FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Platforma pro chytrou domácnost využívající WIFI spojení jednotek s RaspberryPI
Student:	Tomáš Trejdl
Vedoucí:	Ing. Pavel Kubalík, Ph.D.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2020/21

Pokyny pro vypracování

1. Prozkoumejte existující řešení.
2. Pomocí metod softwarového inženýrství navrhněte vlastní řešení vyhovující níže uvedeným požadavkům.
3. Navržené řešení zrealizujte, naprogramujte, řádně ho zdokumentujte a otestujte.
4. Požadavky:
- základem celého zařízení bude webová aplikace běžící na platformě RaspberryPI,
- pro jednotlivá zařízení v domácnosti použijte modul s ESP8266 a implementujte vlastní firmware v prostředí Arduino,
- zařízení spolu budou komunikovat prostřednictvím wifi sítě,
- v domácnosti bude možné sledovat teplotu a detekovat otevření dveří,
- ovládat bude možné zásuvky a světla,
- aplikace bude umožňovat konfiguraci zařízení s pomocí webové aplikace,
- dále bude možné graficky zobrazovat průběh teploty,
- zaměřte se hlavně na efektivní práci s větším množstvím modulů,
- jednoduchý protokol pro komunikaci s možností snadného rozšíření.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 4. února 2020

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Bakalářská práce

Platforma pro chytrou domácnost využívající Wi-Fi spojení jednotek s Raspberry Pi

Tomáš Trejdl

Katedra softwarového inženýrství
Vedoucí práce: Ing. Pavel Kubalík, Ph.D.

3. června 2020

Poděkování

Chtěl bych poděkovat svému vedoucímu práce, rodině, Mistru Ipovi a komunitě vývojářů na Stack Overflow.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principu při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisu. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisu, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programu, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo"), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli zpusobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelum). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 3. června 2020 . . . . . .

Odkaz na tuto práci

Trejdl, Tomáš. *Platforma pro chytrou domácnost využívající Wi-Fi spojení jednotek s Raspberry Pi*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.

Abstrakt

Chytrá domácnost je pojem, který označuje moderní domácnosti, kde lze spotřebiče, osvětlení nebo jiná elektronická zařízení ovládat na dálku prostřednictvím počítačové sítě a automatizovat jejich funkce. Tato práce se zaměřuje na řešení problému interoperability mezi produkty pro chytrou domácnost od různých výrobců. Práce zkoumá existující komerční řešení a řešení typu *Udělej si sám* na trhu a dále se zabývá návrhem a implementací inteligentní domácí platformy, která se snadno používá a lze ji rozšířit o podporu různých zařízení. Výsledná platforma využívá připojení Wi-Fi k ovládání světel a inteligentních zásuvek a čtení dat ze senzorů teploty a vlhkosti a senzorů otevření dveří. Uživatel interaguje s platformou prostřednictvím progresivní webové aplikace spustitelné v moderním webovém prohlížeči.

Klíčová slova chytrá domácnost, inteligentní domácnost, Wi-Fi, RaspberryPi, ESP8266

Abstract

Smart home is a term that refers to modern homes where appliances, lighting or other electronic devices can be controlled remotely through a computer network and their functions automated. This thesis focuses on solving the interoperability problem between Smart home products from different vendors. The approach chosen by the author is researching existing commercial and Do it Yourself solutions currenly on the market and than designing and implementing a Smart home platform that is easy to use and can be extended to support various devices. The resulting platform leverages a Wi-Fi connection to control lights and smart sockets and read data from temperature and humidity sensors and door sensors. The user interacts with the platform through a progressive web application running in any modern web browser.

Keywords smart home, intelligent home, Wi-Fi, RaspberryPi, ESP8266

Obsah

Seznam obrázků

Úvod

Tato práce se zabývá problematikou chytré domácnosti (dále jen *Smarthome*). *Smarthome*, oproti běžné domácnosti, nabízí uživateli snadné ovládání elektronických zařízení v domácnosti (např. zapínání/vypínání světel) pomocí mobilní aplikace nebo hlasového asistenta. Dále *Smarthome* umožňuje centrální sbírání dat z různých senzorů v domě a optimalizaci např. vytápění pro nižší spotřebu energie. Další z funkcí, které *Smarthome* podporuje je zobrazení stavu zařízení (lze například zkontrolovat, zda jsou zavřené dveře). *Smarthome* také nabízí automatizaci úkonů (např. zhasnutí všech světel při odchodu z domu) a vzdálený přístup (např. kontrola domácích mazlíčků přes kameru).

Typická zařízení, které lze pomocí *Smarthome* ovládat, jsou světla, zásuvky, žaluzie, zábavní systém (televize a reproduktory), vytápění a klimatizace. *Smarthome* poskytuje pohodlí a bezpečnost pro obyvatele domácnosti. Nevýhodou může být strach ze ztráty soukromí.

Mezi senzory poskytující uživateli data patří teploměr a vlhkoměr, zabezpečovací senzory (např. kouřový senzor, senzor úniku vody, senzor otevření dveří), senzory kvality vzduchu a další.

Následující kapitoly se budou věnovat analýze již existujících řešení a dále návrhem a implementací vlastního řešení, které bude vycházet z poznatků dosavadních řešení a bude je dále rozšiřovat. Nakonec bude výsledné řešení nasazeno a otestováno.

Kapitola 1

Cíle práce

Cílem této práce je návrh a implementace platformy pro chytrou domácnost. Platforma bude založená na spojení Wi-Fi jednotek s Raspberry Pi [1], bude mít intuitivní uživatelské rozhraní a bude umožňovat jednoduchou konfiguraci koncových zařízení. Platforma bude navíc *otevřená*, tedy nabídne API, které bude moci kdokoli využít k vývoji nových koncových zařízení a také k napojení na další platformy (např. Apple HomeKit [2]).

Hlavním přínosem práce bude vytvoření alternativy ke komerčním produktům, kde důležitým aspektem je nižší pořizovací cena tohoto řešení. Takzvaná DIY (Do it Yourself, Udělej si sám) řešení bývají často levnější než komerční produkty, ale vyžadují od uživatele sestavení, zapojení a pokročilou konfiguraci. Tato práce se zaměřuje na konfiguraci zařízení v chytré domácnosti bez nutnosti zasahovat do konfiguračních souborů, bez ruční instalace pluginů atd. Cílovým uživatelem/zákazníkem je zkušený programátor i laik (vyžadována je minimální technická zdatnost).

Vzhledem k rozsahu bakalářské práce není cílem nahradit již zavedené open source projekty jako např. *Home Assistant* [3] nebo *OpenHab* [4], které nabízí mnoho funkcí, ale mohou být pro některé uživatele příliš složité.

Kapitola 2

Existující řešení pro chytré domácnosti

Tato kapitola se zabývá definicí *Smarthome* a průzkumem existujících řešení. V závěru této kapitoly jsou popsány konkrétní poznatky z rešerše, které budou dále využity při návrhu vlastní platformy.

2.1 Definice chytré domácnosti

Chytrá domácnost je definována jako *„Dům s komunikační sítí, která spojuje klíčové elektrické spotřebiče a služby a umožňuje je dálkově ovládat nebo sledovat, vzdáleně zde znamená v bytě i z vnějšku bytu. V souladu s tím domov, který je inteligentní, musí obsahovat tři prvky, které jsou vnitřní síť, inteligentní ovládání a domácí automatizace. Vnitřní síť je základem inteligentního domu a může být drátová nebo bezdrátová. Inteligentní ovládání znamená brány pro správu systému. Domácí automatizace znamená produkty v domácnostech a napojení na služby a systémy mimo domov.“* [5, přeložil autor].

V této práci bude dále *Smarthome*, kromě výše uvedeného, označovat i produkty a služby (tedy hardware a software), které nabízejí alternativní způsoby ovládání elektronických zařízení v domácnosti. Pomocí technologií síťové komunikace umožňuje *Smarthome* značně více flexibility oproti klasickému způsobu ovládání zařízení. Např. světla v domě jsou standardně ovládána pomocí vypínačů, pevně ukotvených ve stěnách, kde každý vypínač rozsvěcí/zhasíná pevně určené světlo/světla. Naproti tomu chytré osvětlení nabízí seskupování světel a libovolné mapování vypínačů na světla / skupiny světel i vzdálené ovládání. Sortiment zařízení pro chytrou domácnost je vidět na obrázku 2.1.

Obrázek 2.1: Soriment IKEA Home Smart

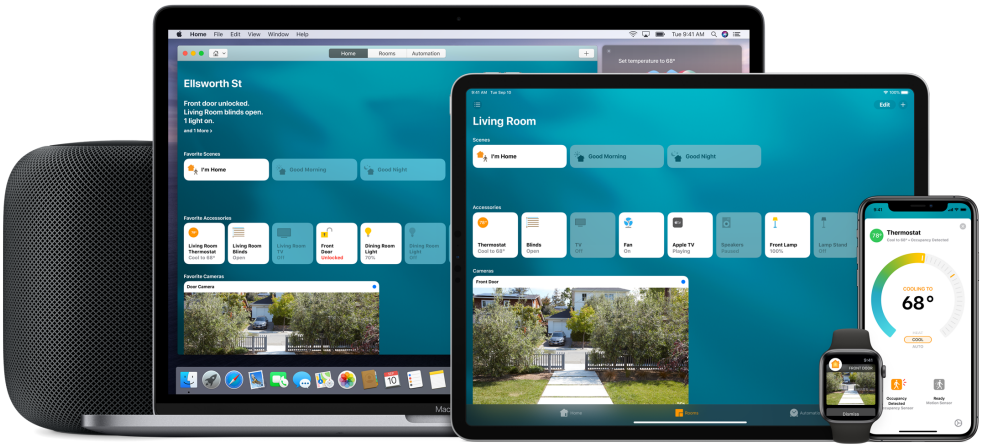
2.1.1 Definice pojmů

V této sekci budou definovány pojmy, které budou využívány v dalších kapitolách. Vysvětlení dalších pojmů z oblasti chytré domácnosti [6, 7].

- **Platforma** – Platformou se rozumí technologické prostředí, jak po stránce hardware, tak i software.
- **Koncové zařízení** – Zařízení, které nabízí nějakou funkcionalitu (např. světlo, zásuvka, senzor) a dokáže komunikovat po síti.
- **Základová stanice (Hub)** – Zařízení připojené k domácí Wi-Fi síti, které zprostředkovává komunikaci mezi jednotlivými koncovými zařízeními.
- **Hlasový asistent** – Uživatelské rozhraní pro komunikaci s počítačem pomocí řeči (např. Amazon Alexa [8], Siri [9]).

2.2 Bezpečnost a soukromí

Existují dva hlavní přístupy ke správě dat a ovládání zařízení. První je cloud, tedy model kde zařízení v domácnosti komunikují prostřednictvím serverů výrobce. Druhý je lokální síť, kde komunikaci zprostředkovává základová stanice připojená do lokální sítě. Výhodou cloudového přístupu je vzdálený přístup prostřednictvím internetu. Nevýhodou je závislost na internetovém připojení a typicky delší doba odezvy příkazů [10].



Obrázek 2.2: Uživatelské rozhraní Apple Home

Lokální přístup nabízí bezpečnost a soukromí, nehrozí únik dat ze serveru poskytovatele služby. Pro vzdálený přístup lze využít např. VPN.

2.3 Kompatibilita

Pokud chce uživatel využívat v jedné domácnosti produkty od více výrobců nastává pro něj několik nepříjemností. Nekompatibilita mezi řešeními různých výrobců se dotýká uživatele mnohdy nutností vlastnit více základových stanic pro různé platformy. Přístup k funkcím je také často roztříštěn do více aplikací.

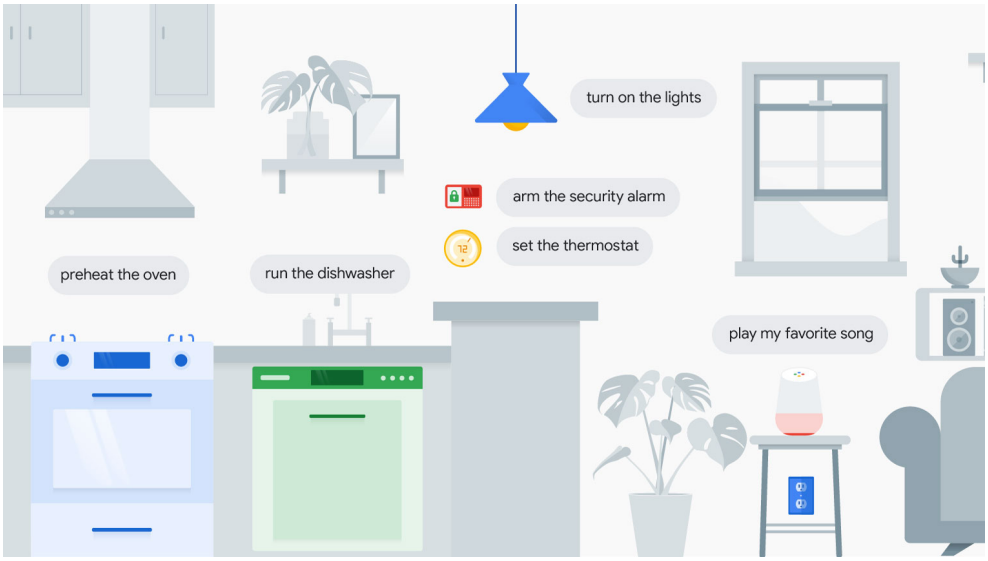
Na trhu existují zařízení, která sdružují běžně nekompatibilní produkty, např. Logitech Harmony Hublogitech-harmony-hub. Nevýhodou tohoto řešení je závislost na serverech výrobce. Ve chvíli, kdy se výrobce zařízení rozhodne již dále nepodporovat, přestane celý systém fungovat [11].

2.4 User experience

Ze studie *User experience in do-it-yourself-style smart homes* [12]: *„Věříme, že přijetí všudypřítomné výpočetní technologie v domácím prostředí může napomoci naplnění uživatelských hodnot a zlepšení kvality života doma.“* [12, přeložil autor] [1]

Dobrý *User experience* začíná balením produktu a končí jednoduchostí každodenního používání, to velmi souvisí s uživatelským rozhraním. Ať už je to rozhraní grafické, viz obrázek 2.2, nebo rozhraní hlasové, viz obrázek 2.3.

[1] Původní text: *„We believe that adopting ubiquitous computing technology in the domestic environment can assist in fulfilling user values and improving the quality of life at home.“*



Obrázek 2.3: Hlasové rozhraní Google Assistant

2.5 Kritéria pro porovnání existujících řešení

Řešení a platformy pro porovnávání byly vybírány na základě popularity a rozšířenosti. Byly zahrnuty produkty velkých hráčů na trhu (vyhodnoceno na základě [13]). Populární open-source platformy vybrány podle popularity na GitHubu, viz tabulka 2.1.

Projekt	Počet hvězd (tisíce)
Home Assistant	33.5
OpenHAB	3.5
Tasmota	11

Tabulka 2.1: Popularita open-source knihoven na GitHubu (květen 2020)

2.6 Existující komerční řešení

Komerční řešení v oblasti chytré domácnosti nabízejí společnosti Google (rodina produktů Nest [14]), Apple (HomeKit [2]), Amazon (Echo [15]), Samsung (SmartThings [16]), IKEA (Home smart [17]) a další. Základním stavebním kamenem většiny řešení je základová stanice, která zprostředkovává komunikaci mezi zařízeními v domácnosti. Komerční řešení často používají různé komunikační protokoly, které mezi sebou nejsou kompatibilní, a základová stanice nabízí také potřebný hardware pro komunikaci s periferiemi (např. Wi-Fi anténa, Zigbee anténa). Tyto platformy nabízejí mnoho možností pro ovládání zařízení. Za prvé jsou to mobilní aplikace, např. Google Home, Apple Home.

Za druhé pak hlasový asistent, buď jako aplikace v mobilním zařízení nebo v podobě chytrého reproduktoru.

Mezi přednosti komerčních řešení patří dostupnost velkého množství koncových zařízení[2], která se dají na tato řešením napojit, a dále jednoduché a přívětivé uživatelské rozhraní aplikací. Zápory mohou být nekompatibilita zařízení od různých výrobců, pořizovací cena či soukromí dat.

Cílovou uživatelskou skupinou těchto produktů a služeb jsou *„The Digital Consumer in every venue and mode they are in“* [3] [19, přeložil autor].

V průběhu psaní této práce (září–květen 2020) se začaly formovat dva standardy v oblasti chytré domácnosti. *OCF Universal Cloud Interface* pod záštitou Open Connectivity Foundation [20] a *Connected Home over IP* s cílem vytvořit nový otevřený standard *Connected Home over IP* [21, 22]. Tyto standardy nebyly, podle mého nejlepšího vědomí v době psaní práce, využity u komerčních produktů či služeb.

2.7 Existující DIY řešení

V kutilské komunitě je oblíbený Home Assistant (Hass.io), platforma pro chytrou domácnost založená na hubu (nejčastěji se používá Raspberry Pi) na kterém běží operační systém HassOS [23] nebo Linux. Hass.io nabízí stovky pluginů pro koncová zařízení v domácnosti (např. světla, různé senzory) a to jak pro generická zařízení, tak i pro konkrétní komerční produkty (např. Philips Hue, propojení s hlasovými asistenty Google Assistant, Amazon Alexa). Jeho uživatelské rozhraní však může být nepřehledné a pro některé uživatele příliš složité.

Další oblíbenou platformou je OpenHab [4]. Ten nabízí velmi podobné spektrum funkcionalit jako Home Assistant. Jako bonus nabízí nativní aplikace pro Android, iOS a Windows.

Jako hardware jsou nejvíce používané jednodeskové počítače Raspberry Pi (popř. Orange Pi, Banana Pi) a vývojové desky s čipem ESP8266 a ESP32.

Také existují open source firmware pro vývojové desky ESP8266 a podobné, např. Tasmota [24]. Tasmota obsahuje integrace na Home Assistant i OpenHab.

Za zmínku stojí také IFTTT (If This Than That) [25], mobilní aplikace umožňující automatizaci úkonů na bázi pravidel a akcí. Pravidla jsou definována uživatelem a specifikují podmínky, při jejichž splnění se spustí definovaná akce, např. *„Když přijdu domů, rozsviť všechna světla v domě“*. Bohužel tato aplikace je závislá na připojení k internetu a služba má vysokou latenci, tedy čas od splnění podmínky nebo zadání příkazu do aktivace požadované akce (osobní zkušenost autora).

Cílovou uživatelskou skupinou těchto řešení jsou domácí kutilové.

[2] Např. vyhledávání na alza.cz [18] zobrazí 208 HomeKit kompatibilních zařízení

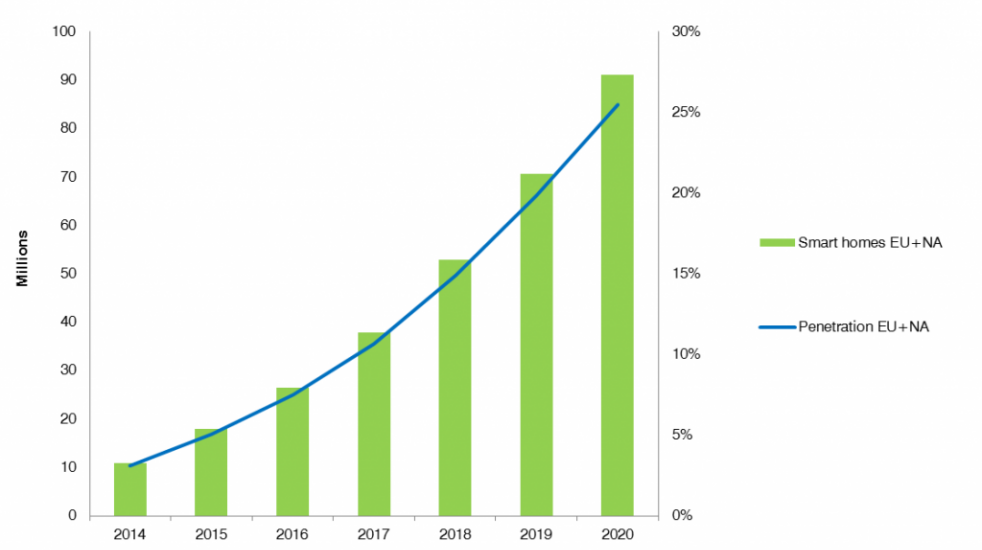
[3] The Digital Consumer in every venue and mode they are in

2.8 Rozdíly mezi komerčními a DIY řešeními

Komerční řešení nabízí uživatelskou přívětivost, nastavení zařízení je snadné a ovládání bezproblémové. V každém daném ekosystému produktů (např. HomeKit) je zaručena kompatibilita, to je velké plus komerčních řešení. Na druhou stranu, výhodou DIY řešení je, že dokáží pracovat s více protokoly a mohou sdružit zařízení několika výrobců, spolu nekompatibilní, do jedné platformy. Nevýhodou DIY řešení je jejich vlastní podstata, tedy že si je uživatel musí *„udělat sám“*. Od uživatele se vyžaduje technická zdatnost při konfiguraci i při běžném používání, např. schopnost odladit drobný technický problém.

Na rozdíl od předchozích domácích automatizačních produktů, umožňují inteligentní DIY výrobky uživatelům proaktivně řešit své složité problémy a plnit tak své potřeby všudypřítomnými technologiemi [12, přeložil autor].

2.9 Cílová skupina



Obrázek 2.4: Vývoj adopce *Smarthome* technologií v Evropě a Severní Americe 2014–2020 [26]

Chytré domácnosti se začaly objevovat na přelomu tisíciletí [27]. V době psaní této práce (září–květen 2020) popularita chytrých domácností roste [26] (viz obrázek 2.4), ale stále si je pořizují spíše nadšenci do technologií. Není však vyloučeno, že s nimi domácnost sdílí i méně technicky založení lidé, a i ti by měli mít možnost s domácností interagovat. Autor se v této práci zaměří na návrh platformy a jejího uživatelského rozhraní, které bude primárně určeno pro DIY nadšence, ale bude intuitivní i pro netechnicky založené uživatele. Cílovou skupinou tak budou kutilové, kteří svou chytrou domácnost chtějí sdílet

s méně technicky založenými členy rodiny. Na konci práce bude provedeno uživatelské testování pro ověření splnění tohoto cíle.

2.10 Získané poznatky z této kapitoly

Poznatky z předchozích odstavců můžeme shrnout do těchto bodů:

- V oblasti chytrých domácností chybí hojně využívaný standard pro komunikaci mezi zařízeními.
- Na trhu je prostor pro otevřené, modulární řešení.

Kapitola 3

Analýza a Návrh

Tato kapitola se zabývá analýzou problémů, řešených v oblasti chytré domácnosti a návrhem architektury řešení, tedy rozdělením řešení na základní bloky, volbou implementačního jazyka, způsobem ukládání dat, výběrem frameworků a knihoven, návrhem komunikačních protokolů a návrhem uživatelského rozhraní.

Budou využity postupy softwarového inženýrství vyučované na FIT ČVUT v předmětu Softwarové inženýrství 1 (BI-SI1). K popisu postupů budou využity diagramy UML (Unified Modeling Language). Mezi tyto postupy patří:

1. Analýza procesů
2. Analýza požadavků
3. Popis problémové domény
4. Analýza případů užití

3.1 Analýza procesů

Mezi základní problémy, které *Smarthome* řeší, patří např. ovládání osvětlení, automatická správa vytápění, zabezpečení, automatizace úkonů. Toto jsou základní stavební kameny, které budou detailně popsané v této sekci.

3.1.1 Osvětlení

Světlo může být v jednom ze dvou stavů – rozsvíceno nebo zhasnuto. Uživatel oznámí pomocí mobilní aplikace, hlasu nebo programovatelného vypínače, že chce rozsvítit a systém se postará o to, aby zapnul požadovaná světla. Akce mohou být kontextové, tedy když se uživatel nachází v kuchyni a řekne *„rozsviť“*, systém rozsvítí světla v kuchyni.

3.1.2 Teplota a vytápění

Chytré vytápění znamená, že se systém učí, kdy je uživatel doma a jakou má rád teplotu. Po nějaké době systém začne vypínat topení, když nikdo není doma, kvůli úspoře energie a nákladů. A zatopí včas před tím něž se uživatel vrací. Teplotu lze kdykoli nastavit na požadovanou hodnotu pomocí aplikace nebo hlasu. Například: *„Nastav teplotu na dvacet dva stupňů“*.

3.1.3 Zabezpečení

Mezi zabezpečovací zařízení se řadí např. chytrý zámek, senzory neoprávněného vniknutí, senzory požáru/vytopení. Chytrý zámek umožňuje odemykání dveří pomocí mobilního zařízení (chytrý telefon, chytré hodinky). Lze spravovat osoby, které mají přístup do domu. Lze také udělit přístup např. kamarádovi pouze na omezenou dobu. Některé chytré zámky umožňují také odemykání pomocí číselného kódu nebo otisku prstu. Hlavní výhoda je eliminace fyzických klíčů a správa přístupu.

3.1.4 Automatizace

Chytrá domácnost může poznat, když se uživatel vrátí domů z práce, že venku už je tma a je potřeba rozsvítit. Uživatel si může nakonfigurovat tzv. scénáře, kde pomocí hodnot ze senzorů je možno podmiňovat chování systému. Například pokud prší, automaticky se zavřou střešní okna. Dalším příkladem může být zhasnutí všech světel a vypnutí určitých spotřebičů večer, potom co jde poslední člen domácnosti spát.

3.2 Analýza požadavků

Tato sekce je rozdělena na analýzu požadavků funkčních (konkrétní funkcionality, které musí výsledné řešení obsahovat) a nefunkčních (další vlastnosti řešení). Tyto požadavky vznikly na základě zadání práce.

3.2.1 Funkční požadavky

F1 - Sledování teploty, grafické zobrazení průběhu v čase
Sledování aktuální teploty, zobrazení historie za poslední den, týden, měsíc atd. Teplota bude znázorněna v podobě grafu vývoje teploty za určitý čas.

F2 - Detekce otevření dveří
Detekce otevření dveří jako způsob zabezpečení proti neoprávněnému vniknutí. Možno aplikovat i na okna/vrata a jiné vstupy do domu.

F3 - Ovládání chytrých zásuvek
Umožňuje rozšíření chytrých funkcionalit na zařízení, která nejsou integrovaná do *Smarthome*. Vypnout/zapnout.

F4 - Ovládání světel
Rozsvěcení/zhasínání vestavěných světel, stojících lamp a chytrých žárovek.

F5 - Konfigurace zařízení s pomocí webové aplikace
Jednotlivá zařízení bude možné pojmenovat, seskupit dle jednotlivých pokojů/místností v domě a dále ovládat celé skupiny najednou.

3.2.2 Nefunkční požadavky

N1 - Základem celého zařízení bude webová aplikace běžící na platformě Raspberry Pi
Webová aplikace bude optimalizovaná pro přístup z mobilních zařízení (telefon, tablet) i desktopu. Uživatelské rozhraní bude responzivní.

N2 - Pro jednotlivá zařízení bude použit modul s čipem ESP8266
Modul s mikrokontrolérem ESP8266 od firmy Espressif se bude připojovat k Wi-Fi síti a bude obsahovat firmware v prostředí Arduino.

N3 - Zařízení spolu budou komunikovat prostřednictvím Wi-Fi sítě
Využita bude domácí Wi-Fi síť. Předpokladem je router, ke kterému lze připojit desítky zařízení.

N4 - Systém bude efektivně pracovat s větším množstvím modulů
Systém bude stabilní a schopný sbírat data z desítek senzorů a ovládat desítky zařízení. Při tomto zatížení nebude docházet k pádům systému ani k výraznému zpoždění komunikace.

N5 - Jednoduchý protokol pro komunikaci s možností snadného rozšíření
Protokol pro interní komunikaci bude umožňovat v budoucnu přidávat nové typy periferních zařízení.

3.3 Popis problémové domény

Následuje popis cílové domény (tj. *Smarthome*). Jednotlivé entity a vztahy mezi nimi jsou popsané v doménovém diagramu na obrázku 3.1.

3.3.1 Doménový model

Tento model popisuje objekty reálného světa, které budou v systému evidovány a se kterými bude systém interagovat (viz obrázek 3.1). Pro popis vlastností fyzických zařízení bude využit koncept tzv. charakteristiky. Charakteristika popisuje jednu danou vlastnost zařízení (např. jednou z charakteristik světla je, zda je zapnuto, či vypnuto). Následuje popis jednotlivých entit.

- **Room (Pokoj)** – Pokoj sjednocuje zařízení do logických celků podle polohy.
- **Attachment (Doplňek)** – Např. světlo, zásuvka, teplotní senzor a další.
- **Device (Zařízení)** – Takové zařízení připojené k systému, které nabízí napájení a bezdrátovou komunikaci pro Doplňky.
- **Characteristic (Parametr)** – Určitá vlastnost Doplňku (např. pro světlo je to rozsvíceno/zhasnuto, pro teploměr je to hodnota teploty).
- **Event (Událost)** – Událost vyslaná z některého zařízení (pro senzor otevření dveří je to např. „Dveře otevřeny“ a „Dveře zavřeny“.

3.3.2 Typy zařízení

K jednomu zařízení může být připojeno více Doplňků. Každé zařízení dokáže obsluhovat libovolný typ Doplňku. Doplňky se dělí na následující typy. Typy mají své charakteristiky, dané dostupnými funkcionalitami.

Světlo (Light)

Klasické vestavěné domácí osvětlení, stojací a stolní lampy nebo LED žárovky.

Charakteristiky:

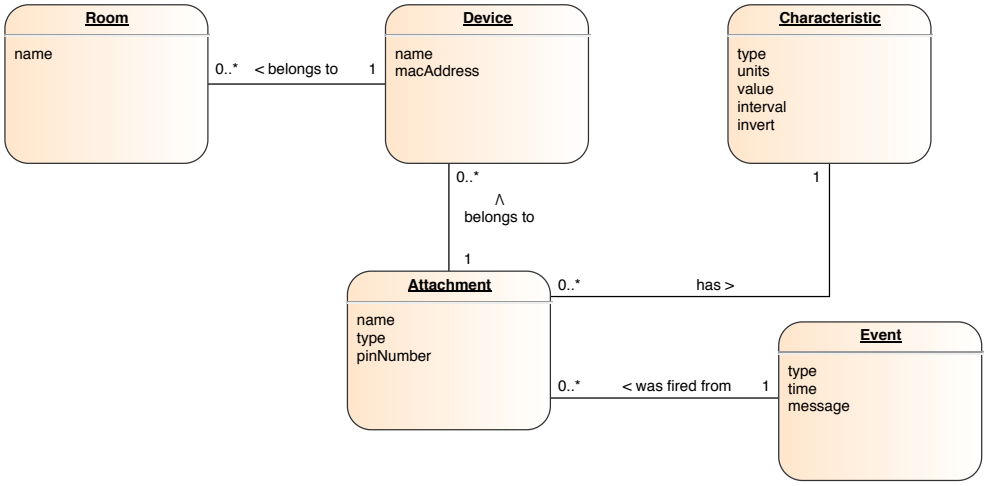
- **isOn (je zapnuto)** Stav světla. Rozsvíceno/Zhasnuto.

Zásuvka (Socket)

Chytrá zásuvka. Zapojí se do klasické zásuvky a do ní se zapojí zařízení. Poté je možné ovládat napájení daného zařízení z aplikace.

Charakteristiky:

- **isOn (je zapnuto)** Stav zásuvky. Zapnutá/Vypnutá.



Obrázek 3.1: Doménový model popisující entity, jejich atributy a vztahy mezi nimi

Senzor otevření dveří (DoorSensor)

Senzor se nachází na zárubních dveří. Koncový spínač pozná stav dveří. Odesílá zprávu při změně stavu. Slouží jako zabezpečovací zařízení pro ochranu proti neoprávněnému vstupu.

Charakteristiky:

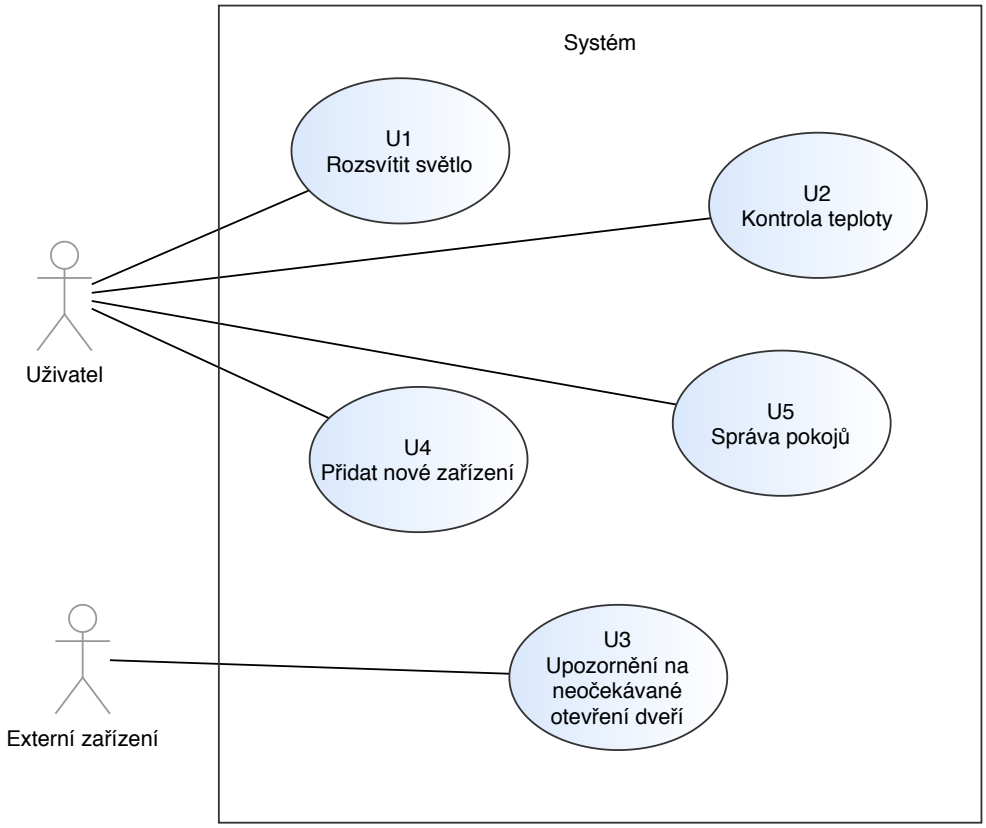
- **isOpen (je otevřeno)** Stav dveří. Otevřené/Zavřené.

Senzor teploty a vlhkosti (TemperatureSensor)

Odesílá hodnoty teploty a vlhkosti vzduchu v určeném (konfigurovatelném) časovém intervalu.

Charakteristiky:

- **temperature (teplota)** Teplota ve stupních Celsia.
- **humidity (vlhkost vzduchu)** Vlhkost vzduchu v procentech.



Obrázek 3.2: Diagram případů užití v notaci UML

3.4 Analýza případů užití

Analýza případů užití se zabývá identifikací a popisem dílčích procesů v systému. Graficky jsou případy užití reprezentovány na obrázku 3.2. Do případů užití vstupují následující aktéři:

- Uživatel
- Externí zařízení

U1 - Ovládání světel
Rozsvěcet a zhasínat světla. Je také možné rozsvítit/zhasnou všechna světla v jednotlivých pokojích nebo v celém domě.

U2 - Ovládání chytrých zásuvek
Zapínat a vypínat zařízení připojené do chytré zásuvky.

U3 - Kontrola teploty
Zobrazení aktuální teploty v domácnosti a také vývoj teploty v čase.

U4 - Kontrola stavu otevření dveří (či oken)
Uživatel zkontroluje v aplikaci stav dveří.

U5 - Přidat nové zařízení
Umožňuje přidat a nakonfigurovat nové zařízení (např. světlo), které bude dále ovládáno prostřednictvím systému (pomocí aplikace).

Hlavní scénář:

1. Uživatel zvolí akci Přidat nové zařízení
2. Uživatel vybere požadované zařízení ze seznamu dostupných (nenakonfigurovaných) zařízení připojených k domácí Wi-Fi
3. Uživatel pojmenuje zařízení
4. Uživatel zvolí ze seznamu pokoj, ve kterém se zařízení nachází. V případě, že daný pokoj není v databázi, vytvoří se nový pokoj s požadovaným názvem.
5. Uživatel vybere typ zařízení ze seznamu (např. světlo, teploměr)
6. Uživatel potvrdí nastavení

U6 - Správa pokojů
Umožňuje vytvářet nové pokoje v domácnosti, přidávat (a odebírat) zařízení k daným pokojům.

U7 - Konfigurace Wi-Fi zařízení
Připojení nového zařízení k domácí Wi-Fi síti.

Hlavní scénář:

1. Zapojení zařízení do napájení
2. Připojení chytrého telefonu k Wi-Fi síti vytvořené zařízením (název ESP-xxxxxx).
3. Klepnutí na notifikaci zobrazenou po připojení
4. Výběr požadované sítě ze seznamu
5. Zadání hesla
6. Potvrzení

3.5 Pokrytí požadavků

Tabulka 3.1 ilustruje pokrytí požadavků případy užití.

	F1	F2	F3	F4	F5
U1				x	
U2			x		x
U3	x				
U4		x			
U5					x
U6					x
U7					x

Tabulka 3.1: Tabulka zobrazující pokrytí požadavků případy užití

3.6 Návrh architektury

Celé řešení bude postaveno na modulárním přístupu. Zaměřeno bude na otevřenost, rozšiřitelnost a uživatelskou konfigurovatelnost. Jádrem systému bude tzv. *Základová stanice* (Hub), v tomto případě Raspberry Pi. Tento Hub bude prostředníkem mezi koncovými zařízeními (periferiemi) v domácnosti a uživatelovými zařízeními (mobilní telefon, tablet, PC). Základní stavební jednotkou bude *Zařízení* (Device). Zařízení zprostředkovává bezdrátovou komunikaci a napájení pro *Doplňky* (Attachment). Doplněk je zařízení, které bude nadále ovládáno systémem. Příkladem Doplňku je např. světlo, teploměr nebo jakýkoli senzor. Zařízení a Doplňky bude možné libovolně kombinovat. K jednomu zařízení bude možné připojit více Doplňků. Rozdělení zařízení a jejich komunikace je znázorněna na obrázku 3.3.

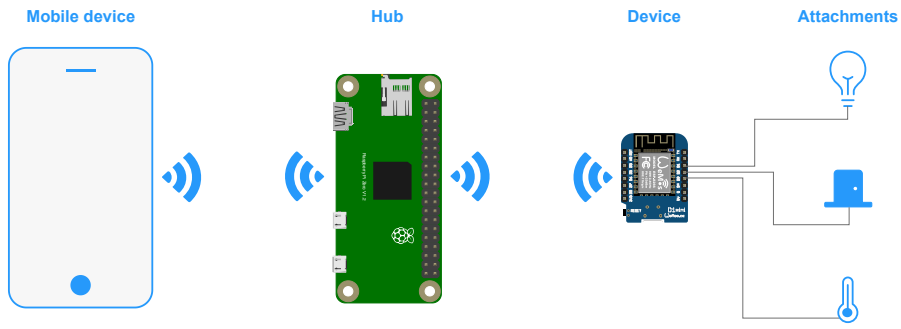
3.7 Modularita

K jednomu zařízení bude možno připojit jeden až čtyři Doplňky. Zařízení a Doplňky se budou propojovat pomocí jednotného rozhraní (portu) a to jak po stránce fyzické, tak z pohledu komunikačního protokolu. Využita bude deska Wemos D1 mini s mikrokontrolérem ESP8266 a tzv. shieldy. Shieldy pro Wemos D1 mini mají osazeno stejné množství pinů jako samotný Wemos. Díky tomu je možné shieldy skládat na sebe.

3.8 Rozdělení řešení na komponenty

Softwarové řešení se bude skládat ze tří komponent, webového serveru, webového frontendu a Arduino firmwaru (viz obrázek 3.4).

Webový server bude implementovat REST [28] API a poběží na Raspberry Pi pod operačním systémem Raspbian (Linux). Základna Raspberry Pi bude sloužit jako komunikační most mezi jednotlivými zařízeními a senzory a odpadá tak navázání na cloud, což je z pohledu bezpečnosti a soukromí značné



Obrázek 3.3: Návrh architektury

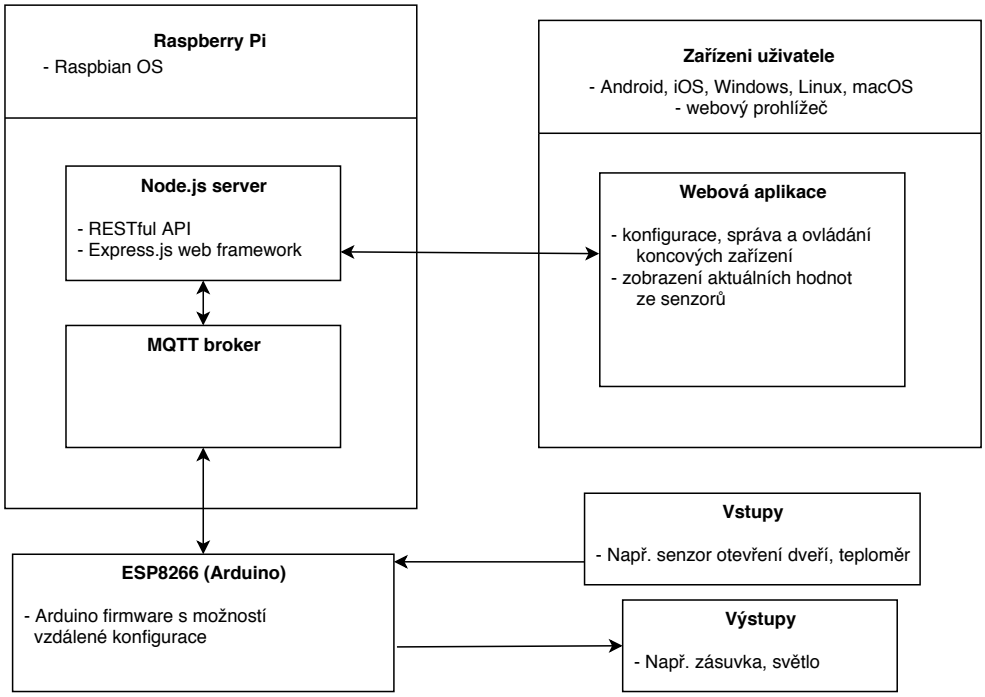
plus. Lze použít Raspberry Pi model A i B, verze 2, 3 i 4 a také miniaturní Raspberry Pi Zero.

Webový frontend bude progresivní webová aplikace (PWA) běžící ve webovém prohlížeči. Benefitem tohoto řešení je hlavně nezávislost na platformě a operačním systému zařízení uživatele či uživatelů v domácnosti. Dále pak odpadá nutnost instalace aplikace, ale aplikaci lze přidat na domovskou obrazovku operačního systému (Android, iOS) díky technologii PWA. Na desktopových operačních systémech (Windows 10, macOS, Linux) je možné PWA spouštět v samostatném okně a vytvořit zástupce na ploše, tedy rozdíl mezi klasickou aplikací a webovou pro uživatele zmizí.

Arduino firmware poběží na mikrokontroléru ESP8266. Prostředí Arduino nabízí velké množství knihoven díky celosvětové komunitě vývojářů. Bude tedy snadné implementovat libovolné komunikační rozhraní. Tento mikrokontrolér byl použit, protože se vyznačuje velmi nízkou cenou [29].

3.9 Volba implementačního jazyka

Aplikace je rozdělena na frontend a backend. Pro vývoj interaktivních aplikaci ve webovém prohlížeči budou využity technologie HTML, CSS a JavaScript. JavaScript má rozsáhlý ekosystém knihoven a frameworků a také velkou komunitu vývojářů. JavaScript je čím dál více populární také na backendu, kde umí běžet na různých platformách pomocí Node.js. Pro vývoj firmwaru bude využito vývojové prostředí Arduino IDE, tedy jazyk C/C++.



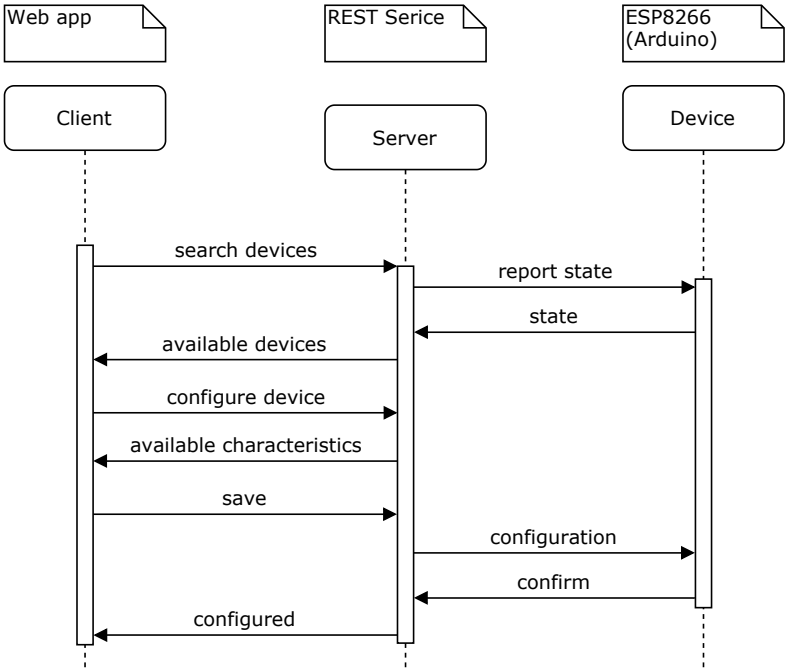
Obrázek 3.4: Návrh architektury detail

3.10 Komunikace mezi komponentami

Komunikace bude probíhat pouze v lokální síti. Systém tedy bude fungovat i v případě výpadku připojení k internetu. Komunikace mezi frontendem a backendem bude probíhat prostřednictvím protokolu HTTP/HTTPS. Pro komunikaci mezi backendem a periferními zařízeními bude využit protokol MQTT (Message Queuing Telemetry Transport) [30]. Ten byl zvolen, protože se vyznačuje malými nároky na výkon a paměť hardwaru a vystačí si i s malou šířkou pásma sítě.

3.11 Vyhledávání zařízení

Při připojení zařízení k napájení, každé zařízení oznámí svou existenci základně MQTT zprávou. MQTT využívá vzor `publisher-subscriber` [31], zpráva ze zařízení tedy jde přes MQTT Brokera, který předává zprávy serveru běžícímu na základně. Broker si zároveň pamatuje poslední přijatou zprávu o stavu každého zařízení a díky tomuto mechanismu jsou poté v uživatelském rozhraní zařízení vidět jako `online` či `offline`. Inicializace komunikace a připojení zařízení k systému je znázorněno na diagramu 3.5.



Obrázek 3.5: Konfigurace nového zařízení

3.12 Způsob ukládání dat

Pro ukládání dat bude využita databáze, ty se dělí na relační a tzv. NoSQL, neboli nerelační. Relační databáze jsou velmi rozšířené, jsou založené na vztahy (relacemi) mezi daty a využívají pro dotazování jazyk SQL. Naopak NoSQL databáze nevyužívají relace, ale používají koncept dokumentu jako úložiště dat. Nevýhodou relačních databází je nutnost mapovat softwarové třídy na relace. Naopak NoSQL databáze dokáží ukládat data ve spoustě různých formátů. Zde byla zvolena NoSQL databáze MongoDB, protože ukládá data ve formátu JSON (JavaScript Object Notation), který je nativní pro JavaScript, odpadá tedy mapování objektů.

3.13 Frameworky a knihovny

Frontendových JavaScriptových frameworků existuje nepřeberné množství. Mezi nejužívanější a neoblíbenější patří Angular [32], React [33] a Vue.js [34]. Využívají návrhový vzor Model-View-ViewModel (MVVM)) [35], který rozděluje aplikaci na části. Model implementuje doménovou logiku aplikace. View má na starosti rozložení a vzhled uživatelského rozhraní. ViewModel zprostředkovává komunikaci mezi View a Modelem.

Frontend bude využívat framework Vue.js, jednoduchý open-source Framework, který však v základu obsahuje funkcionality, které se v jiných frameworcích doplňují dalšími knihovnami. Jsou to např. vue-router (směrování požadavků na frontendu), vuex (state management). Ze všech frameworků má Vue nepřívětivější syntaxi a obsahuje všechny základní funkcionality, které jsou potřeba, proto bude aplikace vyvíjena ve Vue.

Pro backend bude využit Node.js [36] (dále jen Node), což je JavaScriptové běhové prostředí postavené na JavaScriptovém enginu V8. Node.js je tzv. *event-driven* (řízený událostmi)a je navržen pro vysoce škálovatelné síťové aplikace [37]. Pro potřeby chytré domácnosti, které bude zpracovávat zprávy z desítek modulů je Node ideální.

3.14 Bezpečnost

V této sekci budou shrnuty postupy a metody, které budou implementovány za účelem zabezpečení celé platformy. Následující text nepředstavuje detailní úvod do problematiky zabezpečení webových aplikací, tímto se zabývá např. D. Pokorný v práci *Zabezpečení webové aplikace* [38].

Jakýkoli software pracující s osobními údaji uživatelů je dle zákona potřeba zabezpečit. A to proti několika druhům útoků, např. Cross Site Scripting (XSS), SQL injection. Útočníci se často snaží získat osobní informace uživatelů a dále je prodat. Zabezpečení proti úniku informací jsou hlavně silná hesla, unikátní pro každý účet a pro každou aplikaci, a šifrování komunikace.

Platforma pro chytrou domácnost *Smarthome*, navržená a implementovaná v této práci, má nepřímý přístup k osobním informacím uživatelů. Ti sice přímo nevyplňují informace o sobě, ale pomocí senzorů chytré domácnosti lze vysledovat návyky uživatele, které by mohly být dále zneužity.

Jako první vlna ochrany bude využita domácí Wi-Fi síť. Ta musí být zabezpečená protokolem WPA2 nebo lépe WPA3 s využitím AES (nikoli TKIP). Dále bude vytvořena druhá Wi-Fi síť pro hosty, která bude izolovaná od primární sítě. Administrace routeru bude zabezpečena silným heslem.

Uživatelské účty na Raspberry Pi budou zabezpečeny silným heslem. Serverová aplikace bude přístupná na portu 3000, na který bude přesměrována komunikace z portu 80. Přístup k aplikaci bude probíhat prostřednictvím protokolu HTTPS. Serverová aplikace obsahuje ochranu pro XSS (Cross Site Scripting) útokům, využity jsou knihovny express-sanitizer a helmet. Využito bylo doporučení z [39]

Ochrana proti útoku SQL injection [40] (tedy zde NoSQL injection, protože je využita databáze MongoDB) je zabezpečena pomocí knihovny express-mongo-sanitize. Dále je v databázi zakázáno spouštění JavaScriptu. Přístup k databázi je zabezpečen heslem.

Všechny uživatelské vstupy jsou validovány (knihovna express-validator) a *sanitizovány*, tedy jsou odstraněny jakékoli části řetězců, které obsahují klí-

čová slova jazyků HTML nebo JavaScript. Co se týče klientské aplikace, zde je využito ochrany ve frameworku Vue.js, v každém textu, který se zobrazuje na obrazovku, jsou escapovány klíčové znaky jazyka HTML (tedy <>).

Platforma Arduino je zabezpečena heslem pro OTA (Over the Air update). Pro další úroveň zabezpečení je potřeba zamezit fyzickému přístupu k zařízení, v tu chvíli by potenciální útočník mohl upravit firmware.

Zařízení samotná se nepřipojují do internetu, tedy zde nehrozí útoky na komunikaci v internetu.

Pro zařízení uživatele se doporučuje zabezpečit telefony, tablety a počítače heslem, nebo biometrickou autentizací.

Dále je na serveru implementována tzv. Content Security Policy, což je technologie implementovaná prohlížeči a zabezpečuje nahrávání pouze povolených typů zdrojů a pouze z povolených url adres. Implementováno na základě doporučení [41]. Povoleny jsou pouze skripty, obsah, obrázky a styly, a to pouze ze stejného zdroje.

3.14.1 Zvýšení bezpečnosti

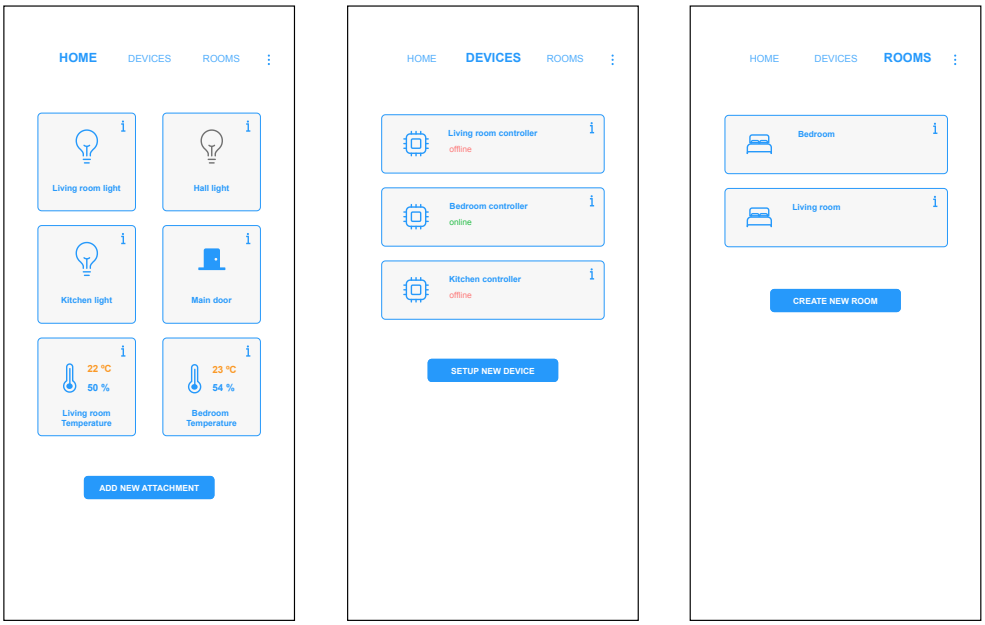
Budoucí možná rozšíření zabezpečení jsou např.

- Ochrana přístupu do aplikace heslem
- Ochrana proti tzv. brute force útokům[4] [42]
- Implementace Cross Origin Resource Sharing (CORS)
- HTTP Strict Transport Security (HTST) [43]
- Referrer-Policy [44]
- Ochrana proti DNS rebinding [45]
- Cross Site Request Forgery (CSRF) [46]
- Cross-Site Script Inclusion (XSSI)
- Zabezpečení MQTT komunikace pomocí HTTPS

3.15 Návrh průchodu aplikací a uživatelského rozhraní

Uživatel bude mít přístup k systému prostřednictvím webové aplikace. Aplikace bude dostupná z libovolného zařízení na definované URL adrese v lokální

[4]Brute force útok (útok hrubou silou) je kybernetickým ekvivalentem zkoušení všech klíčů na klíčence a nakonec nalezení toho správného.



Obrázek 3.6: Návrh obrazovek

síti. Při prvním vstupu aplikace přivítá uživatele a nabídne krátké vysvětlení funkci aplikace, tzv. Onboarding.

Na domovské obrazovce, viz obrázek 3.6 (první zleva), budou zobrazeny *Doplňky*, které si uživatel přidal. Ve výchozím stavu zobrazují Doplňky svůj stav. Ty Doplňky, které umožňují ovládat nějaké zařízení (např. světla) umožní jedním kliknutím (dotykem) přepínat mezi stavy zapnuto/vypnuto. Ty Doplňky, které produkují data, zobrazí na domovské obrazovce aktuální stav a po rozkliknutí nabídnou detailní přehled, včetně historie. Dále domovská obrazovka umožní přidat nové Doplňky k již nakonfigurovaným zařízením.

V horní části obrazovky se bude nacházet navigační lišta, která umožní navigaci mezi domovskou obrazovkou, seznamem nakonfigurovaných zařízení a seznamem pokojů.

Obrazovka se seznamem zařízení, viz obrázek 3.6 (druhá zleva), zobrazí název zařízení a jeho stav (online/offline). Umožní také vyhledat nová nenakonfigurovaná zařízení a přidat je do systému.

Seznam pokojů, viz obrázek 3.6 (třetí zleva), zobrazí název pokoje a umožní vytvářet a odstraňovat pokoje.

Kompletní návrh všech obrazovek naleznete v příloze B.

Aplikace je optimalizována pro použití na mobilních telefonech i desktopech. Pro zařízením s malou obrazovkou nebo nízkým rozlišením je optimalizována velikost písma. Pro zařízení s dotykovou obrazovkou je optimalizována velikost akčních ploch tlačítek a odkazů.

Kapitola 4

Implementace

Tato kapitola obsahuje dokumentaci implementace celého řešení. Implementovány byly funkcionality specifikované v požadavcích. Implementován byl webový server v technologii Node.js, dále webová aplikace s využitím frameworku Vue.js a firmware v prostření Arduino.

4.1 Server

Webový server implementuje standardní RESTful API pomocí HTTP metod. Využito je klasické schéma Create, Read, Update, Delete (CRUD) [47], kdy server poskytuje metody pro vytváření, čtení, modifikaci a odstranění tzv. zdrojů. Komunikace probíhá prostřednictvím síťových protokolů TCP/IP a protokolu HTTP. Data jsou ve formátu JSON (JavaScript Object Notation). API je dostupné na url `http://smarthome.local/api/vX/`, kde `smarthome.local` je název domény a `vX` je číslo verze API (např. `/api/v1/`). RESP API je dokumentováno pomocí nástroje Swagger [48] (dokumentační komentáře ve formátu JSDoc [49]), který nabízí také grafické zobrazení. To je dostupné na url `http://smarthome.local/api/vX/docs`.

4.1.1 Souborová struktura

- `src`
  - `api/v1` ..... zdrojové kódy implementace API
    - `routes` ..... zdrojové kódy pro jednotlivé endpointy
    - `model` ..... definice databázových schémat
    - `declarations` ..... deklarace typů
    - `docs` ..... dokumentace v nástroji Swagger
  - `mqtt` ..... zdrojové kódy pro komunikaci přes MQTT

4.1.2 Popis aplikačního rozhraní

- **POST /devices** – vytvoří nové zařízení
- **GET /devices** – vrátí seznam všech zařízení
- **GET /devices/deviceId** – vrátí seznam všech zařízení
- **UPDATE /devices/deviceId** – aktualizuje zařízení
- **DELETE /devices/deviceId** – odstraní zařízení

Obdobně také pro Doplňky (`/attachments`) a pokoje (`/rooms`). Více v dokumentaci zdrojového kódu (viz přiložené paměťové médium).

Dále jsou dostupné akce:

- **POST /attachments/:attachmentId/toggle** – Přepne stav Doplňku s ID `attachmentId`, pokud má Doplněk pouze dva stavy. Přepne do opačného stavu (zapnuto → vypnuto).
- **POST /rooms/:roomId/toggleAllLights** – Akce `toggle` pro všechna světla v daném pokoji s ID `roomId`.
- **GET /attachments/:attachmentId/getTemperatureData** – Vrátí data o teplotě za posledních 24 hodin ze senzoru s ID `attachmentId`.

Server odpovídá na požadavky pomocí stavových kódů HTTP:

- 200 – Úspěch
- 201 – Zdroj vytvořen
- 400 – Chybný požadavek
- 404 – Zdroj nenalezen

4.2 Databáze

Pro ukládání dat je využita NoSQL databáze MongoDB [50]. Ta pracuje s dokumenty a kolekcemi. Pro komunikaci s databází bude využita knihovna Mongoose [51].

„Mongoose nabízí přímočaré řešení pro modelování aplikačních dat, založené na schématech. Zahrnuje vestavěnou konverzi typů, validaci a vytváření dotazů.“ [51, přeložil autor] Proto byla tato knihovna společně s databázi MongoDB zvolena.

Mongoose modeluje data jako tzv. Schémata. Schéma je ekvivalentem entity v relačním modelu. Data budou v tomto řešení reprezentována jako následující schémata:

- Device
- Attachment
- Room
- Event

4.3 Klientská aplikace

Klientská aplikace je implementována jako progresivní webová aplikace (PWA), tedy bude přístupná z chytrých telefonů, tabletů i desktopů. Jediným požadavkem bude moderní webový prohlížeč se zapnutým JavaScriptem (Google Chrome [52] od verze 80 včetně mobilních verzí, nový Microsoft Edge [53] založený na Chromium [54], Safari [55] od verze 13.1). Technologie PWA nabízí pro webové aplikace porovnatelný uživatelský prožitek jako nativní aplikace. Odpadá tedy nutnost vyvíjet aplikaci pro každou platformu a tím se značně urychlí vývoj.

Aplikace je vyvinuta pomocí webových technologií HTML, CSS a JavaScriptu. Dále je použit JavaScriptový framework Vue.js.

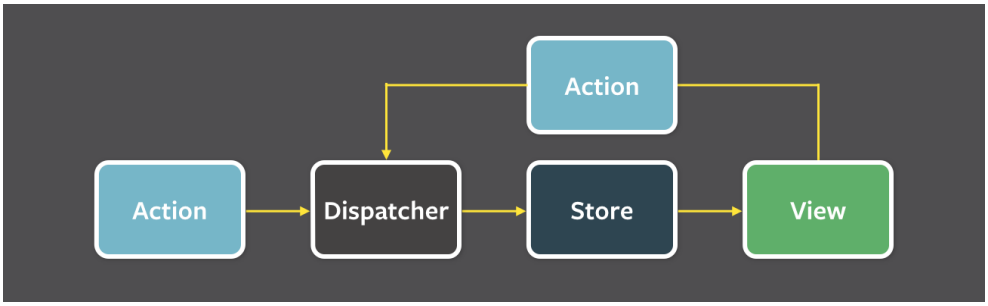
4.3.1 Souborová struktura Vue.js

- `src`
  - `components` ..... šablony komponent
  - `router` ..... definice směrování
  - `store` ..... deklarace Vuex modulů
  - `views` ..... šablony
  - `main.js` ..... vstupní soubor aplikace, inicializace frameworku
  - `App.vue` ..... výchozí komponenta
- `public` ..... statické soubory (obrázky, styly)

4.3.2 State management

Součástí Vue.js je i Vuex, knihovna pro správu stavu aplikace. Vuex je založeno na konceptu Flux [56] (model pro správu toku dat v aplikaci). Flux funguje na principu jednosměrného toku dat (viz obrázek 4.1). Store ukládá data, která mohou být měněna pouze pomocí akcí. Akce zachycují různé způsoby, kterými může cokoliv interagovat s aplikací. View zobrazuje data ze Storu uživateli a aktualizuje je když se změní.

Správa dat v aplikaci je rozdělena do čtyř modulů: `app, devices, rooms, attachments`.



Obrázek 4.1: Flux – tok dat [56]

4.3.3 Komunikace s API

Samotné HTTP požadavky má na starosti knihovna axios. Ta nabízí Promise-based rozhraní nad XMLHTTP požadavky. Komunikace s API je prováděna skrze Vuex metody. Když Store přijme akci, automaticky odešle korespondující request na API. Stav klientské aplikace je tak vždy synchronizován s daty v databázi na serveru.

4.3.4 Routing

Aplikace je tzv. Single Page App (SPA), tedy je třeba řešit směrování požadavků v prohlížeči. Využita je knihovna *vue-router*, která je součástí Vue.js. Aplikace je rozdělena na tři základní cesty.

- **Home** – zobrazuje seznam Doplňků a umožňuje jedním kliknutím/tapnutím přepínat např. světla, úprava a mazání Doplňků
- **Rooms** – seznam Pokojů, přidávání, úprava a mazání
- **Devices** – seznam Zařízení, konfigurace nových zařízení

4.3.5 Komponenty

Komponenty jsou základním stavebním blokem Vue.js aplikací. Vue komponenta obsahuje HTML šablonu, styly a Javascript pro jeden funkční celek. Každá komponenta by měla mít právě jednu funkci (podobně jako single responsibility principle z OOP). Komponenty lze dělit na tzv. Container a Presentational. Container Komponenty jsou napojeny na Store, starají se o odesílání akcí a předávají data Presentational komponentám. Presentational jsou ty komponenty, které se starají pouze o zobrazení dat a styly. Příkladem Container komponenty je *AttachmentList*, tedy seznam Attachmentů zobrazující data ze storů. Příkladem Presentational komponenty je *AttachmentItem*, tedy položka seznamu, pouze zobrazuje data získaná od komponenty *AttachmentList*.

Obrázek 4.2: Smímky obrazovky aplikace

4.3.6 Styly

Pro stylování je využit jazyk CSS a také nástroj TailwindCSS [57]. Snímky obrazovek hotové aplikace jsou na obrázku 4.2.

„Namísto předem navržených součástí poskytuje Tailwind třídy nástrojů nízké úrovně, které vám umožňují vytvářet zcela vlastní návrhy, aniž byste opustili HTML.“ [57, přeložil autor].

Tailwind je sada CSS tříd, s předdefinovanými styly. Komponenty se tedy nestylují přímo pomocí CSS, ale přidáváním CSS tříd (vygenerovaných nástrojem TailwindCSS) přímo k HTML elementům. Tailwind byl zvolen, protože umožňuje velmi rychlé prototypování uživatelských rozhraní.

4.3.7 Grafy

Pro vykreslení grafu teploty je využita knihovna Plotly.js [58]. Na obrázku 4.3 je vidět vykreslení grafu teploty a vlhkosti vzduchu.

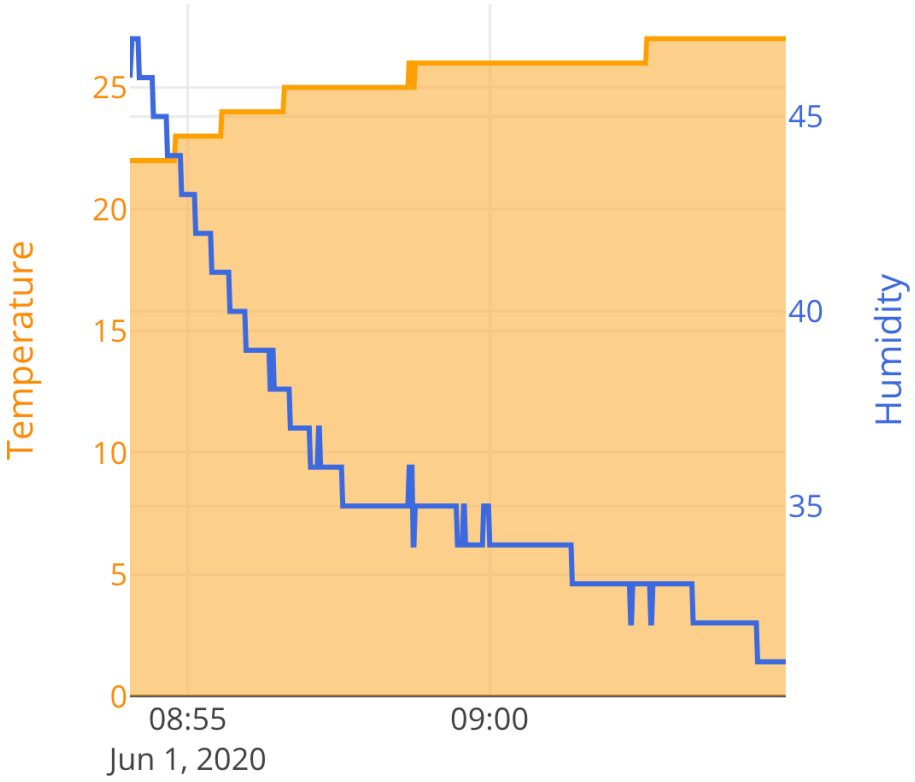
4.4 Firmware pro ESP8266

Firmware bude implementován ve vývojovém prostředí ArduinoIDE. Využito bude několik knihoven:

WiFiManager Automatizace připojení k Wi-Fi

ArduinoOTA Over the Air update (aktualizace firmwaru přes Wi-Fi)

EspMQTTClient Komunikace pomocí protokolu MQTT



Obrázek 4.3: Vykreslení teploty a vlhkosti vzduchu

ArduinoJson Parsování příchozích zpráv a kódování odchozích zpráv ve formátu JSON

Adafruit DHT Čtení dat ze senzoru teploty a vlhkosti vzduchu

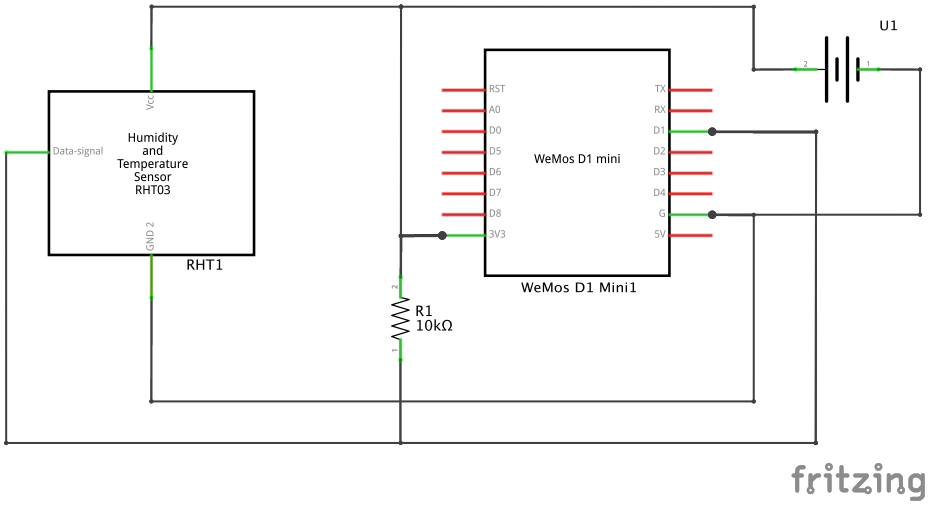
4.5 Návrh zapojení hardwaru

Pro realizaci zařízení v domácnosti budou využity vývojové desky Wemos D1 mini. Napájení desek bude možné řešit několika způsoby, podle potřeby daného zařízení. Světla mohou být napájena ze zdroje a pro senzory otevření dveří jsou vhodnější baterie.

Možnosti napájení:

- Micro USB port na Wemos D1 mini
- Napájení ze spínaného zdroje stejnosměrného napětí o hodnotě 5V
- Li-on baterie typu 18650 s přidaným převodníkem napětí s výstupní hodnotou 5V

Externí zdroj nebo baterie budou připojeny přímo na piny 5V a GND desky Wemos D1 mini, viz obrázek 4.4.Následující diagramy byly vytvořeny v open-source softwaru Fritzing [59]

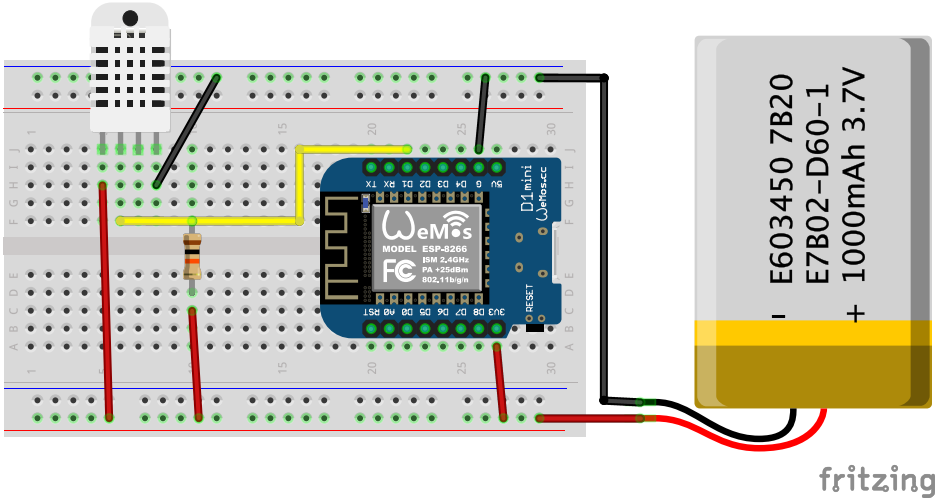


Obrázek 4.4: Návrh zapojení pro teplotní senzor

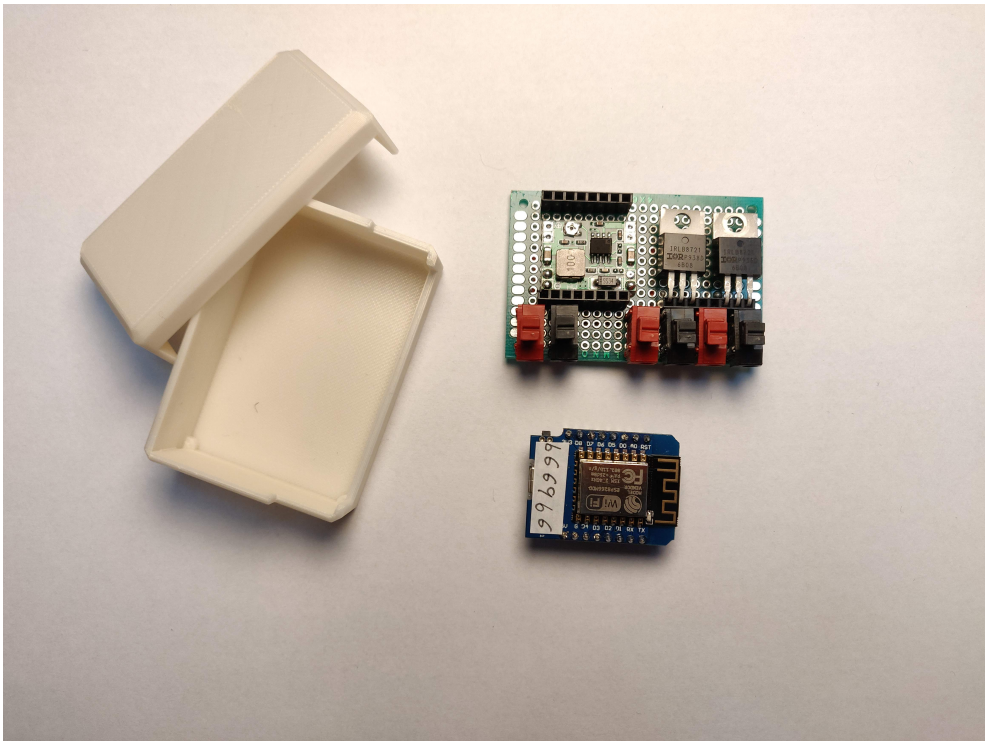
4.6 Výroba hardware

Na obrázku 4.5 je vidět zapojení senzoru teploty a vlhkosti vzduchu. Obrázek byl vytvořen v open-source software Fritzing [59].

Pro prototypování byla využita nepájivá kontaktní pole a prototypové PCB desky. Pro výrobu krabičky byl použit 3D tisk z materiálu PLA Polylactic Acid. Výsledek viz 4.6.



Obrázek 4.5: Nákres prototypu teplotního senzoru



Obrázek 4.6: Krabička vytištěná na 3D tiskárně Original Prusa i3 MK2.5S + hardware ovladače led pásků

Kapitola 5

Testování

5.1 API Testy

Node.js API bylo testováno za pomoci nástroje Postman [60]. Tento nástroj umožňuje simulovat odesílání HTTP požadavků a zobrazuje stavový kód (tj. 200 pro úspěch, 400 pro chybný požadavek, 500 pro chybu na serveru) a odpověď od serveru. Testovány byly požadavky viz tabulka 5.1.

Cesta	GET	POST	PUT	DELETE
/devices	x			
/devices/id	x	x	x	x
/rooms	x			
/rooms/id	x	x	x	x
/attachments	x			
/attachments/id	x	x	x	x
/attachments/id/toggle		x		
/attachments/id/getTemperatureData	x			
/rooms/id/toggleAllLights		x		

Tabulka 5.1: API testy

5.2 Zotavení systému po nenadálé události

Platforma jako celek byla testována na výpadky Wi-Fi připojení a kompletní výpadek elektrického napájení. Všechny součásti jsou schopny se z těchto událostí plně a automaticky zotavit. Spojení se automaticky obnoví po obnově Wi-Fi připojení nebo restartu zařízení.

5.3 Uživatelské testování

Za pomoci uživatelského testování bylo validováno splnění cílů vytyčených na začátku práce. Uživatelské testování probíhalo za dozoru, ale bez zásahu autora práce. Testovací uživatelé dostali uživatelskou příručku, viz příloha C, základnu, napájecí kabel, prototyp chytrého teploměru a baterii. Cílem testování bylo zjistit, zda jsou uživatelé schopni samostatně zapojit a zprovoznit a užívat základní prvky chytré domácnosti.

Po dokončení předchozí části, byly uživatelé požádáni o vyplnění zpětné vazby v podobě dotazníku viz příloha D. Výsledky testování s dvěma uživateli jsou zaneseny v tabulce 5.2. První z uživatelů byl typický představitel cílové skupiny, muž, 48 let, se zájmem o moderní technologie. Druhým testovacím uživatelem byla žena, 50 let.

Otázka	Uživatel 1	Uživatel 2
Seskupení zařízení do Pokojů a hromadné ovládání	4	5
Zobrazení stavu zařízení (připojeno / nepřipojeno, stav baterie)	5	5
Ovládání hlasem	5	4
Modularita (tj. k jednomu bezdrátovému modulu je možné připojit více senzorů)	2	5

Tabulka 5.2: Výsledky uživatelského testování. Hodnocení na škále od 1(nejméně důležité) do 5(nejdůležitější)

Otázka	Uživatel 1	Uživatel 2
Bylo složité vyznat se v instrukcích	nesouhlasím	spíše nesouhl...
Bylo složité vykonat požadované úlohy	nesouhlasím	nesouhlasím
Nevěděl jsem jak spustit aplikaci	nesouhlasím	spíše nesouhl...
Nevěděl jsem jak zapnout/vypnout světla	nesouhlasím	souhlasím
Nevěděl jsem jak zjistit aktuální teplotu	nesouhlasím	nesouhlasím
Nerozuměl jsem co se po mě chce	nesouhlasím	nesouhlasím

Tabulka 5.3: Výsledky uživatelského testování. Hodnocení na škále nesouhlasím - spíše nesouhlasím - nevím - spíše souhlasím - souhlasím

5.4 Vyhodnocení testování

Z pozorování uživatelů při provádění testovacích scénářů vyplynuly následující poznatky, které bude třeba zapracovat do budoucích verzí aplikace a manuálu.

Při požádání o přiložení telefonu k základně kvůli načtení URL adresy aplikace z NFC tagu se ukázalo, že uživatelé nevědí, kde se u jejich telefonu nachází NFC anténa a přikládali telefon takovým způsobem, že nebyli schopni tag načíst.

Uživatelé projevili zmatení při nastavování nového zařízení. V aplikaci jsou pole pro jméno zařízení předvyplněna výchozími hodnotami. Uživatelé nevěděli zda mají výchozí hodnoty smazat a v tomto kroku se na nějakou dobu zastavili než nakonec přešli k dalšímu kroku v manuálu. Jinak proběhlo nastavení nových zařízení bez problému a bez potřeby vnějšího zásahu. Uživatelé chválili graficky zpracovaný manuál.

Při nastavení nového Doplňku uživatelé nevěděli jaký mají nastavit Pin. Toto je možnost pro pokročilé a pokud je připojeno zařízení s předem zapojenými piny (tato zařízení byla použita pro testování), tak bude vhodné volbu Pinu skrýt a pokročilá nastavení zobrazit pouze na vyžádání uživatele.

Závěr

Cílem této práce bylo navrhnout a implementovat inteligentní domácí platformu založenou na Wi-Fi spojení jednotek s Raspberry Pi. Platforma měla mít intuitivní uživatelské rozhraní a nabízet jednoduchou konfiguraci zařízení. Platforma měla být navržena tak, aby byla otevřená a přístupná pro použití a vývoj rozšíření pro připojení k jiným zařízením a platformám.

Tohoto cíle bylo dosaženo, výsledná platforma umožňuje uživatelům ovládat chytrá světla a zásuvky, seskupovat je podle místností a ovládat celé místnosti najednou. Platforma poskytuje data ze senzorů teploty a vlhkosti a senzorů otevření dveří. Toho bylo dosaženo pomocí intuitivního uživatelského rozhraní, které bylo navrženo tak, aby vyhovovalo potřebám uživatelů a bylo ověřeno uživatelským testováním. Mezi důležité vlastnosti navrženého řešení patří jednoduché ovládání, přívětivé uživatelské rozhraní a otevřenost celé platformy.

Smarthome platforma implementovaná jako součást této práce je navržena k rozšiřování. Pomocí REST API, lze rozšířit o další klientské aplikace nebo napojit na hlasové asistenty. Flexibilní MQTT protokol umožňuje rozšíření o další typy koncových zařízení. Celé řešení bude uvolněno k použití a dalšímu rozšiřování pod licencí MIT a bude dostupné na adrese: `https://tomastrejdl.github.io/smart-home-platform/`.

Bibliografie

1. *Raspberry Pi* [online] [cit. 2020-05-08]. Dostupné z: https://www.raspberrypi.org/.
2. APPLE, INC. *HomeKit* [online] [cit. 2020-05-08]. Dostupné z: https://www.apple.com/ios/home/.
3. *Home Assisstant* [online] [cit. 2020-05-08]. Dostupné z: https://www.home-assistant.io/.
4. OPENHAB. *OpenHAB* [online] [cit. 2020-05-08]. Dostupné z: https://www.openhab.org/.
5. JIANG, Li; LIU, Da-You; YANG, Bo. Smart home research. In: *Proceedings of 2004 international conference on machine learning and cybernetics (IEEE Cat. No. 04EX826)*. 2004, sv. 2, s. 659–663.
6. FOR THE SMART HOME. *The Essential Smart Home Glossary of Terms* [online] [cit. 2020-05-08]. Dostupné z: https://forthesmarthome.com/smart-home-glossary/.
7. JOSH.AI. *Smart Home Terms Everyone Should Know* [online]. Medium [cit. 2020-05-08]. Dostupné z: https://medium.com/@joshdotai/smart-home-terms-everyone-should-know-8094823043b7.
8. AMAZON. *Amazon Alexa* [online] [cit. 2020-05-08]. Dostupné z: https://www.amazon.com/b/ref=aeg_d_nav_cat/ref=s9_acss_bw_cg_acpnav_md1_w?node=17934671011&pf_rd_m=ATVPDKIKX0DER&pf_rd_s=merchandised-search-top-1&pf_rd_r=KT5CE4QK9D8YPS0JSSDZ&pf_rd_t=101&pf_rd_p=60145196-d061-449a-be91-1cf9efd9dad2&pf_rd_i=9818047011.
9. APPLE, INC. *Siri* [online] [cit. 2020-05-08]. Dostupné z: https://www.apple.com/siri/.

10. HUBITAT. *Home Automation: The Local vs. Cloud Question* [online] [cit. 2020-05-08]. Dostupné z: https://hubitat.com/blogs/home-automation-blog/home-automation-local-vs-cloud-dependence.

11. WELCH, Chris. *Logitech will brick its Harmony Link hub for all owners in March* [online]. The Verge [cit. 2020-05-08]. Dostupné z: https://www.theverge.com/circuitbreaker/2017/11/8/16623076/logitech-harmony-link-discontinued-bricked.

12. WOO, Jong-bum; LIM, Youn-kyung. User Experience in Do-It-Yourself-Style Smart Homes. In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. Osaka, Japan: Association for Computing Machinery, 2015, s. 779–790. UbiComp '15. ISBN 9781450335744. Dostupné z DOI: 10.1145/2750858.2806063.

13. MARKETWATCH, INC. *Smart Home Market 2018-2023* [online] [cit. 2020-05-08]. Dostupné z: https://www.marketwatch.com/press-release/smart-home-market-2018-2023overview-and-scope-industry-size-market-share-leading-players-and-forecast-2020-04-15?mod=mw_quote_news.

14. GOOGLE, INC. *Nest* [online] [cit. 2020-05-08]. Dostupné z: https://nest.com.

15. AMAZON. *Echo and Alexa* [online] [cit. 2020-05-08]. Dostupné z: https://www.amazon.com/Amazon-Echo-And-Alexa-Devices/b/?ie=UTF8&node=9818047011&ref_=sv_devicesubnav_1.

16. SAMSUNG. *Smart Things* [online] [cit. 2020-05-08]. Dostupné z: https://www.samsung.com/cz/apps/smartthings/.

17. IKEA. *Home smart* [online] [cit. 2020-05-08]. Dostupné z: https://www.ikea.com/cz/cs/product-guides/ikea-home-smart-system/.

18. ALZA A.S. *Alza.cz* [online] [cit. 2020-05-08]. Dostupné z: https://www.alza.cz/.

19. ABLONDI, William. *Apple HomeKit: Poised to Storm the Smart Home Market?* [online] [cit. 2020-05-08]. Dostupné z: https://www.strategyanalytics.com/strategy-analytics/blogs/devices/smart-home/smart-home/2017/03/20/apple-homekit-poised-to-storm-the-smart-home-market.

20. HELPNETSECURITY. *First international smart home standard ensures secure connectivity between devices* [online] [cit. 2020-05-08]. Dostupné z: https://www.helpnetsecurity.com/2020/01/06/smart-home-standard/.

21. ZIGBEE ALLIANCE. *Project Connected Home over IP* [online] [cit. 2020-05-08]. Dostupné z: https://zigbeealliance.org/news_and_articles/connectedhomeIP/.

22. ZIGBEE ALLIANCE. *Project Connected Home over IP* [online] [cit. 2020-05-08]. Dostupné z: https://www.connectedhomeip.com/.

23. HOME ASSISTANT, INC. *home-assistant/operating-system* [software]. Verze 4.8 [cit. 2020-05-08]. Dostupné z: https://github.com/home-assistant/operating-system.

24. ARENDST, Theo. *Tasmota* [software]. Verze 8.3.1 [cit. 2020-05-08]. Dostupné z: https://tasmota.github.io/docs/.

25. IFTTT. *IFTTT* [online] [cit. 2020-05-08]. Dostupné z: https://ifttt.com/.

26. IOT NOW. *Smart Homes technology on the verge of mass adoption* [online] [cit. 2020-05-08]. Dostupné z: https://www.iot-now.com/2016/05/16/47326-smart-homes-technology-on-the-verge-of-mass-adoption/.

27. ASSOCIATION FRANCO-CHINOISE DU DÉVELOPPEMENT URBAIN DURABLE, urldate=2020-05-08. *The History of Smart Homes*. Dostupné také z: https://www.afcdud.com/fr/smart-city/422-how-the-history-of-smart-homes.html.

28. FIELDING, Roy Thomas. *Architectural styles and the design of network-based software architectures*. 2000. Disertační práce. University of California.

29. ESPRESSIF. *ESP8266* [online] [cit. 2020-05-08]. Dostupné z: https://www.espressif.com/en/products/socs/esp8266ex/overview.

30. MQTT. *MQTT* [online] [cit. 2020-05-08]. Dostupné z: https://mqtt.org/.

31. HILLAR, Gastón Cr. *Publisher-Subsciber Pattern* [online] [cit. 2020-05-08]. Dostupné z: https://www.embedded.com/mqtt-essentials-scenarios-and-the-pub-sub-pattern/.

32. GOOGLE, INC. *Angular*. Angular [software]. 2020. Verze 9.1.7 [cit. 2020-05-08]. Dostupné z: https://angular.io/.

33. FACEBOOK, INC. *React* [software]. Verze 16.13.1 [cit. 2020-05-08]. Dostupné z: https://reactjs.org/.

34. YOU, Evan. *Vue.js* [software]. Verze 2.6.11 [cit. 2020-05-08]. Dostupné z: https://vuejs.org/.

35. MICROSOFT. *Model–view–viewmodel* [online] [cit. 2020-05-08]. Dostupné z: https://docs.microsoft.com/cs-cz/xamarin/xamarin-forms/enterprise-application-patterns/mvvm.

36. OPENJS FOUNDATION. *Node.js*. Node.js [software]. 2020. Verze 12.11.1 [cit. 2020-05-08]. Dostupné z: https://nodejs.org.

37. OPENJS FOUNDATION. *About Node.js* [online] [cit. 2020-05-08]. Dostupné z: https://nodejs.org/en/about/.

38. POKORNÝ, David. České vysoké učení technické v Praze, Fakulta informačních technologií, 2019. Dostupné také z: https://dspace.cvut.cz/handle/10467/83128. Diplomová práce.

39. OWASP. *Cross Site Scripting Prevention* [online] [cit. 2020-05-08]. Dostupné z: https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html.

40. OWASP. *SQL Injection* [online] [cit. 2020-05-08]. Dostupné z: https://owasp.org/www-community/attacks/SQL_Injection.

41. FOUNDEO, INC. *Content Security Policy Reference* [online] [cit. 2020-05-08]. Dostupné z: https://content-security-policy.com/.

42. HOLOWAYCHUK, T.J. *ratelimiter* [software]. Verze 3.4.1 [cit. 2020-05-08]. Dostupné z: https://www.npmjs.com/package/ratelimiter.

43. MOZILLA. *HTTP Strict Transport Security* [online] [cit. 2020-05-08]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security.

44. MOZILLA. *Referrer-Policy* [online] [cit. 2020-05-08]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referrer-Policy.

45. SPAGNUOLO, Michele. *The power of DNS rebinding: stealing WiFi passwords with a website* [online] [cit. 2020-05-08]. Dostupné z: https://blog.miki.it/2015/4/20/the-power-of-dns-rebinding-stealing-wifi-passwords-with-a-website/.

46. OPENJS FOUNDATION. *csurf* [online] [cit. 2020-05-08]. Dostupné z: https://www.npmjs.com/package/csurf.

47. CODECADEMY. *What is CRUD?* [online] [cit. 2020-05-08]. Dostupné z: https://www.codecademy.com/articles/what-is-crud.

48. SMARTBEAR. *API Development for Everyone* [online] [cit. 2020-05-08]. Dostupné z: https://swagger.io/.

49. JSDOC. *JSDoc* [software]. Verze 3.6.4 [cit. 2020-05-08]. Dostupné z: https://jsdoc.app/.

50. MONGODB, INC. *MongoDB* [software]. Verze 3.2 [cit. 2020-05-08]. Dostupné z: https://www.mongodb.com/.

51. LEARNBOOST. *Mongoose* [software]. Verze 5.9.17 [cit. 2020-05-08]. Dostupné z: https://mongoosejs.com/.

52. GOOGLE, INC. *Google Chrome* [software]. Verze 83.0.4103.61 [cit. 2020-05-08]. Dostupné z: https://www.google.com/chrome/.

53. MICROSOFT. *Microsoft Edge* [software]. Verze 83.0.478.44 [cit. 2020-05-08]. Dostupné z: `https://www.microsoft.com/cs-cz/edge`.

54. GOOGLE, INC. *Chromium* [software]. Verze 83.0.4103.61 [cit. 2020-05-08]. Dostupné z: `https://www.chromium.org/`.

55. APPLE, INC. *Safari* [software]. Verze 13.1.1 [cit. 2020-05-08]. Dostupné z: `https://www.apple.com/safari/`.

56. FACEBOOK, INC. *Flux* [online] [cit. 2020-05-08]. Dostupné z: `https://facebook.github.io/flux/`.

57. WATHAN, Adam. *Tailwind CSS* [online] [cit. 2020-05-08]. Dostupné z: `https://tailwindcss.com/`.

58. PLOTLY. *Plotly JavaScript Graphing Library* [software]. Verze 1.52.3 [cit. 2020-05-08]. Dostupné z: `https://plotly.com/javascript/`.

59. FRIENDS-OF-FRITZING FOUNDATION. *Fritzinge* [software]. Verze 0.9.5 [cit. 2020-05-08]. Dostupné z: `https://fritzing.org/`.

60. POSTMAN, INC. *Postman* [software]. Verze 7.25.0 [cit. 2020-05-08]. Dostupné z: `https://www.postman.com/`.

Seznam použitých zkratek

ČVUT České vysoké učení technické v Praze 13

AES Andvancer Encryption Standard 24

API Application Programming Interface 3, 20, 27, 30

BI-SI1 Softwarové inženýrství 1 13

CORS Cross Origin Resource Sharing 25

CRUD Create, Read, Update, Delete 27

CSRF Cross Site Request Forgery 25

CSS Cascading Style Sheets 21, 29, 31

DIY Do It Yourself (Udělej si sám) 3, 10

FIT Fakulta informačních technologií 13

HTML Hypertext Markup Language 21, 25, 29–31

HTST HTTP Strict Transport Security 25

HTTP Hypertext Transfer Protocol 22, 27, 28, 30, 35

HTTPS Hypertext Transfer Protocol Secure 22, 24, 25

IDE Integrated Developement Environment 21

JSON JavaScript Object Notation 23, 27, 32

MQTT Message Queuing Telemetry Transport 22, 25, 27, 31

MVVM Model-View-ViewModel 23

NFC Near Field Communication 37

NoSQL Non SQL or Not only SQL (Structured Query Language) 23, 24, 28

OOP Object Oriented Programming 30

OTA Over The Air 25, 31

PC Personal Computer 20

PCB Printed Circuit Board 33

PLA Polylactic Acid 33

PWA Progressive Web App 21, 29

REST REpresentational State Transfer 20, 27

SPA Single Page App 30

TCP/IP Transmission Control Protocol/Internet Protocol 27

TKIP Temporal Key Integrity Protocol 24

UML Unified Modeling Language 13

URL Uniform Resource Locator 25, 37

VPN Virtual Private Network 7

WPA Wi-Fi Protected Access 24

XSS Cross Site Scripting 24

XSSI Cross-Site Script Inclusion 25

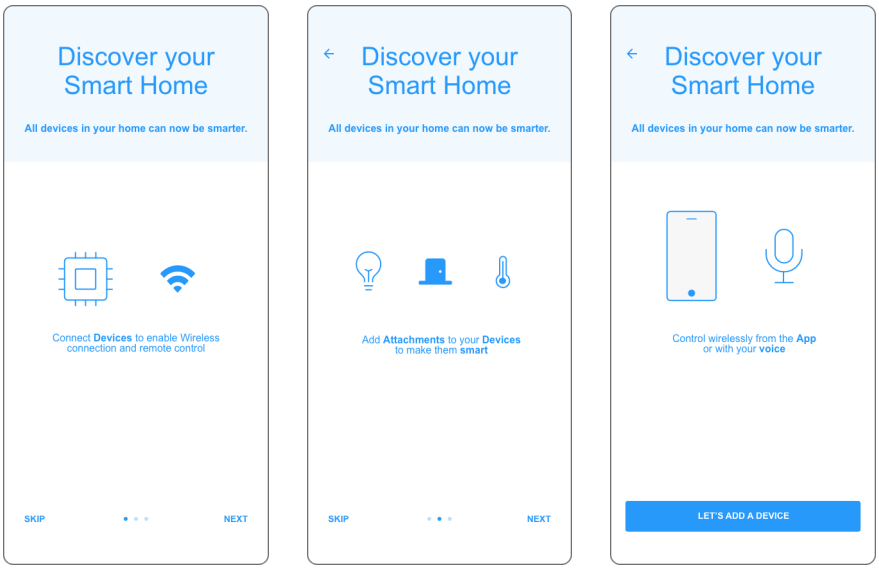
Příloha A

Obsah přiloženého paměťového média

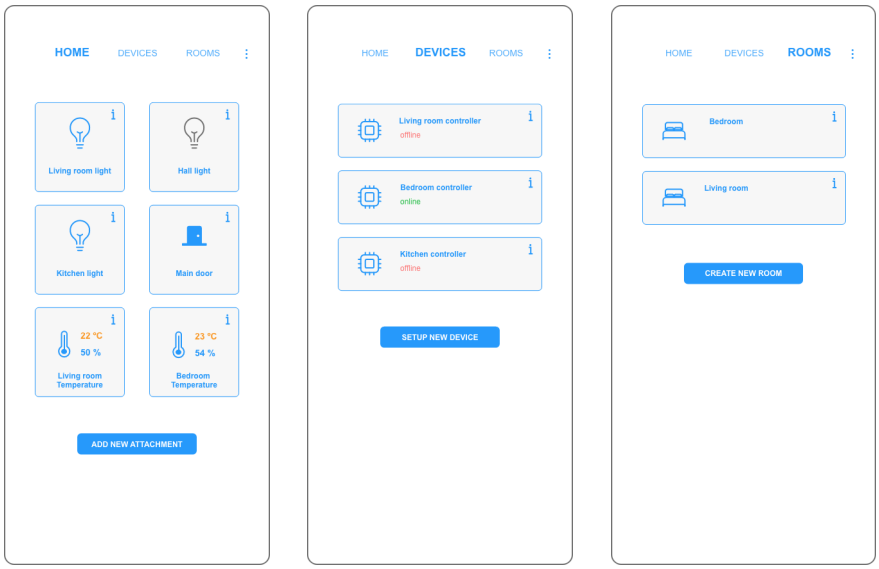
- `readme.txt` ..... stručný popis obsahu
- `src` ..... zdrojové kódy implementace
  - `frontend` ..... zdrojové kódy webové aplikace
  - `backend` ..... zdrojové kódy serveru
  - `firmware` ..... zdrojové kódy firmwaru
  - `test` ..... soubory pro testy
- `docs` ..... dokumentace
- `thesis` ..... text práce
  - `src` ..... zdrojová forma práce ve formátu LaTeX
  - `thesis.pdf` ..... text práce ve formátu PDF
- `assets`
  - `models` ..... 3D modely pro tisk
  - `graphics` ..... grafické návrhy

Příloha B

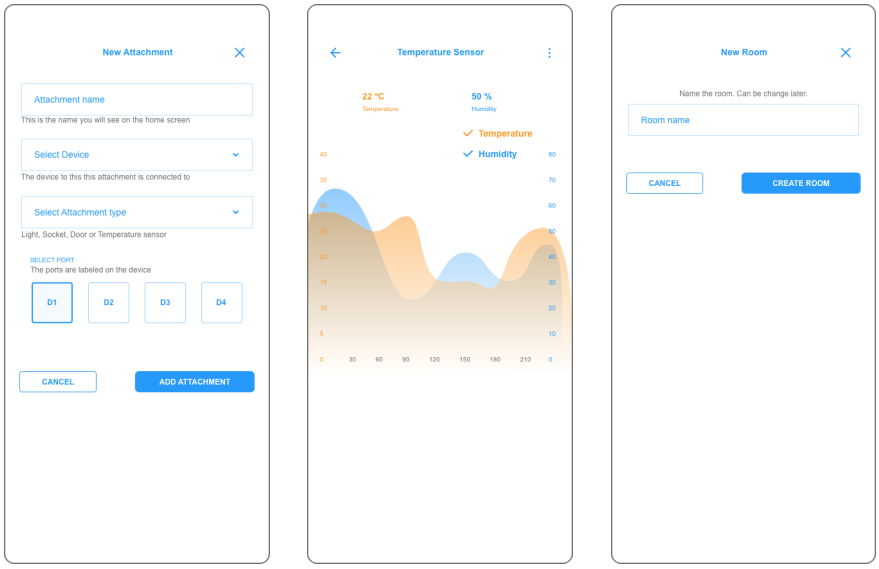
Návrh obrazovek



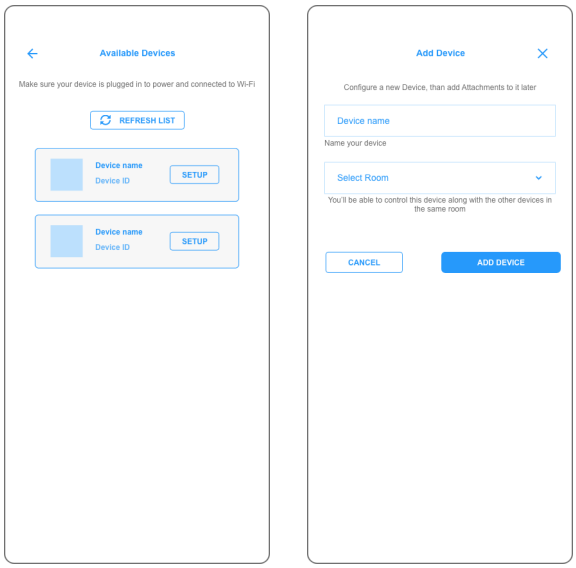
Obrázek B.1: Onboarding – návod pro uživatele, jak používat aplikaci



Obrázek B.2: Domovská obrazovka, seznam zařízení, seznam pokojů



Obrázek B.3: Přidání nového Doplňku, zobrazení teploty, vytvoření pokoje



Obrázek B.4: Konfigurace nového zařízení

Příloha C

Uživatelská příručka

smarthome
Uživatelská příručka

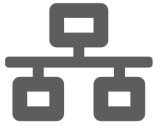
Zapojení základny

1

Připojte základnu k napájení přiloženým napájecím kabel s konektorem microUSB

2

Připojte základnu k síti LAN (Přeskočte tento krok pokud používáte Wi-Fi)



3

Počkejte cca 5 minut než se zařízení zapne

4

Přejděte k nastavení vašeho prvního zařízení

Připojení nového zařízení k Wi-Fi

1

Připojte zařízení k napájení

2

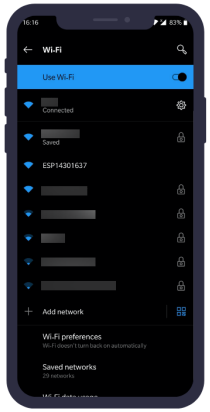
Ve svém smartphone otevřete nastavení Wi-Fi

Nastavení -> Wi-Fi & Internet

Nastavení -> Wi-Fi

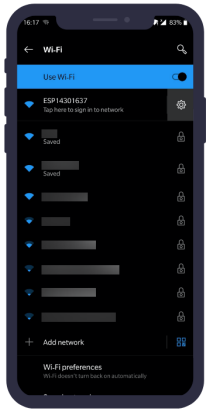
3

Vyberte síť začínající na ESP, toto je vaše zařízení



4

Vyberte "Přihlásit se k síti Wi-Fi"

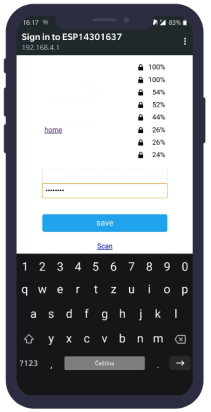


5

Zvolte "Configure WiFi"

6

Vyberte vaši domácí Wi-Fi síť ze seznamu a zadejte heslo, tímto se dané zařízení připojí k vaší domácí Wi-Fi



smarthome
Uživatelská příručka

Konfigurace zařízení v aplikaci

1

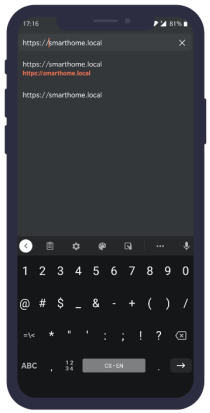
Pokud máte telefon s podporou NFC (většina moderních telefonů, iPhone od verze 6), přiložte telefon k základně, budete přesměrování přímo do aplikace a pokračujte

2

Pokud nemáte NFC, nebo předchozí krok selhal, otevřete svůj webový prohlížeč

3

Přejděte na adresu **http://smarthome.local**

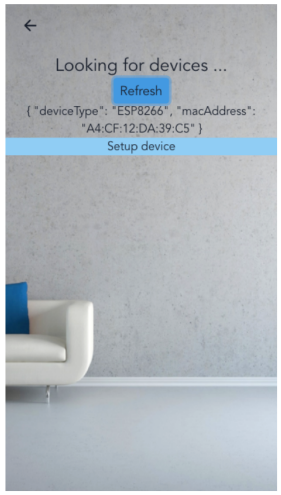


4

Zvolte “Devices”, dále zvolte “Add device”

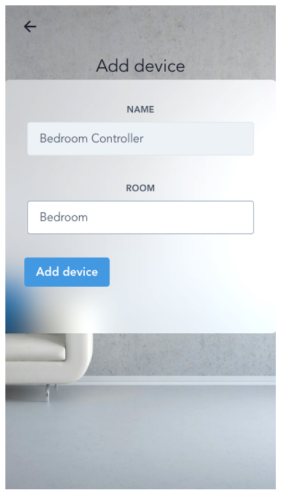
5

Zde vidíte seznam zařízení připojených k vaší síti, pokud žádné nevidíte ujistěte se že jste provedli předchozí kroky a zvolte “Refresh”, až se zařízení zobrazí zvolte “Setup Device”



6

Zařízení si pojmenujte, popřípadě zvolte pokoj ve kterém bude umístěno a zvolte “Add device”

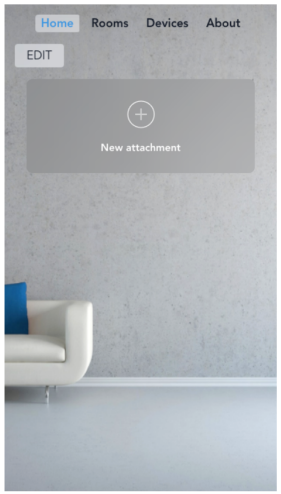


smarthome
Uživatelská příručka

Konfigurace zařízení v aplikaci

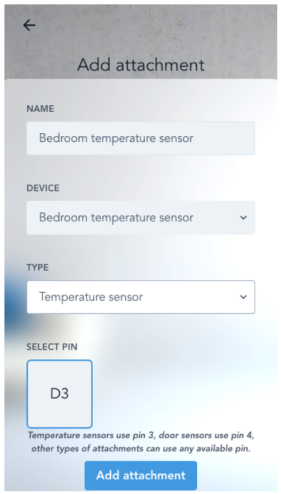
1

Přejděte na “Home” a zvolte “New Attachment”



2

Tímto vytvoříte Doplněk, který uvidíte na domovské obrazovce, Doplněk pojmenujte, zvolte zařízení a Pin, ke kterému je na zařízení Doplněk připojen



3

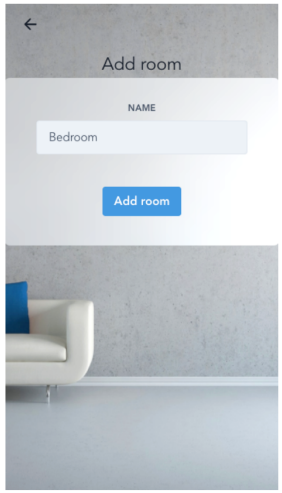
Na domovské obrazovce nyní vidíte přidaný Doplněk, v tomto případě teploměr a hodnoty aktualizované v reálném čase

4

Pro vytvoření nového pokoje přejděte na obrazovku “Rooms” a zvolte “New room”

5

Pokoj pojmenujte a zvolte “Add room”



6

Jste na konci, gratulujeme! Užijte si svou novou chytrou domácnost

Příloha D

Dotazník uživatelského testování

Smart home User Session

Datum	
Pohlaví	
Věk	

Dotazník

▼ Chcete ovládat světla pomocí chytrého telefonu (Seřaďte vlastnosti takového produktu, který toto umožňuje od nejdůležitější po nejméně důležitou)

1. Má mobilní aplikaci, která lze nainstalovat (oproti ovládání z webového prohlížeče)
2. Lze ovládat jak z telefonu tak i z PC
3. Snadnost instalace (nevyžaduje odbornou znalost a obsahuje podrobný návod)
4. Rozsáhlé možnosti konfigurace (na úkor jednoduchosti nastavení)
5. Aplikace má moderní design (vypadá hezky)
6. Stejná společnost prodává velké množství dalších příslušenství (Chytré žárovky, teploměry, senzory pohybu, senzory úniku vody/kouře)
7. Kompatibilní s již vlastněnými zařízeními (Google Home, HomePod, Amazon Echo)

Na škále od 1(nejméně důležité) do 5(nejdůležitější) ohodnoťte:

Name	Důležitost
Seskupení zařízení do Pokojů a hromadné ovládání	
Zobrazení stavu zařízení (připojeno/nepřipojení, stav baterie)	
Ovládání hlasem	
Modularita (tj. k jednomu bezdrátovému modulu je možné připojit více senzorů)	

Nastavení zařízení

Aa Name	Odpověď
Bylo složité vyznat se v instukcích	
Bylo složité vykonat požadované úlohy	
Nerozuměl/a jsem co se po mě chce	

Ovládání

Aa Name	Odpověď
Nevěděl/a jsem jak spustit aplikaci	
Nevěděl/a jsem jak zapnout/vypnout světla	
Nevěděl/a jsem jak zjistit aktuální teplotu	
Nerozuměl/a jsem co se po mě chce	