

**Univerzita Hradec Králové**  
**Fakulta informatiky a managementu**  
**Katedra informačních technologií**

**Návrh a implementace Smart Home řešení s prvky IoT**

Diplomová práce

Autor: Bc. Libor Plíšek

Studijní obor: Aplikovaná informatika

Vedoucí práce: prof. Ing. Ondřej Krejcar, Ph.D.

Centrum základního a aplikovaného výzkumu

Hradec Králové

duben 2021

**Prohlášení:**

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.



.....

V Hradci Králové dne 14. 4. 2021

Bc. Libor Plíšek

**Poděkování:**

Prvně děkuji svému vedoucímu prof. Ing. Ondřejovi Krejcarovi, Ph.D. za ukázkový přístup při tvorbě diplomové práce, za veškeré rady, a hlavně za jeho čas, který mi vždy ochotně a okamžitě věnoval. Neméně děkuji své přítelkyni Veronice Straškové za znamenitou podporu během celého studia. A v neposlední řadě děkuji svým rodičům za to, že mi umožnili studium na této prestižní a moderní škole.



## **Anotace**

Tato diplomová práce je zaměřena na vývoj elektronického modulu, pomocí kterého lze levně a chytře ovládat různé druhy jiných elektrických zařízení, a to vzdáleně pomocí mobilní aplikace a internetu. Obsahem této práce je analýza již existujících řešení a jejich srovnání spolu s analýzou nového řešení. Další součástí této práce je detailní návrh a implementace nového informačního systému. V rámci tohoto systému vznikl firmware pro několik druhů modulů značky Sonoff naprogramovaný v jazyce C a multiplatformní mobilní aplikace vyvinutá ve frameworku Ionic a prostředí Capacitor. Dále vznikla serverová aplikace, vytvořená v CMS Drupal a programovacím jazyce PHP, která obsahuje administrační a komunikační rozhraní pro obě zmíněné platformy. Jako poslední je v této práci popsáno testování implementovaného řešení, jeho limity a možnosti dalšího rozšíření.

## **Annotation**

### **Title: Design and Implementation of Smart Home solution with IoT**

This master thesis is focused on the development of an electronic module, which can be used as a cheap and smart control of other electrical devices, remotely using a mobile application and the Internet. The content of this work is the analysis of existing solutions and their comparison with the analysis of new solutions. Another part of this work is a detailed design and implementation of a new information system. Within this system was created firmware programmed in C language for several types of Sonoff modules and a multiplatform mobile application developed in the Ionic framework and the Capacitor environment. Furthermore, a server application was created in CMS Drupal and the PHP programming language, which contains an administration and communication interface for both mentioned platforms. Lastly, this work describes the testing of the implemented solution, its limits, and the possibility of further expansion.

## Obsah

1	Úvod.....	1
2	Cíl práce.....	3
3	Analýza .....	4
3.1	Možnosti připojení modulů k internetu .....	4
3.2	Existující řešení.....	5
3.2.1	Přehled existujících řešení .....	5
3.2.2	Porovnání existujících řešení s novým řešením .....	8
3.3	Nové řešení.....	9
3.3.1	Použitý hardware pro moduly.....	10
3.3.2	Funkční požadavky .....	12
3.3.3	Nefunkční požadavky .....	15
4	Návrh.....	17
4.1	Komponenty informačního systému .....	17
4.1.1	Firmware modulu.....	19
4.1.2	Mobilní aplikace .....	19
4.1.3	Serverová aplikace .....	20
4.2	Uživatelské role informačního systému.....	20
4.2.1	Správce systému .....	21
4.2.2	Běžný uživatel .....	21
4.2.3	System.....	21
4.3	Případy užití.....	21
4.3.1	Základní správa modulů.....	22
4.3.2	Ovládání modulů.....	23
4.3.3	Stav modulů a systému.....	25
4.3.4	Správa připojení a sítí v modulech.....	26

4.3.5	Konfigurace a nastavení modulů .....	28
4.4	Databáze serverové aplikace .....	30
4.4.1	Databázový model .....	30
4.5	Komunikační rozhraní serverové aplikace .....	32
4.5.1	Dokumentace .....	32
4.5.2	Požadavky .....	34
4.5.3	Odpovědi .....	35
4.6	Automatizované úlohy .....	37
4.6.1	Automatické ovládání modulů a odesílání upozornění .....	37
4.6.2	Automatické povolení zablokovaných IP adres .....	38
4.6.3	Automatické čištění historie .....	38
5	Implementace .....	39
5.1	Moduly .....	39
5.2	Serverová aplikace .....	43
5.3	Mobilní aplikace .....	44
5.4	Komunikace .....	45
5.5	Bezpečnost .....	46
5.6	Nasazení .....	47
5.7	Zálohování .....	48
6	Testování .....	49
6.1	Firmware .....	49
6.2	Serverová aplikace .....	51
6.3	Mobilní aplikace .....	52
6.4	Dostupnost .....	53
6.5	Zálohování .....	54
6.6	Výkon .....	54

7	Shrnutí .....	55
7.1	Limity řešení .....	56
7.2	Možnosti rozšíření .....	56
8	Závěr .....	57
9	Seznam použité literatury .....	58
10	Přílohy .....	64
10.1	Seznam použitých zkratk .....	64
10.2	Obsah praktické části práce .....	65
10.3	Odkazy na praktickou část práce .....	66
10.4	Grafické uživatelské rozhraní mobilní aplikace .....	67
10.5	Databázové tabulky serverové aplikace .....	71



## Seznam obrázků

Obrázek 1: Moduly značek Philips [5], Fibaro [6] a Legrand [7] .....	6
Obrázek 2: Moduly značek Gosund [9], Shelly [12], Woose [10] a iQtech [13] .....	7
Obrázek 3: Původní hardwarové komponenty modulu .....	10
Obrázek 4: Možnosti zapojení modulu do elektrického obvodu .....	11
Obrázek 5: Vlastnoručně upravený plastový kryt modulu .....	12
Obrázek 6: Přehled komponent informačního systému .....	18
Obrázek 7: Hierarchie uživatelských rolí v informačním systému .....	20
Obrázek 8: Případy užití základní správy modulů .....	22
Obrázek 9: Případy užití ovládání modulů .....	24
Obrázek 10: Případy užití stavu modulů a systému .....	25
Obrázek 11: Případy užití správy připojení a sítí v modulech .....	27
Obrázek 12: Případy užití konfigurace a nastavení modulů .....	28
Obrázek 13: Navržený databázový model .....	31
Obrázek 14: Náhled API dokumentace – detail modulu .....	33
Obrázek 15: Náhled API dokumentace – změna uložených Wi-Fi sítí .....	33
Obrázek 16: Příklad HTTP požadavku na koncový bod REST API .....	34
Obrázek 17: Příklad standardní HTTP odpovědi .....	36
Obrázek 18: Příklad chybové HTTP odpovědi .....	36
Obrázek 19: Vývojový diagram automatizace serverové aplikace .....	37
Obrázek 20: Formulář pro nastavení blokování IP adres .....	38
Obrázek 21: Ukázka zapojení modulu při jeho programování .....	40
Obrázek 22: Příprava a zapojení dvou teplotních senzorů k modulu .....	41
Obrázek 23: Výsledný modul obsahující dva teplotní senzory .....	41
Obrázek 24: Stav paměti modulu během aktualizace firmwaru .....	42
Obrázek 25: Ukázka administračního portálu .....	44
Obrázek 26: Snímky obrazovky mobilní aplikace .....	45
Obrázek 27: Vývojový diagram obecného případu užití .....	46
Obrázek 28: Ukázka online GitLab repozitáře .....	47
Obrázek 29: Ukázka uložených záloh databáze serverové aplikace [74] .....	48
Obrázek 30: Ukázka sériového výstupu firmwaru modulu .....	50

Obrázek 31: Ukázka průběžné integrace serverové aplikace [23] .....	51
Obrázek 32: Ukázka rozhraní interního testování v konzoli Google Play [73] .....	52
Obrázek 33: Náhled externího monitoringu dostupnosti serveru [76] .....	53
Obrázek 34: Ukázka automatického testování REST API [23] .....	54
Obrázek 35: Snímky obrazovek mobilní aplikace (1) .....	67
Obrázek 36: Snímky obrazovek mobilní aplikace (2) .....	68
Obrázek 37: Snímky obrazovek mobilní aplikace (3) .....	68
Obrázek 38: Snímky obrazovek mobilní aplikace (4) .....	69
Obrázek 39: Snímky obrazovek mobilní aplikace (5) .....	69
Obrázek 40: Snímky obrazovek mobilní aplikace (6) .....	70
Obrázek 41: Snímky obrazovek mobilní aplikace (7) .....	70
Obrázek 42: Snímky obrazovek mobilní aplikace (8) .....	71

## Seznam tabulek

Tabulka 1: Porovnání existujících řešení s novým řešením.....	9
Tabulka 2: Parametry základních použitých Sonoff modulů.....	11
Tabulka 3: Seznam funkčních požadavků informačního systému.....	13
Tabulka 4: Seznam nefunkčních požadavků informačního systému.....	15
Tabulka 5: Databázová tabulka MODULE.....	72
Tabulka 6: Databázová tabulka FIRMWARE .....	72
Tabulka 7: Databázová tabulka APP .....	72
Tabulka 8: Databázová tabulka MODULE_APP .....	72
Tabulka 9: Databázová tabulka STATUS.....	73
Tabulka 10: Databázová tabulka LAST_VALUES .....	73
Tabulka 11: Databázová tabulka NEW_VALUES.....	73
Tabulka 12: Databázová tabulka CONDITIONS.....	74
Tabulka 13: Databázová tabulka SCHEDULE.....	74
Tabulka 14: Databázová tabulka NOTIFICATIONS.....	74
Tabulka 15: Databázová tabulka CONNECT_TO .....	74
Tabulka 16: Databázová tabulka NETWORK_SAVE.....	74
Tabulka 17: Databázová tabulka NETWORK_SCAN .....	75
Tabulka 18: Databázová tabulka SETTIGNS.....	75
Tabulka 19: Databázová tabulka CONFIG .....	75
Tabulka 20: Databázová tabulka CONFIG_PIN_LED .....	76
Tabulka 21: Databázová tabulka CONFIG_TEMP_SENSOR.....	76
Tabulka 22: Databázová tabulka LOG_ACCESS .....	76
Tabulka 23: Databázová tabulka LOG_RELAY_STATE.....	76
Tabulka 24: Databázová tabulka LOG_TEMPERATURES .....	77
Tabulka 25: Databázová tabulka BAN_IP .....	77
Tabulka 26: Databázová tabulka BAN_IP_LOG .....	77

# 1 Úvod

V dnešní moderní době, kdy je svět poměrně uspěchaný a čas je pro většinu lidí skoro tím nejcennějším co mají, se každý snaží žít co nejlépe, co nejvíce zábavně, ale hlavně co nejsnadněji a nejpohodlněji. Dnes, kdy začíná být připojené k internetu opravdu už skoro téměř vše, co má v sobě alespoň jenom nějaký malý kus elektroniky, se v chytrých domácnostech nemůžeme divit tomu, jak je v celku snadné ovládat přes mobilní telefon například pračku, sušičku, rychlovarnou konvici nebo robotický vysavač.

Mobilní telefony v podobě, v jaké je známe dnes, skoro každý z nás vlastní a nosí pravidelně či téměř neustále u sebe. A díky těmto chytrým kapesním zařízením si dnes dokážeme splňovat a usnadňovat poměrně velké množství každodenních a základních potřeb, například objednávání jídla, potravin, hygienických potřeb, ale i monitorování našeho zdraví apod. Je zcela jistě pro většinu populace naprosto nepředstavitelná úvaha, že by takový produkt na světě vůbec nikdy neexistoval, protože každý z nás si již dávno zvykl na jejich všudypřítomnost a jejich poměrně neomezených možnostech využití. Je tedy naprosto logické, že mobilní telefon nemá být pouze jakési centrum spojení člověka s ostatními lidmi, nýbrž právě i pomocník a spojení s chytrou elektronikou a jinými věcmi. I z tohoto důvodu se tento projekt věnuje právě této problematice internetu věcí a chytrých domácností a snaží se dále rozvíjet a usnadňovat život lidem v tomto odvětví a směru.

Na internetu existuje vzdálených chytrých vypínačů a ovladačů doslova naprosto nepřeborné množství, jenže každý z nich se primárně soustředí pouze na standardní prostředí běžných domácností a kanceláří. V rámci analýzy existujících řešení, která je popsána ve třetí kapitole této práce, nebylo nalezeno ani jedno levné již existující řešení, které by splňovalo funkční požadavky a případy užití tohoto projektu. Právě přesně proto vznikl tento projekt, který se soustředí především přesně na opačný směr, tedy na vzdálené ovládání elektrických zařízení, které nejsou rutinní záležitostí, nebo které se nenacházejí v běžných prostředích. Nestandardními prostředími jsou myšlena například taková prostředí, ve kterých může často docházet k výpadkům Wi-Fi signálu nebo celkově k výpadkům připojení k internetu. V takových prostředích můžeme chtít vzdáleně ovládat něco, co není

široce rozšířenou a rutinní záležitostí a potřebujeme, abychom se na tuto technologii mohli spolehnout, aby byla bezpečná a aby byla plně konfigurovatelná, nastavitelná a připravená na tyto nestandardní případy. Navíc, pokud chceme zachovat nízkou cenu modulů, je potřeba aby fungovaly samostatně a aby byly jednoduše rozšířitelné o další senzory bez přidání dalších jednotek nebo bez přidání centrálního prvku.

Po podrobné analýze existujících řešení, použitého hardwaru a funkčních i nefunkčních požadavků nového informačního systému, následuje ve 4. kapitole této práce detailní návrh, který obsahuje schéma a popis jednotlivých komponent informačního systému, seznam uživatelských rolí a kompletní návrh případů užití přímo navazujících na analyzované funkční požadavky ze 3. kapitoly. Dále v rámci návrhu vzniklo databázové schéma, přehled komunikace mezi komponentami systému a popis automatizovaných úloh, které pravidelně probíhají na pozadí informačního systému. Všechna teoreticky navržená řešení byla v rámci této práce prakticky implementována, testována a kompletně produkčně zprovozněna. Tato praktická část je přiložena v elektronické podobě k této práci a je volně dostupná na internetu. Po kompletním návrhu se v šesté kapitole této práce nachází postup implementace navrženého řešení, kde jsou mimo jiné shrnuty použité technologie, postup nasazení a popis zálohování. Jako předposlední se v této práci nachází popis testování implementovaného řešení včetně testování vhodných nefunkčních požadavků a hlavního komunikačního rozhraní. Závěrem, po kapitole „Testování“, následuje kapitola sedmá, která shrnuje limity nového řešení a nastiňuje možnosti jeho dalšího rozšíření.

## 2 Cíl práce

Hlavním cílem této práce je vytvoření nového informačního systému, který umožňuje levně a chytře ovládat elektrická zařízení, a to vzdáleně pomocí hardwarových modulů a mobilních telefonů přes internet. Tyto hardwarové moduly by navíc měly být samostatné, lehce rozšiřitelné o další senzory, a to bez přidání dalších jednotek nebo bez použití centrálního prvku.

Cílem teoretické části práce je analýza již existujících řešení spolu s analýzou možností připojení modulů k internetu. Dále analýza použitého hardwaru spolu s analýzou funkčních i nefunkčních požadavků nového systému. V rámci návrhu informačního systému a jeho jednotlivých komponent, vytvořit popis uživatelských rolí a detailní popis případů užití plně navazujících na analyzované funkční požadavky. Dále navrhnout databázové schéma, komunikaci mezi komponentami informačního systému včetně návrhu API dokumentace a návrh automatizovaných úloh, které probíhají na pozadí systému. V neposlední řadě je cílem teoretické části této práce popsat implementaci a testování implementovaného řešení a závěrem shrnout limity nového řešení a nastínit možnosti jeho dalšího rozšíření.

Cílem praktické části této práce je implementovat navržená řešení, včetně nasazení a zprovoznění zálohování kódu i dat. V rámci implementace hardwaru, je cílem pokrytí nízkého stejnoměrného i vysokého střídavého napětí. A nakonec je cílem praktické části této práce také testování implementovaného řešení včetně vhodných nefunkčních požadavků a hlavního komunikačního rozhraní.

### **3 Analýza**

Tato kapitola obsahuje shrnutí možností připojení hardwarových modulů k internetu, přehled vybraných nejprodávanějších a nejlépe hodnocených levných již existujících produktů, které se dají pořídit z největších českých e-shopů a u každého z nich uvádí jeho krátký popis s jeho výhodami a jeho nedostatky oproti novému řešení. U existujících řešení tato kapitola především vysvětluje, v čem je právě nové řešení unikátním a za určitých podmínek lepším projektem. Jako poslední obsahuje tato kapitola podrobnou analýzu nového řešení, analýzu použitého hardwaru a analýzu funkčních i nefunkčních požadavků.

#### **3.1 Možnosti připojení modulů k internetu**

Existují dvě nejvhodnější možnosti připojení chytrých modulů k internetu. Jednou z nich je pořízení nebo sestavení hardwaru obsahující GSM modul [1], do kterého se dá vložit, nebo který již obsahuje zabudovanou SIM kartu a který přijímá signály pomocí sítě mobilního operátora. Často se tento typ připojení k internetu objevuje v prostředích, kde je velmi slabý nebo nestabilní Wi-Fi signál anebo kde Wi-Fi signál není vůbec. To z pravidla nastává v domácnostech u zařízení na monitoring solárních systémů [2], protože na střeších domů nebo například zahradách už může být pokrytí Wi-Fi signálem mnohem slabší nebo dokonce vůbec žádné. I přesto, že mají mobilní operátoři mnohem lepší pokrytí signálem než Wi-Fi sítě, použití tohoto řešení není vhodné pro tento projekt. Vzhledem k tomu, že je potřeba k GSM modulu použít SIM kartu a mobilní data (tedy že není možné pouze zprovoznit GSM modul bez SIM karty), vychází toto řešení cenově zbytečně moc nákladně, protože nutně navíc obsahuje pravidelné měsíční náklady na provoz SIM karty a mobilních dat. Aktuální řešení fungující pouze pomocí již existující Wi-Fi sítě, které popisuje tato práce, nevyžaduje navíc vůbec žádné další měsíční náklady. Právě proto byly pro tuto práci použity pouze takové moduly, které obsahují hardwarový Wi-Fi modul [3], tedy které umožňují připojení k internetu pouze pomocí již existujícího Wi-Fi přístupového bodu, nikoliv pomocí sítě mobilního operátora.

## 3.2 Existující řešení

Tato práce řeší problém vzdáleného ovládání elektrických zařízení přes internet pomocí levných a chytrých modulů a mobilních telefonů. Takových řešení existuje na světě již velmi mnoho, nicméně ty nejprodávanější, nejlépe hodnocené a nejlevnější z nich, co se dají pořídit na území České republiky, nevyhovují funkčním požadavkům a případům užití tohoto projektu.

Již existující řešení nevyhovují určitým nestandardním nebo zhoršeným podmínkám, které mohou v určitých situacích v celku pravidelně nastávat. Dále často nedokážou pracovat samostatně nebo se nedají jednoduše rozšířit o externí senzory bez přidání dalších jednotek nebo centrálního prvku. A když už nějaké řešení nabízí přidání senzoru hardwarově napřímo k modulu, tak lze přidat pouze jeden kus senzoru (například pouze jeden teplotní senzor, nikoliv dva a více).

Dále již existující moduly zpravidla nenabízejí konfiguraci akce, která má nastat, jakmile modul ztratí Wi-Fi signál nebo připojení k internetu obecně. Naprostá většina existujících nejlevnějších řešení se po ztrátě signálu buďto vypnou, nebo zůstanou ve stavu v jakém jsou, bez ohledu na to, že může zařízení zůstat odpojené i několik hodin nebo dní. U těchto typů zařízení je třeba mít možnost nastavení maximální doby, po kterou je možné zůstat v sepnutém stavu po ztrátě připojení k internetu, aby uživatel nebyl nucen při takového ztrátě signálu jít ovládat modul manuálně a mohl ušetřit tak svůj čas i peníze. Je pochopitelné, že se podniky na tyto mezní případy nesoustředí a soustředí se spíše na podmínky, které nastávají v drtivém množství případů, protože a jednoduše by se jim to pravděpodobně vůbec nevyplatilo, cílit například jenom na několik setin procent koncových zákazníků.

### 3.2.1 Přehled existujících řešení

V této sekci je shrnut přehled již existujících řešení. Celkem bylo vybráno osm nejprodávanějších a nejlépe hodnocených značek. Za každou dále zmíněnou značku byl vybrán právě jeden nejvhodnější a zároveň nejlevnější produkt, který byl následně podrobněji analyzován a popsán. Pro lepší přehlednost jsou značky rozděleny na dvě skupiny do jednotlivých podsekcí. Všechny obrázky existujících modulů byly převzaty z internetového e-shopu „Alza.cz“ [4], mohou tedy obsahovat například vodotisk obchodu apod.



## Philips, Fibaro a Legrand

Firma Phillips [5] se v této oblasti soustředí primárně především na ovládání světel a zásuvek v chytrých domácnostech než na libovolně přizpůsobitelné vzdáleně ovládané spínače, čímž naprosto vůbec nevyhovuje tomuto projektu. Navíc se pohybuje v již poměrně vyšší cenové kategorii.

Firmy Fibaro [6] a Legrand [7] mají své výrobky v ještě vyšších cenových kategoriích než firma Phillips. Navíc zařízení od firmy Legrand se spíše soustředí na nahrazování stávajících prvků ve zdech již existující elektroinstalace domácností a komunikují především pomocí rádiového protokolu ZigBee [8], takže potřebují ke svému chodu speciální domácí centrální jednotku. Obě firmy se také soustředí především na chytré domácnosti jako firma Phillips a jejich výrobky jsou vskutku pěkné a velmi kladně hodnocené v internetových uživatelských recenzích. Jenže právě z těchto zmíněných důvodů jsou také naprosto nevyhovujícím a zároveň nedostatečným řešením pro tento projekt, a proto nemohou být použity.



**Obrázek 1: Moduly značek Philips [5], Fibaro [6] a Legrand [7]**

## Gosund, Woon, Sonoff, Shelly a iQtech

Ostatní značky chytrých vypínačů, Gosund [9], Woon [10], Sonoff [11], Shelly [12] a iQtech [13] vyrábí vzdáleně ovládané vypínače přesně takového typu a v takové cenové kategorii, které jsou vhodné i pro tento projekt. Vždy mohou fungovat samostatně, jsou levné a dají se ovládat vzdáleně přes internet pomocí Wi-Fi sítě a mobilní aplikace. Jenže ani jedna z nich neposkytuje podporu přímého přidávání více kusů externích senzorů a nastavení akcí, které se mají aplikovat po výpadku napájení, při výpadku signálu Wi-Fi nebo při výpadku připojení

k internetu. Rozšíření o další senzory nebo funkcionality většinou poskytují pouze v omezených případech nebo pomocí domácí centrální jednotky. Značky se tedy soustředí především na běžná prostředí klasických domácností, a ne na jednotlivé mezní situace, které zmiňuje tento projekt. Proto ani tato zařízení nejsou dobře využitelná v rámci tohoto projektu.

Malou výjimkou jsou zařízení značky Sonoff, kterých už je na trhu téměř stovky druhů a v poslední verzi mobilní aplikace eWeLink [14], která je pravidelně několikrát měsíčně aktualizována, mají opravdu poměrně velké množství funkcionalit. Například nově přibýlo i nastavení výchozího stavu vypínače po výpadku napájení apod. Nicméně některé stěžejní funkcionality, které jsou nezbytné pro tento projekt, jim aktuálně stále scházejí.

Hardware modulů značky Sonoff se nakonec využil pro tento projekt. Náhled modulu se nenachází na následujícím obrázku, protože produkty této značky se nacházejí ještě v dalších kapitolách této práce.



**Obrázek 2: Moduly značek Gosund [9], Shelly [12], Woox [10] a iQtech [13]**

### 3.2.2 Porovnání existujících řešení s novým řešením

Úplně každému z analyzovaných existujících řešení chybí nějaký funkční požadavek nebo případ užití, který tento projekt nezbytně potřebuje. Nejvíce však cena, rozšiřitelnost o externí senzory bez centrálního prvku, nastavení výchozího stavu, nebo nastavení stavu po výpadku napájení. Přesně proto vzniklo nové řešení, jehož analýza, návrh, implementace a testování se nachází dále v této práci.

#### Výčet hlavních výhod nového řešení:

- samostatné fungování bez centrálního prvku
- vzdálené ovládání přes internet pomocí Wi-Fi a mobilní aplikace
- automatizované ovládání na základě časového plánu
- podpora přímého připojení více externích senzorů (teplotních čidel)
- automatizované ovládání na základě hodnot senzorů
- mobilní aplikace v češtině pro operační systém Android
- nastavení výchozího stavu a akcí po výpadku internetu nebo napájení
- vzdálená aktualizace firmwaru přes internet

Všechna srovnání v této práci vycházejí z oficiálních webů, e-shopů a technických dokumentací jednotlivých produktů, které jsou volně dostupné na internetu. V rámci tohoto projektu nebyly všechny moduly zakoupeny a osobně vyzkoušeny či otestovány.

V následující tabulce jsou kompletně porovnány funkcionality již existujících řešení spolu s novým řešením. Samostatné fungování modulů bez centrálního prvku a ovládání modulů přes internet pomocí Wi-Fi a mobilní aplikace poskytují úplně všechny moduly. Dále všechny moduly podporují automatizované ovládání podle předem nastaveného plánu, některé dokonce automaticky podle východu a západu slunce. Většina značek je kompatibilní s chytrými hlasovými asistenty jako jsou Google Assistant [15] a Anazon Alexa [16], ale ani jedna značka není kompatibilní se softwarovým frameworkem Apple HomeKit [17]. Pouze dražší značky obsahují mobilní aplikaci pro Android v češtině, ale skoro žádná ze značek nepodporuje přímé přidání externích senzorů, automatizované ovládání podle těchto senzorů a

nastavení akcí po výpadku signálu Wi-Fi nebo připojení k internetu nebo po výpadku napájení. Nové řešení neobsahuje cenu, protože produkt není na prodej, a tak cena nikdy nebyla stanovena. Nicméně každý kus produktu značky Sonoff, který byl objednan ze zahraničí a jehož hardware byl použit v rámci této práce, vyšel i s dopravou do 150 Kč.

**Tabulka 1: Porovnání existujících řešení s novým řešením**

<b>Funkce/zařízení</b>	<b>Gosund [9]</b>	<b>Woox [10]</b>	<b>Sonoff [11]</b>	<b>Shelly [12]</b>	<b>iQtech [13]</b>	<b>Nové řešení</b>
Samostatné fungování	ANO	ANO	ANO	ANO	ANO	ANO
Ovládání přes internet	ANO	ANO	ANO	ANO	ANO	ANO
Podpora časového plánu	ANO	ANO	ANO	ANO	ANO	ANO
Mobilní aplikace v češtině	NE	NE	ANO	ANO	ANO	ANO
Přidání externích senzorů	NE	NE	NE	NE	NE	ANO
Ovládání podle senzorů	NE	NE	NE	NE	NE	ANO
Nastavení po výpadku	NE	NE	NE	NE	NE	ANO
Nastavení výchozího stavu	NE	NE	ANO	NE	NE	ANO
Aktualizace přes internet	NE	NE	NE	NE	NE	ANO
Orientační pořizovací cena	299 Kč	259 Kč	349 Kč	399 Kč	599 Kč	-

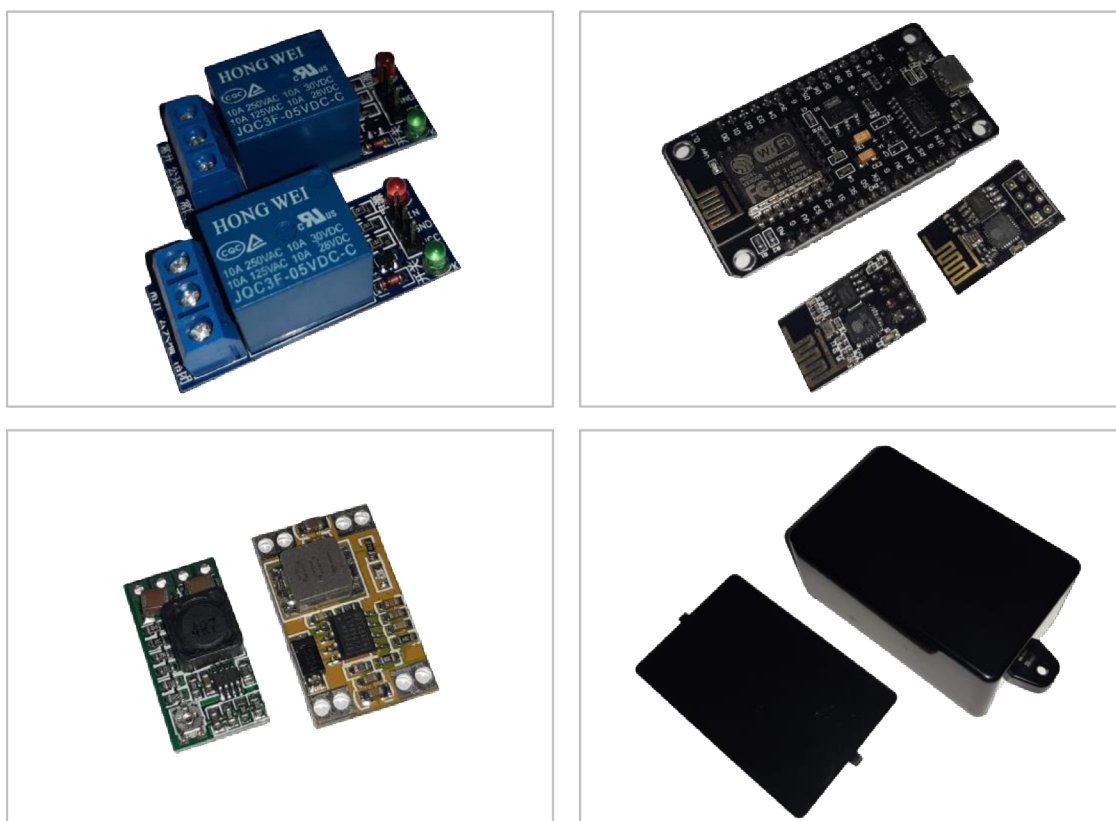
### 3.3 Nové řešení

Z důvodů popsanych v analýze existujících řešení před touto podkapitolou, bylo nezbytně nutné, aby pro tento projekt vzniklo úplně nové řešení. V této podkapitole se nejprve nachází analýza použití nejvhodnějšího hardwaru pro tento projekt. Následně se v této podkapitole nachází analýza nového informačního systému včetně funkčních a nefunkčních požadavků.

Nový informační systém zahrnuje vývoj firmwaru pro nejvhodnější analyzovaný hardware a vývoj mobilní aplikace, která umožňuje uživateli přístup k ovládání a nastavení jeho modulu a to odkudkoliv, kde má uživatel přístup k internetu. V neposlední řadě tento projekt také implementuje serverovou aplikaci, která celý systém zastřešuje tím, že obsahuje komunikační rozhraní pro obě zmíněné platformy a administrační portál, pomocí kterého lze celý systém ovládat a nastavovat.

### 3.3.1 Použitý hardware pro moduly

Původně mělo v rámci tohoto projektu vzniknout hned několik úplně nových hardwarových zařízení, složených z několika menších hardwarových komponent propojených mezi sebou. Mezi tyto komponenty patří například elektromagnetické relé, měnič napětí, ESP Wi-Fi moduly [3], Arduino moduly [18] a další. Tato cesta se ale nakonec ukázala jako zbytečně nákladná, komplikovaná a tím pádem i úplně slepá, protože v rámci předchozí analýzy bylo zjištěno, že zařízení značky Sonoff obsahují všechny hardwarové komponenty, které vyžadují i analyzované požadavky tohoto projektu, a to přímo na jednom tištěném spoji.



**Obrázek 3: Původní hardwarové komponenty modulu**

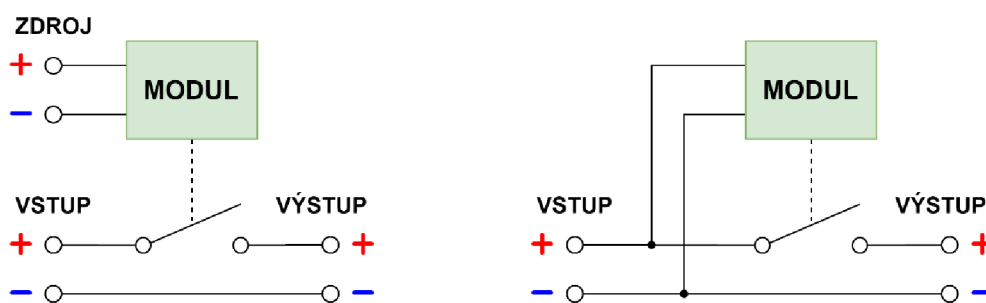
V rámci předchozí analýzy bylo zjištěno, že nejbližší se tomuto projektu podobají svou funkcí a hardwarem zařízení od firmy Sonoff, dokonce patří opravdu i mezi ty nejlevnější. Zařízení od firmy Sonoff lze velmi jednoduše přeprogramovat, a to i v domácích podmínkách. Proto právě zařízení značky Sonoff se staly ideálním hardwarem pro tento projekt.

Pro účely tohoto projektu byly použity dva základní Sonoff moduly, jejichž parametry jsou shrnuty v následující tabulce.

**Tabulka 2: Parametry základních použitých Sonoff modulů**

Parametry/zařízení	Sonoff Basic R2	Sonoff SV
Provozní napětí	90–250 V AC	5–24 V DC
Maximální zátěž	10 A	10 A
Provozní teplota	0–40 °C	0–40 °C
Rozměry	90x41x27 mm	65x34x16 mm
Hmotnost	59 g	15 g
Materiál obalu	Polykarbonát	Bez obalu

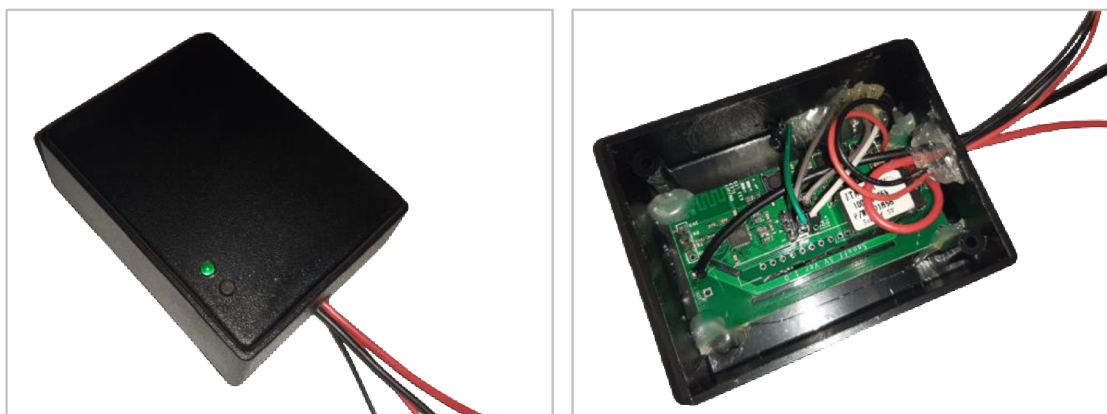
Každý z obou typů modulu umožňuje po v celku jednoduché hardwarové úpravě následující dva druhy zapojení do elektrického obvodu. První ze zapojení má zcela oddělené elektrické napájení od zbytku elektrického obvodu, čímž dokáže spínat obvod s jiným napětím, než ze kterého je napájen. Na druhém schématu zapojení je modul připojen do elektrického obvodu tak, že využívá napájení přímo z elektrického obvodu, který ovládá. Takové zapojení umožňuje modulu spínat jen obvod s takovým napětím, ke kterému je připojen a tento typ zapojení je výchozím zapojením obou typů modulu.



**Obrázek 4: Možnosti zapojení modulu do elektrického obvodu**

Modul Sonoff SV pracuje nízkým stejnosměrným napětím 5–24 V, proto se hodí především do elektronických zařízení, jako jsou například automobily nebo garážová vrata. S takto nízkým stejnosměrným je modul pro člověka naprosto bezpečný, z toho důvodu se prodává výhradně bez krabičky nebo jiného krytu, který

by alespoň částečně zamezoval kontaktu zařízení s jinou elektronikou, nebo jinými vodivými předměty a materiály, jako jsou například kovy nebo voda. Proto byly k těmto modulům nakonec v rámci tohoto projektu i zakoupeny a upraveny plastové krabičky nebo kryty, do kterých byl zabudován LED indikátor a tlačítko na vypnutí, zapnutí, sepnutí nebo přepnutí modulu do jiného režimu.



**Obrázek 5: Vlastnoručně upravený plastový kryt modulu**

Ostatní moduly pracují výhradně se střídavým napětím 90–250 V hodící se např. do domácností, garáží, kanceláří, sklepů a jiné elektroinstalace s vysokým napětím. Nové řešení tedy nakonec nespočívá v návrhu a vývoji hardwaru, ale maximálně v případné úpravě hardwaru modulů Sonoff a v naprogramování firmwaru pro oba výše zmíněné druhy tohoto zařízení. Naprogramováním firmwaru pro tyto dva typy zařízení je pokryta široká škála Sonoff modulů. Konkrétní seznam těchto modulů je uveden u firmwaru v elektronické příloze této práce.

### **3.3.2 Funkční požadavky**

Tato sekce obsahuje analýzu funkčních požadavků informačního systému. Všechny funkční požadavky tohoto IS jsou shrnuty v následující tabulce. U každého funkčního požadavku je v tabulce uveden identifikátor (ID), název požadavku a jeho priorita. Všechny funkční požadavky mají vysokou prioritu, protože všechny je nezbytné navrhnout a implementovat pro správné a pohodlné fungování celého systému. Na všechny funkční požadavky následně přímo navazují případy užití navržené v následující kapitole.

**Tabulka 3: Seznam funkčních požadavků informačního systému**

ID	Název	Priorita
FR1	Základní správa modulů	Vysoká
FR2	Ovládání modulů	Vysoká
FR2	Stav modulů a systému	Vysoká
FR4	Správa připojení a sítí v modulech	Vysoká
FR5	Konfigurace a nastavení modulů	Vysoká

### **FR1 – Základní správa modulů**

První funkční požadavek obsahuje základní CRUD operace nad moduly, jako je například přidání, zobrazení detailu, úprava, nebo odstranění modulu. Správci systému umožňuje přidávat, zobrazovat, editovat a odstraňovat libovolné moduly napříč celým informačním systémem. Běžnému uživateli pak umožňuje provádět CRUD operace pouze nad moduly pouze v rámci jeho mobilní aplikace.

Tento funkční požadavek nepokrývá žádné další operace nad moduly, jako je například ovládání modulů, zjištění stavu modulů, správa sítí v modulech nebo konfigurace, nastavení, aktualizace a restart modulů. Všechny tyto další operace zahrnují až následující funkční požadavky.

### **FR2 – Ovládání modulů**

Tento funkční požadavek obsahuje nejstěžejnější funkci tohoto informačního systému, což je samotné ovládání modulů. Moduly je možné ovládat manuálně, ale i automatizovaně. Manuální ovládání je umožněno hardwarově pomocí tlačítka, které je umístěno přímo na samotném modulu, nebo softwarově všem oprávněným uživatelům skrze internet přes mobilní aplikaci nebo administrační portál. Automatizované ovládání modulů poskytuje ve všech případech samotná serverová aplikace podle nastavené automatizace modulu. Tím mají uživatelé systému nad automatizací plnou kontrolu i v případě nečekané ztráty připojení k internetu nebo výpadku serverové aplikace. Automatizaci lze nastavit na základě kombinace hodnot senzorů modulu nebo na základně časového plánu, kde obě tyto možnosti je možné používat najednou. Při zapnuté automatizaci je možné moduly i nadále ovládat manuálně.



### **FR3 – Stav modulů a systému**

Jednou z dalších velmi důležitých součástí tohoto informačního systému je zobrazení detailního stavu jednotlivých modulů, ale i celého IS. Tento funkční požadavek umožňuje oprávněným uživatelům informace o aktuálně připojené síti, síle signálu a hodnotách všech vstupů i výstupů modulu. Neumožňuje pouze zobrazení těchto informací, nýbrž také jejich kompletní krátkou historii. Správci systému umožňuje zobrazení detailnějších informací a protokolů například o různých pádech nebo restartech modulů apod. Široké veřejnosti poskytuje přehled o dostupnosti celého informačního systému pomocí externí monitorovací služby. Dále je v rámci tohoto funkčního požadavku umožněno uživatelům nastavení upozornění (notifikací) na kritické (prahové) hodnoty senzorů modulu.

### **FR4 – Správa připojení a sítí v modulech**

Vzhledem k tomu, že moduly komunikují s mobilními aplikacemi pouze pomocí Wi-Fi standardů a internetu, potřebují obsahovat správu přístupových bodů (AP). Tento funkční požadavek umožňuje oprávněným uživatelům kompletní a detailní správu těchto přístupových bodů. Tato správa obsahuje modifikovatelný seznam uložených sítí včetně jejich názvů (SSID) a hesel. Dále poskytuje připojení k libovolné síti a v neposlední řadě také získání seznamu nejbližších dostupných sítí včetně jejich SSID, síly signálu (RSSI), BSSID, kanálu a typu zabezpečení. Dále je umožněno přidat nový přístupový bod i v off-line režimu (bez přístupu k internetu), pouze pomocí mobilní aplikace, která je napřímo připojená k příslušnému modulu.

### **FR5 – Konfigurace a nastavení modulů**

Aby mohl pro všechny typy modulů existovat pouze jeden firmware, je stěžejní zajistit pro správce systému možnost konfigurace firmwaru. Správce systému tedy před prvním přeprogramováním modulu (ale i kdykoliv později pomocí internetu), může nakonfigurovat hardwarové komponenty modulu. Běžný uživatel pak může skrze mobilní aplikaci jednotlivě nakonfigurované komponenty softwarově vypínat nebo zcela úplně měnit jiné nastavení modulu, jako jsou například výchozí hodnoty výstupů modulu, frekvence aktualizace stavu apod. Dále tento funkční požadavek umožňuje vzdálenou aktualizaci firmwaru modulu přes

internet, vzdálený restart modulu (v případě jakýchkoliv potíží) a správcům systému vydávat nové aktualizace firmwaru.

### 3.3.3 Nefunkční požadavky

V této sekci se nachází analýza nefunkčních požadavků nového informačního systému. V následující tabulce je obsažen kompletní seznam všech nefunkčních požadavků, kde u každého požadavku je uveden identifikátor (ID), jeho název a jeho priorita. Nejvyšší priorita je u tohoto projektu kladena především a pouze na bezpečnost a spolehlivost celého informačního systému, všem ostatním nefunkčním požadavkům byla přidělena priorita střední.

**Tabulka 4: Seznam nefunkčních požadavků informačního systému**

ID	Název	Priorita
NR1	Výkon a optimalizace	Střední
NR2	Dostupnost a zálohování	Střední
NR3	Udržitelnost a rozšiřitelnost	Střední
NR4	Bezpečnost a spolehlivost	Vysoká

#### NR1 – Výkon a optimalizace

První nefunkční požadavek se svou střední prioritou zajistí načítání libovolných požadavků na REST API [19] serverové aplikace do 1 sekundy. Všechny koncové body obsahující různé delší seznamy (např. seznam modulů, nebo historii stavu modulu apod.) by měly kompletně reagovat vždy alespoň do 2 sekund. Na koncové body přistupují uživatelé pouze prostřednictvím mobilní aplikace nebo administračního portálu, takže vždy je možné libovolným způsobem data z REST API načítat dopředu, ještě před vyvoláním konkrétní akce uživatelem nebo zobrazit na krátkou dobu nějakou animaci načítání. Všechny výkonnostní testy jsou podrobněji popsány a pokryty v šesté kapitole této práce s názvem „Testování“.

#### NR2 – Dostupnost a zálohování

Serverová aplikace, která zastřešuje téměř úplně všechny funkcionality celého informačního systému, by měla být vždy dostupná alespoň 99,9 % celkového času (za posledních 90 dní), včetně veškerých plánovaných odstávek systému pro

údržbu, aktualizaci apod. Dostupnost je monitorována pomocí externí služby [20], aby monitoring nebyl ovlivňován výpadky serveru, na kterém aplikace běží.

Zálohování veškerého programového kódu zajišťuje samotný repozitář verzovacího systému Git [21]. Veškeré ostatní soubory, které nejsou v repozitáři jsou soubory, které lze znovu vygenerovat. Databáze serverové aplikace je pravidelně a kompletně zálohována jedním z modulů serverové aplikace, a to v nočních hodinách v denních intervalech. O veškeré zálohování souborů i databáze se rovněž stará (na týdenní bázi) i samotný webhosting [22], který ovšem používá tyto zálohy pouze pro své vlastní účely v případě havárie části serveru apod. Testování dostupnosti je taktéž ještě konkrétněji popsáno v kapitole „Testování“ na konci této práce.

### **NR3 – Rozšiřitelnost**

Stejně jako většina informačních systémů i tento IS je schopen poměrně dobře přijímat nové aktualizace, a to bez ovlivnění stávajících požadavků a funkcí. Historie GitLab repozitáře [23] a postupné vydávání aktualizací mobilní aplikace na platformu Google Play [24] dokazuje to, že systém byl v provozu na produkci již od samého počátku (kdy měl pouze několik málo API koncových bodů) a byl za provozu po celou dobu postupně rozšiřován. Systém je naprogramovaný napříč více různými softwarovými moduly, které lze za provozu aplikace zapínat, vypínat či vyměňovat. Jiný softwarový modul je například na obsluhu komunikačního rozhraní, jiný je na blokování nežádoucích požadavků apod.

### **NR4 – Bezpečnost**

Všechny komponenty v celém informačním systému spolu komunikují pomocí bezpečného protokolu HTTPS [25] a autentizačních tokenů, které generuje samotná serverová aplikace. Tímto je zajištěna bezpečnost celé komunikace napříč všemi komponentami IS. Toto je bez pochyby nejdůležitější nefunkční požadavek, a proto na něj byl kladen největší důraz. Webhosting [26] chrání serverovou aplikaci před DDoS [27] a jinými útoky, ale i samotný informační systém je chráněn tak, že pokud dojde k několikanásobnému neplatnému použití autentizačního tokenu z jedné IP adresy, je tato adresa dočasně zablokována. Libovolnou IP adresu může správce systému zablokovat i natrvalo přes administrační portál.

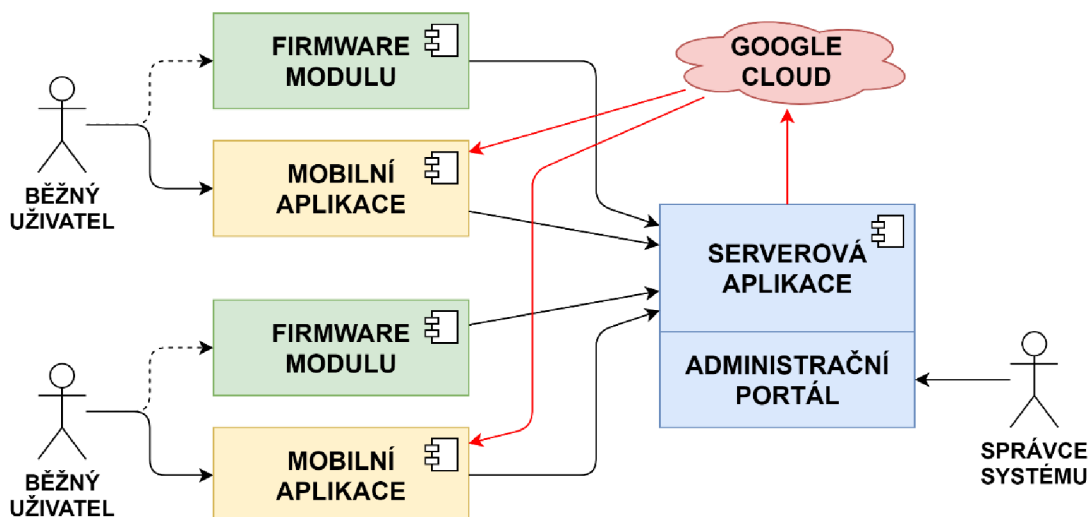
## 4 Návrh

V této kapitole se nachází kompletní návrh nově vzniklého informačního systému. Nejprve je zde detailně popsán nový informační systém včetně všech jeho komponent a uživatelských rolí. Dále byly v rámci této teoretické části práce navrženy případy užití a databázové schéma serverové aplikace. Jako poslední se nachází na konci této kapitoly návrh komunikačního rozhraní serverové aplikace a návrh automatizovaných úloh, které probíhají na pozadí tohoto systému.

### 4.1 Komponenty informačního systému

Nový informační systém se skládá z celkem tří základních softwarových komponent. Všechny tyto komponenty jsou nedílnou součástí celého informačního systému a absence jakékoliv z nich vede k jeho nesprávné funkčnosti. Základními komponentami jsou „Firmware modulu“, „Mobilní aplikace“ a „Serverová aplikace“. Všechny tyto komponenty byly analyzovány, navrženy, implementovány a testovány v rámci této diplomové práce. Každá z těchto komponent se reálně nachází na úplně jiném typu hardwaru a každá má úplně jiný účel, tudíž pro každou z těchto komponent bylo vhodné použít zcela unikátní řešení, čisto diametrálně odlišné od ostatních. Přesný průběh implementace jednotlivých komponent je popsán v páté kapitole této práce s názvem „Implementace“.

Na následujícím obrázku se nachází schéma nového informačního systému, ve kterém jsou znázorněny všechny komponenty i včetně základní komunikace, která mezi nimi probíhá. Nedílnou součástí informačního systému je později přidaná komponenta „Google Cloud“ [28], která umožňuje informačnímu systému odesílat push notifikace [29] ze serverové aplikace do mobilních aplikací, tedy na mobilní zařízení běžných uživatelů. Tok komunikace pomocí těchto push notifikací je v navrženém schématu znázorněn červenou šipkou. Každá mobilní aplikace si po svém prvním spuštění automaticky vyžádá z Google Cloud serveru registrační token, který následně odešle do serverové aplikace. Tento token umožňuje serverové aplikaci odesílat push notifikace dané konkrétní mobilní aplikaci nebo na dané mobilní zařízení. Komunikace, která zajišťuje tento registrační token, nebyla v rámci zachování přehlednosti schématu znázorněna.



**Obrázek 6: Přehled komponent informačního systému**

Běžný uživatel primárně přistupuje k informačnímu systému pomocí mobilní aplikace, může ale přistupovat k informačnímu systému i pomocí firmwaru modulu. Na každém z modulů se totiž zpravidla fyzicky nachází hardwarové tlačítko, díky kterému může běžný uživatel modul zapnout, vypnout nebo sepnout. Po stisknutí hardwarového tlačítka se změna automaticky projeví a aplikuje v úplně celém informačním systému, stejně jako kdyby uživatel zapnul, vypnul nebo sepnul modul pomocí mobilní aplikace. Běžných uživatelů, modulů a mobilních aplikací se může teoreticky v systému nacházet neomezené množství, nicméně ve schématu byly pro stručnost vyobrazeny pouze dva zástupci od každého z nich. Jeden běžný uživatel může ovládat úplně stejné hardwarové moduly i z více jak jedné mobilní aplikace, nebo i z více jak jednoho mobilního zařízení.

Centrem celého informačního systému je serverová aplikace, která obsahuje komunikační rozhraní pro všechny ostatní komponenty IS. Správce systému přistupuje ke správě informačního systému skrze administrační portál, který je nedílnou součástí přímo samotné serverové aplikace. Jedinou výjimkou, kdy komunikace neprobíhá skrze serverovou aplikaci je první spuštění modulu, kdy je třeba přes mobilní aplikaci nahrát název přístupového bodu (SSID) a heslo přímo do modulu v off-line režimu, protože modul v tuto chvíli ještě není připojen k internetu, a tudíž ani nemůže být připojen k serverové aplikaci. Tato komunikace pro lepší přehlednost nebyla znázorněna v přehledu komponent informačního systému.

#### **4.1.1 Firmware modulu**

Firmware modulu je speciální program nebo operační systém, který se nachází a běží přímo v rámci hardwaru jednotlivých modulů. Tento firmware umožňuje obsluhovat dostupné hardwarové komponenty chytrého modulu a tím plnit požadavky přicházející ze serverové aplikace. Jedním z nejdůležitějších požadavků je čtení vstupů a ovládání výstupů. Díky tomu dokáže modul zapínat, vypínat, nebo spínat elektromagnetické relé, pracovat s LED indikátory, nebo zjistit například teplotu z externích digitálních čidel apod. Další hlavní požadavek je práce s Wi-Fi komponentou, tedy připojovat se k přístupovému bodu a komunikovat přes internet. Díky tomu dokáže firmware přijímat požadavky a odesílat data. Například automaticky odesílat svůj stav a data ze senzorů do serverové aplikace apod. V oblasti firmwaru je cílem této práce, aby vznikl pouze jeden univerzální firmware, který bude použitelný pro větší škálu různých hardwarových modulů s různými senzory, a to pouze pomocí konfigurace bez zásahu do jeho programového kódu.

#### **4.1.2 Mobilní aplikace**

Přes mobilní aplikaci přistupují na komunikační rozhraní serverové aplikace primárně pouze běžní uživatelé. Všem běžným uživatelům má mobilní aplikace za úkol umožňovat veškerou a rychlou správu všech jejich dostupných modulů. Tato mobilní aplikace by měla být vyvíjena multiplatformní metodou, ale měla by být vyprodukována především pro operační Android, neboť reální uživatelé právě vyžadují aplikaci zatím pouze pro tuto platformu.

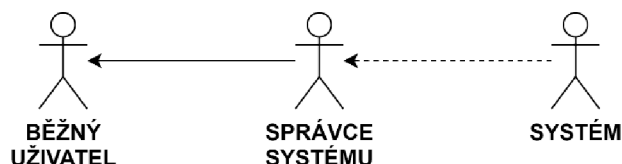
Tato komponenta je hlavní prezentací tohoto projektu. Proto by měla obsahovat kvalitní a jednoduché grafické uživatelské rozhraní (GUI). V rámci tohoto GUI by určitě měla umožňovat volbu téma vzhledu (světlý nebo tmavý režim), nastavení zobrazení modulů (mřížka nebo seznam), nastavení zabezpečení vstupu do samotné mobilní aplikace (pomocí číselného kódu nebo pomocí otisku prstu), zobrazení nápovědy k modulům apod. Kompletní náhled vzniklého grafického uživatelského rozhraní mobilní aplikace se nachází v předposlední příloze na konci této teoretické části diplomové práce.

### 4.1.3 Serverová aplikace

Tato komponenta je středem celého informačního systému, protože obsahuje komunikační rozhraní pro všechny ostatní hlavní komponenty, tedy nejen pro všechny moduly, ale i pro všechny mobilní aplikace. Serverová aplikace navíc obsahuje i administrační portál, do kterého se může přihlásit správce systému a díky kterému může tento uživatel celý systém spravovat. Neméně důležitou součástí serverové aplikace je taktéž databáze, která obsahuje veškerá nejdůležitější data. Dále na pozadí serverové aplikace probíhají automatizované úlohy, které nejen že automaticky obsluhují moduly, ale i udržují systém, zabezpečují komunikaci a pravidelně zálohují databázi. Komunikační rozhraní, databáze i automatizované úlohy jsou ještě dále podrobněji popsány na konci této kapitoly.

## 4.2 Uživatelské role informačního systému

Jako v každém informačním systému i v tomto se používají uživatelské role. Každému uživateli je přidělena právě jedna z následujících rolí. Jednotlivé uživatelské role jsou si navzájem různě nadřazené. Hierarchie nadřazenosti uživatelských rolí je znázorněna na následujícím obrázku. Jinými slovy, správce systému má všechna oprávnění jako běžný uživatel a uživatel s rolí „System“ má úplně všechna dostupná oprávnění IS, ještě více oprávnění, než kolika oprávněními disponuje správce systému. Za uživatelskou rolí „System“ se neskrývá žádný fyzický uživatel. Tato uživatelská role je virtuální uživatel celého informačního systému. Hlavním užitím uživatelské role „System“ jsou pouze automatizované úlohy, které běží na pozadí informačního systému, tedy tento uživatel vznikl především pro zachování přehlednosti případů užití, které jsou popsány v této práci přímo v následující podkapitole.



Obrázek 7: Hierarchie uživatelských rolí v informačním systému

### **4.2.1 Správce systému**

Uživatel s rolí „Správce systému“ může do systému vstupovat pouze pomocí administračního portálu, který je nedílnou součástí serverové aplikace. Tento portál umožňuje správcům systému nastavovat a konfigurovat celý informační systém, a to včetně registrace nových modulů, přidávání nových aktualizací firmwaru modulů, veškeré správy a ovládání všech dostupných modulů i mobilních aplikací.

### **4.2.2 Běžný uživatel**

Běžný uživatel může vstupovat do systému pouze pomocí mobilní aplikace. Mobilní aplikace zpřístupňuje běžným uživatelům kompletní správu a ovládání jejich hardwarových modulů, a to vzdáleně pomocí internetu. Při prvním spuštění modulu umožňuje mobilní aplikace také nahrát první Wi-Fi přístupový bod do modulu, a to napřímo v off-line režimu (bez přístupu k internetu).

### **4.2.3 Systém**

V rámci zachování celkové přehlednosti informačního systému byla navržena speciální virtuální uživatelská role „Systém“, která nereprezentuje žádného fyzického uživatele, nýbrž naopak reprezentuje automatické akce celého informačního systému. Uživatelská role „Systém“ tedy na pozadí může například zasahovat do ovládání modulů, automaticky zasílat upozornění na kritické hodnoty senzorů pomocí push notifikací do dané konkrétní mobilní aplikace nebo na dané mobilní zařízení. Dále také umožňuje provádět údržbu a optimalizace informačního systému, zajišťuje bezpečnost komunikace nebo historii dat, které pravidelně moduly odesílají do serverové aplikace.

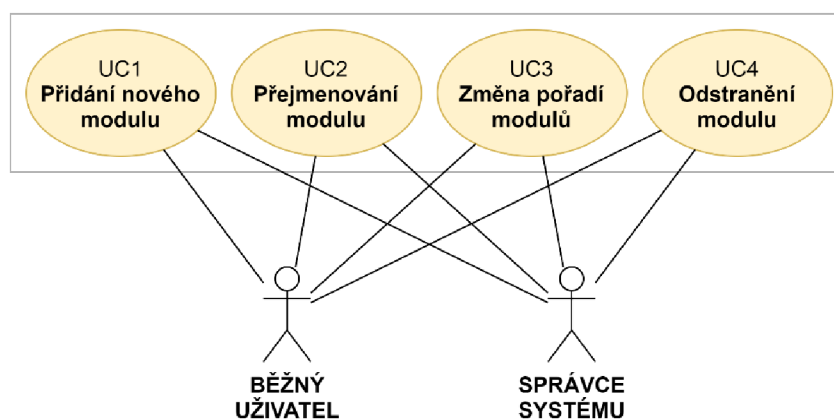
## **4.3 Případy užití**

V této podkapitole se nachází zcela vyčerpávající seznam celkem dvaceti případů užití, které jsou implementovány v praktické části této práce. Tyto případy užití přímo navazují na analyzované funkční požadavky z předchozí kapitoly. Každá sekce této kapitoly shrnuje právě jeden analyzovaný funkční požadavek. U každého případu užití je uveden identifikátor (ID), název, seznam uživatelů, cíl a případně ještě navíc jeho předpoklady.



### 4.3.1 Základní správa modulů

V této sekci se nachází seznam případů užití pokrývající funkční požadavek FR1 „Základní správa modulů“. Tyto případy užití jsou dostupné pro správce systému a pro běžné uživatele. Správce systému má skrze administrační portál kompletní plnou kontrolu nad všemi moduly v informačním systému. Běžný uživatel může spravovat pouze moduly v rámci své mobilní aplikace, ke kterým vlastní přístupové kódy. Základní správa modulů, kterou provádí běžný uživatel, probíhá pouze v rámci jedné mobilní aplikace a není nikam jinam synchronizována. Tato kapitola se týká pouze základních CRUD operací nad moduly. Ovládání modulů, jejich stav, konfigurace, nastavení, restart, aktualizace a další jsou obsaženy až v následujících sekcích této podkapitoly.



Obrázek 8: Případy užití základní správy modulů

#### UC1 – Přidání nového modulu

**Uživatelé:** Správce systému a běžný uživatel

**Cíl:** Umožňuje správci systému zaregistrovat nový modul do systému (neboli vygenerovat nové přístupové kódy a potřebné databázové tabulky). Běžnému uživateli pak umožňuje přidat existující modul v IS do jeho mobilní aplikace. Přidat nový modul do mobilní aplikace lze naskenováním QR kódu nebo manuálním zadáním ovládacího kódu. QR kód nebo ovládací kód je vždy pevně vázán s konkrétním modulem a vždy je s ním automaticky dodáván. Správce systému je oprávněn v krajních případech ovládací kód změnit a poslat běžnému uživateli nový.

## **UC2 – Přejmenování modulu**

**Uživatelé:** Správce systému a běžný uživatel

**Cíl:** Umožňuje běžnému uživateli přejmenovat modul v jeho mobilní aplikaci. Jeden modul může mít napříč mobilními aplikacemi různý název. Správce systému může v administračním portálu moduly také přejmenovávat, avšak na názvy v mobilních aplikacích běžných uživatelů to nemá žádný vliv. Správce systému navíc může pro snadnější identifikaci přidávat k modulům formátované textové poznámky.

## **UC3 – Změna pořadí modulů**

**Uživatelé:** Správce systému a běžný uživatel

**Cíl:** Umožňuje běžnému uživateli změnit pořadí přidanych modulů v jeho mobilní aplikaci. Správce systému může seřadit moduly v administračním portálu podle dostupných kritérií, např. podle jejich názvu, nebo podle času posledního ovládní.

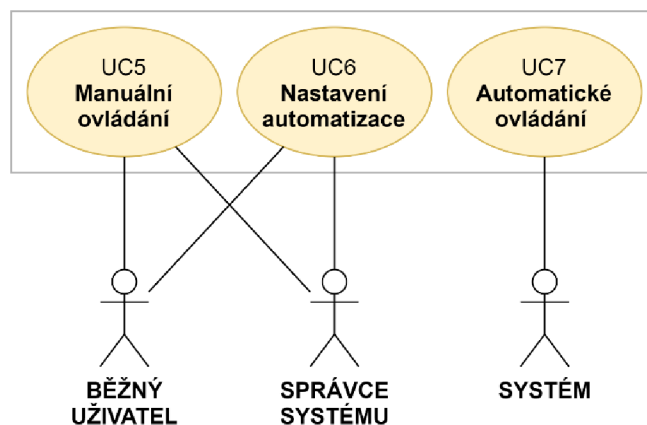
## **UC4 – Odstranění modulu**

**Uživatelé:** Správce systému a běžný uživatel

**Cíl:** Umožňuje správci systému odstranit modul z celého informačního systému, takový modul přestane být funkční i pro běžné uživatele v jejich mobilních aplikacích. Běžnému uživateli umožňuje odstranit modul pouze z jeho jedné konkrétní mobilní aplikace, tedy po odstranění modulu z jedné mobilní aplikace se neodstraní z jiných.

### **4.3.2 Ovládání modulů**

Tato sekce obsahuje případy užití navazující na analyzovaný funkční požadavek FR2 „Ovládání modulů“. Tyto případy užití jsou dostupné pro všechny uživatelské role. Správce systému má skrze administrační portál plnou kontrolu nad manuálním ovládním i nastavením automatizace pro všechny dostupné moduly v celém informačním systému. Běžný uživatel může manuálně ovládat nebo spravovat nastavení automatizace pouze pro moduly v rámci své mobilní aplikace, a ke kterým vlastní přístupové kódy. Uživatel „Systém“ poté automaticky ovládá všechny moduly v informačním systému podle nastavení jejich automatizace podle hodnot senzorů nebo podle předem nastaveného časového plánu.



**Obrázek 9: Případy užití ovládání modulů**

### **UC5 – Manuální ovládání**

**Uživatelé:** Správce systému a běžný uživatel

**Cíl:** Umožňuje ovládat modul, tedy zapnout, vypnout nebo na krátkou dobu sepnout jeho elektromagnetické relé. Ovládat modul je možné vzdáleně nebo fyzicky. Pro fyzické ovládání modulu se musí uživatel osobně nacházet u modulu a modul musí disponovat nezakázaným hardwarovým tlačítkem.

### **UC6 – Nastavení automatizace**

**Uživatelé:** Správce systému a běžný uživatel

**Cíl:** Umožňuje nastavit automatizaci ovládání modulu. Automatizaci lze nastavit na základě hodnot senzorů (momentálně moduly podporují přidání neomezeného množství teplotních senzorů), nebo na základě časového plánu (rozvrhu). Automatizaci lze libovolně kombinovat i s manuálním ovládáním.

### **UC7 – Automatické ovládání**

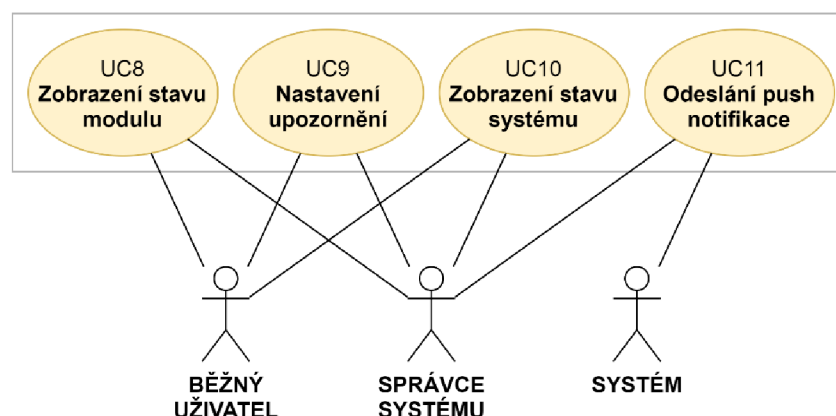
**Uživatelé:** Systém

**Cíl:** Umožňuje automaticky vzdáleně ovládat elektromagnetické relé modulu podle předem nastavené automatizace. Automatické ovládání je možné nastavit podle prahových hodnot teplotních senzorů nebo podle týdenního časového plánu.

**Předpoklady:** Modul musí mít správcem systému nebo běžným uživatelem zapnuté automatické ovládání podle hodnot senzorů nebo podle časového plánu.

### 4.3.3 Stav modulů a systému

V této sekci se nacházejí případy užití pokrývající analyzovaný funkční požadavek FR3 „Stav modulů a systému“. Tyto případy užití jsou dostupné pro správce systému i pro běžné uživatele. Správce systému má skrze administrační portál kompletní plnou kontrolu nad všemi moduly v informačním systému. Běžný uživatel může spravovat pouze moduly ve své mobilní aplikaci, ke kterým vlastní přístupové kódy. Veškeré stavy modulů se pravidelně ukládají do databáze informačního systému, a díky tomu lze sledovat i jejich historii. V rámci historie se uchovává za každý modul pouze několik tisíc záznamů a pouze záznamy za uplynulé období několika týdnů.



Obrázek 10: Případy užití stavu modulů a systému

#### UC8 – Zobrazení stavu modulu

**Uživatelé:** Správce systému a běžný uživatel

**Cíl:** Umožňuje zobrazit stav modulu (zapnutý/vypnutý) a jeho historii. Dále poskytuje stav sítě, tedy název připojeného Wi-Fi přístupového bodu a sílu signálu, včetně délky dostupnosti (nebo nedostupnosti) modulu i včetně historie. Dále zobrazuje hodnoty senzorů modulu (pokud nějakými disponuje) a jejich krátkou historii včetně grafu. Graf hodnot je zobrazen pouze v mobilní aplikaci. Správci systému dále umožňuje zobrazit aktuální stav všech modulů v systému, a navíc poskytuje počet restartování modulu v daných intervalech, což automaticky poukazuje na případné kritické chyby nebo výpadky napájení modulu.

### **UC9 – Nastavení upozornění**

**Uživatelé:** Správce systému a běžný uživatel

**Cíl:** Umožňuje nastavit upozornění na kritické hodnoty senzorů modulu. Ke každému modulu lze uložit textovou zprávu, která se odešle pomocí push notifikace do všech mobilních aplikací (nebo na všechny mobilní zařízení), které daný modul v sobě obsahují.

**Předpoklady:** Modul musí mít správcem systému nebo běžným uživatelem nastavené a zapnuté upozornění na prahové hodnoty teplotních senzorů včetně textové zprávy, která se má v případě překročení těchto prahových hodnot odeslat.

### **UC10 – Zobrazení stavu systému**

**Uživatelé:** Správce systému a běžný uživatel

**Cíl:** Umožňuje uživatelům zobrazit aktuální stav celého informačního systému včetně jeho historie. Aktuální stav i jeho historie se načítají z externí monitorovací služby, která serverovou aplikaci zdarma kontroluje každých 5 minut.

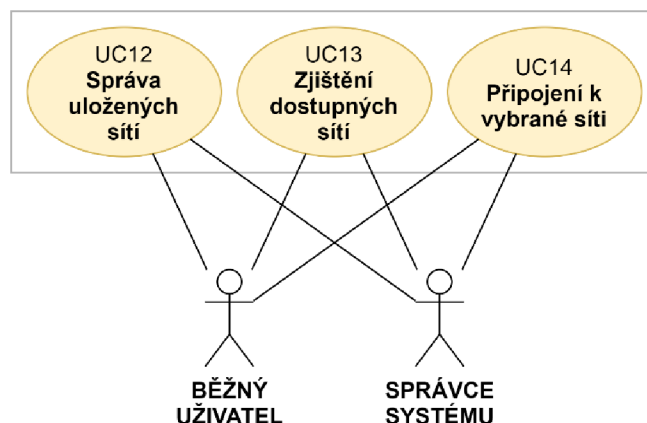
### **UC11 – Odeslání push notifikace**

**Uživatelé:** Systém a správce systému

**Cíl:** Umožňuje odeslat push notifikaci do konkrétní mobilní aplikace (pomocí GCM tokenu). Správci systému umožňuje zobrazit všechny mobilní aplikace a poslat do libovolné mobilní aplikace push notifikaci s libovolnou zprávou.

## **4.3.4 Správa připojení a sítí v modulech**

Tato sekce obsahuje seznam případů užití navazující na analyzovaný funkční požadavek FR4 „Správa připojení a sítí v modulech“. Tyto případy užití jsou dostupné pro správce systému i pro běžné uživatele. Správci systému umožňuje spravovat připojení a Wi-Fi sítě v úplně všech dostupných modulech v celém informačním systému. Běžnému uživateli umožňuje spravovat připojení a Wi-Fi sítě pouze v modulech, ke kterým má přístup skrze nainstalovanou mobilní aplikaci. Moduly umožňují vzdálené uložení nových Wi-Fi sítí (přístupových bodů), zjištění dostupných sítí nebo připojení ke konkrétní vybrané síti. Uložení nových Wi-Fi sítí (přístupových bodů) moduly umožňují i v off-line režimu po přímém napojení mobilní aplikace.



**Obrázek 11: Případy užití správy připojení a sítí v modulech**

### **UC12 – Správa uložených sítí**

**Uživatelé:** Správce systému a běžný uživatel

**Cíl:** Umožňuje spravovat Wi-Fi přístupové body (AP) v modulu (přes tyto AP se modul připojuje k internetu). Běžnému uživateli umožňuje uložit přístupový bod i v off-line režimu. Tato funkcionality se může hodit například v případě, že je modul spuštěn úplně poprvé (s čistou pamětí) a nemá zatím uložené žádné přístupové body. Dokud není uložena alespoň jedna Wi-Fi síť (přístupový bod), neexistuje žádný způsob, pomocí kterého by se modul mohl připojit k internetu.

### **UC13 – Zjištění dostupných sítí**

**Uživatelé:** Správce systému a běžný uživatel

**Cíl:** Poskytuje zjištění aktuálně dostupných Wi-Fi přístupových bodů v okolí modulu a jejich podrobných informací (SSID, síla signálu, BSSID, kanál a typ zabezpečení).

### **UC14 – Připojení k vybrané síti**

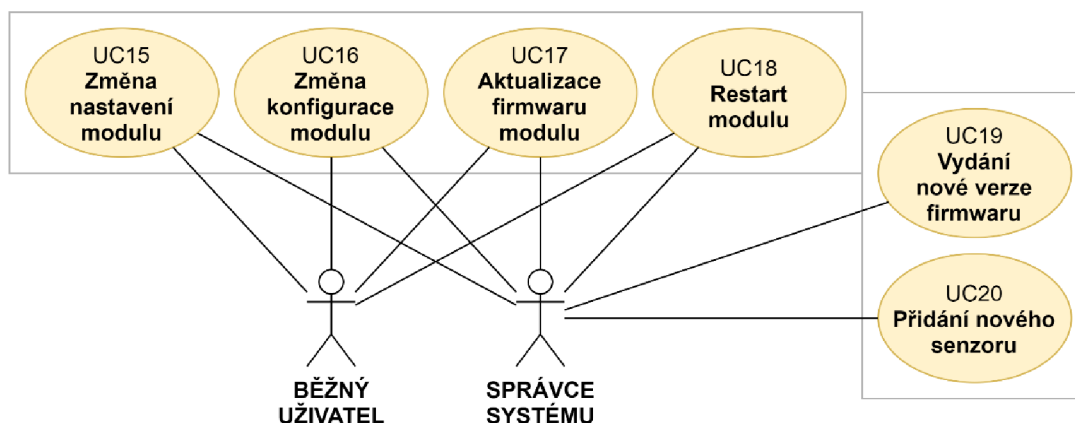
**Uživatelé:** Správce systému a běžný uživatel

**Cíl:** Umožňuje připojení ke zvolenému přístupovému bodu (pomocí SSID a hesla).

**Předpoklady:** Moduly umožňují připojení pouze k sítím bez zabezpečení nebo s běžným typem zabezpečení (WEP, WPA nebo WPA2) [30]. Pokud například síť vyžaduje pro přihlášení přihlašovací jméno a heslo, modul takový typ zabezpečení nerozpozná a není schopen se k takové síti připojit.

### 4.3.5 Konfigurace a nastavení modulů

V této sekci se nacházejí případy užití, které pokrývají analyzovaný funkční požadavek FR5 „Konfigurace a nastavení modulů“. Tyto případy užití jsou dostupné jak pro správce systému, tak i pro běžné uživatele. Správci systému umožňují konfigurovat a nastavovat moduly v celém informačním systému. Běžným uživatelům umožňují nastavovat a konfigurovat moduly pouze v rámci jejich instance mobilní aplikace. Správci systému dále umožňují vydat do informačního systému nový firmware modulu nebo zaregistrovat nový hardwarový senzor, který následně může být nakonfigurován k libovolnému z dostupných modulů. Běžný uživatel i správce systému poté může vzdáleně pomocí internetu aktualizovat firmware ve vybraných modulech a hardwarové senzory může běžný uživatel u svých modulů pouze softwarově zapínat nebo vypínat.



Obrázek 12: Případy užití konfigurace a nastavení modulů

#### UC15 – Změna nastavení modulu

**Uživatelé:** Správce systému a běžný uživatel

**Cíl:** Umožňuje změnit nastavení modulu včetně módu (vypínač nebo tlačítko), výchozího stavu elektromagnetického relé, pravidelného intervalu spojení modulu se serverovou aplikací, maximální doby zapnutí elektromagnetického relé, nebo nastavení doby krátkého sepnutí. Správce systému může nastavovat všechny moduly v rámci celého informačního systému přes administrační portál. Běžný uživatel může plně nastavovat pouze vlastní moduly přes mobilní aplikaci.

### **UC16 – Změna konfigurace modulu**

**Uživatelé:** Správce systému a běžný uživatel

**Cíl:** Umožňuje správci systému softwarově přidávat (nebo odebrat) hardwarové komponenty modulu. Běžnému uživateli umožňuje pouze softwarově zapínat (nebo vypínat) jednotlivé hardwarové komponenty modulů přes jeho mobilní aplikaci.

### **UC17 – Aktualizace firmwaru modulu**

**Uživatelé:** Správce systému a běžný uživatel

**Cíl:** Umožňuje uživatelům vzdáleně přes internet aktualizovat firmware vybraných modulů na dostupnou nejnovější verzi.

**Předpoklady:** Nová verze firmwaru musí být vydaná správcem systému.

### **UC18 – Restart modulu**

**Uživatelé:** Správce systému a běžný uživatel

**Cíl:** Umožňuje uživatelům provést vzdálený restart modulu. Tento případ užití se může hodit v případě výskytu nějakého softwarového problému apod. Většinou při kritické chybě firmware modulu provede restart zcela automaticky. Tyto restarty modulů jsou monitorovány a jsou dostupné správcům systému.

### **UC19 – Vydání nové verze firmwaru**

**Uživatelé:** Pouze správce systému

**Cíl:** Umožňuje správci systému přidat novou verzi firmwaru do informačního systému a umožnit tak běžným uživatelům aktualizaci jejich modulů.

**Předpoklady:** Firmware musí být do informačního systému uložen ve zkompileované binární verzi připravený pro přímé nahrání do paměti modulu.

### **UC20 – Přidání nového senzoru**

**Uživatelé:** Pouze správce systému

**Cíl:** Umožňuje správci systému zaregistrovat do informačního systému údaje o novém hardwarovém senzoru, který následně může být přidán (nakonfigurován) k libovolnému dostupnému modulu. Aktuálně jsou přidávány k modulům pouze teplotní senzory, avšak po malé úpravě informačního systému mohou být přidávány k modulům i jiné. Maximální počet teplotních senzorů, které mohou být přidány k jednomu modulu není omezen.



## 4.4 Databáze serverové aplikace

Tato podkapitola obsahuje návrh databáze serverové aplikace. Jedná se pouze o návrh, který byl v rámci implementace ještě do značné míry upraven kvůli zachování standardů a nejlepších praktik použitých technologií.

### 4.4.1 Databázový model

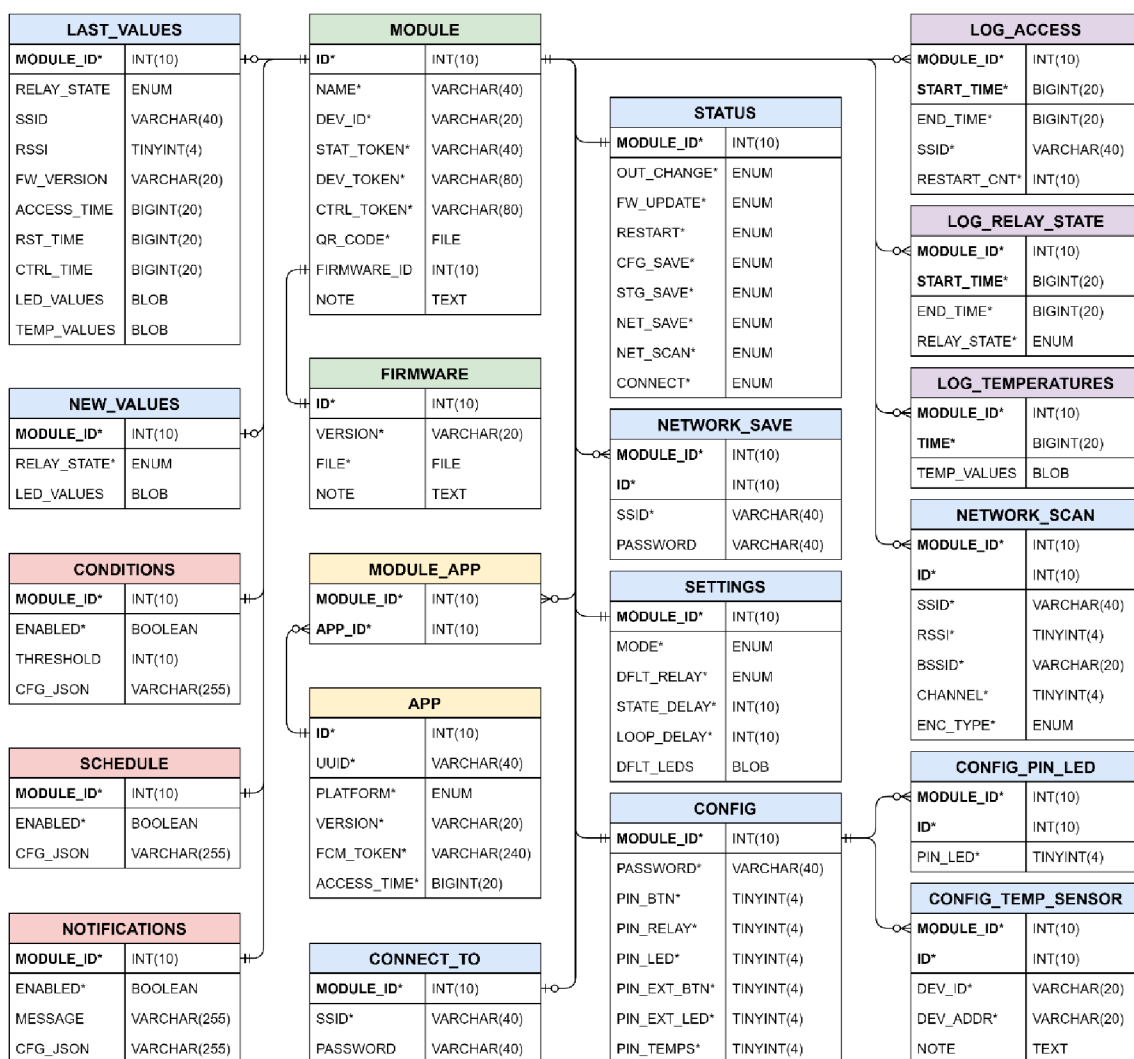
Hlavní entitou databáze je tabulka modulů s názvem „MODULE“. Vzhledem ke komplexnosti modulů je většina ostatních databázových tabulek pouze jejím „one-to-one“ rozšířením. V následujícím databázovém modelu jsou tabulky modulů a jejich firmwarů označeny zelenou barvou. Modrou barvou jsou znázorněny tabulky, které tabulku modulů primárně rozšiřují. Fialovou barvou jsou označeny tabulky, které protokolují nejrůznější stavy modulů, jako jsou například časy dostupnosti a hodnoty vstupů a výstupů. Červenou barvou jsou zvýrazněny tabulky, které slouží k ukládání nastavení automatizace. V informačním systému existuje automatizace podle hodnot senzorů, jejíž nastavení se nachází v tabulce „CONDITIONS“. Dále také automatizace podle časového plánu, jejíž nastavení se nachází v tabulce „SCHEDULE“. V tabulce s názvem „NOTIFICATIONS“ se nachází nastavení automatizace pro zasílání push notifikací do mobilních aplikací (nebo na mobilní zařízení). Tabulky související s mobilními aplikacemi jsou v následujícím modelu označeny barvou žlutou.

Moduly (tabulka „MODULES“) a mobilní aplikace (tabulka „APPS“) jsou spolu ve vztahu „many-to-many“, tedy jedna mobilní aplikace může obsahovat více modulů a zároveň jeden modul se může nacházet ve více jak jedné mobilní aplikaci. Tuto skutečnost umožňuje tabulka s názvem „MODULE\_APP“. Relace (vztahy) mezi tabulkami jsou v následujícím modelu znázorněny pomocí Crow's Foot notace [31] a pro zachování přehlednosti byly vynechány jejich popisky.

Dále v rámci této práce vznikly ještě další tabulky, například tabulky pro automatické blokování uživatelů. Tyto tabulky pro zachování přehlednosti nejsou v následujícím databázovém modelu znázorněny. Seznam všech 22 navržených tabulek se nachází v poslední příloze na konci této práce a obsahuje úplně všechny tabulky, které vznikly v rámci tohoto návrhu. U každé tabulky je uveden název, seznam atributů, jejich typ, vlastnosti a popis. Většina databázových tabulek

obsahuje číselný primární klíč, který je automaticky generovaný při vkládání nového datového záznamu. Všechny textové atributy jsou doplněny o maximální počet znaků a stejně tak jsou upřesněny i rozsahy číselných atributů. Pro časové hodnoty primárně platí, že se ukládají v číselném formátu „Unix timestamp“ [32] s přesností na jednotky sekund.

Vybraný typ technologie obsahuje i další možnosti uložení dat, například „key-value“ [33] uložení, kterých serverová aplikace taktéž využívá. Tyto hodnoty ale ve skutečnosti (bez zprovoznění tohoto typu databáze) mohou být rovněž uloženy do speciální relační tabulky, což je i případ tohoto projektu. Takovéto relační tabulky, které automaticky vznikají použitou technologií (například tabulka uživatelů), v následujícím modelu rovněž nejsou znázorněny.



Obrázek 13: Navrhovaný databázový model

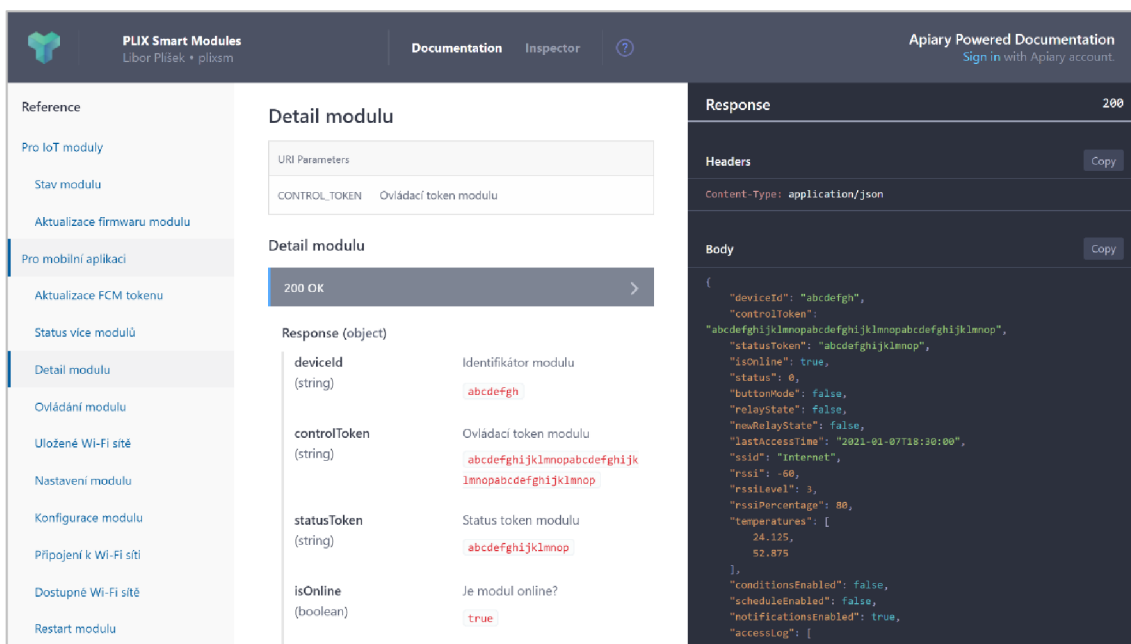
## 4.5 Komunikační rozhraní serverové aplikace

Tato podkapitola obsahuje návrh komunikačního rozhraní. Jak již bylo v této práci dříve popsáno, hlavní komunikační komponentou informačního systému je serverová aplikace, která obsahuje komunikační rozhraní pro všechny ostatní platformy. V této podkapitole je popsáno vytváření dokumentace komunikačního rozhraní, shrnutí typů požadavků, které serverová aplikace podporuje a formát odpovědí, které serverová aplikace po zpracování dotazu vrací zpět. Hlavním komunikačním rozhraním bylo pro tento informační systém zvoleno REST API [19]. Pro rychlejší odezvy by bylo vhodné ještě dále mezi komponentami systému navrhnout a implementovat WebSocket API [34], nicméně v rámci této diplomové práce toto nebylo uskutečněno.

### 4.5.1 Dokumentace

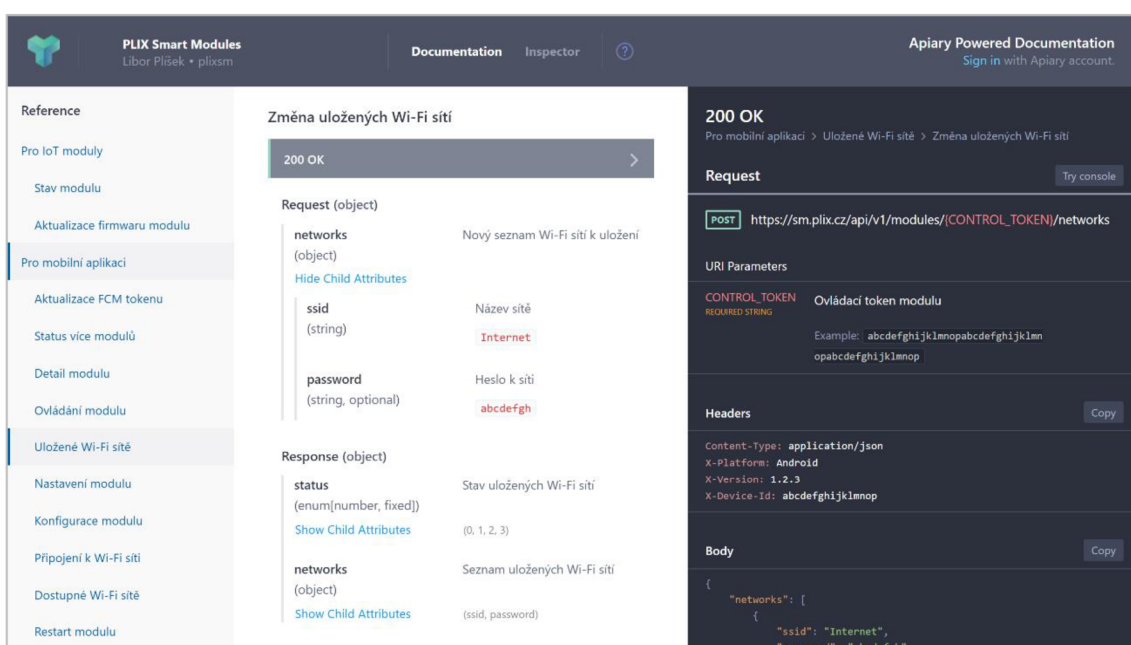
API dokumentace byla vyhotovena v rámci návrhu této práce pomocí platformy Apiary [35]. Tato dokumentace obsahuje veškeré informace potřebné k napojení libovolné další komponenty do informačního systému a je volně dostupná online [36]. Kdokoliv tak může informační systém na základě této dokumentace rozšířit. Zdrojový kód této dokumentace je napsán ve formátu API Blueprint [37] a nachází se taktéž v elektronické příloze této práce.

Navržená dokumentace komunikačního rozhraní je rozdělena na 2 základní části. První částí je úvod, kde je podrobně popsáno, jaké požadavky komunikačního rozhraní serverové aplikace podporuje a jaké odpovědi vrací včetně nejruznějších parametrů, formátů apod. Druhá část dokumentace obsahuje seznam a detaily jednotlivých API koncových bodů. Tyto koncové body dále byly rozděleny ještě na další dvě skupiny. První skupina s názvem „Pro IoT moduly“, obsahuje pouze takové koncové body, které aktuálně využívají především jednotlivé hardwarové moduly. Druhá skupina s názvem „Pro mobilní aplikaci“ obsahuje všechny koncové body, které využívá mobilní aplikace. Zde je nutné podotknout, že mobilní aplikace nevyužívá pouze koncové body komunikačního rozhraní serverové aplikace, nýbrž navíc ještě koncové body externí monitorovací služby a koncové body samotných modulů pro uložení parametrů první Wi-Fi sítě do modulu v off-line režimu (bez připojení k internetu).



**Obrázek 14: Náhled API dokumentace – detail modulu**

V levé části dokumentace se nachází menu, které slouží k rychlé navigaci po celé dokumentaci. Uprostřed dokumentace se nachází seznam všech API koncových bodů včetně jejich podrobností. V pravé části se na tmavém pozadí, po rozkliknutí konkrétního koncového bodu, ještě navíc vždy zobrazuje jeho detail a příklad.



**Obrázek 15: Náhled API dokumentace – změna uložených Wi-Fi sítí**

## 4.5.2 Požadavky

REST API serverové aplikace přijímá HTTP [38] požadavky pouze v JSON [39] formátu. Na všechny ostatní požadavky může být vrácena chybová odpověď, nebo mohou být i v určitých situacích zpracovány nějakým způsobem chybně. Veškeré časové údaje serverová aplikace přijímá pouze v ISO 8601 [40] formátu.

### Přijímané metody

Přijímané metody HTTP požadavků jsou především GET a POST. Metoda GET v tomto informačním systému primárně slouží k získávání dat ze serverové aplikace. Metoda POST je naopak primárně využívána k odesílání dat do serverové aplikace. V tomto informačním systému nebyly implementovány napříč platformami žádné klasické CRUD operace, proto se zde vůbec nepoužívají metody PUT a DELETE. Samozřejmostí je povolení ostatních základních metod jako jsou například metody HEAD nebo OPTIONS.

### Hlavička požadavků

V hlavičce požadavku často bývají povinné atributy „X-Platform“, „X-Version“ a „X-Device-Id“. Atribut „X-Platform“ obsahuje označení platformy, která na daný koncový bod přistupuje (například „android“). Atribut „X-Version“ obsahuje verzi mobilní aplikace a atribut „X-Device-Id“ obsahuje unikátní univerzální identifikátor zařízení, které na daný koncový bod přistupuje.

```
GET /api/v1/modules/abcdefghijklmnopabcdefghijklmnopabcdefghijklmnop/networks HTTP/1.1
Host: sm.plix.cz
X-Platform: Android
X-Version: 1.2.3
X-Device-Id: abcdefghijklmnop
Content-Type: application/json
Content-Length: 210

{
  "networks": [
    {
      "ssid": "Internet",
      "password": "abcdefgh"
    },
    {
      "ssid": "Wi-Fi",
      "password": "ijklmnop"
    }
  ]
}
```

Obrázek 16: Příklad HTTP požadavku na koncový bod REST API

## **Autentizace**

Jednotlivé komponenty informačního systému se mohou autentizovat pomocí různých tokenů. Tyto tokeny mohou být přímou součástí URL adresy, nebo mohou být zasílány v jednom z atributů hlavičky požadavku. Pro zasílání autentizačního tokenu v hlavičce dotazu se používá atribut s názvem „X-Token“. Pod takovými tokeny přistupují na komunikační rozhraní serverové aplikace například samotné hardwarové moduly nebo mobilní aplikace.

Běžní uživatelé mají k dispozici ovládací klíče (tokeny), které po uložení do mobilní aplikace využívají ke správě a ovládání svých modulů. Správce systému se přihlašuje do administračního portálu klasicky přes přihlašovací formulář pomocí svého přihlašovacího jména a hesla. Po odeslání formuláře se správci systému v prohlížeči automaticky uloží „session“ do souborů cookies [41]. Přihlašovací formulář ani celý přihlašovací mechanismus nebyl v rámci této práce navržen a implementován. V rámci této práce byl využit standardní přihlašovací mechanismus, který automaticky poskytuje použitá technologie, proto ani nadále v této práci není přihlašovací mechanismus dále popisován. Ve výjimečných případech může k autentizaci postačit pouze univerzální unikátní identifikátor mobilní aplikace, který mobilní aplikace automaticky zasílá v atributu s názvem „X-Device-Id“ v hlavičce požadavku. Veškeré podrobnosti ohledně autorizace jsou vždy upřesněny v API dokumentaci zvláště jednotlivě u každého koncového bodu.

### **4.5.3 Odpovědi**

Stejně tak jako HTTP [38] požadavky i odpovědi serverové aplikace jsou vráceny pouze v JSON [39] formátu a veškeré časové údaje jsou taktéž vráceny pouze v ISO 8601 [40] formátu.

#### **Standardní odpovědi**

Standardní odpovědi používá serverová aplikace v případě, pokud došlo ke správnému zpracování požadavku. Mezi standardními odpověďmi se používá především odpověď se stavovým kódem „200 OK“. Odpovědi „201 Created“ nebo „204 No Content“ nejsou v tomto informačním systému vůbec využívány. Na následující obrázku se nachází příklad standardní HTTP odpovědi.

```

1 {
2   "deviceId": "abcdefgh",
3   "isOnline": true,
4   "status": 0,
5   "buttonMode": false,
6   "relayState": false,
7   "newRelayState": false,
8   "lastAccessTime": "2021-01-07T18:30:00",
9   "temperatures": [
10    24.125,
11    52.875
12  ],
13   "conditionsEnabled": false,
14   "scheduleEnabled": false,
15   "notificationsEnabled": true
16 }

```

**Obrázek 17: Příklad standardní HTTP odpovědi**

### Chybové odpovědi

Chybové odpovědi jsou vráceny v okamžiku, kdy se nedaří dokončit požadovanou operaci. Jako nejčastější chybové odpovědi se používají odpovědi se stavovými kódy „400 Bad Request“, „401 Unauthorized“, „403 Forbidden“, „404 Not Found“ nebo chybové odpovědi serveru, které začínají stavovým kódem 500 a více. Chybové odpovědi nebývají zasílány příliš často a jsou primárně v JSON Error Objects [42] formátu. Každá taková chybová odpověď obsahuje unikátní identifikátor, status, kód, název a detail. Detail chybové odpovědi obsahuje zprávu, která programátorovi blíže specifikuje chybu.

```

1 {
2   "errors": [
3     {
4       "id": 1000,
5       "status": 400,
6       "code": "MISSING_HEADER_PARAMETER",
7       "title": "Missing header parameter",
8       "detail": "Header parameter DEVICE_TOKEN is missing."
9     }
10  ]
11 }

```

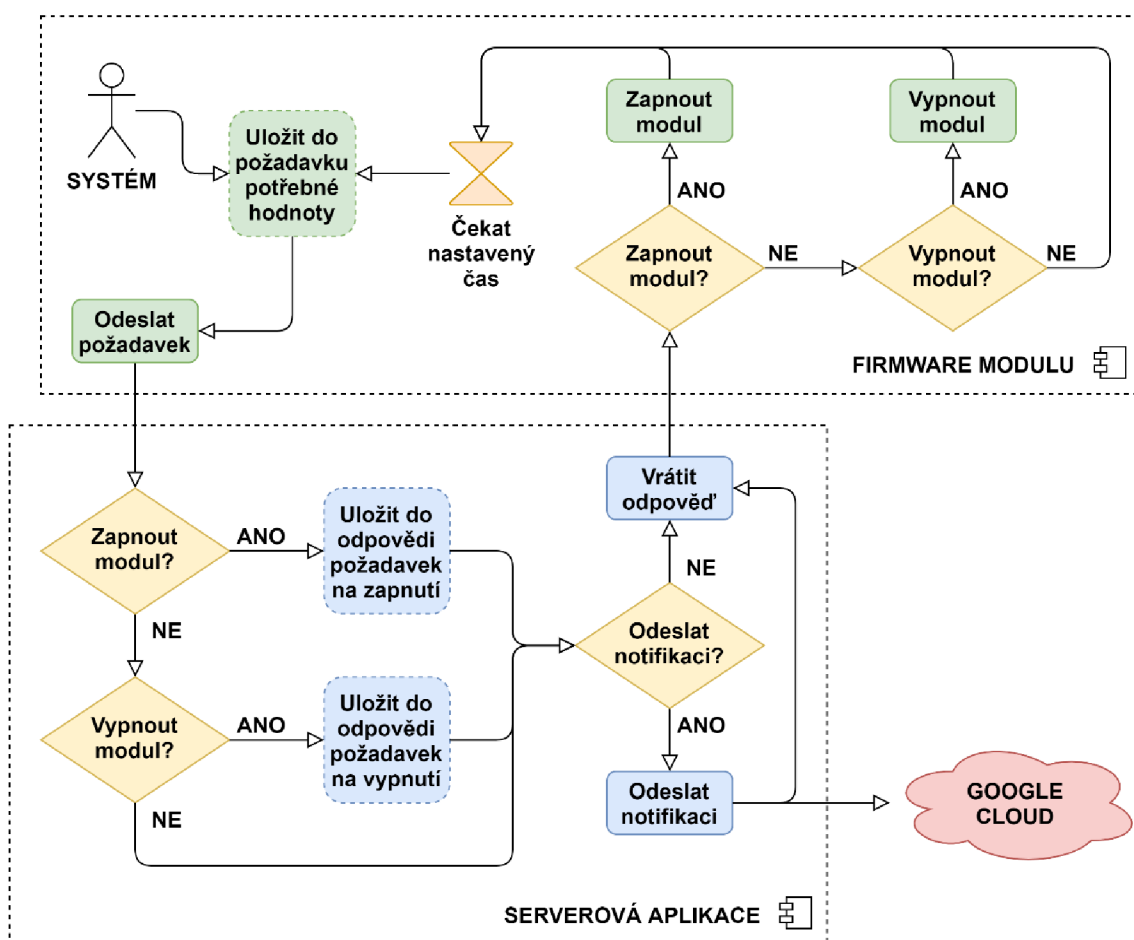
**Obrázek 18: Příklad chybové HTTP odpovědi**

## 4.6 Automatizované úlohy

V této podkapitole se nachází popis automatizovaných úloh, které pravidelně probíhají na pozadí serverové aplikace. Tyto úlohy jsou spouštěny pomocí cronu, jehož pravidelné volání poskytuje samotný webhosting [43].

### 4.6.1 Automatické ovládání modulů a odesílání upozornění

Moduly aktuálně posílají v pravidelných intervalech informace o svém stavu. Mimo jiné například o stavu připojení k internetu nebo o stavu aktuálních hodnot ze senzorů, které jsou na jeho vstupech. Pokaždé kdy dojde k takovému požadavku na serverovou aplikaci, serverová aplikace zkontroluje nastavení automatizace ovládání nebo odeslání push notifikací na mobilní zařízení. Podrobněji znázorněný vývojový diagram automatizace se nachází na následujícím obrázku.

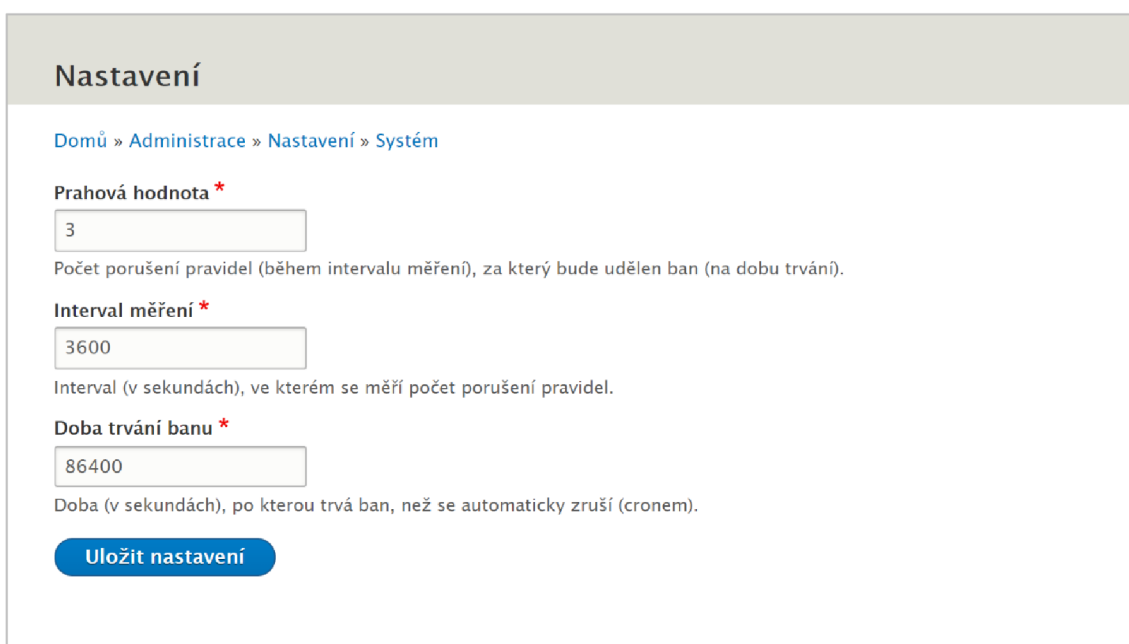


Obrázek 19: Vývojový diagram automatizace serverové aplikace



## 4.6.2 Automatické povolení zablokovaných IP adres

Pokud dojde na libovolných koncových bodech serverové aplikace ze stejné IP adresy k chybné autentizaci, je tato IP adresa okamžitě dočasně zablokována. Automatizované úlohy, které běží na pozadí serverové aplikace, poté automaticky po daném čase zablokované IP adresy opět povolují. Nastavit tento čas může správce systému přes administrační portál. O tuto bezpečnostní funkcionalitu se stará softwarový modul, který vznikl rovněž v rámci této diplomové práce. Na následujícím obrázku se nachází formulář, který obsahuje nastavení tohoto modulu.



The image shows a web interface for configuring IP address blocking. The page title is 'Nastavení'. The breadcrumb trail is 'Domů » Administrace » Nastavení » Systém'. There are three input fields, each with a red asterisk indicating it is required:

- Prahová hodnota \***: Input field containing the value '3'. Below it is the text: 'Počet porušení pravidel (během intervalu měření), za který bude udělen ban (na dobu trvání).'.
- Interval měření \***: Input field containing the value '3600'. Below it is the text: 'Interval (v sekundách), ve kterém se měří počet porušení pravidel.'.
- Doba trvání banu \***: Input field containing the value '86400'. Below it is the text: 'Doba (v sekundách), po kterou trvá ban, než se automaticky zruší (cronem).'.

At the bottom of the form is a blue button labeled 'Uložit nastavení'.

**Obrázek 20: Formulář pro nastavení blokování IP adres**

## 4.6.3 Automatické čištění historie

Automatizované úlohy, které běží na pozadí serverové aplikace dále automaticky pravidelně odstraňují záznamy v databázi, které se nacházejí v historických nebo protokolových tabulkách. Odstraňují se především takové záznamy, kterých je větší počet než zadaná hranice, nebo takové, které jsou starší než zadaný časový údaj. Úloha na čištění databáze je spouštěna přibližně 1x za hodinu na pozadí systému pomocí cronu.

## 5 Implementace

V této kapitole je popsána praktická implementace, která byla rozdělena celkově do tří částí podle jednotlivých komponent informačního systému, kterých se daná implementace týká a také chronologicky podle toho, jak postupně byly vyvíjeny za sebou. Na konci této kapitoly se ještě navíc nachází shrnutí komunikace, bezpečnosti, nasazení a zálohování všech vyvinutých řešení.

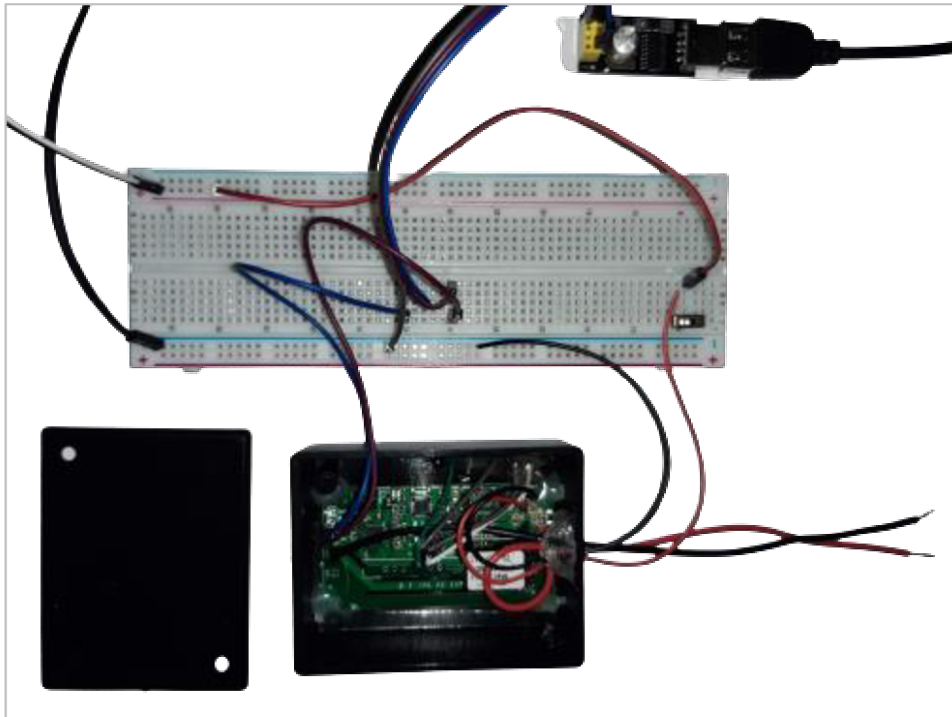
### 5.1 Moduly

Jako první byla provedena implementace firmwaru pro hardwarové moduly značky Sonoff [11], kde se zjistilo, zda vůbec půjdou zrealizovat všechny navržené případy užití. Nakonec bylo zjištěno, že půjde pokrýt úplně všechny navržené případy užití, které byly navrženy a podrobně popsány v rámci předchozí kapitoly této práce s názvem „Návrh“.

I přes to, že byl původně v rámci praktické části této práce plánován i vývoj vlastního hardwarového modulu složeného z různých HW komponent, nakonec bylo rozhodnuto, že pro účely tohoto projektu postačí pouze přeprogramování modulů značky Sonoff. Přeprogramování modulů této značky se ukázala jako jednodušší, praktičtější, a především rychlejší a levnější cesta. Detaily k tomuto rozhodnutí se nacházejí ve třetí kapitole této práce s názvem „Analýza“.

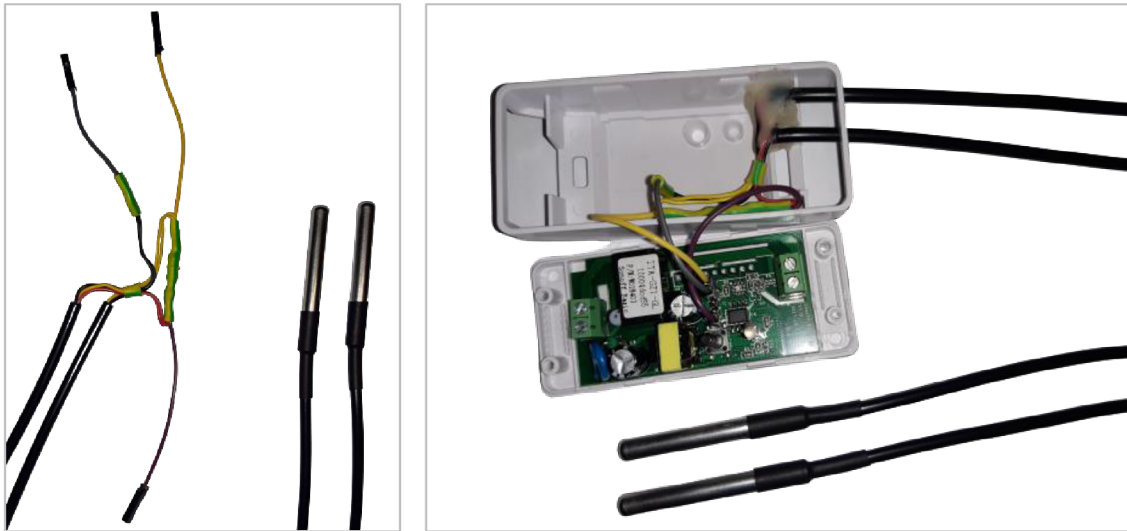
Přeprogramovat modul, tedy Wi-Fi čip ESP8285 [44], který se na zařízeních této značky nachází, lze poměrně jednoduše, a to přímo v domácích podmínkách pomocí USB rozhraní počítače, FTDI [45] kabelu a vývojové desktopové aplikace Arduino IDE [46]. Vývojové prostředí Arduino IDE musí mít před prvním přeprogramováním nejprve stažené všechny potřebné balíčky a musí být správně nakonfigurována komunikace mezi počítačem a modulem, přesně podle typu použitého čipu. Detail konfigurace se nachází v elektronické příloze této práce a zároveň i v online repozitáři základního firmwaru pro moduly [47].

Na následujícím obrázku se nachází náhled zapojení modulu, který je připojen k počítači pomocí USB rozhraní a FTDI kabelu a je připraven pro následné přeprogramování nebo pro čtení sériového ladícího výstupu. Ukázka výstupu je blíže popsána v následující kapitole s názvem „Testování“.



**Obrázek 21: Ukázka zapojení modulu při jeho programování**

Jeden z reálných produkčních modulů se nachází u krbových kamen, které zároveň umožňují nahřát teplou vodu v bojleru pro celou domácnost. Tento modul vyžadoval připojení a práci se dvěma teplotními senzory, podle kterých se přes mobilní aplikaci nastaví pravidelná automatizace. Jeden teplotní sensor snímá teplotu vody v kamnech a druhý teplotní sensor snímá teplotu vody v bojleru. Poté lze jednoduše v mobilní aplikaci nastavit například to, že čerpadlo, které přečerpává vodu mezi kamny a bojlerem je spuštěno pouze v tom případě, když teplota v kamnech je o 5 °C vyšší než teplota v bojleru a jakmile tato podmínka neplatí, je čerpadlo ihned vypnuto, aby nedocházelo k ochlazení teplé vody. Zároveň je také možné nastavit, aby se voda přečerpávala pouze v tom případě, když má voda v kamnech více jak 50 °C, aby se kamna stihla dostatečně nahřát, než dojde k přečerpávání. Většina modulů značky Sonoff nepodporují přímé připojení teplotních sensorů a když už je podporují, tak podporují pouze připojení jednoho senzoru. Z toho důvodu bylo potřeba provést manuální hardwarovou úpravu jednoho ze Sonoff modulů. K tomuto modulu byly připojeny dva teplotní senzory DS18B20 [48], ze kterých nový firmware dokáže pravidelně načítat teplotu. Ukázka reálného zapojení teplotních sensorů k modulu se nachází na následujícím obrázku.



**Obrázek 22: Příprava a zapojení dvou teplotních senzorů k modulu**

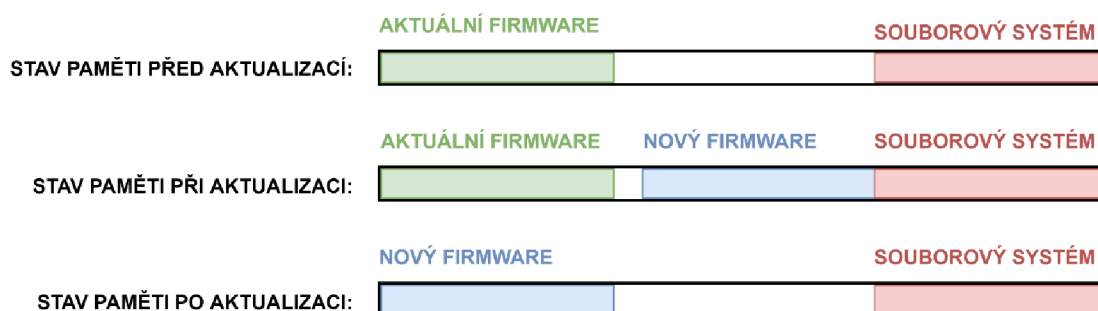
Výsledným produktem je modul, který v libovolném intervalu měří reálnou teplotu ze dvou teplotních senzorů a dokáže automaticky reagovat na jejich změny. Počet těchto senzorů navíc není ničím omezený (kromě maximálního proudového zatížení), protože tyto teplotní senzory mohou být všechny připojeny pouze přes jeden vstupní pin modulu, a to díky sériové komunikaci a použití hardwarových adres. Hardwarové adresy senzorů jsou předem zjišťovány pomocí speciálního firmwaru na testování senzorů, který taktéž vznikl v rámci této práce [49]. Na následujícím obrázku se nachází fotografie výsledného produktu.



**Obrázek 23: Výsledný modul obsahující dva teplotní senzory**

Pro hardwarové moduly značky Sonoff byl vyhotoven pouze jeden firmware, který se v jednotlivých modulech liší pouze ve své konfiguraci. Tím se velmi zjednodušil následný vývoj všech potřebných funkcionalit pro všechny typy modulů, kde se jedním programovým kódem pokrývají úplně všechny druhy zařízení, které byly použity v rámci tohoto projektu. Programový kód pro čip ESP8285 se sice vyvíjí v programovacím jazyce C [50], ale v rámci této implementace je třeba postupovat přesně v rámci příslušné dokumentace [51] těchto Wi-Fi čipů. Tyto čipy totiž potřebují dostatek prostoru pro operace a komunikaci spojenou s udržením aktivního internetového připojení, který může být se špatně optimalizovaným programovým kódem velmi rychle vyčerpán.

V rámci implementace firmwaru byla také implementována vzdálená aktualizace, která funguje pomocí serverové aplikace, a tedy vzdáleně přes internet. V tomto ohledu je tento projekt naprosto zcela unikátní a z ostatních levných řešení je jediný, který si na pokyn serveru dokáže stáhnout předem připravený binární soubor a potom z něj spustit nový firmware. Aby mohl modul tuto akci provést, musí mít k dispozici takovou velikost volného místa v paměti, která odpovídá velikosti nového firmwaru. V tomto projektu tedy bylo dodržováno, aby se výsledný programový kód vešel do poloviny volné paměti po odečtení prostoru, který zabírá souborový systém. Aktuálně programový kód zabírá včetně všech ladicích zpráv a nástrojů přibližně 48 % paměti. Nicméně pokud se tyto zprávy a nástroje odstraní, zabírá programový kód přibližně 45 % paměti. Přibližně 35 % paměti zabírají pouze použité knihovny, které všechny byly voleny s pečlivou rozvahou, aby bylo ušetřeno co nejvíce místa. Na následujícím obrázku se nachází diagram, který znázorňuje paměť modulu během jeho vzdálené aktualizace.



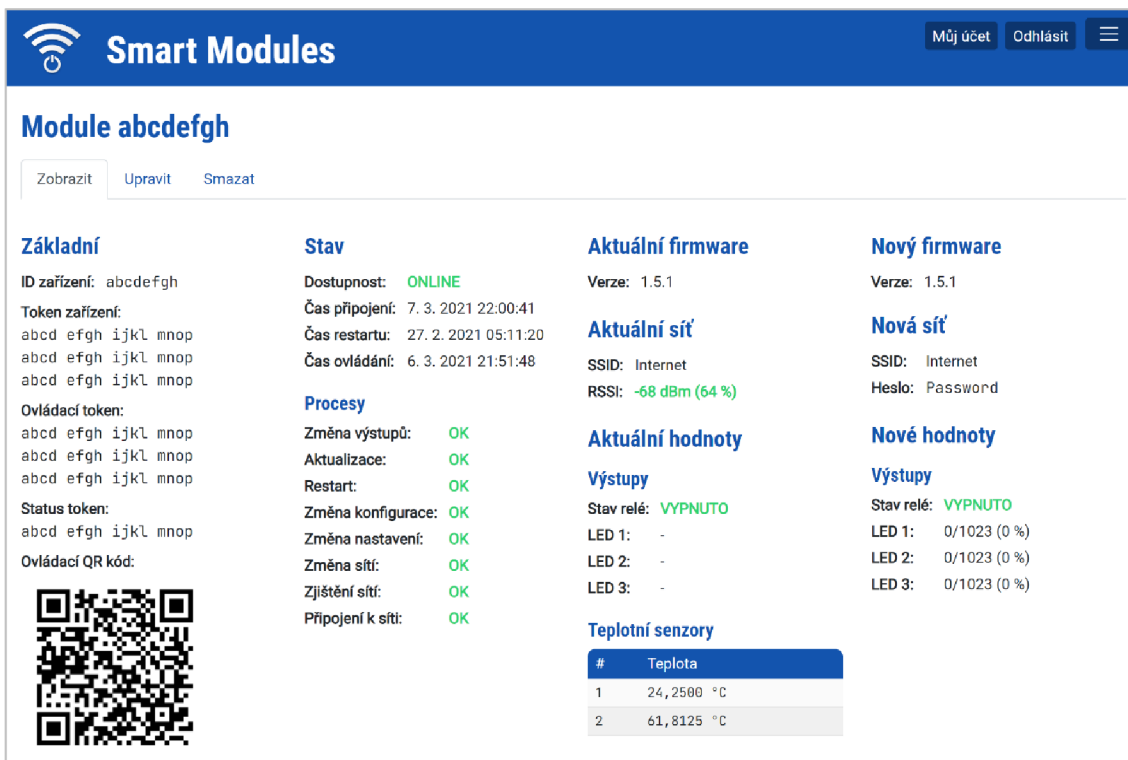
**Obrázek 24: Stav paměti modulu během aktualizace firmwaru**

## 5.2 Serverová aplikace

Dále byla implementována serverová aplikace, která zastřešuje celý informační systém tím, že obsahuje komunikační rozhraní, pomocí kterého s ní komunikují všechny ostatní softwarové komponenty. Dále je nezbytnou součástí IS také z toho důvodu, že obsahuje administrační portál pro správce systému, který díky němu může celý systém ovládat a nastavovat.

Po delší analýze a zvažování byla serverová aplikace nakonec implementována pomocí CMS Drupal 9 [52] v programovacím jazyce PHP [53] nad frameworkem Symfony [54]. Původně bylo plánováno, že bude tato aplikace implementována v Node.js [55] a nasazena na platformě Heroku [56], jenže kvůli nepodpoře SSL certifikátu [57] na vlastní doméně a pravidelnému ukončování aplikace při neaktivitě a limitu maximálního počtu dostupných měsíčních minut běhu, bylo rozhodnuto, že bude serverová aplikace vyvinuta jako klasická webová aplikace na stejném jádru ještě s jinými projekty. Bylo rozhodnuto, že to tak bude nejen nejjednodušší, ale hlavně finančně úspornější, protože tímto provoz serverové aplikace právě teď vyjde měsíčně úplně zcela zdarma. Dělí se tedy o programové jádro Drupalu a měsíční náklady s ostatními aplikacemi, které by stejně na stejném serveru o stejném výkonu a kapacitě běžely tak jako tak. Navíc díky tomuto je projekt spuštěný na vlastní doméně s SSL certifikátem a bylo celkem jednoduché pro něj vytvořit webový administrační portál. Nicméně, kdyby se projekt více rozšířil a obsahoval by několik stovek nebo dokonce tisíců vzdáleně ovládaných zařízení, tato serverová aplikace by zcela jistě začala být nedostačující a muselo by se přejít na nějaké ušlechtlejší řešení. Takové rozšíření se aktuálně v dohledné době zcela vůbec neplánuje, takže momentálně bylo možné bez problému zrealizovat serverovou aplikaci právě tímto způsobem.

Pro vývoj serverové aplikace byl použit nástroj Composer [58], který se stará o automatickou instalaci externích balíčků (závislostí) včetně komunitních softwarových Drupal modulů. Lokálně i na produkci byl použit webový server Apache [59]. Databáze byla použita ve všech prostředích MariaDB [60] a pro pohodlnou práci s CMS Drupal byl použit nástroj Drush [61]. Následující obrázek obsahuje ukázkou detailu modulu, který je otevřen v administračním portálu.

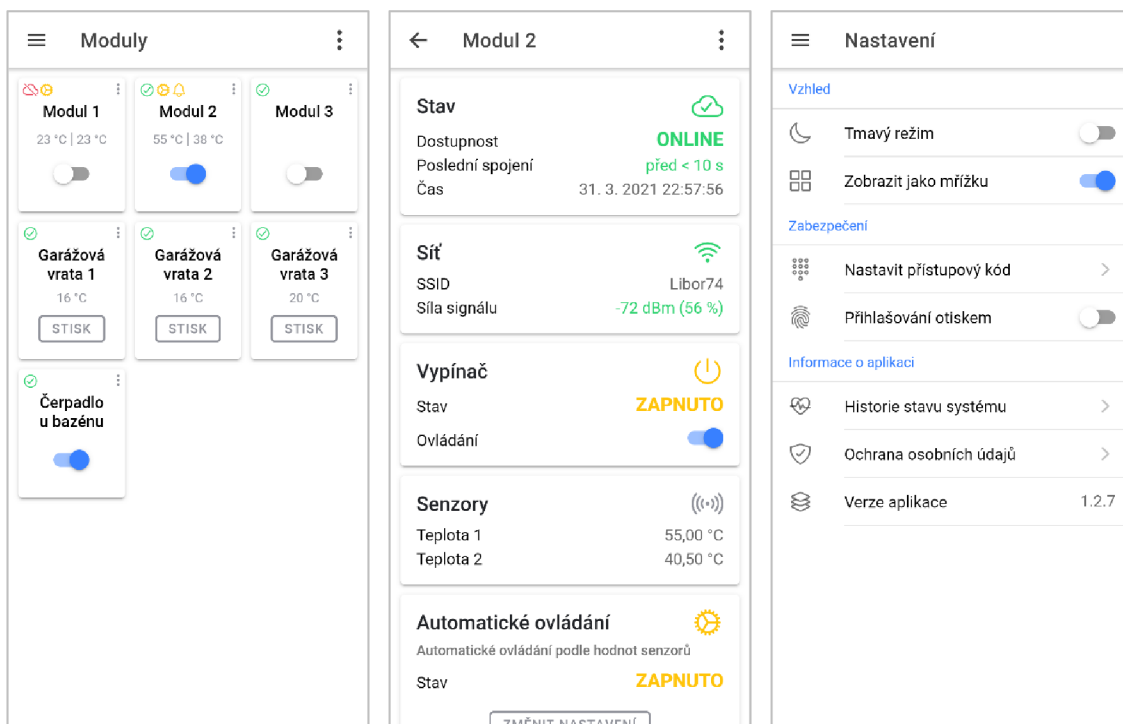


Obrázek 25: Ukázka administračního portálu

### 5.3 Mobilní aplikace

Úplně nakonec byla implementována mobilní aplikace, která je hlavní prezentací tohoto projektu. Tato aplikace komunikuje se serverovou aplikací, díky čemuž uživatel dokáže velmi pohodlně ovládat, nastavovat, konfigurovat, nebo aktualizovat své moduly. Mobilní aplikace vznikla pomocí frameworku Ionic [62] a pomoci poměrně nového prostředí Capacitor [63] jako multiplatformní hybridní aplikace, kterou lze nasadit jako nativní iOS, Android i PWA aplikaci [64]. V rámci tohoto projektu byla aplikace nakonec produkčně nasazena pouze na platformu Google Play jako nativní aplikace pro operační systém Android. Na ostatní platformy zatím nasazena nebyla, protože to nevyžadoval žádný reálný uživatel. Pro vývoj této mobilní aplikace byly využity webové technologie, jako například Vue.js [65] framework anebo programovací jazyk TypeScript [66].

Na následujícím obrázku se nachází vybrané snímky obrazovek mobilní aplikace. Všechny snímky se nachází ve čtvrté příloze na konci této práce a také v elektronické příloze ve tmavém i světlém režimu.



**Obrázek 26: Snímky obrazovky mobilní aplikace**

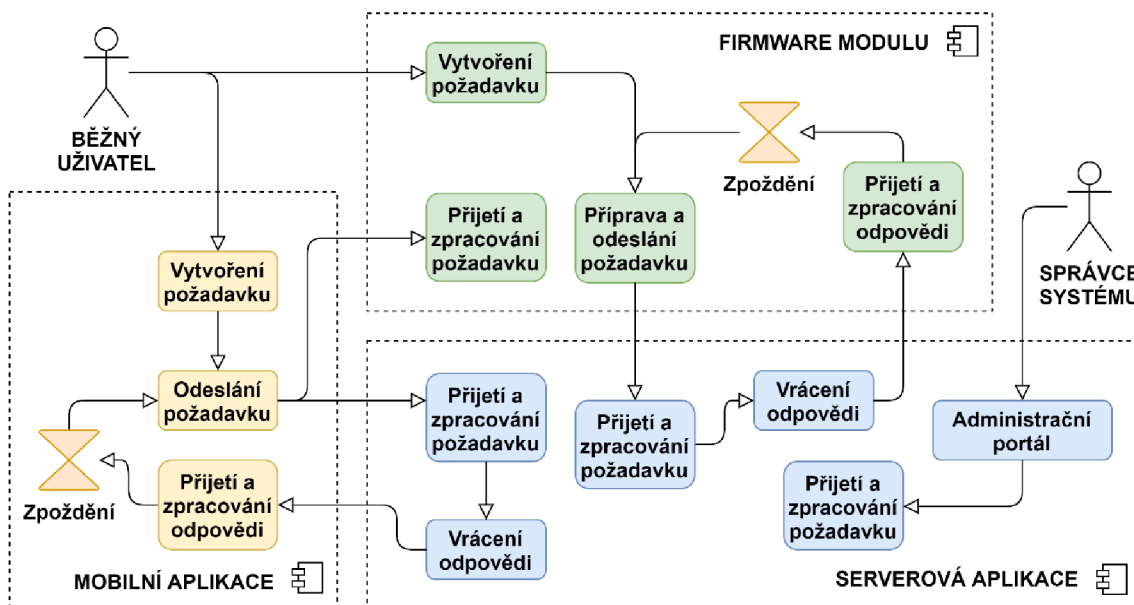
## 5.4 Komunikace

Hlavním komunikačním rozhraním systému je serverová aplikace. K tomuto rozhraní přistupuje pomocí zabezpečeného spojení jak firmware hardwarových modulů, tak i mobilní aplikace.

Běžný uživatel primárně ovládá svoje moduly pomocí mobilní aplikace, a to vzdáleně pomocí serverové aplikace a internetu. Ve výjimečných případech může běžný uživatel ovládat modul napřímo pomocí hardwarového tlačítka, které se nachází na všech modulech, které vznikly v rámci této práce. Dále mobilní aplikace umožňuje přímé požadavky na komunikační rozhraní modulů pomocí lokálního spojení modulu a mobilní aplikace pro účely nahrání údajů k první Wi-Fi síti. Správce systému pak přistupuje k celému informačnímu systému pouze skrze administrační portál, který je součástí serverové aplikace.

Na následujícím obrázku se nachází tok obecných požadavků informačním systémem. Z vývojového diagramu byla pro větší přehlednost úmyslně odstraněna komunikace zajišťující odesílání push notifikací přes Google Cloud [28] služby.





Obrázek 27: Vývojový diagram obecného případu užití

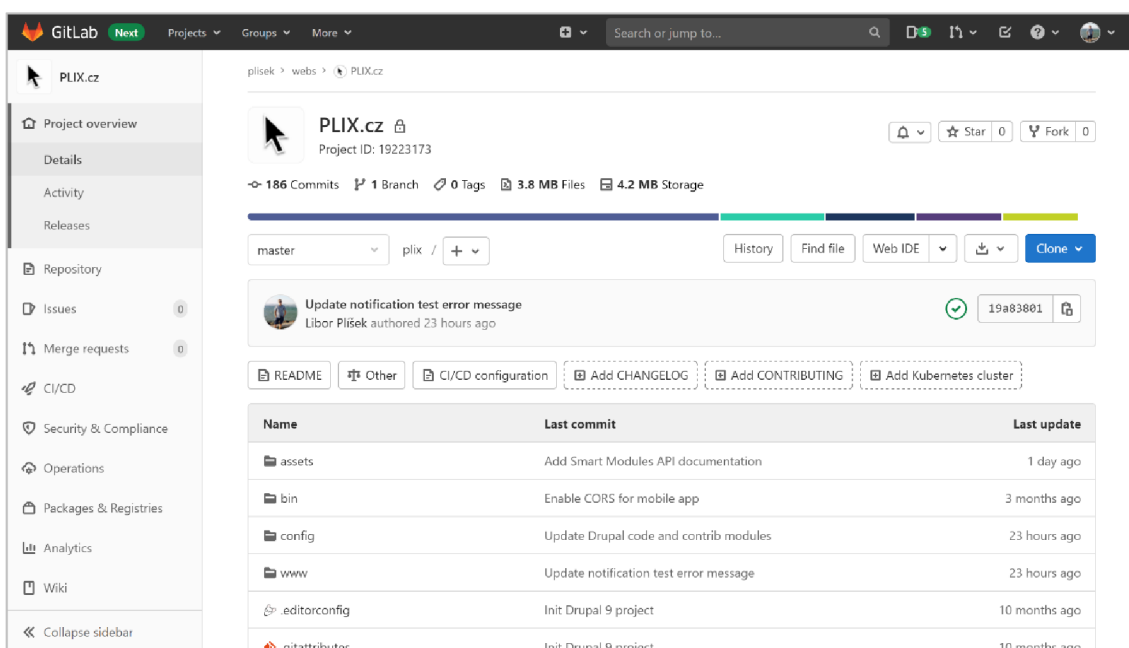
## 5.5 Bezpečnost

Na bezpečnost spojení mezi všemi komponentami informačního systému byl kladen co nejvyšší důraz. Postupně se komunikace mezi serverovou aplikací a firmwarem modulů vyvíjela přes obyčejné nezabezpečené a zcela odposlechnutelné a modifikovatelné spojení HTTP. Později probíhala komunikace přes HTTPS [25] bez ověřování certifikátu serveru. Tím se stala komunikace šifrovanou, ale nebyla odolná vůči „Men in the middle“ útoku [67]. Následně se podařilo zprovoznit komunikaci s ověřováním certifikátu alespoň pomocí kryptografického otisku. Nicméně toto řešení stále nebylo zcela dostatečné, proto se nakonec projekt dostal do fáze, kdy firmware modulů se serverem komunikuje přes zcela naprosto zabezpečené HTTPS spojení pomocí ověřování certifikátu serveru u kořenové certifikační autority. Pro správu certifikátů na serveru byla použita služba Let's Encrypt [68] a její hlavní kořenové certifikáty jsou přímou součástí programového kódu firmwaru. Aby mohl firmware modulů ověřit správnost certifikátů, potřebuje si po svém prvním spuštění a připojení k Wi-Fi síti zjistit aktuální čas. Dále komunikaci zastřešují dlouhé autentizační tokeny, které obsahuje každý hardwarový modul už od inicializačního přeprogramování. V rámci mobilní aplikace bylo zabezpečené spojení se serverem automatické, protože všechny důležité

kořenové certifikáty obsahují mobilní telefony přímo ve svém operačním systému. Dlouhý autentizační token pro ovládání modulů získá mobilní aplikace automaticky po načtení QR kódu, který je součástí každého hardwarového modulu.

## 5.6 Nasazení

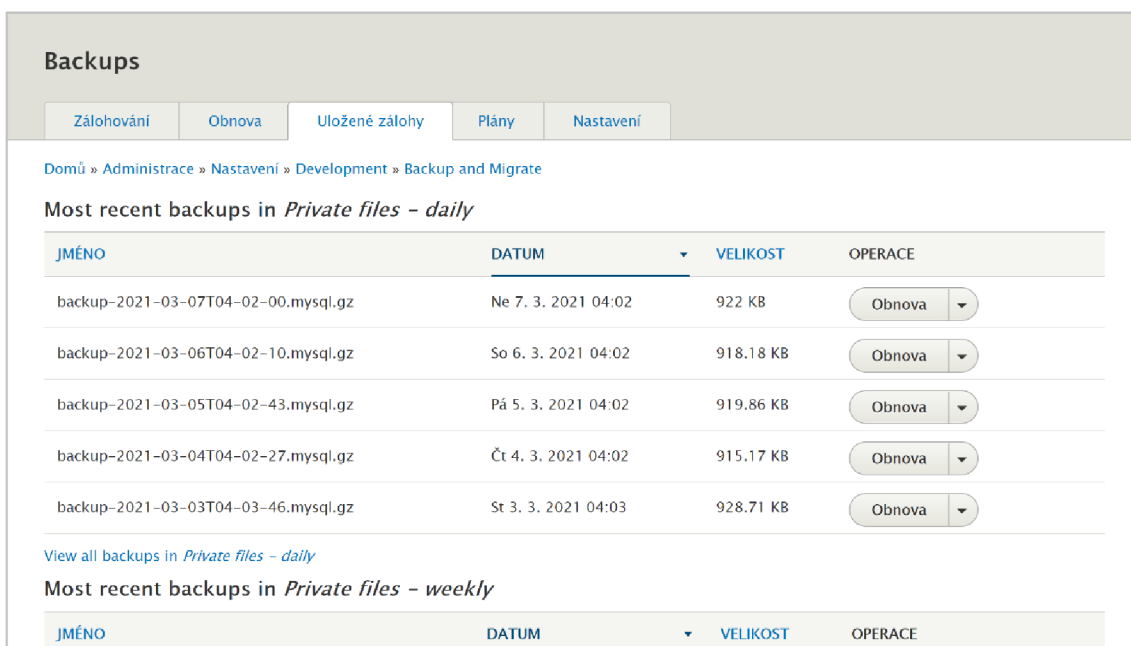
Pro vývoj veškerého softwaru, který byl vyvíjen v rámci této diplomové práce byl po celou dobu používán verzovací systém Git [21] a online repozitář GitLab [69]. U projektů serverové a mobilní aplikace byla v tomto online verzovacím systému částečně zprovozněna průběžná integrace (CI/CD) [70]. Při každé změně v tomto online repozitáři se kód vždy automaticky otestuje. U serverové aplikace se testují „coding standards“ [71] a její REST API. U mobilní aplikace byly nastaveny pouze základní testy kompilace. Bližší informace k testování se nacházejí v následující kapitole. Nasazení firmwaru probíhá manuálně pomocí postupu popsáno na začátku této kapitoly. Nasazení mobilní aplikace probíhá manuálně přes Android Studio [72] a Google Play konzoli [73]. Nasazení serverové aplikace probíhá pomocí automatizovaného shellového skriptu, který je součástí elektronické přílohy této práce. Všechna implementovaná řešení jsou dostupná online a jejich URL adresy jsou uvedeny ve třetí příloze na konci této práce.



Obrázek 28: Ukázka online GitLab repozitáře

## 5.7 Zálohování

V produkční verzi serverové aplikace je nastaveno pravidelné zálohování. Zálohování probíhá pravidelně každý den v nočních hodinách pomocí externího softwarového modulu „Backup and Migrate“ [74]. Pokud by tedy došlo k nějaké kritické havárii serverové aplikace a bylo by potřeba obnovit databázi, lze tak učinit jedním kliknutím v nastavení tohoto softwarového modulu. Dále díky těmto zálohám lze zpětně dohledat jakákoliv důležitá data i když se v samotném produkčním systému již nenacházejí. Na serveru jsou aktuálně ponechávány zálohy starší méně než 64 týdnů. Vzhledem k tomu, že serverová aplikace je se zálohováním spuštěna teprve od začátku listopadu 2020, zatím nebyla kapacita záloh ani z poloviny naplněna. Všechny zálohy jsou tedy uloženy v rámci jedné serverové aplikace a nejsou ani serverově a ani geograficky odděleny. Pokud havaruje celý server webhostingu, jsou tímto zálohy serverové aplikace nenávratně ztraceny. Nicméně proto existují ještě zálohy, které provádí samotný webhosting, které jsou již bezpečně a moderně zálohovány a webhosting je automaticky využívá k obnově serveru po jeho havárii. Tyto zálohy provádí webhosting pouze 1x týdně a nejsou primárně určeny pro běžné zákazníky [22]. Na následujícím obrázku se nachází ukázka uložených záloh databáze serverové aplikace.



The screenshot shows a web interface for managing backups. At the top, there are tabs for 'Zálohování', 'Obnova', 'Uložené zálohy', 'Plány', and 'Nastavení'. Below the tabs, there is a breadcrumb trail: 'Domů » Administrace » Nastavení » Development » Backup and Migrate'. The main section is titled 'Most recent backups in Private files - daily'. It contains a table with the following data:

JMÉNO	DATUM	VELIKOST	OPERACE
backup-2021-03-07T04-02-00.mysql.gz	Ne 7. 3. 2021 04:02	922 KB	Obnova
backup-2021-03-06T04-02-10.mysql.gz	So 6. 3. 2021 04:02	918.18 KB	Obnova
backup-2021-03-05T04-02-43.mysql.gz	Pá 5. 3. 2021 04:02	919.86 KB	Obnova
backup-2021-03-04T04-02-27.mysql.gz	Čt 4. 3. 2021 04:02	915.17 KB	Obnova
backup-2021-03-03T04-03-46.mysql.gz	St 3. 3. 2021 04:03	928.71 KB	Obnova

Below the table, there is a link 'View all backups in Private files - daily'. The next section is titled 'Most recent backups in Private files - weekly' and shows a table header with columns 'JMÉNO', 'DATUM', 'VELIKOST', and 'OPERACE'.

Obrázek 29: Ukázka uložených záloh databáze serverové aplikace [74]

## 6 Testování

Implementované řešení bylo produkčně zprovozněno ihned po vzniku základních funkcionalit. Nejprve informační systém obsahoval pouze základní funkcionality a fungoval zcela úplně bez mobilní aplikace. Namísto mobilní aplikace informační systém původně obsahoval dočasné jednoduché webové rozhraní, přes které bylo možné ovládat základní funkce modulů a ke kterému měli přístup pouze správcem systému vybraní uživatelé. Toto dočasné webové rozhraní postupně zaniklo a nahradila ho multiplatformní mobilní aplikace, která postupně naplnila úplně všechny navržené případy užití tohoto informačního systému.

Každé praktické řešení, které vzniklo v rámci této práce bylo testováno nejprve postupně ve vývojových podmínkách, a to vždy ihned po vytvoření nové funkcionality. Jakmile byla v programovém kódu implementována nová funkce, byly ihned provedeny vhodné manuální testy, které dané softwarové komponentě vyhovovaly nejvíce. Například pokud byla ve firmwaru pro hardwarové moduly implementována funkce pro zjištění dostupných sítí, byla tato funkce před nasazením do zbylé části informačního systému nejprve manuálně několikrát otestována. Toto otestování probíhalo například i včetně náhlých odpojení nebo připojení jednotlivých hardwarových Wi-Fi přístupových modů, které se nacházely v blízkosti modulu apod. Takto probíhalo testování drtivé většiny případů užití firmwaru pro moduly, u kterého byl vždy sledován jeho vlastní sériový výstup.

Podrobnější popis testování jednotlivých softwarových komponent byl v této kapitole dále rozdělen do podkapitol podle jednotlivých komponent chronologicky přesně tak, jak postupně byly implementovány za sebou. Automatizované testování, které probíhá na platformě GitLab, je spouštěno automaticky po každé změně kódu v online repozitáři. Pokud při tomto automatickém testování dojde k chybě, vývojář je automaticky ihned upozorněn pomocí e-mailu.

### 6.1 Firmware

Jako první byl postupně testován firmware, a to pomocí ladících nástrojů samotné platformy Arduino IDE [46] včetně výjimečného dekodéru. Bylo třeba hlídat, aby měl modul dostatečné množství prostoru pro procesy běžící na pozadí,

kteřé udržují neustálé spojení s Wi-Fi přístupovým bodem a tím i připojení k internetu apod. Stávalo se, že k nějaké kritické chybě došlo i několikrát denně, ale firmware už je v pokročilé a plně odladěné verzi, která již měsíce nevyžaduje žádnou úpravu. Moduly byly několikrát dlouhodobě testovány ve vývojovém prostředí (například přes noc), kde byla následně objevena chyba, u které se zastavily a tím pádem bylo i zjištěno, co je potřeba opravit. Největší problémy mají tyto moduly především v mezních situacích, jako je náhlé odpojení od Wi-Fi přístupového bodu, nebo ztráta připojení k internetu či nečekaná nedostupnost serveru. Následně byly tyto mezní situace simulovány, čímž se nakonec předešlo úplně všem přetrvávajícím problémům. A i když by se teď přeci jenom stalo, že k nějaké kritické chybě dojde, modul je nastavený tak, že po takové chybě automaticky provede restart, čímž se kompletně vyčistí operační paměť a může zase pracovat dál. Server všechny tyto výpadky automaticky zaznamenává a jsou dostupné pro správce systému.

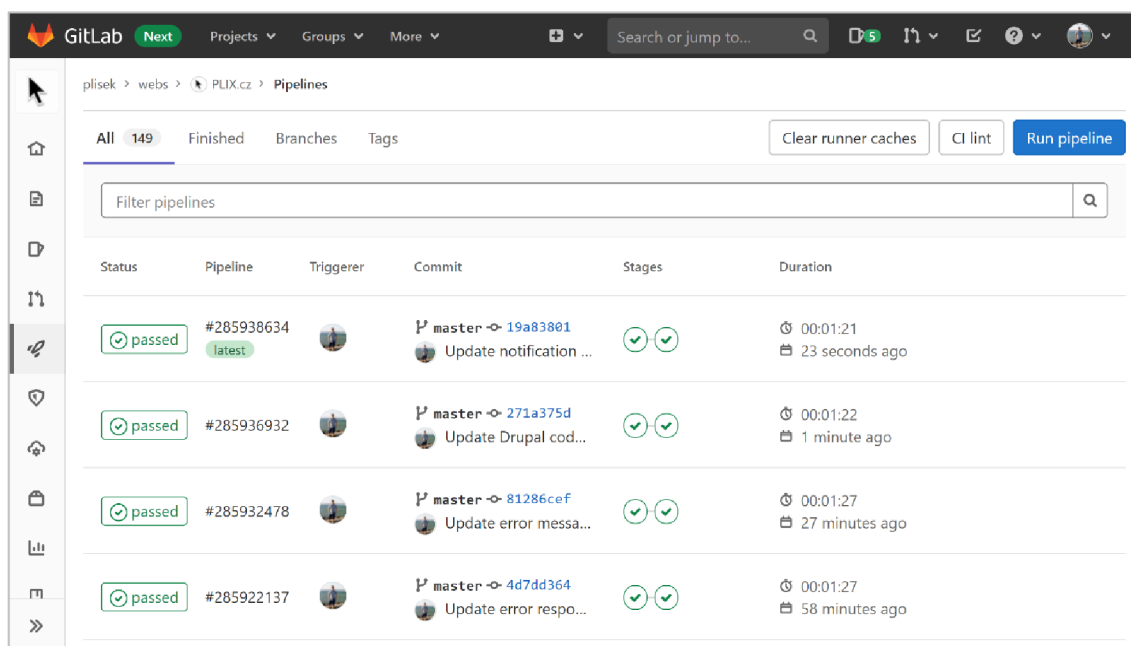
```
Setup info:
Flash chip ID: 1327198
Flash chip size: 1048576 B
Flash chip speed: 40000000 Hz
Flash chip mode: 3
Free sketch space: 483328 B
Free heap: 29520 B
Boot mode: 1
Boot version: 31
Core version: 2_7_4
SDK version: 2.2.2-dev(38a443e)
Reset reason: Power On
Reset info: Power On
File system total size: 65536 B
File system used size: 32768 B
File system free size: 32768 B
Firmware version: 1.5.3
Setup mode: CLIENT
Button pin: 0
Relay pin: 12
Led pin: 13
External button pin: -1
External LED pin: -1
Temperature sensors pin: 14
Device ID: 'abcdefgh'
Device name: 'PLIX-SM-abcdefgh'
Device token: 'abcdefghijklmnopabcdefghijklmnopabcdefghijklmnop'
```

**Obrázek 30: Ukázka sériového výstupu firmwaru modulu**

## 6.2 Serverová aplikace

Jako druhá se začala vyvíjet a testovat serverová aplikace. U serverové aplikace jsou striktně dodržovány a testovány „Drupal coding standards“. Tyto standardy programovacího kódu pomáhají vývojářům udržovat stejně formátovaný kód napříč všemi zařízeními, které pro vývoj aplikace používají. Tyto testy jsou automaticky spouštěny po každé změně v online repozitáři v rámci průběžné integrace (CI/CD). Dále se u serverové aplikace pravidelně testuje samotné REST API, což je nejdůležitější součást celého informačního systému, přes kterou komunikují všechny ostatní softwarové komponenty. Toto komunikační rozhraní (REST API) je testováno pomocí nástroje Dredd [75], který pro tyto testy používá přímo samotný zdrojový kód vytvořené API dokumentace z platformy Apiary [35]. Pokud dojde k chybě, může to znamenat pouze dvě věci. Buď může být chyba v serverové aplikaci nebo pouze v její API dokumentaci.

Na následujícím obrázku se nachází GitLab repozitář, kde jsou otevřeny pro ukázkou poslední provedené automatické testy. Bližší náhled na testování REST API se nachází ještě na konci této kapitoly, kde je zobrazen samotný kód výstupu tohoto testování.

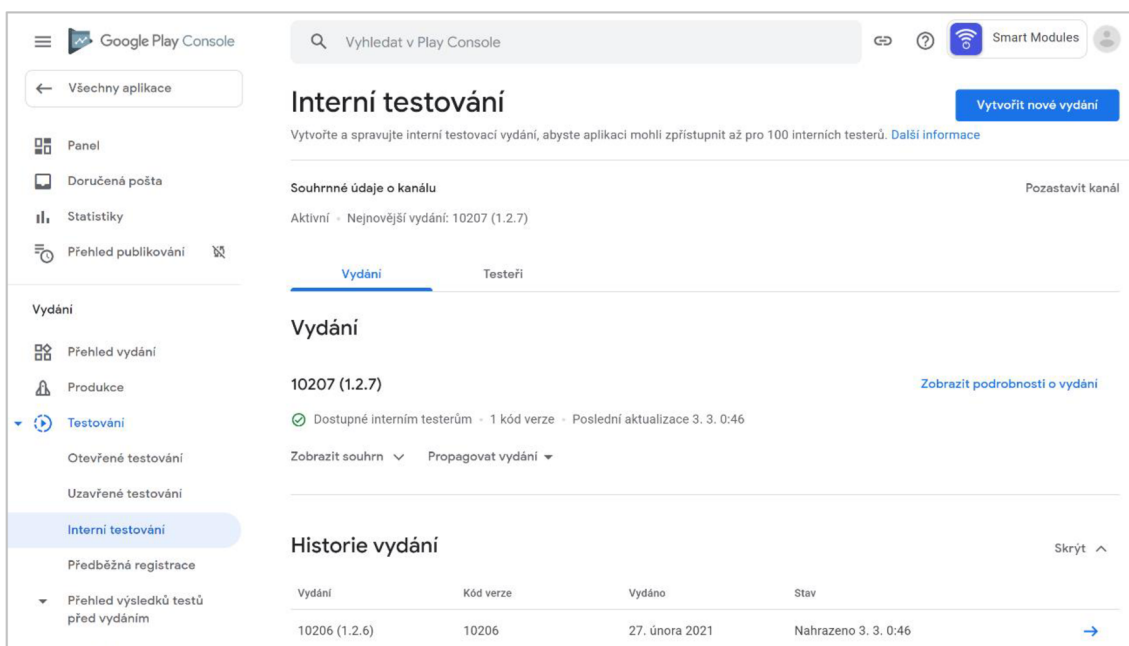


Obrázek 31: Ukázka průběžné integrace serverové aplikace [23]

## 6.3 Mobilní aplikace

Mobilní aplikace je testována především pomocí interního testování, které je dostupné přímo v rámci Google Play konzole [73]. Nová verze aplikace je vždy vydána nejprve interním testerům, což je skupina vybraných reálných uživatelů, kteří mobilní aplikaci aktivně používají. Těmto uživatelům jsou aktualizace mobilní aplikace pravidelně vydávány přednostně, ještě dříve, než jsou schváleny samotnou platformou a mohou tak novou verzi vyzkoušet mnohem dříve, než se dostane k ostatním uživatelům. Díky tomu je možné otestovat aplikaci v uzavřeném kruhu interních testerů a pokud interní testeři nenajdou žádnou chybu, může být nová verze mobilní aplikace následně bezpečně publikována široké veřejnosti přímo na platformě Google Play. Tímto způsobem je aplikace testována už od samotného počátku a díky tomuto testování bylo v minulosti detekováno hned několik závažných chyb, které se nepromítly úplně u všech uživatelů nýbrž pouze u interních testerů, kteří s tím automaticky předem dobrovolně počítali.

Na následujícím obrázku se nachází ukázka rozhraní interního testování v konzoli Google Play, kde je vidět, že poslední testovaná verze aplikace byla „1.2.7“, která se začala testovat 3. března.



The screenshot shows the Google Play Console interface for internal testing. The main heading is 'Interní testování' (Internal Testing). Below the heading, there is a search bar and a 'Vytvořit nové vydání' (Create new release) button. The page is divided into two tabs: 'Vydání' (Releases) and 'Testeři' (Testers). The 'Vydání' tab is active, showing a list of releases. The current release is '10207 (1.2.7)', which is 'Dostupné interním testerům' (Available to internal testers). Below this, there is a 'Historie vydání' (Release history) table with the following data:

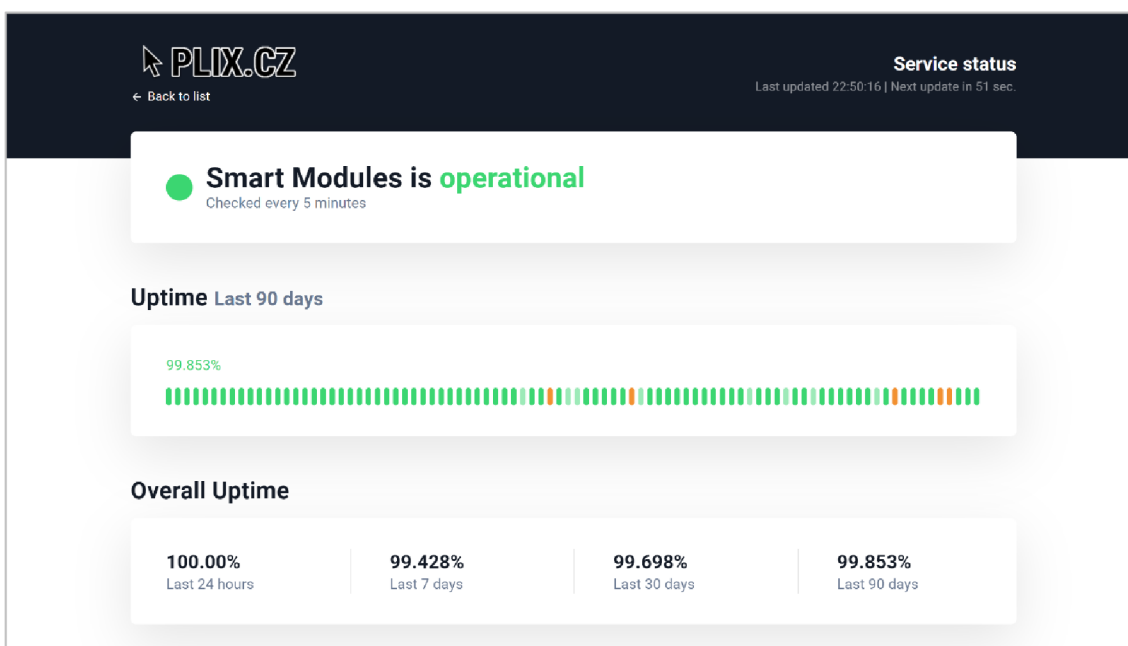
Vydání	Kód verze	Vydáno	Stav
10206 (1.2.6)	10206	27. února 2021	Nahrazeno 3. 3. 0:46

**Obrázek 32: Ukázka rozhraní interního testování v konzoli Google Play [73]**

## 6.4 Dostupnost

Dostupnost celého informačního systému závisí především na dostupnosti samotné serverové aplikace. Proto je u serverové aplikace striktně testována dostupnost. Podle externí monitorovací služby má v poslední době webhosting poměrně velké výpadky, čímž není zaručena garantovaná dostupnost webhostingu 99,99 %. Reálná dostupnost projektu je za posledních 90 dní přibližně 99,853 %. Tato skutečnost byla diskutována s webhostingem a webhosting odpověděl, že se výpadky mohou týkat pouze externího monitoringu, že žádné výpadky v systému sami nevidují. Psali, že v poslední době zpříšňovali opatření proti robotům, kam může občas spadat i externí monitorovací služba. To by potvrzovalo i to, že žádný z reálných uživatelů, který používá chytré moduly, nikdy žádný výpadek nezaregistroval. Nicméně tímto není důkazně potvrzeno splnění nefunkčního požadavku tohoto projektu, kterým je dostupnost minimálně 99,9 %.

Následuje ukázka externí monitorovací služby dostupnosti serverové aplikace. Na tomto snímku obrazovky je vidět, že služba byla nepřetržitě dostupná bez jediného výpadku od 11. ledna do 17. února, ale od té doby došlo ke značným výpapkům, které sesadili dostupnost serverové aplikace za posledních 90 dní na přibližně 99,853 %.



Obrázek 33: Náhled externího monitoringu dostupnosti serveru [76]

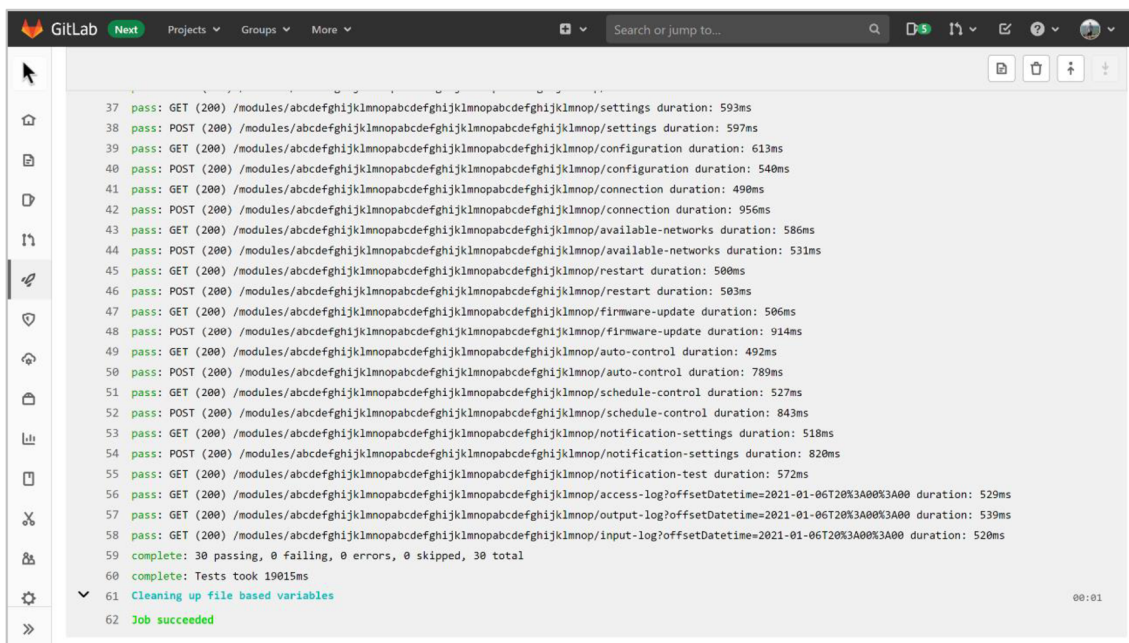


## 6.5 Zálohování

Informace o zálohování jsou podrobně popsány na konci předchozí kapitoly. V rámci otestování zálohování bylo obnovení databáze několikrát vyzkoušeno a zkontrolováno na lokálním serveru, tedy v podobném prostředí a na stejných nebo velmi podobných technologiích, na kterých běží serverová aplikace i v rámci produkčního serveru webhostingu.

## 6.6 Výkon

Jak již bylo zmíněno v podkapitole „Serverová aplikace“, v rámci průběžné integrace probíhá automatické testování REST API serverové aplikace. Tyto testy jsou zároveň i výkonnostními testy, protože u každého testu je měřeno, jak dlouhou dobu celý požadavek a jeho kontrola trvaly. Na následujícím obrázku je ukázka posledního automatického testování, kde je vidět že drtivá většina požadavků je zpracována okolo 500 milisekund, a to i včetně samotného testování hlavičky a obsahu vrácené odpovědi.



```
37 pass: GET (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/settings duration: 593ms
38 pass: POST (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/settings duration: 597ms
39 pass: GET (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/configuration duration: 613ms
40 pass: POST (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/configuration duration: 540ms
41 pass: GET (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/connection duration: 490ms
42 pass: POST (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/connection duration: 956ms
43 pass: GET (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/available-networks duration: 586ms
44 pass: POST (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/available-networks duration: 531ms
45 pass: GET (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/restart duration: 500ms
46 pass: POST (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/restart duration: 503ms
47 pass: GET (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/firmware-update duration: 506ms
48 pass: POST (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/firmware-update duration: 914ms
49 pass: GET (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/auto-control duration: 492ms
50 pass: POST (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/auto-control duration: 789ms
51 pass: GET (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/schedule-control duration: 527ms
52 pass: POST (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/schedule-control duration: 843ms
53 pass: GET (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/notification-settings duration: 518ms
54 pass: POST (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/notification-settings duration: 820ms
55 pass: GET (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/notification-test duration: 572ms
56 pass: GET (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/access-log?offsetDatetime=2021-01-06T20%3A00%3A00 duration: 529ms
57 pass: GET (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/output-log?offsetDatetime=2021-01-06T20%3A00%3A00 duration: 539ms
58 pass: GET (200) /modules/abcdefghijklmnopqrsxyzabcdefghijklmnopqrsxyz/input-log?offsetDatetime=2021-01-06T20%3A00%3A00 duration: 520ms
59 complete: 30 passing, 0 failing, 0 errors, 0 skipped, 30 total
60 complete: Tests took 19015ms
61 Cleaning up file based variables
62 Job succeeded
```

Obrázek 34: Ukázka automatického testování REST API [23]

## 7 Shrnutí

Jako první byla na začátku projektu provedena analýza a návrh, které jsou popsány v předchozích kapitolách. V této analýze a návrhu se mohlo zjistit, že řešení uspokojující funkční požadavky této práce nakonec nebude možné zrealizovat, nebo že již takové řešení existuje. Nicméně opak se stal pravdou a projekt se dostal až do fáze implementace a testování. I v této fázi ostatně mohl být zamítnut, protože zde se také mohlo zjistit, že některé funkcionality prostě a jednoduše nebude možné v domácím prostředí jednoduše zrealizovat. I proto se návrh a implementace dost překrývaly, ve smyslu, že se nejdříve zkusila daná funkcionality alespoň nějak částečně zrealizovat, než byla kompletně naplánována a zakomponována do celého systému. Nicméně u žádné navržené funkcionality nebyl nalezen žádný problém, který by se nepodařilo nějakým způsobem vyřešit.

Prakticky vzniklo pro koncové uživatele rovnou několik druhů elektrických modulů v celkovém množství 8 kusů a mobilní aplikace, kterou dnes používá k ovládání svých chytrých modulů 5 reálných uživatelů. Dále vznikla serverová aplikace, která zastřešuje celý informační systém tím, že pro celý systém poskytuje komunikační a administrační rozhraní.

Tento projekt dokáže v konkrétních případech ušetřit spoustu peněz i času. V rámci tohoto projektu, je jeden z produkčních modulů umístěn například ve vozidle, kde nahrazuje klasické rádiové dálkové ovládání k přídatnému topení. V zimě se tak pomocí tohoto modulu dá pro řidiče předem vytopit kabina tohoto vozidla na příjemnou teplotu, i když je venku krutá zima. Nejen, že vozidlo je na originální dálkový ovladač příliš daleko, aby k němu dosáhl signál pomocí rádiového signálu napřímo z domu jeho uživatele, ale originální dálkový radiofrekvenční ovladač k tomuto přídatnému topení se pohybuje ve vyšší cenové relaci, záleží na konkrétním typu, dovednostech a možnostech ovladače.

I když to tak ze začátku možná trochu bylo zamýšleno, a i přes to, že moduly a celý systém splňují všechny bezpečnostní podmínky a certifikace, nakonec není plánováno, že by se moduly a systém staly nějakou komerční záležitostí a byly nabízeny široké veřejnosti. Pravděpodobně zůstanou pouze k uspokojení potřeb autora této práce a jeho rodiny či přátel.

## 7.1 Limity řešení

Jedním z největších limitů vyvinutého řešení je použití REST API jakožto hlavního komunikačního rozhraní informačního systému. Díky tomu má systém poměrně dlouhou odezvu a běžný uživatel tak může na změnu stavu svého modulu čekat i několik sekund. Dalším limitem řešení by mohly být použité technologie a použitý webhosting, který nepodporuje řádnou integraci průběžného nasazení serverové aplikace. Nicméně tyto technologie byly použity především z důvodu nulových měsíčních nákladů na jejich nepřetržitý provoz. Tyto použité technologie dále zdaleka nejsou nejvýkonnější a dokážou přes aktuální rozhraní současně obsluhovat pouze několik stovek nebo maximálně tisíců hardwarových modulů.

Jedním z dalších limitů řešení je nasazení mobilní aplikace pouze pro operační systém Android, nicméně je tomu tak primárně z toho důvodu, že mobilní aplikace pro jiný operační systém zatím ještě nebyla poptávána. Dále mobilní aplikace obsahuje pouze český jazyk, a to ze stejného důvodu. I z tohoto důvodu je mobilní aplikace distribuována na platformě Google Play pouze pro Českou republiku. Dalším limitem řešení by mohla být dokumentace, která je taktéž pouze v češtině a díky tomu nemůže jednoduše rozšiřovat tento informační systém žádný zahraniční vývojář, který neovládá český jazyk.

## 7.2 Možnosti rozšíření

Všechny limity řešení lze do budoucna vyřešit. Pro rychlejší odezvy lze nasadit WebSocket API. Pro obsluhu většího množství modulů lze využít cloudové řešení s orchestračními nástroji a virtualizací. Mobilní aplikace byla naprogramována multiplatformní metodou, a tak může být v případě potřeby v celku jednoduše distribuována i na platformu iOS. Aby mohli systém případně používat i zahraniční uživatelé, bylo by vhodné přeložit mobilní aplikaci ještě alespoň do anglického jazyka a povolit distribuci mobilní aplikace po celém světě. Díky volně přístupným kódům a dokumentaci může být provedeno libovolné rozšíření systému o novou komponentu, a to libovolným vývojářem na světě. Proto by bylo vhodné API dokumentaci komunikačního rozhraní rovněž přeložit ještě také alespoň do anglického jazyka.

## 8 Závěr

Hlavním cílem této práce byla tvorba informačního systému, který umožňuje levně a chytře ovládat elektrická zařízení, a to vzdáleně pomocí hardwarových modulů a mobilních telefonů přes internet. Tyto hardwarové moduly fungují samostatně a jsou lehce rozšiřitelné o další senzory, a to bez přidání dalších jednotek nebo bez použití centrálního prvku. V rámci praktické části této práce například vznikl hardwarový modul, který obsahuje 2 teplotní senzory a automaticky dokáže reagovat na jejich hodnoty.

Nejprve byla v rámci teoretické části této práce provedena analýza, kde byly shrnuty možnosti připojení modulů k internetu a kde byly analyzovány již existující řešení. Po analýze existujících řešení bylo provedeno srovnání a analýza nového řešení včetně funkčních a nefunkčních požadavků. Následně byl proveden návrh informačního systému a jeho jednotlivých komponent. Byl vytvořen popis uživatelských rolí a detailní popis případů užití plně navazujících na analyzované funkční požadavky. Dále bylo navrženo databázové schéma komunikace mezi jednotlivými softwarovými komponentami informačního systému včetně návrhu API dokumentace a automatizovaných úloh, které probíhají na pozadí systému. Po návrhu byl popsán postup implementace a testování implementovaného řešení. Závěrem byla shrnuta celá práce a definovány její limity spolu s nastíněním možností jejího dalšího budoucího rozšíření.

Cílem praktické části této práce bylo implementovat všechna navržená řešení, včetně nasazení a zprovoznění zálohování kódu i dat. V rámci implementace hardwaru bylo cílem pokrytí nízkého stejnosměrného i vysokého střídavého napětí. Nakonec bylo cílem praktické části této práce také testování implementovaného řešení včetně vhodných nefunkčních požadavků komunikačního rozhraní. Všechna implementovaná řešení jsou dostupná online a jejich zdrojový kód se nachází v digitální příloze této práce.

Všechny cíle této diplomové práce byly úspěšně naplněny a informační systém byl produkčně zprovozněn. Hardwarové moduly a mobilní aplikace jsou již po dlouhé měsíce aktivně používány reálnými uživateli.

## 9 Seznam použité literatury

- [1] What is GSM Module and GPRS Module? [Online] *EXPERT CLASS NOTES*, 2019 [cit. 2021-03-29]. Dostupné z:  
<https://www.electronicsforu.com/resources/gsm-module>
- [2] KABALCI et al. *Design and implementation of a solar plant and irrigation system with remote monitoring and remote control infrastructures*. Solar Energy, 2016 [cit. 2021-03-29]. Dostupné z:  
<http://www.sciencedirect.com/science/article/pii/S0038092X16304911>
- [3] DELPORT, Renier. Introducing the inexpensive ESP-01 Wi-Fi module [Online] *Behind The Scenes*, 2020 [cit. 2021-03-29]. Dostupné z: <https://behind-the-scenes.net/introducing-the-inexpensive-esp-01-wi-fi-module/>
- [4] Základní informace [Online] *Alza.cz a.s.*, 2021 [cit. 2021-03-30]. Dostupné z:  
<https://www.alza.cz/zakladni-informace>
- [5] O nás [Online] *Philips Hue*, 2021 [cit. 2021-03-30]. Dostupné z:  
<https://www.philips-hue.com/cs-cz/about-us>
- [6] O nás [Online] *FIBARO*, 2021 [cit. 2021-03-30]. Dostupné z:  
<https://www.fibaro.com/cz/about-us/>
- [7] O nás [Online] *Legrand*, 2021 [cit. 2021-03-30]. Dostupné z:  
<https://www.legrand.cz/o-nas>
- [8] What is Zigbee? [Online] *Zigbee Alliance*, 2021 [cit. 2021-03-30]. Dostupné z:  
<https://zigbeealliance.org/solution/zigbee/>
- [9] About us [Online] *Gosund*, 2021 [cit. 2021-03-31]. Dostupné z:  
<https://www.gosund.com/about>
- [10] About us [Online] *Woox*, 2021 [cit. 2021-03-31]. Dostupné z:  
<https://wooxhome.com/about-us-i1>
- [11] SONOFF Official Homepage [Online] *Sonoff*, 2021 [cit. 2021-03-31]. Dostupné z: <https://sonoff.tech/>
- [12] Home [Online] *Shelly Cloud*, 2021 [cit. 2021-03-31]. Dostupné z:  
<https://shelly.cloud/>
- [13] O nás [Online] *Agora DMT*, 2021 [cit. 2021-03-31]. Dostupné z:  
<https://www.agora.cz/cs/o-nas/>

- [14] Your Smart Home Center [Online] *eWeLink*, 2021 [cit. 2021-03-31].  
Dostupné z: <https://www.ewelink.cc/en/>
- [15] Google Assistant, your own personal Google [Online] *Google*, 2021 [cit. 2021-04-02]. Dostupné z: <https://assistant.google.com/>
- [16] Amazon Alexa – Learn what Alexa can do [Online] *Amazon.com*, 2021 [cit. 2021-04-02]. Dostupné z:  
<https://www.amazon.com/b?ie=UTF8&node=21576558011>
- [17] iOS Home [Online] *Apple*, 2021 [cit. 2021-04-02]. Dostupné z:  
<https://www.apple.com/ios/home/>
- [18] What is Arduino? [Online] *Arduino*, 2021 [cit. 2021-04-06]. Dostupné z:  
<https://www.arduino.cc/en/Guide/Introduction>
- [19] What is REST [Online] *REST API Tutorial*, 2021 [cit. 2021-04-06]. Dostupné z:  
<https://restfulapi.net/>
- [20] Free Website Monitoring [Online] *UptimeRobot*, 2021 [cit. 2021-04-06].  
Dostupné z: <https://uptimerobot.com/>
- [21] About [Online] *Git*, 2021 [cit. 2021-04-06]. Dostupné z: <https://git-scm.com/about>
- [22] Zálohování [Online] *WEDOS.cz*, 2021 [cit. 2021-04-06]. Dostupné z:  
<https://www.wedos.cz/zalohovani>
- [23] Commits · master · PLISELI1 / PLIX.cz [Online] *GitLab*, 2021 [cit. 2021-04-06]. Dostupné z: <https://gitlab.com/pliseli1/plix/-/commits/master>
- [24] Smart Modules [Online] *Aplikace na Google Play*, 2021 [cit. 2021-04-06].  
Dostupné z: <https://play.google.com/store/apps/details?id=cz.plix.sm>
- [25] What is HTTPS? [Online] *SSL.com*, 2021 [cit. 2021-04-06]. Dostupné z:  
<https://www.ssl.com/faqs/what-is-https/>
- [26] Webhosting NoLimit [Online] *WEDOS.cz*, 2021 [cit. 2021-04-06]. Dostupné z:  
<https://www.wedos.cz/webhosting>
- [27] What Is a Distributed Denial-of-Service (DDoS) Attack? [Online] *Cloudflare*,  
2021 [cit. 2021-04-06]. Dostupné z:  
<https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/>
- [28] Why Google Cloud [Online] *Google Cloud*, 2021 [cit. 2021-04-07]. Dostupné z:  
<https://cloud.google.com/why-google-cloud>

- [29] Push notifications overview [Online] *Google Developers*, 2021 [cit. 2021-04-07]. Dostupné z: <https://web.dev/push-notifications-overview/>
- [30] What Is Wireless Encryption and Why Is It Used? [Online] *Sony USA*, 2021 [cit. 2021-04-07]. Dostupné z: <https://www.sony.com/electronics/support/articles/00009475>
- [31] Crow's Foot Notation [Online] *Vertabelo Database Modeler*, 2016 [cit. 2021-04-07]. Dostupné z: <https://www.vertabelo.com/blog/crow-s-foot-notation/>
- [32] Unix Timestamp [Online] *Epoch Converter*, 2021 [cit. 2021-04-07]. Dostupné z: <https://www.unixtimestamp.com/>
- [33] What is a Key-Value Database? [Online] *MongoDB*, 2021 [cit. 2021-04-07]. Dostupné z: <https://www.mongodb.com/key-value-database>
- [34] The WebSocket API (WebSockets) [Online] *MDN*, 2021 [cit. 2021-04-07]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Web/API/WebSockets\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API)
- [35] How Apiary Works: Fast-track your API Design Process [Online] *Apiary*, 2021 [cit. 2021-04-08]. Dostupné z: <https://apiary.io/how-apiary-works>
- [36] PLIX Smart Modules [Online] *Apiary*, 2021 [cit. 2021-04-08]. Dostupné z: <https://plixsm.docs.apiary.io/>
- [37] API Blueprint [Online] *API Blueprint*, 2021 [cit. 2021-04-08]. Dostupné z: <https://apiblueprint.org/>
- [38] An overview of HTTP [Online] *MDN*, 2021 [cit. 2021-04-08]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>
- [39] Introducing JSON [Online] *JSON*, 2021 [cit. 2021-04-08]. Dostupné z: <https://www.json.org/json-en.html>
- [40] ISO 8601 – Date and time format [Online] *ISO*, 2021 [cit. 2021-04-08]. Dostupné z: <https://www.iso.org/iso-8601-date-and-time-format.html>
- [41] What are session cookies used for? [Online] *All About Cookies*, 2021 [cit. 2021-04-08]. Dostupné z: <https://www.allaboutcookies.org/cookies/session-cookies-used-for.html>
- [42] Specification v1.1 (Still in Development): Errors [Online] *JSON:API*, 2021 [cit. 2021-04-08]. Dostupné z: <https://jsonapi.org/format/1.1/#errors>

- [43] Webhosting - CRON [Online] *WEDOS.cz*, 2021 [cit. 2021-04-08]. Dostupné z: <https://www.wedos.cz/webhosting/cron>
- [44] ESP8285 Datasheet [Online] *Espressif Systems*, 2020 [cit. 2021-04-09]. Dostupné z: [https://www.espressif.com/sites/default/files/documentation/0a-esp8285\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8285_datasheet_en.pdf)
- [45] Corporate Profile [Online] *FTDI*, 2021 [cit. 2021-04-09]. Dostupné z: <https://ftdichip.com/corporate-profile/>
- [46] Software [Online] *Arduino*, 2021 [cit. 2021-04-09]. Dostupné z: <https://www.arduino.cc/en/software>
- [47] PLISELI1 / PLIX ESP8285 module firmware [Online] *GitLab*, 2021 [cit. 2021-04-09]. Dostupné z: <https://gitlab.com/pliseli1/esp8285-firmware>
- [48] DS18B20 - Programmable Resolution 1-Wire Digital Thermometer [Online] *Maxim Integrated Products*, 2019 [cit. 2021-04-09]. Dostupné z: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- [49] PLISELI1 / PLIX ESP8285 module sensor tester [Online] *GitLab*, 2021 [cit. 2021-04-09]. Dostupné z: <https://gitlab.com/pliseli1/esp8285-sensor-tester>
- [50] C reference [Online] *cppreference.com*, 2017 [cit. 2021-04-12]. Dostupné z: <https://en.cppreference.com/w/c>
- [51] Welcome to ESP8266 Arduino Core's documentation! [Online] *ESP8266 Arduino Core*, 2017 [cit. 2021-04-12]. Dostupné z: <https://arduino-esp8266.readthedocs.io/en/latest/index.html>
- [52] About [Online] *Drupal.org*, 2021 [cit. 2021-04-12]. Dostupné z: <https://www.drupal.org/about>
- [53] What is PHP? [Online] *PHP.net*, 2021 [cit. 2021-04-12]. Dostupné z: <https://www.php.net/manual/en/intro-what-is.php>
- [54] What is Symfony [Online] *Symfony*, 2021 [cit. 2021-04-12]. Dostupné z: <https://symfony.com/what-is-symfony>
- [55] About Node.js [Online] *Node.js*, 2021 [cit. 2021-04-12]. Dostupné z: <https://nodejs.org/en/about/>
- [56] About Heroku [Online] *Heroku*, 2021 [cit. 2021-04-12]. Dostupné z: <https://www.heroku.com/about>



- [57] What is an SSL certificate? [Online] *Cloudflare*, 2021 [cit. 2021-04-12]. Dostupné z: <https://www.cloudflare.com/learning/ssl/what-is-an-ssl-certificate/>
- [58] Introduction [Online] *Composer*, 2021 [cit. 2021-04-12]. Dostupné z: <https://getcomposer.org/doc/00-intro.md>
- [59] About the Apache HTTP Server Project [Online] *The Apache HTTP Server Project*, 2021 [cit. 2021-04-12]. Dostupné z: [https://httpd.apache.org/ABOUT\\_APACHE.html](https://httpd.apache.org/ABOUT_APACHE.html)
- [60] About MariaDB Server [Online] *MariaDB.org*, 2021 [cit. 2021-04-12]. Dostupné z: <https://mariadb.org/about/>
- [61] Introduction [Online] *Drush*, 2021 [cit. 2021-04-12]. Dostupné z: <https://www.drush.org/latest/>
- [62] Ionic - Cross-Platform Mobile App Development [Online] *Ionic*, 2021 [cit. 2021-04-13]. Dostupné z: <https://ionicframework.com/>
- [63] Capacitor - build cross platform apps with the web [Online] *Capacitor*, 2021 [cit. 2021-04-13]. Dostupné z: <https://capacitorjs.com/>
- [64] What are Progressive Web Apps? [Online] *Google Developers*, 2021 [cit. 2021-04-13]. Dostupné z: <https://web.dev/what-are-pwas/>
- [65] Introduction [Online] *Vue.js*, 2021 [cit. 2021-04-13]. Dostupné z: <https://vuejs.org/v2/guide/>
- [66] TypeScript: Typed JavaScript at Any Scale [Online] *TypeScript*, 2021 [cit. 2021-04-13]. Dostupné z: <https://www.typescriptlang.org/>
- [67] What is MITM (Man in the Middle) Attack [Online] *Imperva*, 2021 [cit. 2021-04-13]. Dostupné z: <https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/>
- [68] About Let's Encrypt [Online] *Let's Encrypt*, 2021 [cit. 2021-04-13]. Dostupné z: <https://letsencrypt.org/about/>
- [69] Iterate faster, innovate together [Online] *GitLab*, 2021 [cit. 2021-04-14]. Dostupné z: <https://about.gitlab.com/>
- [70] What is CI/CD? [Online] *RedHat*, 2021 [cit. 2021-04-14]. Dostupné z: <https://www.redhat.com/en/topics/devops/what-is-ci-cd>

- [71] Coding standards [Online] *Drupal Wiki guide on Drupal.org*, 2020 [cit. 2021-04-14]. Dostupné z: <https://www.drupal.org/docs/develop/standards/coding-standards>
- [72] Meet Android Studio [Online] *Android Developers*, 2021 [cit. 2021-04-14]. Dostupné z: <https://developer.android.com/studio/intro>
- [73] Google Play Console [Online] *Android Developers*, 2021 [cit. 2021-04-14]. Dostupné z: <https://developer.android.com/distribute/console>
- [74] Backup and Migrate [Online] *Drupal.org*, 2021 [cit. 2021-04-14]. Dostupné z: [https://www.drupal.org/project/backup\\_migrate](https://www.drupal.org/project/backup_migrate)
- [75] Dredd — HTTP API Testing Framework [Online] *Dredd*, 2021 [cit. 2021-04-15]. Dostupné z: <https://dredd.org/en/latest/>
- [76] Smart Modules - service status [Online] *PLIX.cz*, 2021 [cit. 2021-04-09]. Dostupné z: <https://status.plix.cz/786265921>

## 10 Přílohy

### 10.1 Seznam použitých zkratk

**AC** Alternating current

**AP** Access Point

**API** Application Programming Interface

**BSSID** Basic Service Set Identifier

**CD** Continuous Delivery

**CI** Continuous Integration

**CMS** Content management system

**CRUD** Create, Read, Update, Delete

**DC** Direct current

**DDoS** Distributed Denial-of-service

**FR** Functional Requirements

**GCM** Google Cloud Messaging

**GUI** Graphical User Interface

**HTTP** Hypertext Transfer Protocol

**HTTPS** Hypertext Transfer Protocol Secure

**ID** Identifikátor

**IoT** Internet of Things

**IP** Internet Protocol

**IS** Informační systém

**JSON** JavaScript Object Notation

**LED** Light-Emitting Diode

**NR** Non-functional Requirements

**REST** Representational State Transfer

**RSSI** Received Signal Strength Indication

**SIM** Subscriber Identity Module

**SSID** Service Set Identifier

**URL** Uniform resource locator

**UUID** Universally unique identifier

## 10.2 Obsah praktické části práce

K této teoretické části práce je v digitální podobě přiložena ještě praktická část, která obsahuje všechny elektronické podklady, které byly použity k tvorbě této práce. Tato praktická část obsahuje programové kódy jednotlivých softwarových komponent informačního systému, dále obsahuje veškeré fotografie, obrázky, schémata a digramy, které byly použity pro tvorbu této práce, včetně všech zdrojových nebo originálních podob. V neposlední řadě obsahuje také odkazy na online repozitáře a produkční verze praktické části této práce a text teoretické části v elektronické podobě včetně zdrojového formátu. Níže se nachází popis obsahu této digitální přílohy, který lze najít i v souboru s názvem „readme.txt“ umístěném v jejím kořenovém adresáři.

Je možné, že příloha, která je přímo připojená k této práci, neobsahuje kvůli velikostnímu limitu nějaké méně důležité soubory, například originální zdrojové formy. Pro úplně zcela kompletní praktickou část je nutné přejít na odkaz, který se nachází v následující příloze.

### Popis obsahu digitální přílohy:

	readme.txt	.....	stručný popis obsahu praktické části práce
	links.txt	.....	odkazy na praktickou část práce
	thesis.pdf	.....	text teoretické části práce
	src	.....	zdrojové kódy
	impl	.....	zdrojové kódy implementace
	firmware	.....	zdrojové kódy firmwarů pro moduly
	mobile-app	.....	zdrojový kód mobilní aplikace
	server-app	.....	zdrojový kód serverové aplikace
	thesis	.....	zdrojová forma teoretické části práce
	doc	.....	zdrojová forma API dokumentace
	sh	.....	zdrojové kódy shellových skriptů
	img	.....	obrázkové přílohy a jejich zdrojové formy
	app	.....	snímky obrazovek mobilní aplikace
	dark	.....	snímky obrazovek ve tmavém režimu
	light	.....	snímky obrazovek ve světlém režimu
	used	.....	snímky obrazovek použité v teoretické části práce
	photos	.....	fotografie
	source	.....	původní (originální) fotografie
	used	.....	upravené fotografie použité v teoretické části práce
	schemes	.....	schémata a digramy
	source	.....	zdrojová forma schémat a diagramů
	used	.....	schémata a digramy použité v teoretické části práce
	screenshots	.....	snímky obrazovek použité v teoretické části práce

### 10.3 Odkazy na praktickou část práce

Tato příloha obsahuje online odkazy na veškerou volně přístupnou praktickou část této diplomové práce. V první řadě obsahuje seznam všech dostupných a použitých GitLab repozitářů. Dále obsahuje odkazy na produkční verze serverové a mobilní aplikace včetně odkazu na základní adresu REST API serverové aplikace a jeho online dokumentaci. Jako poslední obsahuje odkazy na externí monitoring dostupnosti serveru.

#### Kompletní forma praktické části této práce:

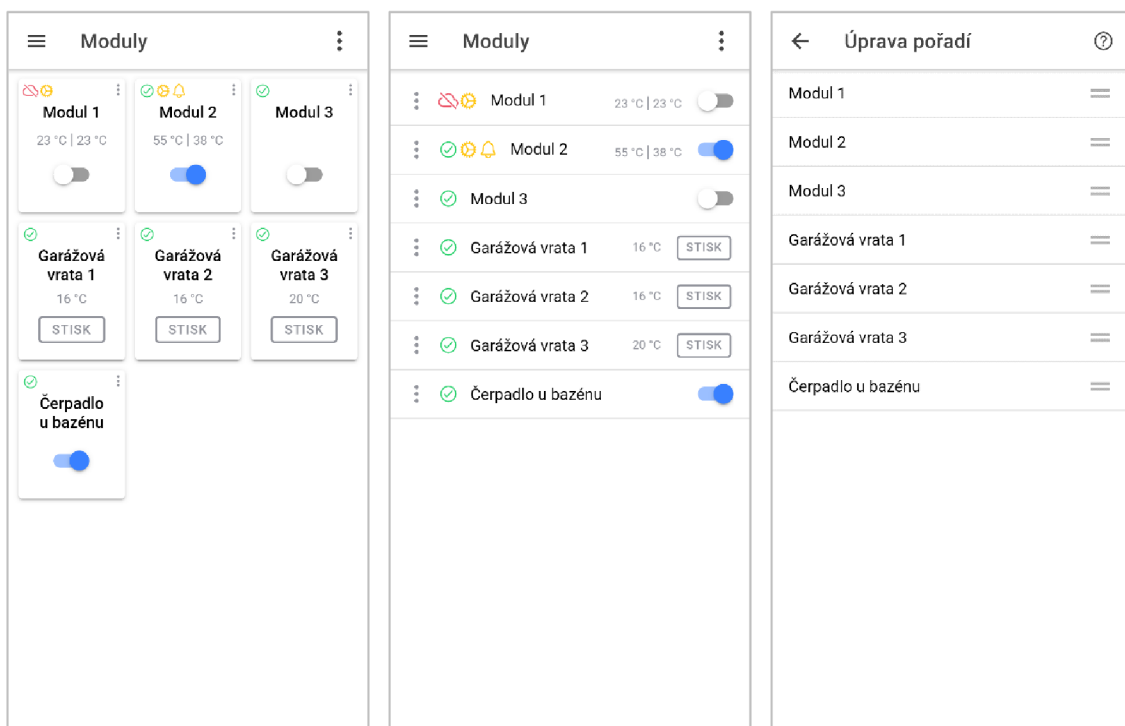
- [https://drive.google.com/drive/folders/1Ym96NK\\_ruyVCpCl0\\_Rm3hTs56o80Z9b](https://drive.google.com/drive/folders/1Ym96NK_ruyVCpCl0_Rm3hTs56o80Z9b) (složka na Google disku)
- <https://drive.google.com/file/d/1Y-cxoaiV0n-U4KQuJrXhTGXm2-4GsWlv/view> (ZIP soubor)

#### Seznam odkazů na praktickou část této práce:

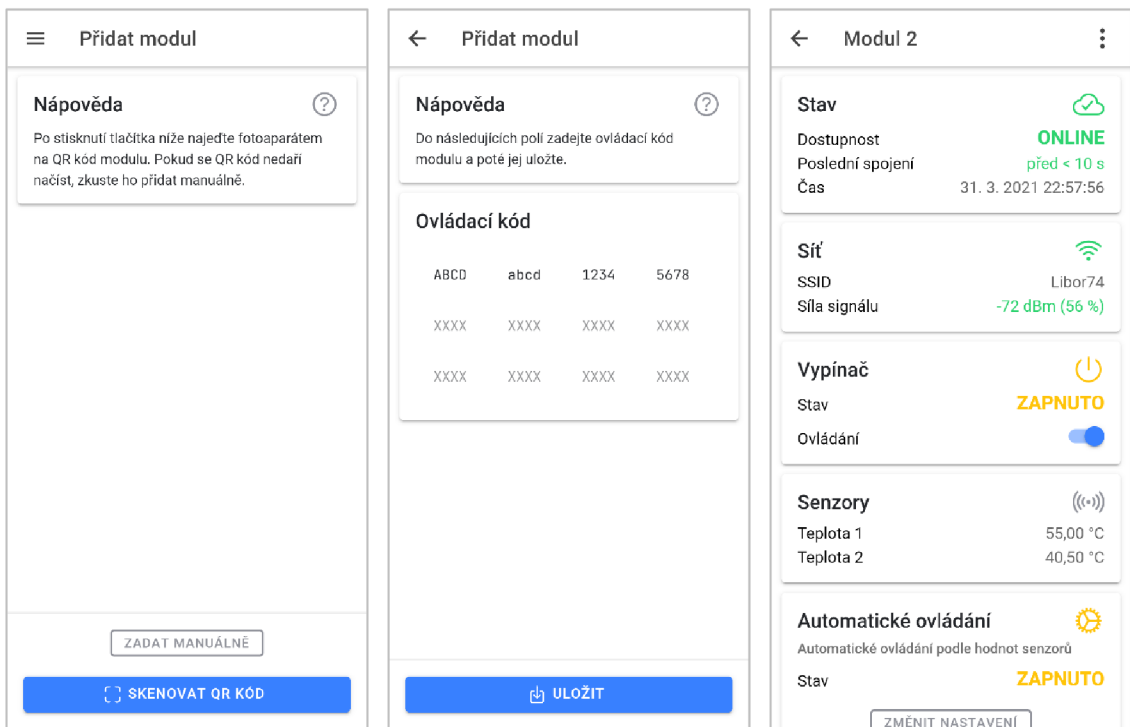
- GitLab skupina s projekty: <https://gitlab.com/pliseli1>
- Repozitář serveru (obsahuje více projektů): <https://gitlab.com/pliseli1/plix>
- Repozitář mobilní aplikace: <https://gitlab.com/pliseli1/smart-modules>
- Repozitář firmwaru: <https://gitlab.com/pliseli1/esp8285-firmware>
- Repozitář firmwaru pro prvotní inicializaci modulu:  
<https://gitlab.com/pliseli1/esp8285-fs-initializer>
- Repozitář firmwaru pro testování senzorů:  
<https://gitlab.com/pliseli1/esp8285-sensor-tester>
- Produkční verze mobilní aplikace pro platformu Android:  
<https://play.google.com/store/apps/details?id=cz.plix.sm>
- Dokumentace REST API serverové aplikace: <https://plixsm.docs.apiary.io/>
- Produkční verze serveru (obsahuje více projektů): <https://plix.cz/>
- Produkční verze serverové aplikace: <https://sm.plix.cz/>
- Základní adresa REST API: <https://sm.plix.cz/api/v1/>
- Dostupnost serveru (obsahuje více projektů): <https://status.plix.cz/>
- Dostupnost serverové aplikace: <https://status.plix.cz/786265921>
- Záložní adresa dostupnosti: <https://stats.uptimerobot.com/LnmkzHPokD>

## 10.4 Grafické uživatelské rozhraní mobilní aplikace

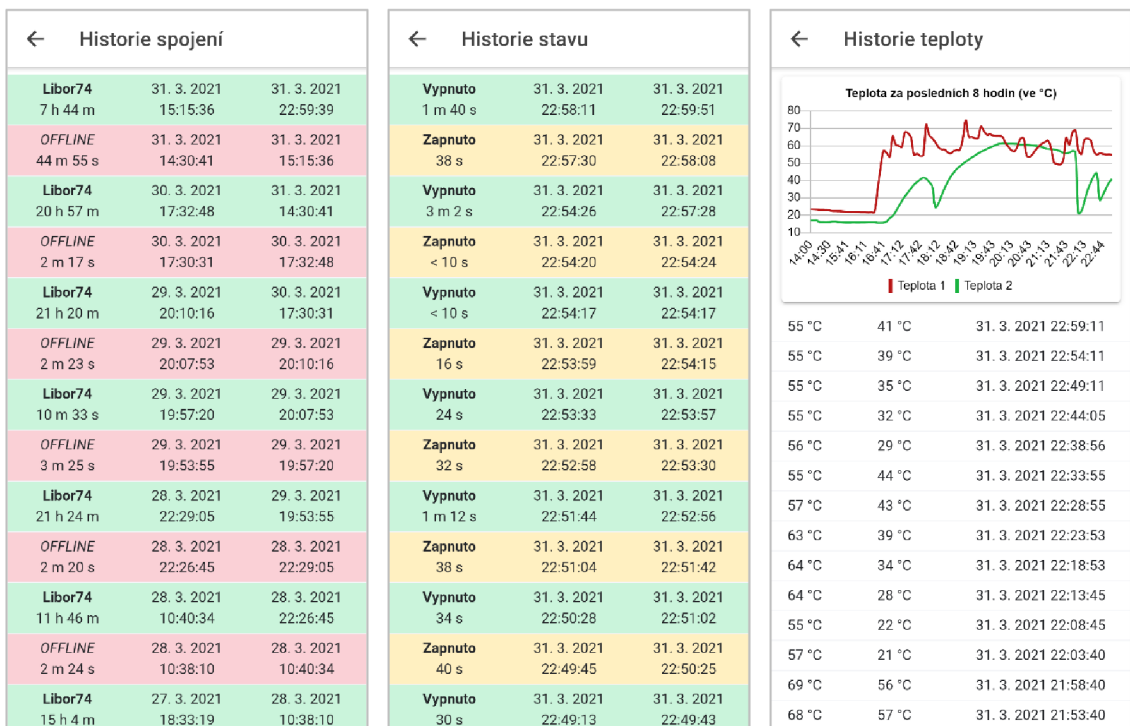
Tato příloha obsahuje reálné snímky obrazovek grafického uživatelského rozhraní (GUI) mobilní aplikace. Mobilní aplikace je hlavní prezentací tohoto projektu a obsahuje celkem 26 druhů různých obrazovek. V rámci nastavení mobilní aplikace si lze vybrat tmavý nebo světlý režim. I když byla aplikace vyvíjena primárně ve tmavém režimu a z analyzovaných uživatelských dat taktéž vyplývá, že uživatelé nejčastěji používají tmavý režim, v rámci úspory a zachování kvality případných výtisků této práce, bylo přiloženo 24 snímků obrazovek mobilní aplikace pouze ve světlém režimu. Snímky všech obrazovek mobilní aplikace ve světlém i tmavém režimu se nacházejí v elektronické příloze této práce. Na některých snímcích byly skryty (rozostřeny) citlivé údaje nebo citlivá data (například hesla k Wi-Fi přístupovým bodům).



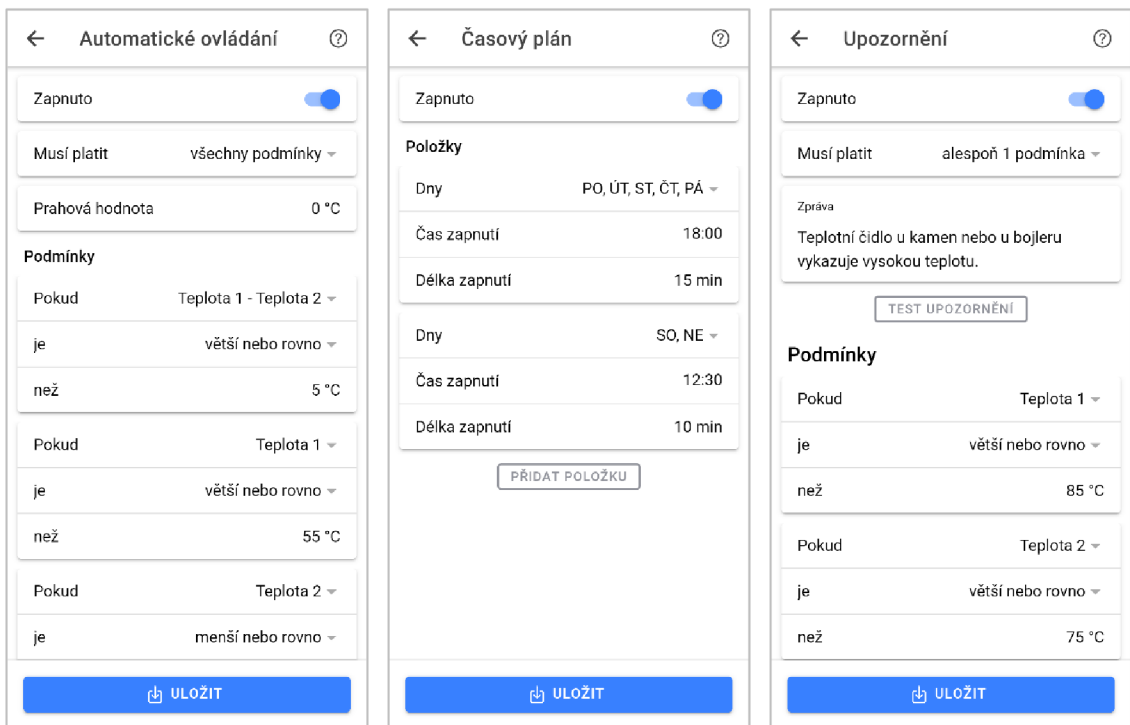
Obrázek 35: Snímky obrazovek mobilní aplikace (1)



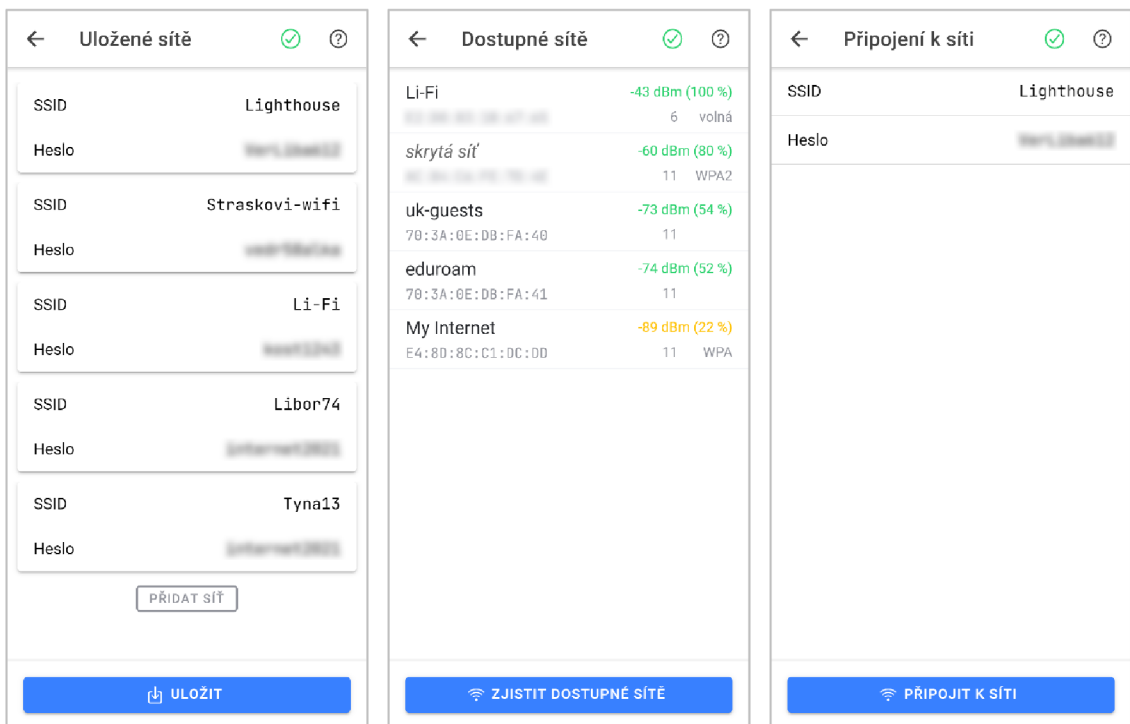
Obrázek 36: Snímky obrazovek mobilní aplikace (2)



Obrázek 37: Snímky obrazovek mobilní aplikace (3)

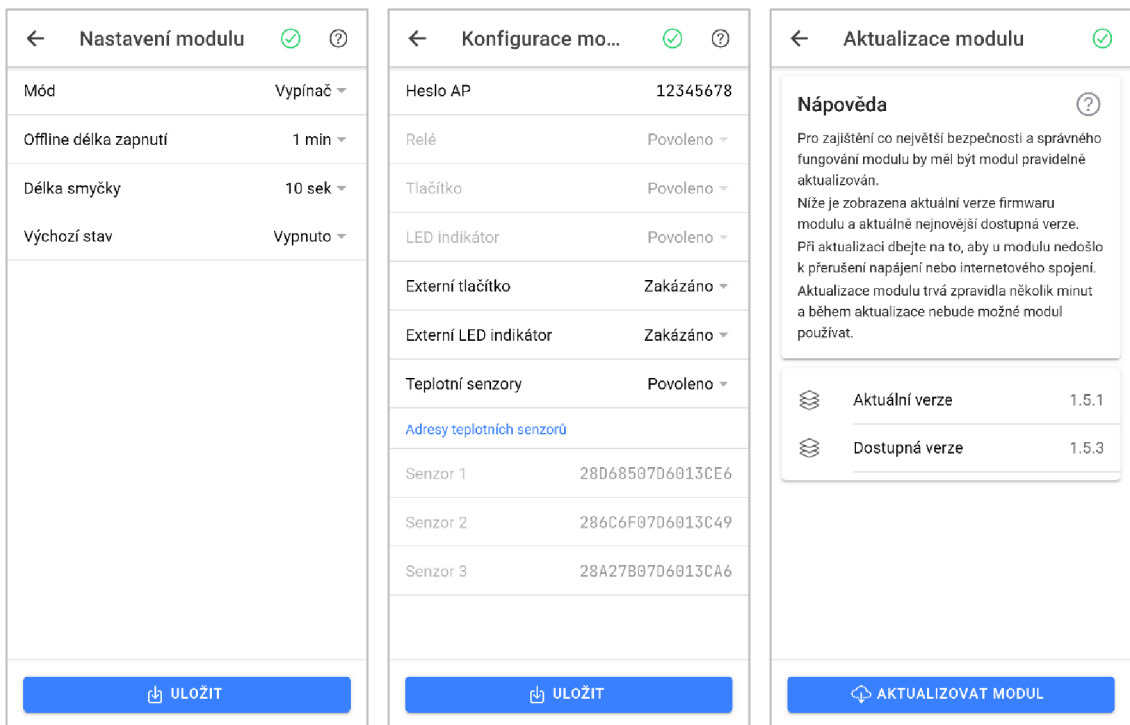


Obrázek 38: Snímky obrazovek mobilní aplikace (4)

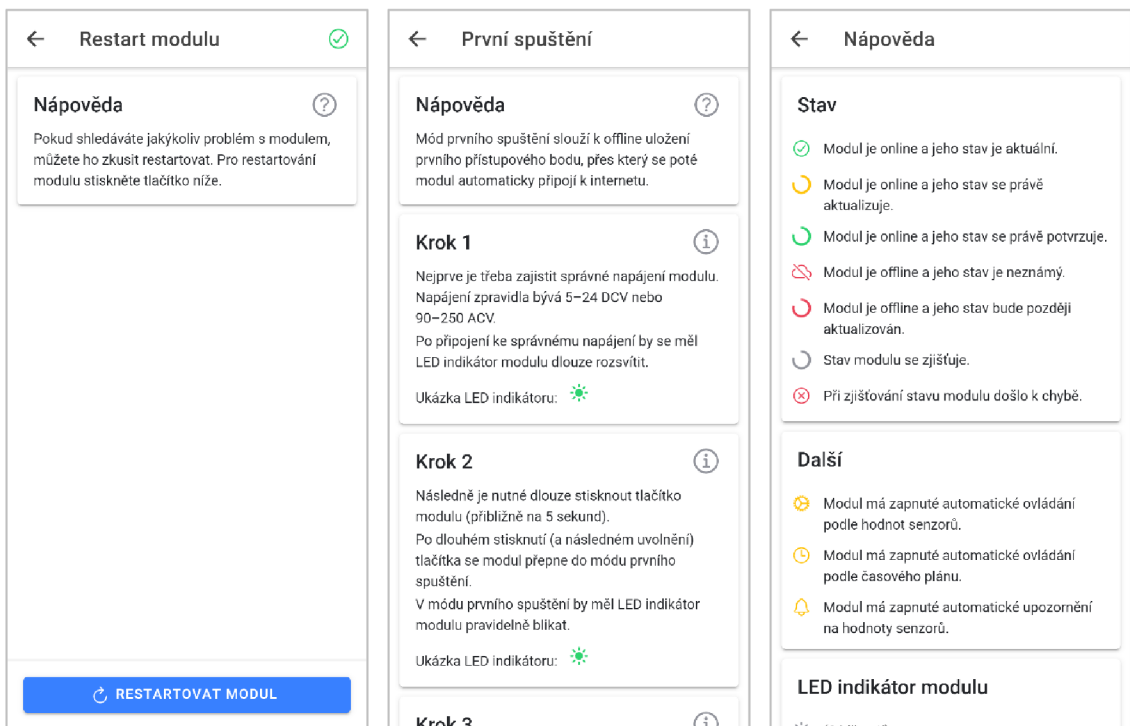


Obrázek 39: Snímky obrazovek mobilní aplikace (5)

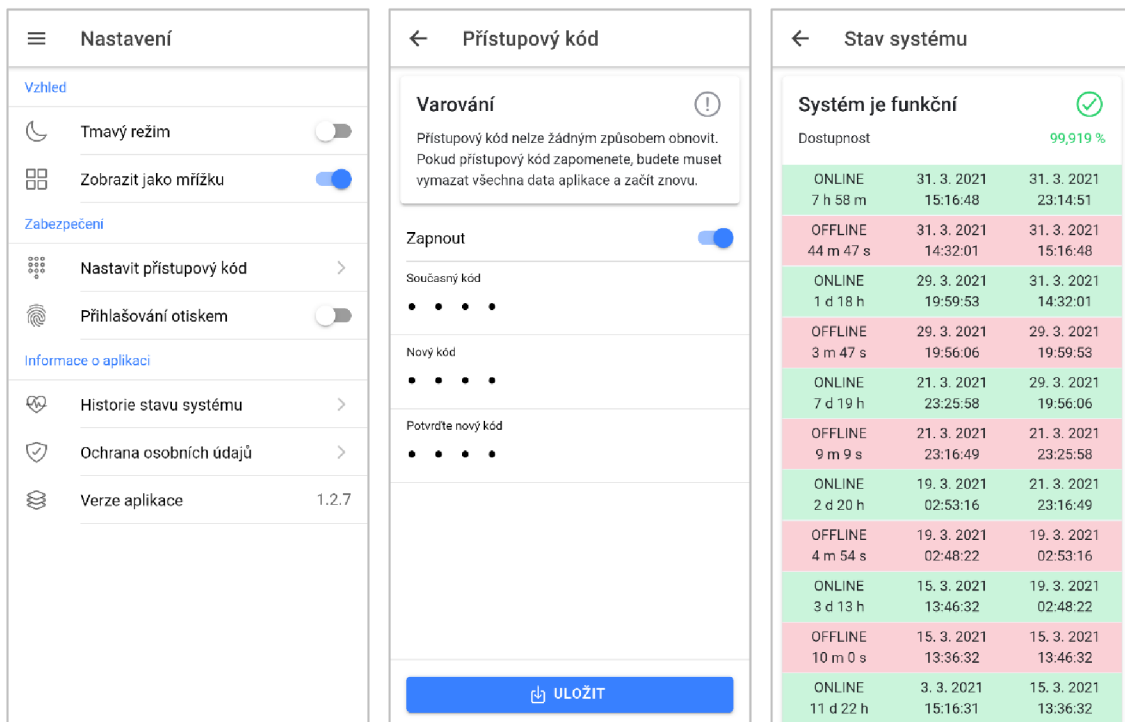




Obrázek 40: Snímky obrazovek mobilní aplikace (6)



Obrázek 41: Snímky obrazovek mobilní aplikace (7)



Obrázek 42: Snímky obrazovek mobilní aplikace (8)

## 10.5 Databázové tabulky serverové aplikace

Tato příloha obsahuje seznam všech databázových tabulek serverové aplikace, které byly navrženy v rámci teoretické části této diplomové práce. Všechny tyto tabulky vycházejí pro lepší přehlednost pouze z návrhu databáze. To znamená, že jejich reálná implementace se může poměrně výrazně lišit kvůli zachování standardů nebo nejlepších praktik použitých technologií. Povinné položky jsou označeny hvězdičkou „\*“. Privátní klíče jsou označeny pomocí symbolů „[PK]“ a jejich název je zvýrazněn tučným fontem. Unikátní klíče jsou označeny pomocí symbolů „[UK]“ a cizí klíče jsou označeny pomocí symbolů „[FK]“. Kompletní databázové schéma, ve kterém jsou znázorněny všechny tabulky i včetně relací mezi nimi, se nachází ve čtvrté kapitole této práce s názvem „Návrh“. Jedinou výjimkou jsou dvě tabulky „BAN\_IP“ a „BAN\_IP\_LOG“, které pro lepší přehlednost nejsou uvedeny v návrhu databáze a které slouží pouze k zajištění bezpečnosti serverové aplikace, nikoliv k naplnění jejich funkčních požadavků.

**Tabulka 5: Databázová tabulka MODULE**

Název	Typ a vlastnosti	Popis
ID*	INT(10) [PK]	Identifikátor
NAME*	VARCHAR(40)	Název
DEVICE_ID*	VARCHAR(20) [UK]	Identifikátor zařízení
STATUS_TOKEN*	VARCHAR(40) [UK]	Autentizační token pro čtení stavu
DEVICE_TOKEN*	VARCHAR(80) [UK]	Autentizační token pro zařízení
CONTROL_TOKEN*	VARCHAR(80) [UK]	Autentizační token pro ovládání
QR_CODE_FILE*	FILE	Obrázek ovládacího QR kódu
FIRMWARE_ID	INT(10) [FK]	Identifikátor firmwaru
NOTE	TEXT	Poznámka

**Tabulka 6: Databázová tabulka FIRMWARE**

Název	Typ a vlastnosti	Popis
ID*	INT(10) [PK]	Identifikátor
VERSION*	VARCHAR(20) [UK]	Verze firmwaru
FILE*	FILE	Binární soubor firmwaru
NOTE	TEXT	Poznámka

**Tabulka 7: Databázová tabulka APP**

Název	Typ a vlastnosti	Popis
ID*	INT(10) [PK]	Identifikátor
UUID*	VARCHAR(40) [UK]	Univerzální unikátní identifikátor
PLATFORM*	ENUM	Platforma (operační systém)
VERSION*	VARCHAR(20)	Verze mobilní aplikace
FCM_TOKEN*	VARCHAR(240) [UK]	FCM token pro zasílání push notifikací
ACCESS_TIME*	BIGINT(20)	Poslední čas přístupu

**Tabulka 8: Databázová tabulka MODULE\_APP**

Název	Typ a vlastnosti	Popis
MODULE_ID*	INT(10) [PK] [FK]	Identifikátor modulu
APP_ID*	INT(10) [PK] [FK]	Identifikátor mobilní aplikace

**Tabulka 9: Databázová tabulka STATUS**

Název	Typ a vlastnosti	Popis
<b>MODULE_ID*</b>	INT(10) [PK] [FK]	Identifikátor modulu
OUT_CHANGE*	ENUM	Stav změny výstupu
FW_UPDATE*	ENUM	Stav aktualizace firmwaru
RESTART*	ENUM	Stav restartu
CFG_SAVE*	ENUM	Stav uložení nové konfigurace
STG_SAVE*	ENUM	Stav uložení nového nastavení
NET_SAVE*	ENUM	Stav uložení nových Wi-Fi sítí
NET_SCAN*	ENUM	Stav skenování dostupných sítí
CONNECT*	ENUM	Stav připojení k nové síti

**Tabulka 10: Databázová tabulka LAST\_VALUES**

Název	Typ a vlastnosti	Popis
<b>MODULE_ID*</b>	INT(10) [PK] [FK]	Identifikátor modulu
RELAY_STATE	ENUM	Stav elektromagnetického relé modulu
SSID	VARCHAR(40)	Název poslední Wi-Fi sítě (AP)
RSSI	TINYINT(4)	Poslední síla signálu Wi-Fi sítě (AP)
FW_VERSION	VARCHAR(20)	Poslední verze firmwaru modulu
ACCESS_TIME	BIGINT(20)	Poslední čas přístupu
RST_TIME	BIGINT(20)	Poslední čas restartu (chyby)
CTRL_TIME	BIGINT(20)	Poslední čas ovládání
LED_VALUES	BLOB	Poslední hodnoty LED kanálů
TEMP_VALUES	BLOB	Poslední hodnoty teplotních senzorů

**Tabulka 11: Databázová tabulka NEW\_VALUES**

Název	Typ a vlastnosti	Popis
<b>MODULE_ID*</b>	INT(10) [PK] [FK]	Identifikátor modulu
RELAY_STATE*	ENUM	Nový stav elektromagnetického relé
LED_VALUES	BLOB	Nové hodnoty LED kanálů

**Tabulka 12: Databázová tabulka CONDITIONS**

Název	Typ a vlastnosti	Popis
<b>MODULE_ID*</b>	INT(10) [PK] [FK]	Identifikátor modulu
ENABLED*	BOOLEAN	Stav automatizace podle podmínek
THRESHOLD	INT(10)	Prahová hodnota
CONFIG_JSON	VARCHAR(255)	Nastavení automatizace podle podmínek

**Tabulka 13: Databázová tabulka SCHEDULE**

Název	Typ a vlastnosti	Popis
<b>MODULE_ID*</b>	INT(10) [PK] [FK]	Identifikátor modulu
ENABLED*	BOOLEAN	Stav časového plánu
CONFIG_JSON	VARCHAR(255)	Nastavení časového plánu

**Tabulka 14: Databázová tabulka NOTIFICATIONS**

Název	Typ a vlastnosti	Popis
<b>MODULE_ID*</b>	INT(10) [PK] [FK]	Identifikátor modulu
ENABLED*	BOOLEAN	Stav automatizovaných push notifikací
MESSAGE	VARCHAR(255)	Text (zpráva) push notifikace
CONFIG_JSON	VARCHAR(255)	Nastavení automatizovaných push notifikací

**Tabulka 15: Databázová tabulka CONNECT\_TO**

Název	Typ a vlastnosti	Popis
<b>MODULE_ID*</b>	INT(10) [PK] [FK]	Identifikátor modulu
SSID*	VARCHAR(40)	Název Wi-Fi sítě (AP)
PASSWORD	VARCHAR(40)	Heslo k Wi-Fi síti (AP)

**Tabulka 16: Databázová tabulka NETWORK\_SAVE**

Název	Typ a vlastnosti	Popis
<b>MODULE_ID*</b>	INT(10) [PK] [FK]	Identifikátor modulu
<b>ID*</b>	INT(10) [PK]	Identifikátor (klíč)
SSID*	VARCHAR(40)	Název Wi-Fi sítě (AP)
PASSWORD	VARCHAR(40)	Heslo k Wi-Fi síti (AP)

**Tabulka 17: Databázová tabulka NETWORK\_SCAN**

Název	Typ a vlastnosti	Popis
<b>MODULE_ID*</b>	INT(10) [PK] [FK]	Identifikátor modulu
<b>ID*</b>	INT(10) [PK]	Identifikátor (klíč)
SSID*	VARCHAR(40)	Název Wi-Fi sítě (AP)
RSSI*	TINYINT(4)	Síla signálu Wi-Fi sítě (AP)
BSSID*	VARCHAR(20)	Adresa Wi-Fi sítě (AP)
CHANNEL*	TINYINT(4)	Kanál Wi-Fi sítě (AP)
ENC_TYPE*	ENUM	Typ zabezpečení Wi-Fi sítě (AP)

**Tabulka 18: Databázová tabulka SETTIGNS**

Název	Typ a vlastnosti	Popis
<b>MODULE_ID*</b>	INT(10) [PK] [FK]	Identifikátor modulu
MODE*	ENUM	Mód modulu
DEFAULT_STATE*	ENUM	Výchozí stav elektromagnet. relé
STATE_DELAY*	INT(10)	Maximální délka stavu
LOOP_DELAY*	INT(10)	Délka opakovací smyčky
DEFAULT_LED_VALUES	BLOB	Výchozí hodnoty LED kanálů

**Tabulka 19: Databázová tabulka CONFIG**

Název	Typ a vlastnosti	Popis
<b>MODULE_ID*</b>	INT(10) [PK] [FK]	Identifikátor modulu
PASSWORD*	VARCHAR(40)	Heslo přístupového bodu (AP)
PIN_BUTTON*	TINYINT(4)	Pin výchozího HW tlačítka
PIN_RELAY*	TINYINT(4)	Pin elektromagnetického relé
PIN_LED*	TINYINT(4)	Pin výchozího LED indikátoru
PIN_EXT_BUTTON*	TINYINT(4)	Pin externího HW tlačítka
PIN_EXT_LED*	TINYINT(4)	Pin externího LED indikátoru
PIN_TEMPS*	TINYINT(4)	Pin teplotních senzorů

**Tabulka 20: Databázová tabulka CONFIG\_PIN\_LED**

Název	Typ a vlastnosti	Popis
<b>MODULE_ID*</b>	INT(10) [PK] [FK]	Identifikátor modulu
ID*	INT(10) [PK]	Identifikátor (klíč)
PIN_LED*	TINYINT(4)	Pin LED kanálu

**Tabulka 21: Databázová tabulka CONFIG\_TEMP\_SENSOR**

Název	Typ a vlastnosti	Popis
<b>MODULE_ID*</b>	INT(10) [PK] [FK]	Identifikátor modulu
ID*	INT(10) [PK]	Identifikátor (klíč)
DEVICE_ID*	VARCHAR(20)	Identifikátor senzoru
DEVICE_ADDRESS*	VARCHAR(20)	Adresa senzoru
NOTE	TEXT	Poznámka

**Tabulka 22: Databázová tabulka LOG\_ACCESS**

Název	Typ a vlastnosti	Popis
<b>MODULE_ID*</b>	INT(10) [PK]	Identifikátor modulu
<b>START_TIME*</b>	BIGINT(20) [PK]	Počáteční čas intervalu
END_TIME*	BIGINT(20)	Konečný čas intervalu
SSID*	VARCHAR(40)	Název Wi-Fi sítě (AP)
RESTART_COUNT*	INT(10)	Počet restartů (chyb) modulu.

**Tabulka 23: Databázová tabulka LOG\_RELAY\_STATE**

Název	Typ a vlastnosti	Popis
<b>MODULE_ID*</b>	INT(10) [PK]	Identifikátor modulu
<b>START_TIME*</b>	BIGINT(20) [PK]	Počáteční čas intervalu
END_TIME*	BIGINT(20)	Konečný čas intervalu
RELAY_STATE*	ENUM	Stav elektromagnetického relé

**Tabulka 24: Databázová tabulka LOG\_TEMPERATURES**

Název	Typ a vlastnosti	Popis
<b>MODULE_ID*</b>	INT(10) [PK]	Identifikátor modulu
<b>START*</b>	BIGINT(20) [PK]	Čas záznamu
TEMP_VALUES	BLOB	Hodnoty teplotních senzorů

**Tabulka 25: Databázová tabulka BAN\_IP**

Název	Typ a vlastnosti	Popis
<b>ID*</b>	INT(10) [PK]	Identifikátor
<b>IP*</b>	VARCHAR(40) [PK]	IP adresa

**Tabulka 26: Databázová tabulka BAN\_IP\_LOG**

Název	Typ a vlastnosti	Popis
<b>TIME*</b>	BIGINT(20) [PK]	Čas přístupu
<b>IP*</b>	VARCHAR(40) [PK]	IP adresa



## Podklad pro zadání DIPLOMOVÉ práce studenta

Jméno a příjmení: **Libor Plíšek**  
Osobní číslo: **I1900286**  
Adresa: **Nad Sokolovnou 239, Nasavrky, 53825 Nasavrky, Česká republika**  
Téma práce: **Návrh a implementace Smart Home řešení s prvky IoT**  
Téma práce anglicky: **Design and Implementation of Smart Home solution with IoT**  
Vedoucí práce: **prof. Ing. Ondřej Krejcar, Ph.D.**  
**Centrum základního a aplikovaného výzkumu**

Zásady pro vypracování:

Osnova:

- Úvod
- Analýza stávajících (současných) řešení
- Analýza nového řešení (funkční a nefunkční požadavky, případy užití)
- Návrh nového řešení (komunikace mezi komponentami systému, databáze)
- Implementace řešení (firmware, mobilní aplikace, serverová aplikace)
- Testování vyvinutého řešení
- Diskuze výsledků, limity řešení a možnosti rozšíření
- Závěr

Seznam doporučené literatury:

dle dohody

Podpis studenta:



Datum: 27.3.2021

Podpis vedoucího práce:



Datum: 28.3.2021