce se skládá ze tří tabulek:

- brana_log
- settings
- users

V tabulce brana_log jsou uloženy záznamy o tom, kdo a kdy jakou bránu otevřel. Čas se do databáze ukládá jako timestamp, což je jakési číslo, které lze převést na lidsky čitelný čas.

Další tabulkou v databázi je tabulka settings, která obsahuje dva sloupce. Díky této tabulce se nastavuje aplikace, přesněji řečeno se díky této tabulce nastavují možnosti nových registrací uživatelů. Buďto se uživatelé mohou zaregistrovat sami, nebo nové uživatele smí zaregistrovat pouze správce, anebo lze registrace zakázat úplně.

Poslední tabulkou z databáze je tabulka users, která obsahuje nejvíce sloupců. V této tabulce jsou uloženi všichni uživatelé webové aplikace včetně jejich ID, jména, hesla, data vytvoření, uživatelské role a možnosti přístupu k jednotlivým branám. Zajímavá je položka heslo, která je z bezpečnostních důvodů ukládána pouze jako hash hesla (v šifrované podobě).

Na další stránce je zobrazen kód, který vytvoří celou databázi. V kódu si všimněte tří tabulek, sloupců a datových typů. Tento soubor jsem vygeneroval z phpMyAdmin rozhraní jako zálohu a lze použít pro rychlé vytvoření databáze na novém serveru. Tento kód udává pouze strukturu databáze a neobsahuje žádné záznamy.

```

1  -- databáze
2  CREATE DATABASE IF NOT EXISTS `iot_brany` /*!40100 DEFAULT CHARACTER SET utf8 COLLATE utf8_czech_ci */;
3  USE `iot_brany`;
4
5  -- tabulka iot_brany.branalog
6  CREATE TABLE IF NOT EXISTS `brana_log` (
7    `id` int(11) NOT NULL AUTO_INCREMENT,
8    `user` tinytext COLLATE utf8_czech_ci NOT NULL,
9    `element` tinytext COLLATE utf8_czech_ci NOT NULL,
10   `time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
11   PRIMARY KEY (`id`)
12 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci;
13
14 -- tabulka iot_brany.settings
15 CREATE TABLE IF NOT EXISTS `settings` (
16   `userRegistrationAllowed` binary(1) NOT NULL DEFAULT '0',
17   `adminRegistrationAllowed` binary(1) NOT NULL DEFAULT '0'
18 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci MAX_ROWS=1;
19
20 -- tabulka iot_brany.users
21 CREATE TABLE IF NOT EXISTS `users` (
22   `id` int(11) NOT NULL AUTO_INCREMENT,
23   `username` varchar(50) COLLATE utf8_czech_ci NOT NULL,
24   `password` varchar(255) COLLATE utf8_czech_ci NOT NULL,
25   `created_at` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
26   `admin` binary(1) NOT NULL DEFAULT '0',
27   `brana1` binary(1) NOT NULL DEFAULT '0',
28   `brana2` binary(1) NOT NULL DEFAULT '0',
29   `garaz` binary(1) NOT NULL DEFAULT '0',
30   PRIMARY KEY (`id`),
31   UNIQUE KEY `username` (`username`)
32 ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci;

```

Obrázek 13 – SQL databáze

5.3 HTML a CSS

Hypertext Markup Language (zkratka HTML) je v informatice název značkovacího jazyka používaného pro tvorbu webových stránek, které jsou propojeny hypertextovými odkazy. HTML je hlavním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu. Jazyk je aplikací dříve vyvinutého rozsáhlého univerzálního značkovacího jazyka SGML (Standard Generalized Markup Language). Vývoj HTML byl ovlivněn vývojem webových prohlížečů, které zpětně ovlivňovaly definici jazyka. [9]

Kaskádové styly (v anglickém originále Cascading Style Sheets se zkratkou CSS) je v informatice jazyk pro popis způsobu zobrazení elementů na stránkách napsaných v jazycích HTML, XHTML nebo XML. Jazyk byl navržen standardizační organizací W3C, autorem prvotního návrhu byl Håkon Wium Lie. Byly vydány CSS1, CSS2 a CSS3. Dne 7. června 2011 byla dokončena revize CSS 2.1[1] Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Původně to měl umožnit už jazyk HTML, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů se vyvinul jinak. Starší verze HTML obsahují celou řadu elementů, které nepopisují obsah a strukturu dokumentu, ale i způsob jeho zobrazení. Z hlediska zpracování dokumentů a vyhledávání informací není takový vývoj žádoucí. [10]

Pro shrnutí, HTML je jakási kostra (udává tvar) a CSS je tělo (udává vzhled). Dohromady této viditelné stránce aplikace říkáme frontend (HTML a CSS). Naopak backend je to, co na první pohled nevidíme (PHP). Obor, který se věnuje vzhledu webových stránek se říká webdesign. V následujících snímcích můžete vidět design webové aplikace, kterou jsem vytvořil.



Obrázek 14 – Hlavní stránka aplikace

This is a registration form titled 'Registrace'. It includes a sub-header 'Vyplňte tento formulář pro založení nového účtu.' Below this, there are three input fields: 'Uživatelské jméno', 'Heslo', and 'Potvrdit heslo'. At the bottom of the form, there are two buttons: 'Potvrdit' and 'Resetovat'. A link 'Máte účet? Přihlašte se zde.' is located at the very bottom.

Obrázek 15 – Registrační formulář

This is a login form titled 'Přihlásit se'. It includes a sub-header 'Vyplňte tento formulář pro přihlášení k účtu.' Below this, there are two input fields: 'Jméno' and 'Heslo'. A 'Přihlásit se' button is positioned below the password field. At the bottom, there is a link 'Nemáte účet? Založit si účet.'.

Obrázek 16 – Přihlašovací formulář

SPS-CL Brána

Log

Uživatelé

Nastavení uživatelů

Nastavení registrací

Registrace uživatelů

Nastavení +

Vítej uživateli kuba!

ID	Uživatel	Prvek	Čas požadavku
204	kuba	brana1	2022-02-21 21:39:56
203	kuba	brana2	2022-02-21 21:39:49
202	kuba	brana2	2022-02-21 21:39:48
201	kuba	brana1	2022-02-21 21:39:47
200	kuba	brana1	2022-02-21 21:39:46
198	kuba	brana2	2022-02-21 21:39:44
199	kuba	brana1	2022-02-21 21:39:44
197	kuba	brana1	2022-02-21 21:39:43
196	kuba	brana1	2022-02-21 21:23:36
194	kuba	garaz	2022-02-21 21:23:35
195	kuba	brana2	2022-02-21 21:23:35
193	kuba	brana1	2022-02-21 21:23:34
192	kuba	brana2	2022-02-21 21:23:33
191	kuba	brana2	2022-02-21 21:23:27
190	kuba	brana2	2022-02-21 20:28:58
189	kuba	brana2	2022-02-21 20:28:12
188	kuba	brana1	2022-02-21 20:28:10
187	kuba	garaz	2022-02-21 20:25:36
185	kuba	garaz	2022-02-21 20:25:35
186	kuba	brana2	2022-02-21 20:25:35
184	kuba	brana1	2022-02-21 20:25:33
183	kuba	brana1	2022-02-21 20:25:27
182	kuba	brana1	2022-02-21 20:25:04
180	kuba	brana1	2022-02-21 20:25:03
181	kuba	brana2	2022-02-21 20:25:03

Obrázek 17 – List záznamů aktivit

SPS-CL Brána

Log

Uživatelé

Nastavení uživatelů

Nastavení registrací

Registrace uživatelů

Nastavení +

Vítej uživateli kuba!

ID	Jméno	Admin	Vytvořen
1	kuba	1	2021-12-03 03:04:37
2	kuba2	0	2021-12-03 04:22:06
3	kuba3	1	2021-12-03 15:03:00
5	kuba6	0	2021-12-05 01:36:14
6	kuba7	0	2021-12-05 01:36:28
7	kuba8	0	2021-12-05 01:37:08
8	kuba9	0	2021-12-05 01:44:08

Obrázek 18 – Seznam uživatelů

SPS-CL Brána Log Uživatelé Nastavení uživatelů Nastavení registrací Registrace uživatelů Nastavení *

Vítej uživateli **kuba**!

Jméno	Admin	Brána 1	Brána 2	Geriz	Odstanění uživatele
kuba	1	1	1	1	x
kuba2	0	1	0	1	x
kuba3	1	1	1	1	x
kuba6	0	0	0	0	x
kuba7	0	0	0	0	x
kuba8	0	0	0	0	x
kuba9	0	0	0	0	x

Obrázek 19 – Oprávnění a odstanění účtů uživatelů

SPS-CL Brána Log Uživatelé Nastavení uživatelů Nastavení registrací Registrace uživatelů Nastavení *

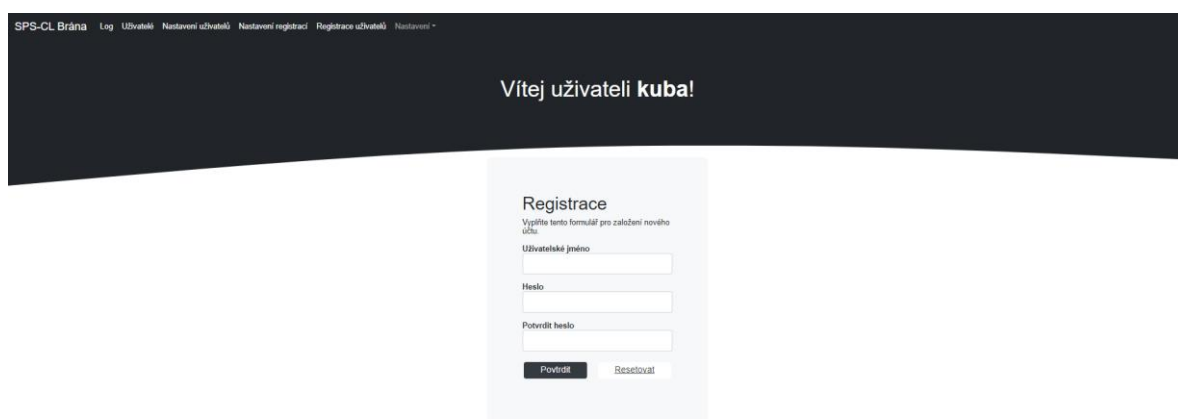
Vítej uživateli **kuba**!

☒ Povolit registrace uživatelů

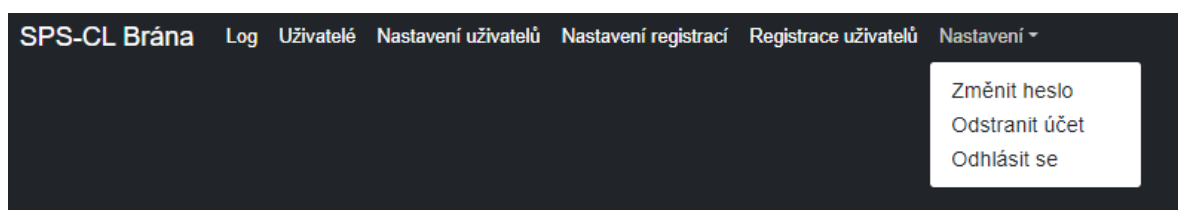
☒ Povolit registrace adminem

Nastavit

Obrázek 20 – Nastavení nových registrací



Obrázek 21 – Registrační formulář dostupný pouze pro správce



Obrázek 22 – Hlavní menu aplikace

6 ZÁVĚR

Ve své práci jsem poukázal na výhody využití internetu věcí spolu s evidenčními systémy v praxi a zprovoznil pomocí jej projekt, který vzdáleně ovládá vjezdovou bránu a zaznamenává provedené akce. Teoretická část obsahuje přehled základů funkcí internetu věcí spolu s evidenčními systémy i principy tvorby samostatného projektu. Mojí snahou je seznámit široké spektrum lidí s tímto konceptem, který si najde své využití téměř všude.

Cílem praktické části byla tvorba „chytré“ brány, která měla splňovat podmínky stanovené ředitelem školy. Tohoto cíle jsem dosáhl, ale do cesty se mi připletlo nemálo problémů a chyb. Jeden z problémů bohužel zůstal nedořešený, ale s tímto problémem nemohu sám nic udělat. Data se mezi zařízeními předávají bezdrátově pomocí školní sítě, která je dobře zabezpečená a disponuje síťovým firewallem. Tento firewall blokuje pakety, které jsou v mém projektu klíčové. Díky tomuto problému není možné na mnou sestavené zařízení odeslat jakýkoliv požadavek. V domácí síti se s touto problematikou nesetkáme, ale ve školní či firemní síti ano. K vyřešení tohoto problému by stačilo přidat do firewallu výjimku pro dané zařízení, ale k této akci bohužel nemám potřebná oprávnění. Aplikace je prozatím dostupná k testovacím účelům na adrese <http://10.1.100.50/sirovy/www>.

Všechny ostatní problémy se mi podařit vyřešilo. Jedním z těchto problémů bylo například hledání ovládacích pinů na řídicí jednotce samotné brány. S tím mi pomohl pan ředitel, který mi poskytl potřebné dokumenty a ukázku zapojení jeho brány, která má podobnou řídicí jednotku. Díky této pomoci pro mne bylo snazší dohledat potřebné piny.

Nad prací jsem se zamýšlel tak, aby toto zařízení bylo schopné pracovat dlouho a v případě technické chyby (například opotřebení relé modulu nebo jeho zničení) je možné prvek svépomocí vyměnit a zařízení uvést zpět do provozu.

V domácích podmínkách je tato práce plně a stabilně funkční, avšak vždy je možné něco zlepšovat a vyvíjet. V budoucnu by bylo například možné k tomuto projektu vytvořit mobilní aplikaci, nebo zlepšit a zautomatizovat vaše aktivity společně s mobilní aplikací (například: „Já v autě mívám aktivní bluetooth na mém mobilním telefonu. Pokud bych se přiblížil k bráně s aktivním bluetooth, tak by se přes mobilní aplikaci vyslal požadavek a brána by se sama automaticky otevřela.“) Toto vše je hudba budoucnosti a ne vždy je každé vylepšení k prospěchu.

7 POUŽITÁ LITERATURA

- [1] Apache HTTP Server. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-04-21]. Dostupné z: https://cs.wikipedia.org/wiki/Apache_HTTP_Server
- [2] Slowloris. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-04-21]. Dostupné z: <https://cs.wikipedia.org/wiki/Slowloris>
- [3] Nginx. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-04-21]. Dostupné z: <https://cs.wikipedia.org/wiki/Nginx>
- [4] Secure Shell. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-04-21]. Dostupné z: https://cs.wikipedia.org/wiki/Secure_Shell
- [5] SSH file transfer protocol. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-04-21]. Dostupné z: https://cs.wikipedia.org/wiki/SSH_file_transfer_protocol
- [6] RotaMatic 2 [online]. 2017 [cit. 2022-04-21]. Dostupné z: https://www.ivrata.cz/files/TR10A275_RotaMatic2_novy_cesky_4_1.pdf
- [7] HOPKINS, Callum. PHP Okamžitě. Cambridge: COMPUTER PRESS, 2014. ISBN 9788025143940.
- [8] WELLING, Luke a Laura THOMSON. Mistrovství - PHP a MySQL. London: COMPUTER PRESS, 2017. ISBN 9788025149010.
- [9] Hypertext Markup Language. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-04-21]. Dostupné z: https://cs.wikipedia.org/wiki/Hypertext_Markup_Language
- [10] Kaskádové styly. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-04-21]. Dostupné z: https://cs.wikipedia.org/wiki/Kask%C3%A1dov%C3%A9_styly

8 SEZNAM OBRÁZKŮ A TABULEK

Obrázek 1 – Schéma projektu	16
Obrázek 2 – Vysvětlivky	16
Obrázek 3 – Výstřižek z manuálu výrobce.....	18
Obrázek 4 – Řídící jednotka vjezdové brány.....	19
Obrázek 5 – Detail pinů řídicí jednotky	20
Obrázek 6 – Detail relé.....	21
Obrázek 7 – NodeMCU V3 (ESP8266)	22
Obrázek 8 – Relé FL-3FF-S-Z.....	22
Obrázek 9 – Schéma zapojení.....	22
Obrázek 10 – Adresář webové aplikace	25
Obrázek 11 – Ukázka části kódu – index.php (funkce zaznamenávající uživatelské akce do databáze).....	26
Obrázek 12 – Ukázka části kódu – isadmin.php (zjišťuje, jestli je uživatel správce).....	26
Obrázek 13 – SQL databáze	28
Obrázek 14 – Hlavní stránka aplikace.....	30
Obrázek 15 – Registrační formulář	30
Obrázek 16 – Přihlašovací formulář.....	30
Obrázek 17 – List záznamů aktivit.....	31
Obrázek 18 – Seznam uživatelů.....	31
Obrázek 19 – Oprávnění a odstanění účtů uživatelů.....	32
Obrázek 20 – Nastavení nových registrací	32
Obrázek 21 – Registrační formulář dostupný pouze pro správce	33
Obrázek 22 – Hlavní menu aplikace	33