



**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

F3

Fakulta elektrotechnická

Bakalářská práce

Optimalizace nákladů na elektřinu pomocí IoT platformy

Jan Veselý

Květen 2020

Vedoucí práce: Ing. Jiří Šebek

Poděkování / Prohlášení

Chtěl bych poděkovat svému vedoucímu Ing. Jiřímu Šebkovi za kvalitní připomínky a vytrvalou pomoc při psaní této práce, dále pak Ing. Jiřímu Veselému za praktické rady při využití této metody.

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

V Praze dne 19. 05. 2020

.....

Abstrakt / Abstract

Tento dokument ukazuje, jak je možné snížit náklady za elektřinu pomocí IoT prvků a základních myšlenek průmyslu 4.0. tj. snaží se propojit rozhodovací mechanismy s různými daty a statistikami včetně online cen.

Teoreticky popisuje způsob optimalizace nákladů na elektrickou energii, která je založená na tarifech, jež počítají cenu elektřiny pomocí velkoobchodních cen, tj. cen z denního trhu s elektřinou. Definuje optimalizační algoritmus, na jehož základě jsou spouštěny jednotlivé spotřebiče. Srovnává simulaci vytvořenou pro ceny z denního trhu a z reálného prostředí. Následně popisuje fungování systému.

Systém se skládá z mobilní aplikace, řídicí jednotky a spouštěcích jednotek. Řešení je distribuované v jednotlivých domácnostech tj. není řízeno centrálně a tím snižuje možnosti neoprávněných zásahů zvenčí do kritické infrastruktury. Mobilní aplikace je implementována pro platformu Android a řídicí jednotka je nasazena na Raspberry Pi 4B.

Klíčová slova: IoT, náklady na elektřinu, optimalizace, Raspberry Pi, mobilní aplikace

This document describes a way of reducing electricity expenditure based on principles of Industry 4.0 and modular IoT devices. It interwinds decision-making mechanisms with available data from sensors, external online sources, and their statistics.

It constructs a theoretical base for reducing total electricity costs by harnessing the benefits of electricity plans based on the day-ahead market. The document defines an optimization algorithm for controlling electric appliances. Finally compares expenditure of fixed electricity plan and plan based on day-ahead prices.

The system is composed of a mobile app, central unit, and power units. The system is distributed in households, there is no central management, therefore it reduces the possibility of overtaking the critical infrastructure. The mobile app is implemented on Android, central unit and power units are programmed for Raspberry Pi 4B.

Keywords: IoT, electricity expenditure, optimization, Raspberry Pi, mobile application

Title translation: Optimizing electricity costs through IoT

Obsah /

1 Úvod	1
2 Elektřina	2
2.1 Trh elektřinou	3
2.1.1 Aktéři	3
2.1.2 Druhy trhů	3
2.1.3 Krátkodobé trhy	3
2.2 Tarif Energie A++	4
3 Architektura systému	5
3.1 Průmysl 4.0	5
3.2 Inspirace	5
3.3 Návrh	5
3.4 Mobilní aplikace	6
3.5 Spínací jednotky	6
3.6 Centrální jednotka	6
4 Teorie optimalizace	7
4.1 Denní data	7
4.2 Optimalizační strategie	7
4.2.1 Block Optimizer	8
4.2.2 Scheduled Optimizer	9
4.3 Optimalizační plán	9
4.3.1 Správné nastavení optimalizačního plánu	9
5 Simulace	10
5.1 Struktura simulace	10
5.2 Implementace simulace	11
5.3 Modely	12
5.3.1 Denní trh	12
5.3.2 Fixní tarif	13
5.4 Srovnání obou simulací	14
5.5 Obecné závěry	16
6 Centrální jednotka	17
6.1 Hardware	17
6.2 Architektura	17
6.2.1 Webový server	17
6.2.2 Persistence dat	18
6.2.3 REST	18
6.2.4 Rozhraní	19
6.3 Výpočet aktivačního plánu	20
6.4 Spouštění spotřebičů	20
7 Mobilní aplikace	21
7.1 Návrh UI UX	21
7.1.1 Material Design	21
7.1.2 Navigace	22
7.1.3 Spotřebič	23
7.1.4 Optimalizační plán	24
7.1.5 Aktivační plán	25
7.1.6 Přehled cen	26
7.2 Architektura	27
7.2.1 Decoupling	27
7.2.2 Propojení na REST	28
8 Prototyp hardwaru	29
8.1 Platformy Raspberry Pi 4B ...	29
8.2 Arduino Uno Wifi rev 2	29
8.3 Prototyp spínání	30
9 Implementační detaily	31
9.1 Modularizace projektu	31
9.2 Jednotná technologie	31
9.3 Jednotný datový model	32
10 Testování	33
10.1 Jednotkové testování	33
10.2 Integrační testy	33
10.2.1 Ktor	33
10.3 Uživatelské testy	34
10.4 Firebase	34
10.5 Výsledek testování	35
10.6 Posouzení kvality UI UX	35
11 Další rozvoj projektu	36
11.1 Nasazení systému	36
11.2 Další rozvoj	36
11.3 Elektromobilita	36
12 Závěr	37
Literatura	38
A Matematické vzorce	41
B Zadání práce	42

Tabulky / Obrázky

2.1. Rozdělení spotřeb energie.....2	3.1. Architektura aplikace5
4.1. Optimalizační plánu.....9	4.1. Optimalizační mechanismus7
5.1. Spouštěcí plán - denní trh 12	4.2. Optimalizační algoritmus8
5.2. Spouštěcí plán - fixní tarif 13	4.3. Průměrná cena 20189
5.3. Ceníku ČEZ 14	5.1. Simulace pro fixní tarif 11
5.4. Simulace 1 14	5.2. Simulace pro denní trh 11
5.5. Cena elektřiny - U Veselých ... 14	6.1. Architektura centrální jed- notky 17
5.6. Simulace 2 14	6.2. Datová entity 19
5.7. Vývoj ceny elektřiny 15	6.3. JSON datové entity 19
10.1. Testovací scénáře 34	6.4. Rozhraní REST 19
	6.5. Výpočet aktivačního plánu..... 20
	7.1. Navigace v aplikaci 22
	7.2. Obrazovka spotřebičů 23
	7.3. Obrazovka spotřebiče 23
	7.4. Obrazovka optimalizačního plánu 24
	7.5. Obrazovka blokové strategie... 24
	7.6. Obrazovka aktivačního plánu .. 25
	7.7. Obrazovka aktivačního plánu editace 25
	7.8. Obrazovka cen el. energie 26
	7.9. Architektura obrazovky 27
	7.10. Životní cyklus obrazovky 28
	7.11. Propojení na REST 28
	8.1. Raspberry Pi 4B 30
	8.2. Relé modul 30
	9.1. JVM jazyky 32
	10.1. Plán testování 33
	10.2. Ukázka intergračního testu 34

Kapitola 1

Úvod

Budoucnost a efektivita leží v principech průmyslu 4.0[1]. Jeho hlavním smyslem je propojit různá data generovaná společnostmi, výrobou nebo získaná z různých senzorů a zpětně je propojit s řídicími mechanismy. Toto spojení vede k lepšímu přidělování zdrojů a tím k zvýšení efektivity práce a úspore zdrojů či nákladů.

Cílem práce je navrhnout systém pro internet věcí na platformě Raspberry Pi včetně uživatelského rozhraní pro platformu android, který na základě cen z denního trhu s elektřinou optimalizuje spouštění elektrických zařízení tak, aby došlo ke snížení celkových nákladů.

Tento záměr vznikl již v roce 2018 v penzionu U Veselých jako odpověď na zvyšující se cenu elektřiny. Nicméně celý systém se dá generalizovat a nasadit zejména v domácnostech nebo na místech, kde není zajištění elektřiny ve špičkách nutné. Systém si klade za úkol automatizovat rutinní operace řízení energie jako například zapínání topení případně na něj navázaných systémů. Uživatel by měl mít jeden centrální bod pro ovládání zařízení skrz svoje mobilní zařízení.

Celý návrh optimalizace je navržen na tarify, které účtují spotřebovanou elektřinu podle výsledků z denního trhu. V našem případě se jedná o tarif A++[2]. Denní trh má tu výhodu pro koncového zákazníka, že cena, která mu bude účtována za spotřebované kWh za jednotlivé hodiny, je známa den předem. To je obrovský benefit, jelikož umožňuje spotřebiče, u kterých můžeme odložit čas spuštění na levnější hodinovou sazbu, spustit později a tím ušetřit za výdaje. Typicky topení nebo nabíjení elektromobilu.

V první části je čtenář krátce seznámen s fungování trhu s elektřinou a je zde představen tarif A++ od společnosti Bohemie Energy pro, který je systém navrhován. Dále je vybudována teorie optimalizace spuštění spotřebičů, které je následně ozkoušena v simulaci na reálných datech a porovnána vůči běžnému spouštění spotřebičů při fixních cenách elektřiny. Simulace ukázala, že po určité situace celý systém může ušetřit náklady na eklektickou energii.

Následně je popsána implementace Mobilní aplikace a centrální řídicí jednotky, které jsou distribuované v jednotlivých odběrných místech. Testování systému probíhalo podle testovacího plánu, který je popsán v poslední kapitole.

Kapitola 2

Elektrina

Podle U.S. Energy Information Administration (EIA)[3], které se zabývá sběrem, analýzou informací ohledně využití různých druhů energií, více než polovina konkrétně 70% procent spotřebované energie v domácnostech je použito na ohřev domu, vody či klimatizování prostor. K přibližně podobným číslům došel i Eurostat[4], který tvrdí, že se jedná o 82,2%. Data Eurostatu jsou podrobnější, jelikož uvádí i rozdělení mezi jednotlivé druhy energií. Elektrická energie je využívána ke stejnému účelu z 6,5 %. Je zde otázka, zdali položka renewables and wastes nezahrnuje i elektrické obnovitelné zdroje jako jsou například tepelná čerpadla. Poté by skutečné využití elektrické energie pro ohřev/chlazení domů a teplou vodu bylo daleko vyšší. Podle Office of Energy Efficiency & Renewable Energy[5] v roce 2011 v USA 34 % procent domácností využívalo v USA elektřinu jako zdroj tepla.

Share of fuels in the final energy consumption in the residential sector by type of end-use, 2017 (%)

EU-28	Total Residential /Households	Space heating	Space cooling	Water heating	Cooking	Lighting and appliances	Other end uses
Electricity	24.3	3.4	0.3	2.8	2.7	14.4	0.7
Derived Heat	7.8	6.1	0.0	1.7	0.0	0.0	0.0
Gas	36.5	27.6	0.0	7.1	1.9	0.0	0.0
Solid Fuels	3.3	3.0	0.0	0.3	0.0	0.0	0.0
Oil & Petroleum Products	11.3	9.0	0.0	1.6	0.7	0.0	0.1
Renewables and Wastes	16.8	15.0	0.0	1.5	0.3	0.0	0.1
Total	100.0	64.1	0.3	14.8	5.6	14.4	0.9

Source: Eurostat

eurostat 

Tabulka 2.1. Rozdělení spotřeb energie

Z předchozí statistiky vyplývá, že většina energie je spotřebována na vytápění, klimatizaci a ohřev teplé vody. Pokud domácnost chce snížit svoje mandatorní výdaje, měla by začít u těchto položek. Nejjednodušší kroky ke snížení výdajů jsou například změna dodavatele nebo vyjednání lepšího fixního tarifu na poskytnutí elektrické energie. To znamená optimalizace využití jednotlivých spotřebičů v rámci nízkého tarifu NT a vysokého tarifu VT. Spotřebiče, u kterých můžeme naplánovat spuštění v čase nízkého tarifu, nám šetří výdaje, jelikož cena za kWh NT je menší než u VT.

Všechny z výše zmíněných kroků nejsou nikterak náročné na zavedení pro běžného spotřebitele, zároveň ale neumožňují vyšší stupeň optimalizace spotřeby, a to z pohledu komfortnosti řešení nebo technické náročnosti. K pochopení optimalizace založené na tarifu A++ od Bohemia Energy¹ je třeba nejdříve pochopit trh s elektřinou, který je také popsán v části 2.2.

¹ <https://www.reseni2plus.cz/>

2.1 Trh elektřinou

Trh s elektřinou se velmi liší v porovnání s jinými trhy jako akciové nebo měnové. Elektřinu není možné uschovat, směřovat či dělit. Elektřinu můžeme uschovat jen částečně například v přečerpávacích elektrárnách nebo velkých bateriových systémech jako například Hornsdale Power Reserve v Austrálii. Elektřina se v síti směřuje k místům s největší spotřebou. Z toho také vyplývá, že jí nemůžeme dělit. A právě díky těmto specifickým podmínkám a jeho významnosti je trh výrazně legislativně regulován.

2.1.1 Aktéři

- **Provozovatel distribuční soustavy:** stará se o distribuci elektrické energie ke konečným spotřebitelům.
- **Provozovatel přenosové soustavy:** zajišťuje funkčnost páteřní soustavy tj. 400kV nebo 220kV vedení.
- **Operátor trhu s elektřinou:** zajištění fungování trhu s elektřinou, v ČR je to společnost OTE.
- **Regulační úřad:** statní úřad dohlížející na dodržování všech podmínek, udělování licencí, regulování cen elektřiny.
- **Dodavatel:** společnost zprostředkovávající nákup a prodej elektrické energie.

2.1.2 Druhy trhů

- **Neorganizovaný trh:** na tomto trhu se uzavírají bilaterální dohody. Jednotlivé strany nejsou vázány externími pravidly a záleží na nich, jaké podmínky si mezi sebou dohodnou
- **Organizovaný trh:** obchodníci uzavírají dohodu s centrální stranou. Ta zajišťuje zprostředkování nabídky a poptávky mezi stranami, zajišťuje provedení obchodu jako takového.
- **Dlouhodobý trh:** slouží pro dlouhodobé kontrakty na cenu elektřiny, které zároveň brání náhlým výkyvům.
- **Krátkodobý trh:** na tomto trhu se obchoduje s fyzickou dodávkou v řádu dní až hodin po uskutečnění obchodu. Tento trh se dělí na následující trhy.

2.1.3 Krátkodobé trhy

- **Blokový trh:** zde se obchodují tři produkty, tj. intervaly, kdy bude elektřina dodána. Base – základ = celý den. Peak – špička = od 8 do 20 hod. Off-Peak – mimo špičku = od 20 do 8 hod.
- **Vyrovňovací trh:** tento trh má centrální nabídku a poptávku a je řízen provozovatelem přenosové soustavy. Obchody se uskutečňují 30 minut před dodávkou elektřiny. Dá se zde koupit tzv. regulační elektřina, která slouží k vykrytí odchylek v přenosové soustavě.
- **Vnitrodenní trh:** trh s fyzickou dodávkou elektřiny. Obchody se zde uskutečňují minimálně hodinu před dodáním energie. Jako ve vyrovňovacím trhu zde mohou dodavatelé řešit svůj nedostatek/přebytek energie.
- **Denní trh:** nejdůležitější trh z pohledu této práce. Trh probíhá formou 24 aukcí na den dopředu. Právě ceny z tohoto trhu jsou účtovány zákazníkům podle tarifu Energie A++, které můžeme následně využít pro optimalizaci spouštění spotřebičů právě z důvodu časového předstihu den před dodávkou. Trh uzavírá obchodování v 11:00 předchozího dne.

2.2 Tarif Energie A++

„Cena silové elektřiny je pro produktovou řadu ENERGIE A++ stanovována pro každou hodinu dodávky zvlášť. Cena každé hodiny se rovná ceně, vzniklé obchodováním na denním trhu zajišťovaným operátorem trhu OTE, a.s. Výše cen je k dispozici na adrese <http://www.ote-cr.cz/kratkodobe-trhy/elektrina/denni-trh>. Ceny jsou uváděny v EUR. Pro přepočítání na Kč se použijí denní kurzy ČNB střed, které jsou k dispozici na téže adrese. Výsledná cena silové elektřiny je určena jako součet součinů hodinových spotřeb a hodinových cen dosažených obchodováním na denním trhu zajišťovaným OTE, a.s., a vydělených naměřenou spotřebou za dané období.“[2]

Celková cena elektřiny se tedy počítá následujícím vzorcem, kde s je spotřeba elektřiny v kWh a c je cena na denním trhu v konkrétní hodinu konkrétního dne.

$$cena = \sum_{d=1}^n \sum_{h=1}^{24} s_{hi} * c_{hi}$$

Vzorec 1: Výpočet ceny elektřiny z cen z denního trhu

Kapitola 3

Architektura systému

3.1 Průmysl 4.0

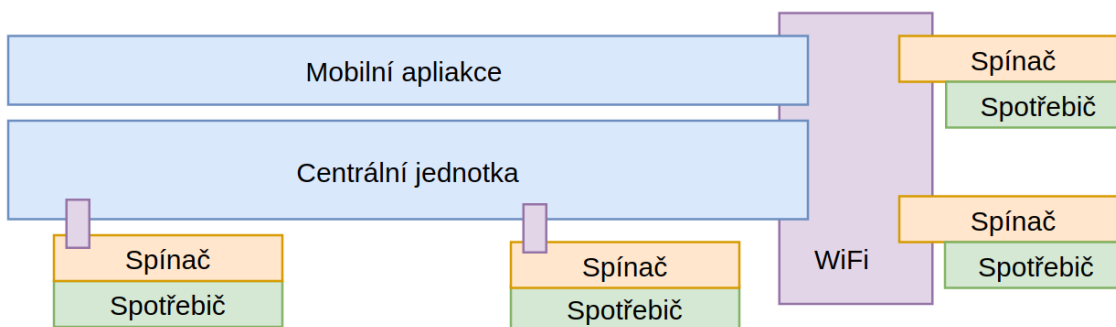
„The modern and more sophisticated machines and tools with advanced software and networked sensors can be used to plan, predict, adjust and control the societal outcome and business models to create another phase of value chain organization and it can be managed throughout the whole cycle of a product. Thus, Industry 4.0 is an advantage to stay competitive in any industry. To create a more dynamic flow of production, optimization of value chain has to be autonomously controlled.“[1]

3.2 Inspirace

„Several companies have designed intelligent HEMS to control energy consumption. For example, General Electric Co. designed a system that controls home appliances using smartphones. The system offers a strong support to customers by controlling and monitoring devices with a smartphone app. Furthermore, Honda US has implemented a software and hardware system that controls, monitors, and optimizes electrical generation and appliance usage at home. This home automation system uses solar PV and batteries to store energy that can be used during peak hours.“[6]

3.3 Návrh

Systém je navržen na tarify, které účtují spotřebovanou elektřinu podle výsledků z denního trhu. Tento trh má tu výhodu pro koncového zákazníka, že cena, která mu bude účtována za spotřebované kWh za jednotlivé hodiny zvlášť a to zdůrazňuji, je známá den předem. To je obrovská výhoda ve srovnání s běžnými tarify např. C25, jelikož můžeme spotřebiče, u kterých je možné odložit čas spuštění na levnější hodinovou sazbu, spustit později a tím ušetřit za výdaje. Optimalizují se výdaje nikoliv spotřebu elektrické energie.



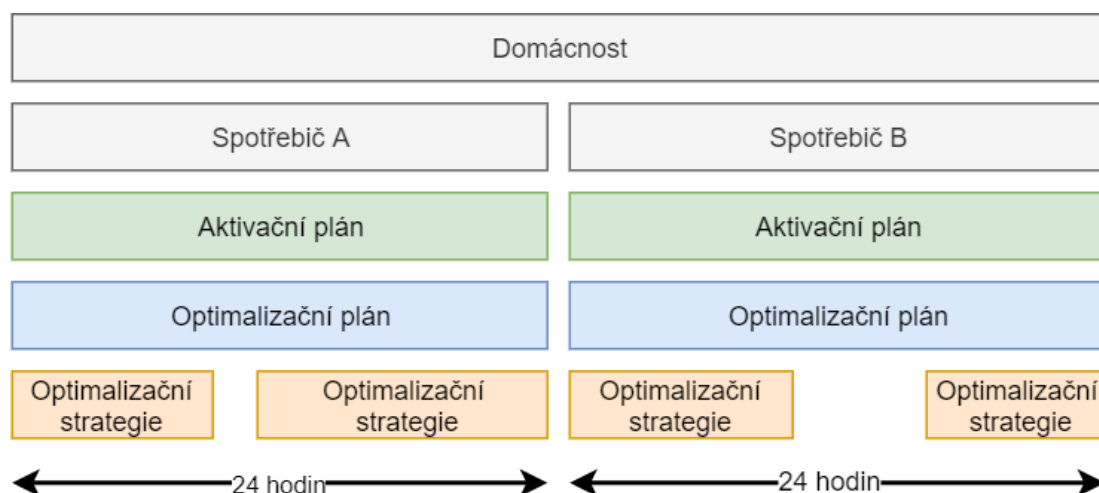
Obrázek 3.1. Architektura aplikace

Kapitola 4

Teorie optimalizace

Cílem algoritmu bude nabídnout uživateli takové množství flexibility a nastavení pro různé druhy spotřebičů, aby mohl optimálně využít příkon spotřebičů, limit hlavního jističe a uživatelské pohodlí. Tyto parametry mají mnohdy větší váhu než samostatné výdaje za elektřinu. Zároveň činí tuto úlohu netriviální. Jelikož už nebude možné použít například model: Vyber N nejlevnějších hodin a zapni topení, protože uživatel preferuje topení například v určitý čas, kdy je ale cena energie vysoká.

Nastavení algoritmu jako takového bude provádět uživatel, jelikož algoritmus nemůže odhadnout nekvantifikovatelné parametry, jako jsou pohodlí uživatele či topný model pro danou nemovitost. Z tohoto důvodu je lepší využít uživatelskou empirickou zkušenost a nechat uživatele nastavit parametry algoritmu dle jeho uvážení.



Obrázek 4.1. Vrstvy optimalizačního mechanismu

4.1 Denní data

Tato data reprezentují datové podklady pro výpočet aktivačního plánu a jsou získávány pomocí web scarpingu nebo z ročních přehledů z webových stránek regulátora trhu.

Jedná se o posloupnost 24 hodinových cen z denního trhu, které mohou začínat libovolnou hodinou, a to z důvodu zvýšení efektivity výpočtu optimalizace. Dělicí úsek mezi denními daty je doba, mezi optimalizačními strategiemi.

4.2 Optimalizační strategie

Optimalizační strategie je metoda výpočtu optimální topné doby na určité části optimalizačního plánu. Metoda bude vyhodnocovat zapnutí a vypnutí spotřebiče podle ceny elektřiny na denním trhu s elektřinou. Je uživatelsky definovaná.

Aktivační plán pro jeden spotřebič může obsahovat více nepřekrývajících aktivačních strategií různého druhu.

Optimalizační strategie musí umožňovat nastavit tyto vlastnosti:

- **Začátku a konce:** Vymezení na jakém úseku za den bude optimalizační algoritmus splňovat níže zmíněné vlastnosti.
- **Minimální dobu běhu:** Opakované zapínání a vypínání spotřebičů může snížit jejich životnost. Minimální doba jednoho aktivačního úseku tj. 15 minut nemusí být dostatečně dlouhá.
- **Celkovou délku běhu spotřebiče:** Jak dlouho musí spotřebič běžet

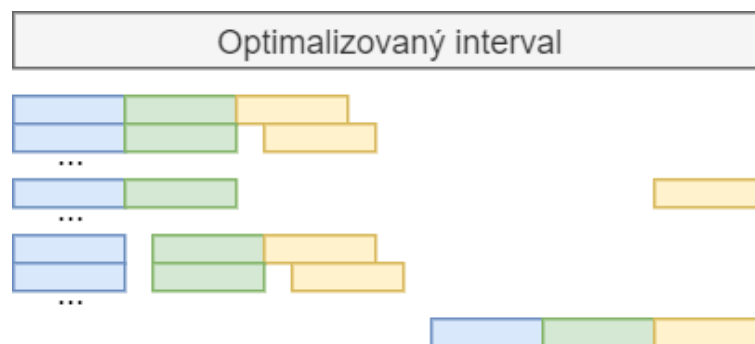
4.2.1 Block Optimizer

Je základní optimalizační strategií v celém systému. Slouží k poskládání několika minimálních dob zapnutí na jednom optimalizovaném intervalu tak, aby celková cena za spotřebovanou elektřinu byla nejmenší možná.

4.2.1.1 Optimalizační algoritmus

- **Délka bloku:** Délka bloku simuluje minimální dobu zapnutí spotřebiče
- **Počet bloků:** Počet bloků říká, kolik bloků má být na optimalizovaném intervalu.

Algoritmus postupně posouvá bloky (interval zapnutí spotřebiče) na optimalizovaném intervalu a tím vytvoří množinu všech možných pozic daných bloků. Při každé iteraci spočítá, zdali daná kombinace bloků má menší cenu než dosud nejlepší známá.



Obrázek 4.2. Optimalizační algoritmus

Složitost algoritmu, tj. počet pozic nutných k porovnání se dá spočítat pomocí vzorce s parametry: délka bloků l , počet bloků k , počet pozic n :

$$\binom{n - k * (l - 1)}{k}$$

Vzorec 2: Výpočet složitosti optimalizace pomocí BlockOptimizeru

Toto řešení není zcela efektivní. Je zde prostor pro další výzkum a optimalizaci algoritmu pomocí kombinatorické optimalizace. Dalším problémem tohoto algoritmu je, že celková doba zapnutí se rovná délce bloku vynásobené počtem bloků. Není možná nastavit jinou dobu běhu. Pro většinu aplikací však minimální délka bloku 15 minut zcela postačuje

4.2.2 Scheduled Optimizer

Nejjednodušší možná strategie zapnutí spotřebiče. Využívá se zejména tam, kde je nutné spustit spotřebič v určitý čas. Vymezuje pouze dobu zapnutí a vypnutí a nedochází zde k žádné optimalizaci nákladů na elektřinu. Jediný způsob, jak uživatel může ušetřit, je dát tuto strategii na lokální minimum v křivce průměrných denních hodinových cen.

4.3 Optimalizační plán

Optimalizační plán je soubor optimalizačních strategií pro konkrétní spotřebič rozprostřených na 24 hodinovém intervalu tak, aby se nepřekrývaly. Aktivační plán se vypočte sjednocením všech optimalizačních strategií.

Strategie	Interval	Počet Bloků	Délka Bloku (15 m)	Pozn.: délka zapnutí
BlockOptimizer	10:00 – 18:00	2	4	2 h
BlockOptimizer	21:30 – 22:45	1	2	30 m
Scheduled	07:30 – 09:15	Fixní zapnutí		1 h 45 m

Tabulka 4.1. Příklad nastavení optimalizačního plánu

4.3.1 Správné nastavení optimalizačního plánu

Algoritmus optimalizuje pouze interval strategie jako takové nikoliv nastavení strategií. Uživatel by měl dbát následujícího grafu, který popisuje průměrnou cenu elektřiny v jednotlivých hodinách v roce 2018 v ostatních letech je graf téměř identický. Algoritmus bude nejvíce efektivní, pokud co nejvíce spotřebičů poběží zhruba od 10 hodiny večer do 7 hodin ráno. Pokud je třeba aktivovat určité spotřebiče odpoledne využít lokálního minima mezi 12 až 15 hodinou.



Obrázek 4.3. Průměrná hodinová cena za rok 2018

Kapitola 5

Simulace

Simulace se jeví jako ideální způsob otestování výhodnosti optimalizace, jelikož nastavení optimalizace závisí na řadě nedeterministických parametrů jako například časové preference zapnutí spotřebiče s ohledem na potřeby uživatele a na průměrnou hodinovou cenu elektřiny během jednoho dne.

Cílem testování je ověřit, zdali se vyplatí spotřebiče zapínat podle tarifu A++, který je založen na cenách z denního trhu s elektřinou nebo plně využít VT a NT v klasických tarifech. Pro kvalitu simulace je třeba věrně popsat, jaké spotřebiče v jakých dobách a při jakých spotřebách reálně fungují v nějakém objektu, aby simulace odpovídala skutečnosti. Faktor času je důležitý, aby simulace nebyla ovlivněna byť velice výhodným, nicméně v mnoha případech uživatelsky nepohodlným nastavením zapnutí spotřebičů na noční hodiny, kdy je cena elektřiny nejlevnější.

Data pro definici simulace byla získána ze stávajícího nastavení zapnutí spotřebičů v penzionu U Veselých. Takto vybudovaná simulace má hned dvě výhody. První je kvalita získaných dat opřená o dlouhodobé zkušenosti majitele s optimalizací nastavení spínání spotřebičů. Druhou výhodou je velikost odběru. Penzion má násobně větší odběr elektrické energie než rodinný dům, což zvyšuje kvalitu výsledku simulace.

Přesto je velice těžké namodelovat provoz kompletního penzionu se všemi spotřebiči a jejich odběry ve správný čas. Proto byl model zjednodušen na předpoklad, že vytápění domu společně s přípravou teplé užitkové vody tvoří stěžejní část spotřeby elektrické energie, a tudíž zde by se měla tvořit největší úspora.

5.1 Struktura simulace

Simulace bude vystavěna jako porovnání dvou modelů. První bude založen na fixním zapínání spotřebičů podle VT a NT a druhý založen na využití optimalizačních strategií. Oba modely budou mít stejný odběr kilowatthodin, pouze jinak rozložený v čase. Tyto modely se pustí na běžných obchodních cenách respektive na cenách z denního trhu za poslední 4 roky. K porovnání obou modelů nám stačí porovnat pouze náklady na silovou elektřinu, jelikož ostatní fixní cenové položky jsou regulované, pro oba tarify stejné a uživatel je nemůže žádným způsobem ovlivnit ve svůj prospěch. Vedlejším efektem optimalizace by pak mohlo být snížení hodnoty hlavního jističe v případě vhodného časového rozložení spotřeby. Tato možnost ale není předmětem uvedené simulace.

5.2 Implementace simulace

Definice simulace je napsaná pomocí jazyka Kotlin DSL domain-specific-language¹. Tato vlastnost jazyka nám umožňuje rychle měnit strukturu definice konkrétní simulace.

Cenová data byla získána od operátora trhu OTE z ročních přehledů ve formátu XLSX². Následně byla převedena do formátu CSV, který je strojově zpracovatelný.

```

appliance("Water Heating - boiler 1", 2.5){
    summer{
        scheduled(t(22), t(1))
        scheduled(t(17), t(19))
    }

    winter{
        scheduled(t(22), t(1))
        scheduled(t(17), t(19))
    }
}

```

Obrázek 5.1. Simulace pro fixní tarif

```

appliance("Water Heating - boiler 1", 2.5){
    summer{
        block_optimizer(t(0), t(7), 6, 2)
        block_optimizer(t(13), t(17), 4, 2)
    }

    winter{
        block_optimizer(t(0), t(7), 4, 3)
        block_optimizer(t(13), t(17), 4, 2)
    }
}

```

Obrázek 5.2. Simulace pro denní trh

K zpřesnění výsledků je celá simulace rozdělena do tzv. season. Season je container na spotřebiče, který je spouští pouze v měsících definovaných ve výčtu. Winter je označení pro měsíce 1, 2, 3 summer je označení pro měsíce 7, 8, 9. Ostatní měsíce tvoří marginální část spotřeby elektrické energie a proto nejsou v simulaci obsaženy.

Výpočet celkových nákladů na silovou elektřinu je již jednoduchý. Stačí pro všechny dny v simulaci spočítat aktivační plán a vynásobit příslušnou hodinovou sazbu s odebranou elektřinou. Podrobný výpočet je popsán na straně 3.

Data pro nastavení této simulace pochází z penzionu U Veselých v Peci pod Sněžkou³. Uvedený rozbor platí pro horský penzion. Pro hotel v Praze nebo penzion v nížině je nutné parametry změnit.

¹ <https://kotlinlang.org/docs/reference/type-safe-builders.html>

² <https://www.ote-cr.cz/cs/statistika/rocni-zprava?date=2019-01-01>

³ <http://www.veselypenzion.cz/>

5.3 Modely

5.3.1 Denní trh

Spotřebič	Sezóna	Doby zapnutí	Příkon (kW)	Denní odběr (kWh)
Topení teplé čerpadlo	Zima	02:00 – 07:00 2x1 h 14:00 – 20:00 2x2 h 20:00 – 00:00 2x1 h	10	80
Tepelné voda teplé čerpadlo	Léto	00:00 – 04:00 1x1,5 h 13:00 – 16:00 1x1,5 h 17:00 – 19:00 1x0,5 h	5	17,5
	Zima	00:00 – 04:00 3x0,5 h 13:00 – 16:00 2x0,75 h 17:00 – 19:00 1x1 h 20:00 – 00:00 2x1 h		30
Boiler 1, 2	Léto	00:00 – 07:00 2x1,5 h 00:00 – 00:00 2x1 h	2,5	2x12,5
	Zima	00:00 – 07:00 3x1 h 00:00 – 00:00 2x1 h		
Akumulační kamna 1, 2	Zima	19:00 – 23:00 3x0,75 h	2 x 3	9
Pračka, sušička	Zima	18:00 – 22:00 1x2 h 00:00 – 07:00 1x2 h	2	10
	Léto	18:00 – 22:00 1x2 h 00:00 – 07:00 1x2 h		
Myčka	Zima	19:00 – 07:00 1x2 h 00:00 – 07:00 1x2 h	2	8
	Léto	19:00 – 07:00 1x2 h 00:00 – 07:00 1x2 h		
Wellness	Zima	15:30 – 18:00 19:00 – 21:00	6	27
Kuchyň	Zima	16:00 – 18:30	5	7,5
	Léto	17:00 – 18:00		12,5

Tabulka 5.1. Spouštěcí harmonogram pro denní trh

5.3.2 Fixní tarif

Spotřebič	Sezóna	Doby zapnutí	Příkon (kW)	Denní odběr (kWh)
Topení teplené čerpadlo	Zima	16:00 – 19:00 22:00 – 00:00 01:00 – 04:00	10	80
Tepelné voda teplené čerpadlo	Léto	02:30 – 04:00 17:00 – 18:30 22:00 – 22:30	5	17,5
	Zima	02:00 – 04:00 14:00 – 16:00 17:00 – 18:00 22:00 – 23:00		30
Boiler 1, 2	Léto	22:00 – 01:00 17:00 – 19:00	2,5	2x12,5
	Zima	22:00 – 01:00 17:00 – 19:00		
Akumulační kamna 1, 2	Zima	17:00 – 20:00	2 x 3	2 x 9
Pračka, sušička	Léto	17:00 – 19:30 22:00 – 00:30	2	10
	Zima	17:00 – 19:30 22:00 – 00:30		
Myčka	Zima	22:00 – 00:00 10:00 – 12:00	2	8
	Léto	22:00 – 00:00 10:00 – 12:00		
Wellness	Zima	15:30 – 18:00 19:00 – 21:00	6	27
Kuchyň	Zima	16:00 – 18:30	5	7,5
	Léto	17:00 – 18:00		12,5

Tabulka 5.2. Spouštěcí harmonogram pro fixní tarif

5.4 Srovnání obou simulací

Výsledky jsou souhrnně uvedeny v následujících tabulkách. Při porovnání výsledků je nutné si však uvědomit několik omezení. Tato práce počítá pouze s daty do roku 2019, přičemž právě rok 2019 se z hlediska vývoje cen zdá být nejzajímavější. Další omezení, které limituje vypovídající schopnost je simulace příkonu jednotlivých spotřebičů. Ta je v tomto případě navržena pro konkrétní penzion a to na základě dlouhodobých zkušeností s provozem. Proto je nepřenositelná a pro každé jiné odběrné místo by se musela provést opět detailní analýza potřeb. Tabulky ukazují dva možné scénáře. První je pro nekatalogový tarif, který byl dohodnut pro odběr u Penzionu u Veselých. Tento tarif je výhodnější z důvodu vyššího odběru elektrické energie. Druhá část porovnání obsahuje katalogové tarify.

Rok	VT (Kč/kWh)	NT(Kč/kWh)	Průměr h. cen za rok VT (Kč/kWh)	NT - čas
2015	1.511	0.895	0.88	<17, 19>, <22, 04>
2016	1.511	0.895	0.84	<17, 19>, <22, 04>
2017	1.511	0.895	0.96	<17, 19>, <22, 04>
2018	1.828	1.082	1.18	<17, 19>, <22, 04>

Tabulka 5.3. Ceny elektřiny podle veřejného ceníku ČEZ [7], [8], [9], [10]

Rok	Simulace deního trhu	Simulace fixního tarfu (D26d)
2015	21411.78 Kč	27237.78 Kč
2016	18687.28 Kč	27309.11 Kč
2017	23102.42 Kč	27237.78 Kč
2018	30740.76 Kč	32938.10 Kč

Tabulka 5.4. Simulace pro penzion pro ceny z tabulky 5.3

Rok	VT (Kč/kWh)	NT(Kč/kWh)	Průměr h. cen za rok VT (Kč/kWh)	NT - čas
2016	1.135	0.730	0.84	<17, 19>, <22, 04>
2017	1.135	0.730	0.96	<17, 19>, <22, 04>
2018	1.828	1.082	1.18	<17, 19>, <22, 04>

Tabulka 5.5. Dohodnutá cena elektřiny pro penzion U Veselých

Rok	Simulace deního trhu	Simulace fixního tarfu (D26d)
2016	18687.28 Kč	21572.44 Kč
2017	23102.42 Kč	21515.92 Kč
2018	30740.76 Kč	32938.10 Kč

Tabulka 5.6. Simulace pro penzion pro ceny z tabulky 5.5



Tabulka 5.7. Vývoj ceny elektřiny [11]

K lepšímu pochopení výsledků simulace je třeba ještě zohlednit vývoj spotových cen elektřiny znázorněný na obrázku 5.7. V tabulce 5.4. je uvedeno srovnání simulace a ceny při použití nejpoužívanějšího tarifu D26d firmy ČEZ. Na první pohled je zřejmé, že při použití denního trhu jsou výsledky vyjádřené v korunách při stejné spotřebě lepší. V letech 2015-2017, kdy se spotová cena držela zhruba na stejné úrovni, případně jen mírně stoupala, jsou výsledky vždy lepší. Zlom nastává v roce 2018, kdy spotová cena začala strmě stoupat. Turbulence ceny zejména v denních hodinách způsobila, že spotová elektřina byla výrazně dražší a tím dlouhodobě znevýhodnila tento způsob odběru. Proto by bylo zajímavé srovnání vývoje zejména v roce 2019 a počátku roku 2020, kdy se trend v druhé polovině roku 2019 obrátil a cena začala opět klesat. Lze jen dovodit, že výsledky by byly opět mnohem příznivější ve prospěch využití spotových cen. V tabulce 5.6., kde je totéž srovnání ovšem pro výhodnější tarifní ceny vyjednané se společností Bohemia Energy, je vidět shodný trend, nicméně výhodnost využití spotových cen je nižší. Ze všech tabulek je pak vidět skok v cenách v době, kdy dochází ke změnám cen tarifu. Při obecně rostoucích cenách elektřiny je tarif vždy nejvýhodnější ve své závěrečné fázi. V případě poklesu cen však platí opak. Menší výhodnost spotových cen v penzionu je též dána faktem, že majitel se snažil relativně jednoduchými způsoby převést podstatnou část odběrů do hodin s nízkým tarifem a tak jej využíval na maximum. Pokud by nebyla tato optimalizace, byly by souhrnné ceny ve fixním tarifu ve výsledku vyšší a výhoda spotových cen by byla patrně zřejmější.

5.5 Obecné závěry

- Použití spotových cen má smysl zejména v objektech, kde lze odložit zapnutí některých spotřebičů na dobu s výhodnější okamžitou cenou (typicky topení, bojler, pračky, myčky).
- Spotové ceny jsou také výhodnější v období s vyšší spotřebou elektřiny, typicky např. topení zimě.
- Pro domácnosti je systém výhodnější neboť vzhledem k celkové spotřebě a tím i vyjednávací síle nejsou schopné dosáhnout na lepší cenové podmínky v tarifech.
- Pokud je v objektu již nějakým způsobem optimalizováno rozložení spotřeby ve prospěch nízkého tarifu, výhodnost použití spotových cen klesá.
- Spotové ceny jsou obecně výhodnější v prostředí klesajících, stálých či případně lehce stoupajících cen. Ve všech těchto případech jsou amplitudy výchylek dolů zajímavé a systém snadno nalézá výhodnější ceny. V prostředí strmě stoupajících cen platí samozřejmě opak.

Kapitola 6

Centrální jednotka

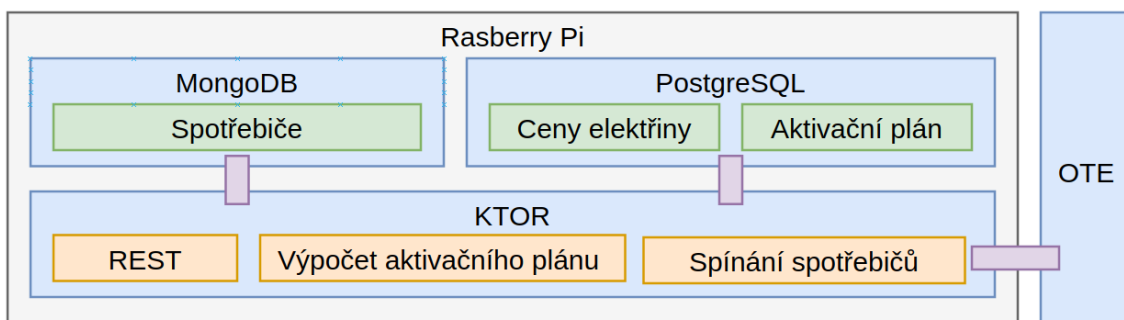
Centrální jednotka se stará o všechny výpočty optimalizace a signalizaci zapnutí připojených spotřebičů. Bude mít na starost dvě časově nezávislé operace. První je výpočet aktivačních plánů pro konkrétní spotřebiče. Druhou činností bude spínání jednotlivých spotřebičů podle aktivačního plánu, každých 15 minut.

6.1 Hardware

Je nainstalován na platformě Raspberry Pi, jež je vhodná pro podobné činnosti, jelikož se jedná o plnohodnotný počítač s operačním systémem, který nám ovšem dovoluje jednoduše připojit hardwarové periferie.

6.2 Architektura

Architektura se skládá z tří hlavních částí. Dva datábázové moduly a to MongoDB pro ukládání informací o spotřebičích a PostgreSQL pro ukládání cen elektřiny a aktivačního plánu. Dalším modulem je Ktor webový server, který sprostředkovává komunikaci mezi aplikací a ostatními moduly a dále se stará o získávání cen elektřiny z webu OTE, Výpočítávání aktivačních plánů a spínání spotřebičů.



Obrázek 6.1. Architektura centrální jednotky

6.2.1 Webový server

„Ktor is a framework built by JetBrains for creating Web applications in Kotlin, making use of coroutines for high scalability and offering an easy-to-use and idiomatic API.“[12]

Jako webový server pro Rozhraní REST byl vybrán Ktor. Byl vybrán proto, že je přímo vyvinut pro programovací jazyk Kotlin a díky vlastnostem jazyka je celý proces generování odpovědi na request asynchronní. Je modulární a ke svému běhu nepotřebuje velké zdroje, což je ideální pro jeho nasazení na Raspberry Pi.

■ 6.2.2 Persistence dat

K ukládání dat byly zvoleny dvě databáze a to MongoDB a PostgreSQL díky svým rozdílným vlastnostem, které se dají ideálně kombinovat pro náš usecase.

■ 6.2.2.1 MongoDB

„Schema Flexibility - SQL, the language used in relational database querying, isn't a great fit for working with time series data. The fixed schema requirement does not allow for the flexibility and agility that modern systems require. For example, adding a new sensor type to an existing application may require table schema changes that result in application downtime or complex multi-resource coordination. ... Storing IoT data in JSON format allows for the most flexibility and having a database system that natively understands and works with JSON data is essential to the success of an IoT project.“[13]

MongoDB je NoSQL databáze, která slouží pro ukládání stromových dokumentů podobných jako je formát JSON. To je pro nás ideální způsob, jak ukládat veškeré informace o spotřebičích pohromadě. Entita spotřebiče se skládá z: optimalizačních plánů, aktivačního plánu, interface pro komunikaci se spínací jednotkou, jména spotřebiče a dalších elementů. Přístup k těmto datům je velice efektivní, jelikož vše je uloženo na jednom místě. Porovnejme podobný způsob uložení dat nicméně v SQL databázi. Ta by musela obsahovat několik tabulek a při každém požadavku na informace k spotřebiči by muselo probíhat jejich spojování, aby byl získán kompletní obraz spotřebiče. Aplikace pro své fungování nejčastěji potřebuje právě kompletní obraz dat.

REST endpointy kopírují entity v mongo DB, dokonce využívají stejné kotlin data classy. Je to značná výhoda, jelikož ve většině případů GET požadavků na server jsou pouze zobrazením databáze.

Centrální jednotka používá KMongo tj. Kotlin nadstavba nad Mongo Java drivery. Tato nadstavba mapuje Mongo BSON objekty přímo na naše datové entity ze sdílené knihovny. Více v kapitole REST.

■ 6.2.2.2 PostgreSQL

PostgreSQL je tabulková relační databáze, která se ideálně hodí pro uložení strukturovaných řádkových dat. Takovýto přístup je využit pro ukládání dat z denního trhu s elektřinou.

■ 6.2.2.3 Aktivační plán

Aktivační plán se ukládá jako tabulka časů, které mohou začínat každou 0, 15, 30, 45 minutu v hodině, a akce pro spínací jednotku. Pro snížení počtu záznamů se spínací akce OFF neukládá. Pokud není záznam v databázi, jedná se vždy o akci OFF.

Data z aktivačního plánu jsou používána nejvíce jednak ke spouštění spínacích jednotek, ale i k zobrazování stavu spotřebiče v mobilní aplikaci. V tomto případě se ideálně doplňují obě databáze. PostgreSQL obsahuje kompletní aktivační plán a dokument spotřebiče a v MongoSQL je pouze aktivační plán za poslední tři dny a jeden následující. To nám stačí k zobrazování nedávné historie a zobrazení výhledu spouštění na další den, pokud již byl aktivační plán vypočítán.

■ 6.2.3 REST

Komunikaci s aplikací zajišťuje rozhraní REST. Celá komunikace je postavena na sdílené knihovně datových entit. Ty jsou serializovatelné pomocí technologie kotlinx serialization. Komunikace díky této technologii probíhá v abstraktnější rovinách, programátor neřešení implementační detaily přenosu informace. Naprostá většina endpointů

vrací nebo přijímá právě objekty ze sdílené knihovny. Proto je práce s RESTem velice jednoduchá, stačí pouze znát příslušný endpoint a datovou entitu, kterou vrací. Zbytek nám statická nápověda jazyka nabídne.

```
@Serializable
data class Appliance(
    @SerializedName("_id")
    @ContextualSerialization
    var id: Id<Appliance>,
    var display_name: String,
    var appliance_interface: PowerInterface,
    val optimization_plans: MutableList<OptimizationPlan>,
    var current_optimization_plan: Int,
    var partial_power_plan: PowerPlan,
    var avg_kwh_consumption: Double = Double.NaN
)
```

Obrázek 6.2. Datová entity

```
{
  "_id": "5e931797e4cf696513b4e817",
  "display_name": "Appliance - 5",
  "appliance_interface": { "type": "not_set" },
  "optimization_plans": [
    {
      "name": "Winter - 8",
      "from": "00:00",
      "to": "00:00",
      "optimizers": [
        { "type": "scheduled_optimizer", "from": "05:00", "to": "09:45" }
      ]
    }
  ],
  "current_optimization_plan": 0,
  "partial_power_plan": {
    "from": "2020-04-10T00:00",
    "to": "2020-04-14T00:00",
    "plan": [ { "time": "2020-04-10T05:30", "action": "ON" } ] }
}
```

Obrázek 6.3. JSON reprezentace datové entity

„Kotlin serialization consists of a compiler plugin, that generates visitor code for serializable classes, runtime library with core serialization API and JSON format, and support libraries with ProtoBuf, CBOR and properties formats. Supports Kotlin classes marked as @Serializable and standard collections. Provides JSON, CBOR, and Protobuf formats. Complete multiplatform support: JVM, JS and Nativ“[14]

Kotlinx serialization je plugin do kotlin kompilátoru, který nám umožňuje nativně parsovat datové zprávy různých formátů, jako jsou například JSON nebo Protobuf, na objekty programovacího jazyka. Děje se tak bez použití reflexe a každá entita má svůj serializátor přímo vytvořený kompilátorem na míru. Serializaci a Deserializaci objektu programátor nemusí vůbec řešit a je velmi efektivní.

e.

6.2.4 Rozhraní

Mobilní aplikace slouží pouze k nastavování spotřebičů a optimalizačního a aktivačního plánu, a proto ke svému běhu nepotřebuje mnoho endpointů. K nastavení spotřebiče se využívá zejména endpoint POST appliance, který obstarává jak vlastní spotřebič tak i úpravu aktivačního plánu.

Z hlediska mobilní aplikace se spotřebič vždy upravuje jako celek nikoli po částech.

Aktivační plán se tedy upravuje ve stejném požadavku jako vlastní spotřebič. JSON objekt spotřebiče obsahuje výšeč aktivačního plánu. Pokud přijde

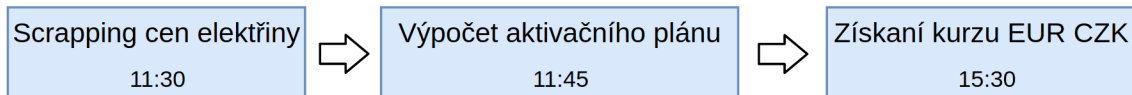
Rozhraní	URL	Popis	Volitelné parametry
Modul spotřebiče			
GET_appliances	/	Získá všechny spotřebiče	from, to: interval aktivačního plánu
GET_appliance	/{id}/	Získá konkrétní spotřebič	from, to: interval aktivačního plánu
POST_appliance	/{id}/	Vytvoří, upraví spotřebič	
DELETE_appliance	/{id}/	Smaže spotřebič	
Modul cen			
GET_price	/{from}/ {to}/	Ceny elektřiny z centrální j.	

Obrázek 6.4. Rozhraní REST

6.3 Výpočet aktivačního plánu

Aktivační plán představuje rozvrh zapínání a vypínání konkrétního spotřebiče během 24 hodin. Je vypočítán na základě optimalizačního plánu. Mezní dobu přechodu mezi dvěma plány lze nastavit pro lepší definování optimalizačních strategií. Plán je rozdělen do 15minutových úseků. Tato hodnota byla stanovena jako kompromis mezi délkou výpočtu optimalizace a možností zapínání spotřebičů v libovolný čas a návrhu architektury.

Plán bude obsahovat dvojice hodnot a to čas a akci. Akce je reprezentována hodnotami: ON, OFF.



Obrázek 6.5. Proces výpočtu aktivačního plánu

Každý den pro každý spotřebič je nutné vypočítat aktivační plán. Tuto činnost je možné naplánovat jednorázově v intervalu od 11:00 do 11:00 dalšího dne a to z důvodu zveřejnění dat z denního trhu v 11:00 předchozího dne.

6.2.4.1 Práce s časem

Ve sdílené knihovně existuje služba PowerPlanManager. Ta se stará o veškerou manipulaci s aktivačním plánem. Ke svému chodu potřebuje znát ceny elektřiny na příslušném intervalu. Výpočet optimalizace vyžaduje častý random access read cen elektřiny. Výpočet založený na časových knihovnách jako je například Java time od JDK 8, 9 by byl velice komplikovaný. PowerPlanManager převede aktivační plán a k němu příslušné ceny elektřiny z denního trhu do dvou polí. Takto uložená data jsou indexovatelná a jednoduše přístupná. Vlastní optimalizační algoritmus například BolockOptimizer poté nepracuje s časem ale pouze s indexy a celý výpočet i algoritmus je čitelnější.

Tato služba obsahuje i několik nástrojů a algoritmů pro zobrazování časového plánu. Například funkce cluster spojí všechny časy spuštění tj. 15 minutové intervaly do skupin.

6.4 Spouštění spotřebičů

Každých 15 minut centrální optimalizační počítač vyšle signalizaci podle aktivačního plánu do každé spínací jednotky, které reprezentuje jednotlivý spotřebič, zdali se má spotřebič zapnout či vypnout. Tato služba běží vedle Ktor web serveru na Raspberry Pi. Architektura datového modulu nám tuto práci usnadňuje, jelikož aktuální část datového plánu je uložena spolu s interfacem spínací jednotky. Jedním databázovým query získáme všechna potřebná data.

Spínací jednotky, které jsou přímo napojeny na Centrální jednotku jsou spuštěny přímo tj. zapnutím příslušný GPIO portu. Bohužel JAVA Knihovna pro práci s GPIO porty Raspberry Pi je velice nestabilní, proto zapnutí probíhá spuštěním vlastního Python kódu umožňující jednoduché ovládání GPIO.

Vzdálené spínací jednotky jsou s centrální jednotkou propojené přes REST WiFi spojení, které jim zašle JSON zprávu, zdali se mají spustit či vypnout.

Kapitola 7

Mobilní aplikace

Jako uživatelské rozhraní byla zvolena mobilní aplikace díky přirozenému prolínání jednotlivých prvků a plynulosti procházení. Aplikace cílí spíše na menší zákazníky, není příliš vhodná jako řídicí systém do velkých chytrých budov, které mají komplexní způsob ovládání, jelikož musí obsluhovat velký počet senzorů a dalších periferních jednotek. V rámci ověření funkčnosti a spolehlivosti celého systému byla vytvořena aplikace pouze pro platformu Android.

7.1 Návrh UI UX

Aplikace slouží jako uživatelské rozhraní celého systému. Klade si za úkol informovat uživatele o stavu systému a případně v něm provádět změny. Jedná se o mobilní aplikaci a uživatelské rozhraní je tomu náležitě přizpůsobené.

7.1.1 Material Design

„Material is an adaptable system of guidelines, components, and tools that support the best practices of user interface design. Backed by open-source code, Material streamlines collaboration between designers and developers, and helps teams quickly build beautiful products.“[15]

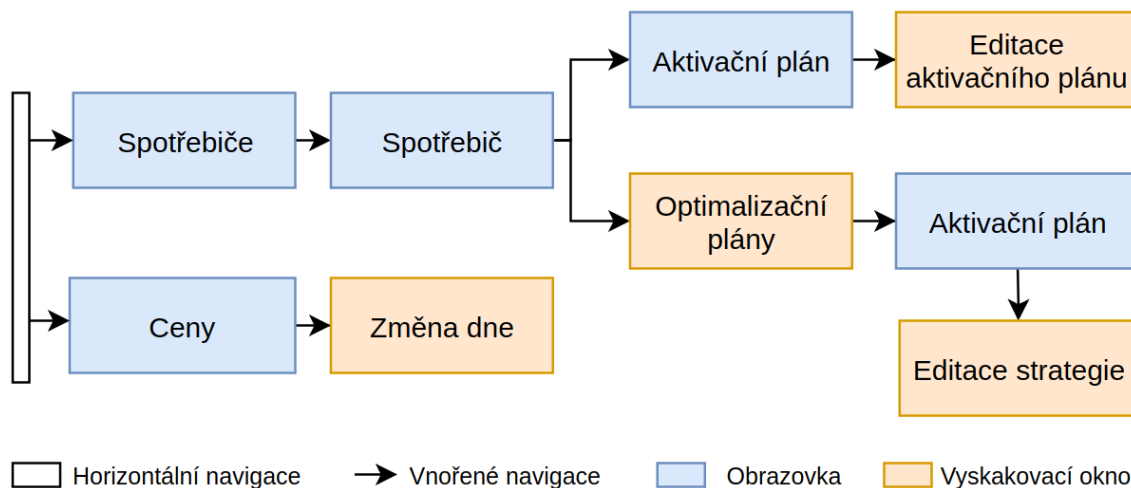
Material Design je soubor návrhů, pravidel a implementací definovaných společností Google pro návrh moderních uživatelských rozhraní. Jednotlivé elementy jsou již implementované pro nejrůznější technologie, jako jsou například web, Android-native, Android-Dart, IOS. Programátor se nemusí starat o jejich implementaci a vzájemné sladění prvků v návrhu. Vše je navrhnuté ve vzájemné jednotě.

Aplikace se snaží řídit těmito pravidly a používá tyto postupy pro lepší uživatelský zážitek. V neposlední řadě je to velké ušetření implementační práce a časové náročnosti celého projektu.

7.1.2 Navigace

Základním navigačním prvkem v aplikaci je horizontální navigace ve spodní části obrazovky. Tou může uživatel přejít na kartu spotřebičů, cen a statistik. Karta statistik je pouze ilustrační a bude rozvíjena v další iteraci toho projektu.

Navigace skrz úpravu spotřebiče se snaží modelovat strukturu dat. Více o konkrétních obrazovkách v následujících kapitolách.



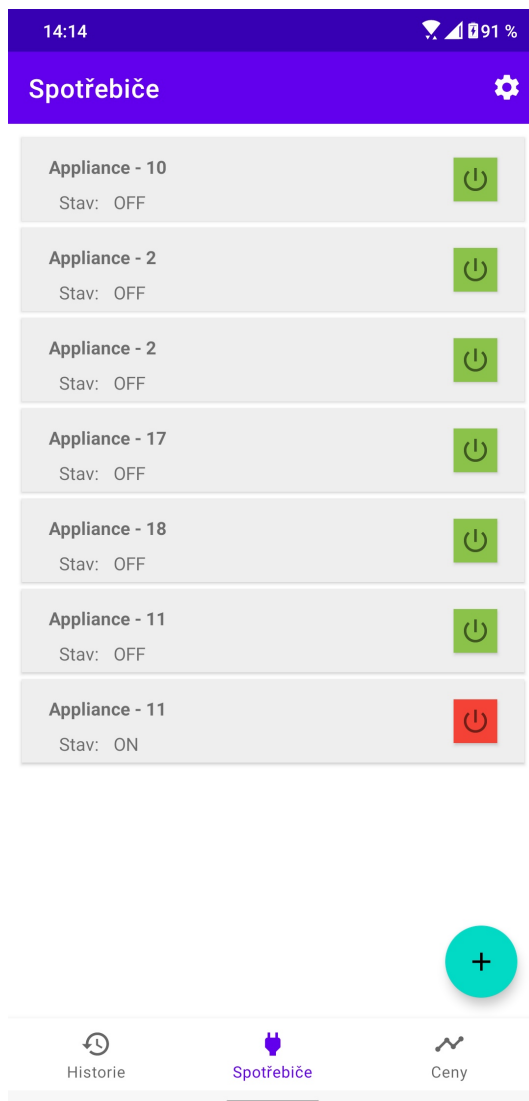
Obrázek 7.1. Navigace v mobilní aplikaci

7.1.3 Spotřebič

Na obrázcích x a y se nachází nejdůležitější část aplikace - přehled spotřebičů a nastavení spotřebiče.

Obrazovka Přehledu spotřebičů je velice jednoduchá a obsahuje pouze jejich seznam. Řádek v seznamu začíná titulkem názvu spotřebiče, pod kterým je drobným textem naznačený status spotřebiče. Na pravé straně je tlačítko okamžitého zapnutí spotřebiče, které nám rovněž ukazuje jeho stav. Informace o stavu je zde duplicitní, nicméně zvyšuje informativnost a v neposlední řadě i estetičnost celého designu. Po kliknutí na řádek je uživatel přeměřován na kartu spotřebiče.

Okamžitě po zobrazení není karta editovatelná. Uživatel musí kliknout na akční tlačítko se symbolem tužky v pravém dolním rohu. Karta spotřebiče začíná dvěma formulářovými poli pro zadání názvu rozhraní spínací jednotky. Následuje výšeč aktivačního plánu, které zobrazuje budoucí spuštění. U aktuálního optimalizačního plánu má Uživatel možnost si předdefinovat více optimalizačních plánů a rychle mezi nimi přepínat.



Obrázek 7.2. Obrazovka spotřebičů



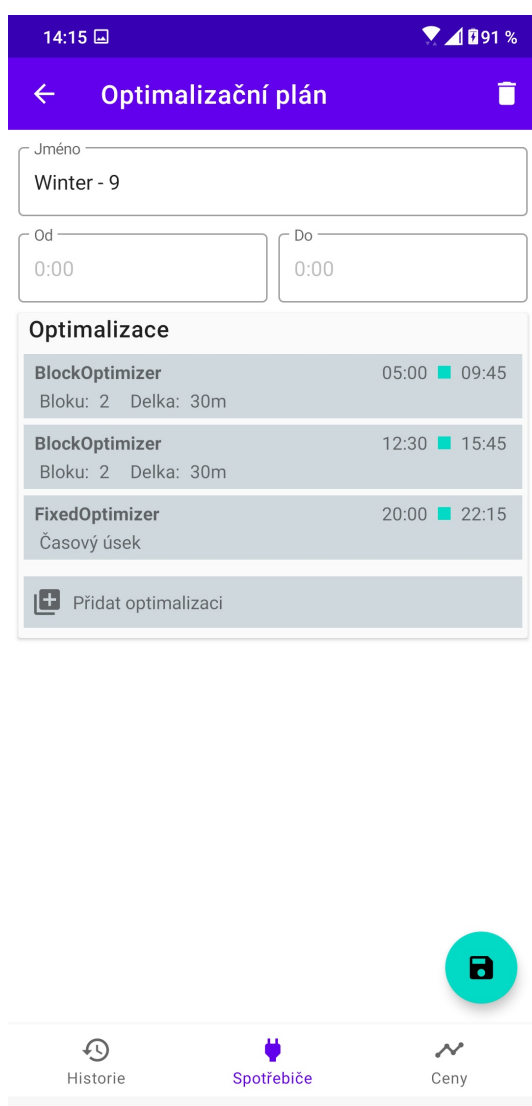
Obrázek 7.3. Obrazovka spotřebiče

7.1.4 Optimalizační plán

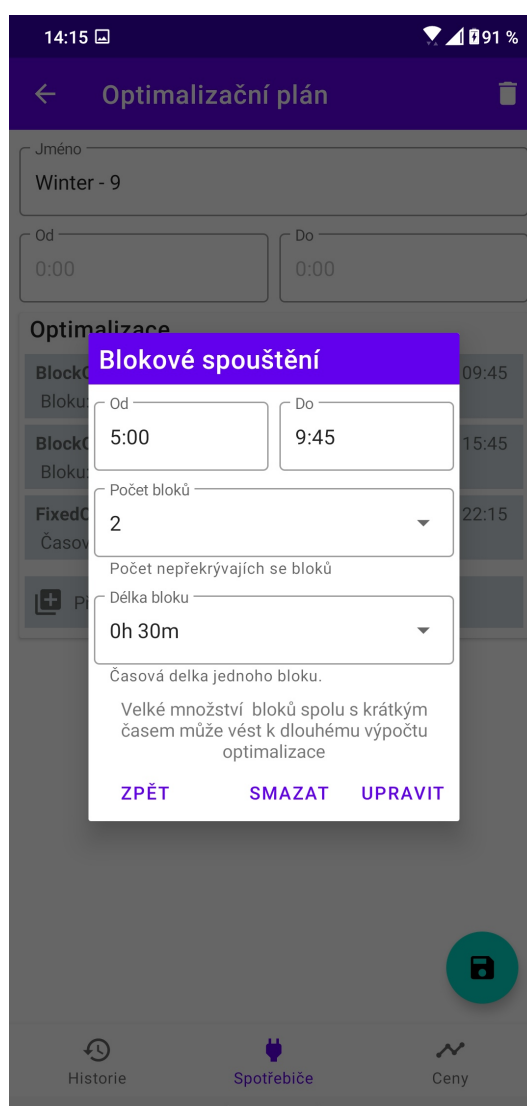
Nastavení optimalizačního plánu probíhá přes stejnojmennou obrazovku. Ta začíná třemi formulářovými poli. První obsahuje název plánu. Dalšími parametry jsou, kdy má plán začít, a kdy skončit, vždy se jedná o 24 hodinový úsek. Následuje karta vlastní optimalizace, která obsahuje jednotlivé optimalizační strategie. Zobrazené položky v seznamu vždy přizpůsobují svůj layout typu zobrazené strategie. Aplikace zatím využívá pouze dvě optimalizační strategie a to BlockOptimizer a ScheduledOptimizer. V levé části seznamu jsou zobrazeny konkrétní informace o optimalizační strategii a v pravé části je, na jakém bloku se má vykonat. Pod seznamem je externí řádek s ikonou se znaménkem plus pro přidání optimalizační strategie.

Uživatel má možnost optimalizační plán uložit akčním tlačítkem v pravém dolním rohu nebo smazat tlačítkem popelnice v TopAppBaru.

Na obrázku X je příklad dialogového okna pro editování BlockOptimizeru.



Obrázek 7.4. Obrazovka optimalizačního plánu



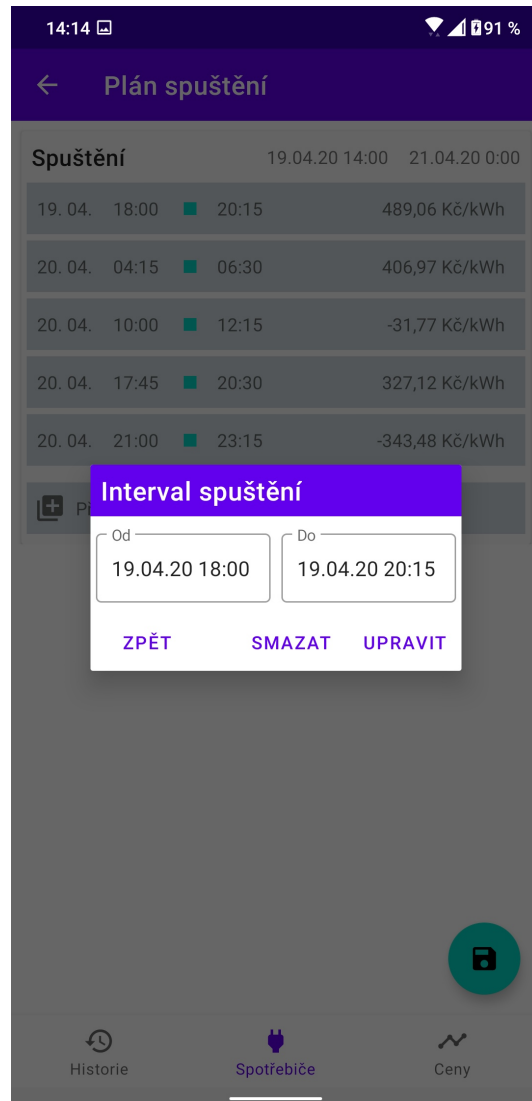
Obrázek 7.5. Obrazovka blokové strategie

7.1.5 Aktivační plán

Zobrazení aktivačního plánu je především k informativním účelům. Uživatel by měl primárně nastavit celou optimalizaci v optimalizačním plánu, který bude vytvářet aktivační plán. Nicméně v ojedinělých případech je třeba aktivační plán upravit. Aplikace dává možnost uživateli na den dopředu změnit plán spouštění podle okamžité potřeby. Aktivační plán se zobrazuje jako seznam úseků tj. sloučených 15 minutových intervalů běhu spotřebiče. Takovéto zobrazení zvyšuje přehlednost a čitelnost, v jakém stavu v konkrétní hodinu je daný spotřebič. Jednotlivé položky lze upravit nebo smazat, po úpravě se znovu přepočítá shlukování a vše je znovu přehledně zobrazeno.



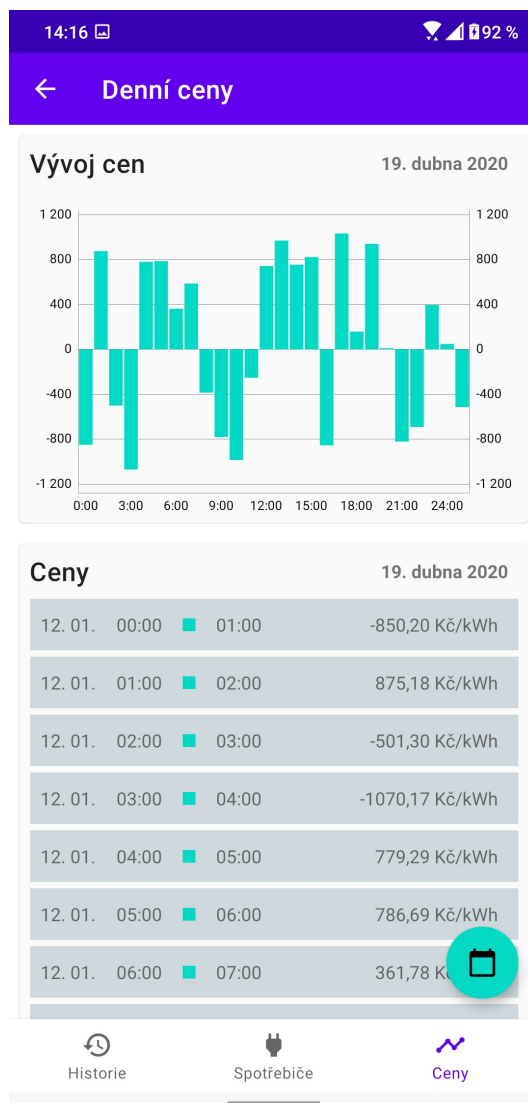
Obrázek 7.6. Obrazovka aktivačního plánu



Obrázek 7.7. Obrazovka úpravy aktivačního plánu

7.1.6 Přehled cen

Poslední důležitou obrazovkou je přehled cen elektřiny z denního trhu. Operátor trhu OTE má na svých stránkách přehlednou tabulku cen. Nicméně pro uživatelské pohodlí aplikace obsahuje obrazovku, kde jsou ceny zobrazeny ve 24 sloupcovém/hodinovém grafu, kde sloupeček znázorňuje cenu za jednu hodinu a megawattu elektřiny. Akční tlačítko lze využít pro navigování do historie cen za jiné dny.



Obrázek 7.8. Obrazovka přehledu cen el. energie

7.2 Architektura

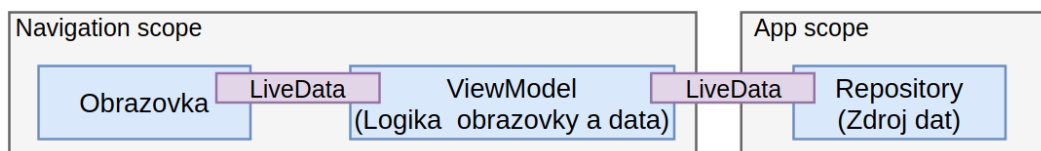
Mobilní aplikace slouží jako ovládací terminál celého systému. Předpokládá se, že celý systém musí být 24 hodin online a přístupný v domácí síti. Nehrozí zde tedy výpadky sítě nebo dlouhá doba odpovědi na požadavky. Na tomto předpokladu je celá architektura vystavena. Jedná se čistě o frontend neobsahující žádnou interní databázi. Aplikace si na začátku stáhne seznam všech spotřebičů, uloží si je do interní paměti a následně s nimi pracuje. Pokud aplikaci nějaká data chybí, například zobrazování historie cen elektřiny, sama si je stáhne. Návrh se snaží kopírovat nejnovější postupy doporučené společností Google, zejména soubor knihoven z Jetpacku.

„Jetpack is a suite of libraries, tools, and guidance to help developers write high-quality apps more easily. These components help you follow best practices, free you from writing boilerplate code, and simplify complex tasks, so you can focus on the code you care about.

Jetpack comprises the androidx.* package libraries, unbundled from the platform APIs. This means that it offers backward compatibility and is updated more frequently than the Android platform, making sure you always have access to the latest and greatest versions of the Jetpack components.“[16]

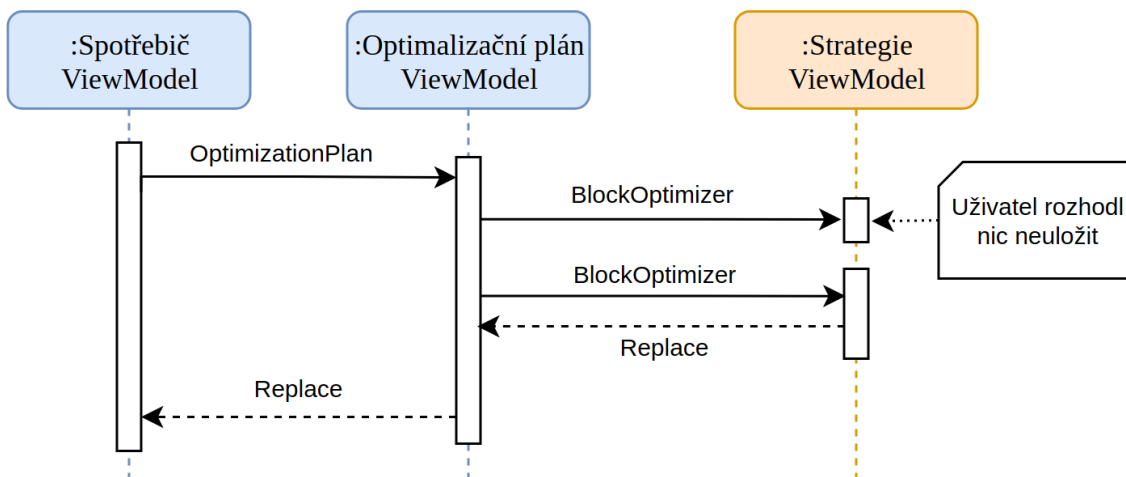
7.2.1 Decoupling

Architektura aplikace využívá nejnovější Androidx lifecycle knihovny, které usnadňují práci s spravováním chování aplikace při různých změnách stavů vyvolaných například navigací, změnou orientace nebo samotným androidem. Hlavní výhodou těchto knihoven je snaha o decoupling obrazovky, její logiky a aplikačních dat. každá obrazovka má k sobě navázaný svůj view model, který pro ní zajišťuje data a její logiku.



Obrázek 7.9. Příklad nastavení optimalizačního plánu

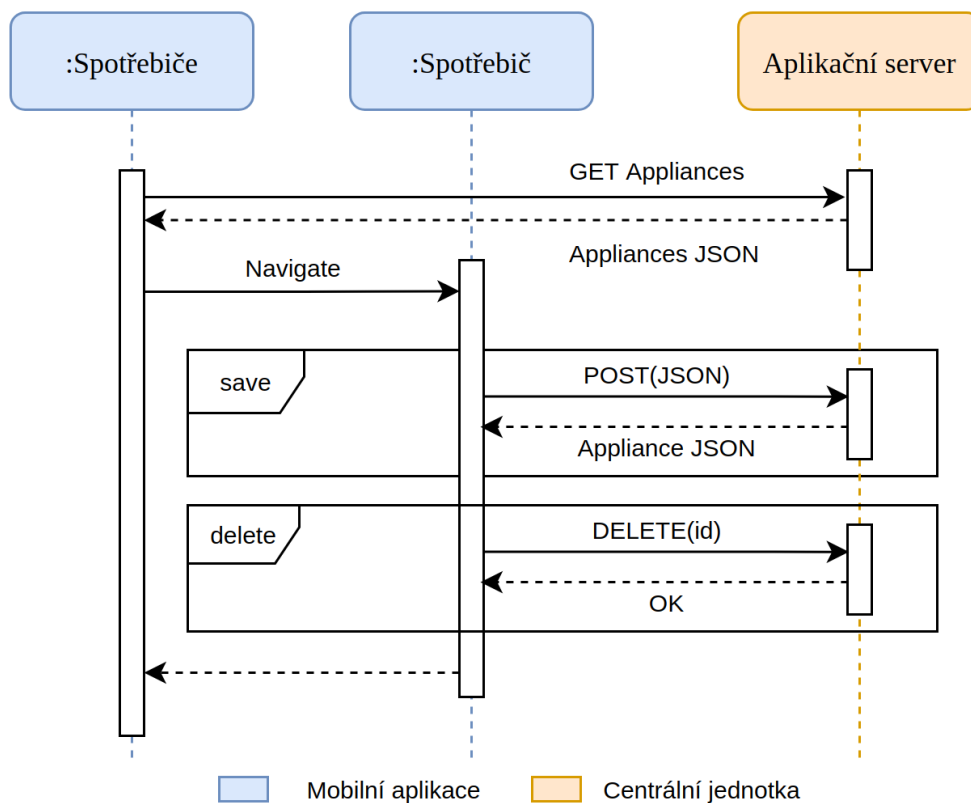
Viewmodel umožňuje změny konfigurace (rotace), nemusíme ho vytvářet znovu a můžeme si do něj ukládat aktuální stav obrazovky. Životnost viewmodelu můžeme navázat na zásobník prošlých obrazovek. To nám napomáhá k lepší správě historie. V různých obrazovkách můžeme získat viewmodely všech předchozích obrazovek (pokud je tak nadefinujeme) a upravovat jejich data. Tento postup je znázorněn na obrázku XY



Obrázek 7.10. Sekvenční diagram propojení na RES

7.2.2 Propojení na REST

Ke komunikaci s aplikačním serverem je použita knihovna Retrofit s rozšířením kotlix serialization. K definování Serverového API je používá anotovaný interface (url, parametry, ...), který Retrofit převezme a vytvoří z něj volatelné rozhraní. Na diagramu XY je znázorněna komunikace aplikace a serveru. Ta zahrnuje spuštění aplikace, stáhnutí spotřebičů, upravení spotřebiče a jeho následné smazání.



Obrázek 7.11. Propojení mobilní aplikace na REST

Kapitola 8

Prototyp hardwaru

Celý projekt bylo nutné také odzkoušet v praktické rovině, která umožní alespoň minimální nasazení v reálných podmínkách. Základní požadavky na hardwarové centrální jednotky a spínacích jednotek vycházely již z první analýzy a poté ze samotného softwaru.

- **Centrální jednotka:** Protože jsou zde používány databáze doplněné výkonným analytickým softwarem, bylo jasné, že tomu musí odpovídat i hardware. Složitost výpočtu optimalizace je blíže popsána na v kapitole 4.2.2. Dalším požadavkem bylo ovládní periférií přímo z centrální jednotky. K řízení systému nemůže být použito klasické PC, ale jednodeskový mikropočítač vybavený jak vlastním operačním systémem tak i IO porty.
- **Bezdrátové spínací jednotky:** K ovládní spínacích jednotek nám stačí jednoduchý mikrokontroler, který se umí připojit do sítě.

Na trhu je dostupných několik druhů těchto zařízení, ale zcela dominantní postavení zde mají mikropočítače platformy Raspberry Pi 4 případně doplněná jednodušší platformou Arduino. Hlavním důvodem pro použití těchto zařízení je kromě výše uvedených parametrů také relativně nízká cena, velká externí podpora a snadná dostupnost.

8.1 Platformy Raspberry Pi 4B

- Procesor: Broadcom BCM2711, 1.5 GHz quad-core ARM Cortex-A72 64-bit
- Paměť: LPDDR4 4GB
- Konektivita: 2,4GHz a 5GHz IEEE 802.11.b/g/n/ac Wi-Fi
- Bluetooth 5.0 (BLE)
- Gigabit Ethernet (1000 Mbit/s)
- GPIO Standardní GPIO header se 40 piny
- 2 x microHDMI 2.0 konektor (až 4Kp60)
- MIPI DSI konektor pro připojení displeje
- Micro SD slot pro načítání operačního systému a ukládání dat
- Oficiální operační systém: ARMové verze linuxových distribucí Debian a Arch

8.2 Arduino Uno Wifi rev 2

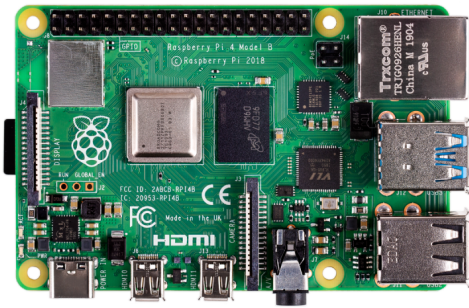
- Mikrokontroler ATMEGA4809
- Digital I/O porty 14
- Flash Memory 48 KB
- SRAM 6,144 B
- EEPROM 256 Bytes
- Clock Speed 16 MHz
- Délka 68 mm
- Šířka 54 mm

8.3 Prototyp spínání

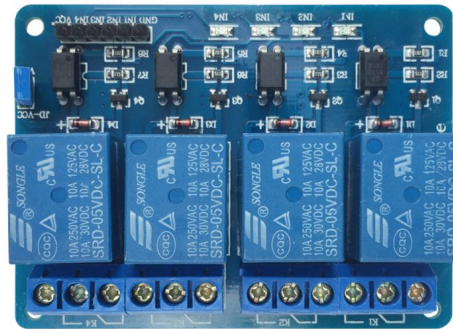
Jako výstupní periferie byl použit 4 kanálový reléový modul HW316. Modul obsahuje 4 relé s výstupním přepínacím kontaktem oddělené optočleny od ovládací části.

- 4-kanálové relé
- Odběr na kanál: 50-60mA
- Výstup: vysokonapěťové relé, AC250V 10A, DC30V 10A a přepínacím kontaktem
- LED indikace pro status výstupu.

Základní zapojení vychází z katalogového doporučení prodejce [17]. Stačilo velmi jednoduše propojit několika kabely IO port Raspberry s výstupní deskou a připojit demonstrační LED. Autor by rád podotknul, že není HW specialista jedná se pouze o prototyp.



Obrázek 8.1. RaspberryPi 4B, obrázek převzat z rpishop.cz[18]



Obrázek 8.2. Relé modul použitý pro spínání, obrázek převzat z arduino-shop.cz[19]

Kapitola 9

Implementační detaily

9.1 Modularizace projektu

Pro lepší správu a kompilaci je celý projekt rozdělen do několika částí. Všechny části jsou spravované build systémem Gradle a mohou se vzájemně používat, pokud to dává smysl. Modularizace projektu přispívá k jeho čitelnosti a snižování velikosti spustitelných výstupních souborů.

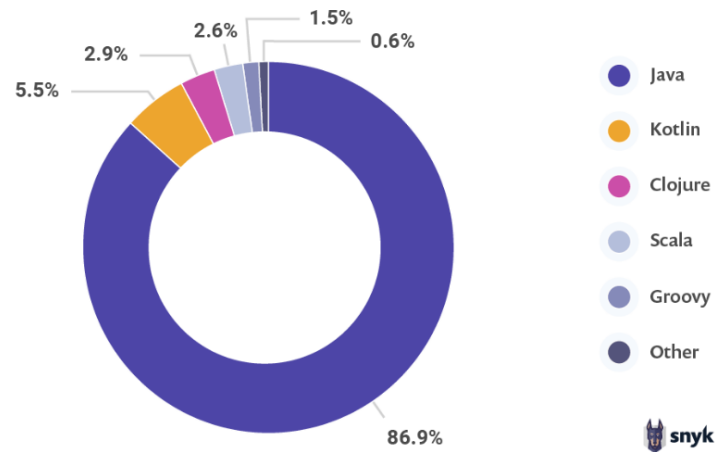
- **Common:** Jedná se o sdílenou knihovnu, ze které čerpají ostatní moduly. Jsou zde položky jako datové entity, parsování OTE dat, služby pro výpočet optimalizace a optimalizační strategie.
- **Server:** Implementace centrální jednotky a služby webscrapping dat a spouštění spotřebičů
- **Mobilní aplikace:** Implementace mobilní aplikace.
- **Simulace:** Simulace spotřebičů, pro ověření rentability celého systému.
- **Spínací jednotka:** Script pro spuštění spotřebiče na dálku programovaný pro Arduino

9.2 Jednotná technologie

Celá architektura systému byla navrhovaná pro programovací jazyk Kotlin. Díky jeho všestrannosti ho můžeme využít jen s několika výjimkami pro téměř celou implementaci. To představuje velkou výhodu, jelikož můžeme sdílet části kódu skrz celý projekt. Jedná se moderní programovací jazyk se vzrůstající oblíbeností. Je plně kompatibilní s Javou. Obsahuje mnoho syntaktických vylepšení, které zvyšují čitelnost. Další výhodou Kotlinu je velké množství oficiálně podporovaných knihoven, které tento projekt hojně využívá.

- Přehledná syntaxe
- Primární jazyk pro Android native
- Kotlinx.serialization: usnadnění práce s RESTem
- Ktor: asynchroní webový framework
- DSL pro přehlednou simulaci
- Gradle build files jsou nyní také psané v kotlinu
- Exposed ORM pro kotlin
- KMongo ORM pro kotlin a MongoDB

„However, despite the strong preference for Java, the use of other JVM languages grew as well. Particularly Kotlin, a language developed by JetBrains, gained a lot of popularity over the last couple of years. In fact, Kotlin grew from 2.4%, according to last year’s report, to an impressive 5.5%.“[20]



Obrázek 9.1. Graf rozložení JVM jazyků podle Průzkumu Snyk

9.3 Jednotný datový model

Cílem implementace bylo opětovné využití datových entit. Téměř jasnou volbou pro Kotlin bylo využití `kotlinx.serialization`. Společné datové entity uložené v modulu `common` jdou velice snadno serializovat mezi centrální jednotkou a mobilní aplikací. ORM pro MongoDB umí také pracovat s `kotlinx.serialization`, což vede k dalšímu snížení boiler plate kódu.

Kapitola 10

Testování

Celý systém má několik úrovní a částí, každé části odpovídá patřičný způsob testování. Hlavním cílem bylo otestovat korektnost počítání aktivačního plánu z optimalizačních strategií, aby byla zajištěna správnost simulace a z ní odvozených dat, na kterých závisí funkčnost celého řešení.

Modul	Unit testy	Integrační testy	Uživatelské testy
Common	Vysoké	-	-
Backend	Nízké	Vysoké	-
Mobilní aplikace	-	Střední	Vysoké

Obrázek 10.1. Plán testování

10.1 Jednotkové testování

Jednotkové testování aplikace bylo uplatněno především na business logiku. Jmenovitě nejvíce na optimalizační strategie a na jejich správu pomocí PowerPlanManageru. Jako testovací framework byl použit standardní JUnit 5. Nejvíce testovanou komponentou touto strategií je modul common, který je sdílený s dalšími moduly a téměř veškerá testovatelná business logika je v něm obsažena. Další podrobnosti o testování:

- **BlockOptimizerTest:** Stěžejní optimalizační strategie. Testování mezních bodů. Nejlepší cena na začátku, na konci, uprostřed, uprostřed s přerušením.
- **ScheduledOptimizerTest:** Testování mezních bodů.
- **PowerPlanManagerTest:** Tvorba a úprava aktivačního plánu. Testování zejména mezních bodů, setrů a operátorů, které manipulují s aktivačním plánem

10.2 Integrační testy

Integračními testy byl nejvíce testován backend, konkrétně REST API, celého systému. Integrační testy jsou velice užitečné, jelikož převážná část kódu backendu je pouze pro zajištění komunikací nebo připojení na databáze. Tyto části se špatně jednotkově testují a je nutné je otestovat komplexně přes integrační testy.

10.2.1 Ktor

Díky zvolenému webovému frameworku Ktor bylo integrační testování velice jednoduché. Ktor umí simulovat REST rozhraní pro integrační testy, jako by Ktor běžel přímo na webovém serveru.

```

@Test fun `create get`() = withTestApplication(Setup) {
    val appliance = Appliance.getEmpty().apply { display_name = "Test name" }

    with(handleRequest(HttpMethod.Post, "/appliances/")){
        setBody(JsonContext().stringify(Appliance.serializer(), appliance))
    }{
        assertEquals(HttpStatusCode.OK, response.status())
    }

    with(handleRequest(HttpMethod.Get, "/appliances/")) {
        assertEquals(JsonContext().stringify(Appliance.serializer().list, listOf(appliance)), response.content)
    }
}

```

Obrázek 10.2. Ukázka intergračního testu

Scénář	Postup
Vytvoření spotřebiče	<ol style="list-style-type: none"> 1. Přejít na kartu spotřebičů 2. Kliknout na akční tlačítko plus 3. Vyplnit údaje o spotřebiči 4. Uložit spotřebič 5. Zkontrolovat přítomnost přidaného psotřebiče na kartě spotřebičů
Upravení spotřebiče	<ol style="list-style-type: none"> 1. Zobrazit náhodný spotřebič 2. Kliknout na akční tlačítko úpravy 3. Upravit jméno a optimalizační plán 4. Uložit spotřebič 5. Zkontrolova stav uložených dat
Okamžité zapnutí spotřebiče	<ol style="list-style-type: none"> 1. Vybrat náhodný spotřebič v seznamu spotřebičů 2. Zapnout spotřebič rychlím zapnutím 3. Zkontrolovat aktivační plán spotřebiče
Smazání spotřebiče	<ol style="list-style-type: none"> 1. Vybrat náhodný spotřebič 2. Kliknout na tlačítko smzat 3. Zkontolovat smazání spotřebiče v seznamu spotřebičů

Tabulka 10.1. Základní přehled testovaných scénářů

10.3 Uživatelské testy

Uživatelskými testy byla pokryta zejména mobilní aplikace, jelikož se jedná pouze o frontend, který neobsahuje business logiku. Byl zde záměr použít automatizované uživatelské testy spolu s integračními testy naprogramované mocí knihovny Espresso. Bohužel tento záměr nebyl z časových důvodů realizován. Pokud se budu projekt dál rozvíjet, je určitě namístě na tento typ testů přejít.

10.4 Firebase

Aplikace je připravená na použití automatického sběru chyb pomocí Google Firebase v produkčním prostředí.

„Firebase Crashlytics is a lightweight, realtime crash reporter that helps you track, prioritize, and fix stability issues that erode your app quality. Crashlytics saves you troubleshooting time by intelligently grouping crashes and highlighting the circumstances that lead up to them.“[21]

10.5 Výsledek testování

Hlavním cílem testováním bylo ověřit funkčnost optimalizační části systému. A to z důvodu, že je hlavní částí pro teoretickou simulaci, ze které jsou získávána data pro další rozhodování o budoucnosti projektu.

Jednotkové testy BlockOptimizeru odhalily jednu zásadní chybu, které byla zanesena do kódu při jeho ladění pro testovací účely. Pomocí jednotkového testu byla chyba nalezena a posléze i odstraněna. Zbývající část testů proběhla v pořádku.

Integrační testy neodhalily žádnou chybu, přesto sloužily jako zpětná vazba pro chyby při nasazování systému.

10.6 Posouzení kvality UI UX

Posouzení kvality UI, UX Tomášem Veselý, product owner ve společnosti Etnetera a Košík:

Mobilní aplikace respektuje hlavní myšlenky systému a to jednoduchost a flexibilitu. Je patrné, že se snaží čistě plnit svojí úlohu a neobsahuje žádné další funkce, které by znesnadňovali přehlednost, ergonomie ovládaní a UX. Aplikace logicky strukturuje ovládaní do přehledných hierarchických bloků, mezi kterými má uživatel možnost navigovat a při každém kroku má možnost vrátit se a resetovat aktuální změny. To je uživatelsky přívětivé, neboť uživatel nemusí hned vědět, co vše je potřeba nastavit a takto na něj není kladen tlak okamžitého rozhodování při přesouvání v hierarchii, jelikož změny jde jednoduše vracet.

UI je jednotné skrz celou mobilní aplikace a na první pohled zde nejsou zjevné chyby v návrhu, které by odporovaly základním standardům a best practises. Aplikace se snaží respektovat zásady Material Designu.

Při krátkém zhodnocení se zdá, že aplikace uživateli nabízí optimální výčet nastavitelných vlastností pro spotřebiče, které zajistí funkčnost celého řešení, ale nematou uživatele velkou množinou parametrů a detailů. Pro optimální uživatelský zážitek a edukaci uživatele ohledně správného nastavení celého systému by bylo vhodné rozšířit nabídku statistik, přehledů a v neposlední řadě informací o správném nastavení celého systému přímo v aplikaci.

Kapitola 11

Další rozvoj projektu

11.1 Nasazení systému

Testovací simulace pro stěžejní části roku vykazovala ziskovost oproti fixním tarifům. Pokud se zásadně nezmění okolnosti projektu v následujícím roce bude následovat praktické nasazení systému v penzionu U Veselých v reálném prostředí, které jasně potvrdí, zdali se finálně vyplatí celý projekt realizovat a provozovat.

11.2 Další rozvoj

- **Oboarding:** Aplikace momentálně nemá žádný onboarding pro nastavení parametrů systému. Je třeba vyřešit prvotní přihlášení uživatele a nastavení celého systému případně obnovu do továrního nastavení.
- **Aktualizování:** Pro komerční nasazení je třeba doplnit jednoduché aktualizování systému, jelikož celý systém je distribuovaný a umístěný v lokálních sítích.
- **Návrh periferií:** Je třeba přepracovat fyzický design hardwarových periferií, aby propojení na spotřebiče zvládl i neznalý uživatel podle návodu. Bude třeba vylepšit prototyp spínání, aby umožňoval spouště i třífázové spotřebiče pod zátěží 20A.
- **Zabezpečení:** Celý systém byl v první fázi vyvinut pro použití v domácí síti a proto zatím neobsahuje širší možnosti zabezpečení komunikace a přihlášení.

11.3 Elektromobilita

V průběhu projektu se začala rozvíjet myšlenka využití stejné optimalizace spouštění pro nabíjení elektromobilů. Z křivky grafu vyplývá, že cena elektřiny je nejlevnější od 23:00 do 7:00. To je ideální doba pro nabíjení elektromobilů. Dle údajů na oficiálních stránkách společnosti Tesla je Model S schopen zvýšit svůj dojezd o 34 milí za hodinu nabíjení [22].

Například podle ceníku ČEZ pro rok 2019 [23] [24] [25] ceny nízké sazby surové elektřiny k tarifu elektromobilita ke dni 01. 10, 01. 03. 01. 11. jsou 1082,95 1777,01 a 1754,50 za MWh. Srovnáme-li je s průměrnou hodinovou cenou s denního trhu mezi 22:00 a 8:00 za celý rok, ta je 910 Kč, jedná se o značnou úsporu.

Kapitola 12

Závěr

V této práci byl teoreticky popsán mechanismus, jakým způsobem lze optimalizovat výdaje za elektřinu za použití velkoobchodních cen z denního trhu s elektřinou. Na základě této teorie byla implementována simulace podle reálné situace z penzionu U Veselých, která porovnávala relativní účtování cen fixně nebo pomocí tarifu Energie A++. Z této simulace vyplynulo, že při správném použití může dojít k úspoře nákladů za elektrickou energii. Tímto byla naplněna teoretická část zadání.

Následně navrhnutá koncepce využití tohoto systému pro domácí použití. Na základě této koncepce byl vytvořen ovládací systém a mobilní aplikace, přes kterou je celý systém kontrolován a nastavován. Při návrhu uživatelského rozhraní byl kladen důraz na jednoduchost ovládání a flexibilitu nastavení. Implementace byla otestována různými úrovněmi kontroly kvality, jmenovitě, jednotkové integrační a uživatelské testování. Testování odhalilo jednu kritickou chybu, které byla zanesena v průběhu vytváření testů, chyba byla na základě testu odstraněna.

Byl navrhnut prototyp hardwarových periférií na IOT platformách, který byl zkušebně zapojen na platformě Raspberry Pi, aby ověřil funkčnost celého zařízení v praktické rovině.

Na konci práce byla znovu formulována koncepce systému ve světle zjištěných výhod a úskalí systému a spolu s dalším rozvojem byla popsána v předposlední kapitole. Byla zde nastíněna myšlenka dalšího rozvoje systému pro nabíjení elektromobilů pomocí cen z denního trhu. Nabíjení probíhá zejména v nočních hodinách. Právě zde můžeme naplno využít průměrné rozložení ceny elektřiny podle grafu, který nám říká, že ceny elektřiny jsou nejnižší právě ve večerních a ranních hodinách a tím dále minimalizovat náklady.

Literatura

- [1] Shu Tay, Lee Te Chuan, A. Aziati a Ahmad Nur Aizat Ahmad. An Overview of Industry 4.0: Definition, Components, and Government Initiatives. *Journal of Advanced Research in Dynamical and Control Systems*. 2018, 10 14.
- [2] *Ceník Produktové řady ENERGIE A++*. Bohemia Energy entity s.r.o.. 2020.
https://www.reseni2plus.cz/wp-content/uploads/2016/08/BE-PD-EC-DO-AP-CZ_20200101-01.pdf.
- [3] *Use of energy explained - Energy use in homes*. U.S. Energy Information Administration. 2015.
<https://www.eia.gov/energyexplained/use-of-energy/homes.php>.
- [4] *Share of fuels in the final energy consumption in the residential sector by type of end-use, 2017*. Eurostat. 2017.
[https://ec.europa.eu/eurostat/statistics-explained/index.php?title=File:Share_of_fuels_in_the_use,_2017_\(%25\).png](https://ec.europa.eu/eurostat/statistics-explained/index.php?title=File:Share_of_fuels_in_the_use,_2017_(%25).png).
- [5] *Home Heating Systems*. Office of Energy Efficiency & Renewable. 2011.
<https://www.energy.gov/energysaver/heat-and-cool/home-heating-systems>.
- [6] H. Shareef, M. S. Ahmed, A. Mohamed a E. Al Hassan. Review on Home Energy Management System Considering Demand Responses, Smart Technologies, and Intelligent Controllers. *IEEE Access*. 2018, 6 24498-24509.
- [7] *Ceník produktové řady COMFORT 2015*. ČEZ, a. s.. 2015.
https://www.cez.cz/edee/content/file/produkty-a-sluzby/obcane-a-domacnosti/elektrina-2015/cez_cz_ele_cenikmoo_2015-01-01_silovka_comfort.pdf.
- [8] *Ceník produktové řady COMFORT 2016*. ČEZ, a. s.. 2016.
https://www.cez.cz/edee/content/file/produkty-a-sluzby/obcane-a-domacnosti/elektrina-2016/new_cez_cz_ele_cenikmoo_2016-31-10_silovka_comfort.pdf.
- [9] *Ceník produktové řady COMFORT 2017*. ČEZ, a. s.. 2017.
https://www.cez.cz/edee/content/file/produkty-a-sluzby/obcane-a-domacnosti/elektrina-2017/web_cez_cz_ele_cenikmoo_2017-1-1_comfort.pdf.
- [10] *Ceník elektřiny na 1 rok*. ČEZ, a. s.. 2018.
https://www.cez.cz/edee/content/file/produkty-a-sluzby/obcane-a-domacnosti/elektrina-2018/moo/web_cenik_elektrina_na_1_rok_moo_2018_2_cezdi.pdf.
- [11] *Elektrina - ceny a grafy elektřiny, vývoj ceny elektřiny 1 KWh - od 30.12.2015 do 01.05.2020 - měna CZK*. Kurzy.cz, spol. s r.o.. 2019.
https://www.kurzy.cz/komodity/cena-elektriny-graf-vyvoje-ceny/1kwh-czk-3-roky?dat_field=30.12.2015&dat_field2=01.05.2020.
- [12] *Using Kotlin for Server-side Development*. Kotlin Foundation. 2019.
<https://kotlinlang.org/docs/reference/server-overview.html>.
- [13] *IoT Reference Architecture*. MongoDB, Inc..
https://webassets.mongodb.com/iot_reference_architecture.pdf.

-
- [14] *Kotlin multiplatform / multi-format reflectionless serialization*. Kotlin Foundation. 2019.
<https://github.com/Kotlin/kotlinx.serialization>.
- [15] *Introduction to Material Design*. Google LLC.
<https://material.io/design/introduction>.
- [16] *Android Jetpack*. Google LLC. 2019.
<https://developer.android.com/jetpack>.
- [17] Robert Faltus. *Vícekanálový relé modul*. ECLIPSERA s.r.o..
<https://navody.arduino-shop.cz/zaciname-s-arduinem/spinaci-prvky-rele-tranzistory-a-tranzistorova-pole.html>.
- [18] *Raspberry Pi 4B*. ECLIPSERA s.r.o.. 2019.
<https://rpishop.cz/raspberry-pi-4b/1598-raspberry-pi-4-model-b-4gb-ram-765756931182.html>.
- [19] *Vícekanálový relé modul*. ECLIPSERA s.r.o.. 2019.
<https://arduino-shop.cz/arduino/2190-arduino-rele-4-kanaly-5v-s-optoclenem.html>.
- [20] *Kotlin overtakes Scala and Clojure, to become the 2nd most popular language on the JVM*. Snyk Ltd.. 2019.
<https://kotlinlang.org/>.
- [21] *Firebase Crashlytics*. Google LLC. 2019.
<https://firebase.google.com/docs/crashlytics>.
- [22] *Tesla Wall Connector*. TESLA. 2019.
<https://www.tesla.com/support/home-charging-installation/wall-connector>.
- [23] *Ceník Elektriny na jeden rok od 01. 01*. ČEZ, a. s.. 2020.
https://www.cez.cz/edee/content/file/produkty-a-sluzby/obcane-a-domacnosti/elektrina-2019/moo/web_new_cenik_elektrina_na_1_rok_moo_201811_vezdi.pdf.
- [24] *Ceník Elektriny na jeden rok od 01. 03*. ČEZ, a. s.. 2020.
https://www.cez.cz/edee/content/file/produkty-a-sluzby/obcane-a-domacnosti/elektrina-2019/moo/web_new_cenik_elektrina_na_1_rok_moo_20191_vezdi.pdf.
- [25] *Ceník Elektriny na jeden rok od 01. 11*. ČEZ, a. s.. 2020.
https://www.cez.cz/edee/content/file/produkty-a-sluzby/obcane-a-domacnosti/elektrina-2019/moo/web_cenik_elektrina_na-1-rok_moo_20199_vezdi.pdf.



Příloha **A**

Matematické vzorce

vzorec 1: Výpočet ceny elektřiny	5
vzorec 2: Počet kombinací k výpočtu optimalizace	9

Příloha B

Zadání práce



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Veselý** Jméno: **Jan** Osobní číslo: **474405**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Optimalizace nákladů na elektřinu pomocí IOT platformy

Název bakalářské práce anglicky:

Optimizing electricity costs through IOT platforms

Pokyny pro vypracování:

V rámci digitalizace a propojování dat díky průmyslu 4.0 cílem této práce je navrhnout software pro IOT platformu Rasbery Pi včetně uživatelského rozhraní, který na základě cen z denního trhu s elektřinou [1, 4], optimalizuje spouštění elektrických zařízení na následující den tak, aby došlo ke snížení nákladů při zachování pohodlí uživatele oproti nákladům, které by byly zaplacený při použití fixního tarifu. K ověření prospěšnosti návrhu bude vytvořena simulace, která porovná náklady účtované běžným fixním tarifem od společnosti ČEZ [6] a tarifem od společnosti Bohemie Energy Energie A++ [5] na reprezentativním vzorku elektrických zařízení (3 – 5 elektrických zařízení, které mají největší vliv na spotřebu).

Seznam doporučené literatury:

- [1] ATTILA, Kürthy. Obchod s přeshraničními kapacitami v Evropě. Praha, 2017. Magisterská práce. ČVUT. Vedoucí práce Krejcar Rostislav.
- [2] Home Heating Systems [online]. Washington: U.S. Department of Energy, 2011 [cit. 2020-01-03]. Dostupné z: <https://www.energy.gov/energysaver/heat-and-cool/home-heating-systems>
- [3] Share of fuels in the final energy consumption in the residential sector by type of end-use, 2017. Eurostat [online]. Eurostat, 2017 [cit. 2020-01-03]. Dostupné z: [https://ec.europa.eu/eurostat/statistics-explained/index.php?title=File:Share_of_fuels_in_the_final_energy_consumption_in_t_e_residential_sector_by_type_of_end-use,_2017_\(%25\).png](https://ec.europa.eu/eurostat/statistics-explained/index.php?title=File:Share_of_fuels_in_the_final_energy_consumption_in_t_e_residential_sector_by_type_of_end-use,_2017_(%25).png)
- [4] Krátkodobé trhy – Denní trh [online]. OTE: OTE, 2020 [cit. 2020-01-03]. Dostupné z: <https://www.ote-cr.cz/cs/kratkodobe-trhy/elektrina/denni-trh?date=2020-01-04>
- [5] Ceník Produktové řady ENERGIE A++ [online]. Praha: Bohemia Energy, 2019 [cit. 2020-01-03]. Dostupné z: https://www.reseni2plus.cz/wp-content/uploads/2016/08/BE-PD-EC-DO-AP-CZ_20200101-01.pdf
- [6] Ceník elektřiny na 1 rok [online]. Praha: ČEZ, 2017 [cit. 2020-01-07]. Dostupné z: https://www.cez.cz/edee/content/file/produkty-a-sluzby/obcane-a-domacnosti/elektrina-2017/web_cenik_elektrina_na_jeden_rok_moo_2017_8_cezdi.pdf

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Jiří Šebek, kabinet výuky informatiky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **14.02.2020**

Termín odevzdání bakalářské práce: **22.05.2020**

Platnost zadání bakalářské práce: **30.09.2021**

Ing. Jiří Šebek
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

