



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií ■

# Online portál pro skladování a analýzu dat PQ monitoru

## Bakalářská práce

*Studijní program:* B2646 – Informační technologie  
*Studijní obor:* 1802T007 – Informační technologie  
*Autor práce:* **Mykhailo Kushchov**  
*Vedoucí práce:* Ing. Jan Kraus, Ph.D.





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

# Online portal for storage and analysis of the PQ monitor data sets

## Bachelor thesis

*Study programme:* B2646 – Information Technology

*Study branch:* 1802T007 – Information technology

*Author:* **Mykhailo Kushchov**

*Supervisor:* Ing. Jan Kraus, Ph.D.





## Zadání bakalářské práce

# Online portál pro skladování a analýzu dat PQ monitoru

|                          |  |
|--------------------------|--|
| <i>Jméno a příjmení:</i> | <b>Mykhailo Kushchov</b>                   |
| <i>Osobní číslo:</i>     | M15000249                                  |
| <i>Studijní program:</i> | B2646 Informační technologie               |
| <i>Studijní obor:</i>    | Informační technologie                     |
| <i>Zadávací katedra:</i> | Ústav mechatroniky a technické informatiky |
| <i>Akademický rok:</i>   | <b>2018/2019</b>                           |

### Zásady pro vypracování:

1. Seznamte se s typickými strukturami dat v archivu PQ monitoru a s dostupnými metodami jejich vzdáleného odečtu, přenosu a ukládání.
2. Navrhněte efektivní datové úložiště pro tato data a s využitím vhodné technologie pro vývoj webových služeb navrhněte a realizujte rozhraní s funkcemi pro analýzu a sdílení dat se vzdálenými klienty.
3. Implementujte rozhraní a jednoduchého klienta s funkcemi pro alespoň základní otestování portálu.
4. V závěru shrňte dosažené výsledky a diskutujte další možnosti rozvoje tématu.

*Rozsah grafických prací:*  
*Rozsah pracovní zprávy:*  
*Forma zpracování práce:*  
*Jazyk práce:*

dle potřeby dokumentace  
30–40 stran  
tištěná/elektronická  
Čeština



### **Seznam odborné literatury:**

- [1] CASCIARO, Mario; MAMMINO, Luciano. Node.js Design Patterns. Packt Publishing Ltd, 2016.
- [2] POWERS, Shelley, 2016. Learning Node: moving to the server-side. Second edition. Boston: O'Reilly. ISBN 1491943122.
- [3] BERTOLI, Michele. React Design Patterns and Best Practices. Packt Publishing Ltd, 2017.
- [4] BOJINOV, Valentin. RESTful Web API Design with Node.js. Packt Publishing Ltd, 2016.

*Vedoucí práce:*

Ing. Jan Kraus, Ph.D.  
Ústav mechatroniky a technické informatiky

*Datum zadání práce:*

10. října 2018

*Předpokládaný termín odevzdání:*

30. dubna 2019

prof. Ing. Zdeněk Plíva, Ph.D.  
děkan

L.S.

doc. Ing. Milan Kolář, CSc.  
vedoucí ústavu

V Liberci dne 10. října 2018

## Prohlášení

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

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

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

Bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že texty tištěné verze práce a elektronické verze práce vložené do IS/STAG se shodují.

23. srpna 2019

Mykhailo Kushchov

## Poděkování

Rad bych poděkoval vědoucemu této bakalářské práce panu Ing. Janu Krausovi, Ph.D. za rady při psaní této práce.

## Abstrakt

Práce je věnována realizaci online portálu, který nabízí běžnému uživateli a správci systému pohodlné rozhraní s funkcemi určenými k analýze a sdílení dat naměřených Power Quality (PQ) monitorem se vzdálenými klienty. Rešerše této práce se zabývá definicí a problematikou kvality elektrické energie, existujícími metodami řešení dané problematiky a návrhu efektivního datového úložiště pro data z PQ monitorů. V závěru rešerše je představen návrh vlastního řešení, kterým je online portál. Vývoj online portálu je popsán v kapitole „Online portál“, jež detailně popisuje rozhraní pro běžného uživatele a správce systému a také použité technologie. Pro klientskou část je to programovací jazyk TypeScript, JavaScriptová knihovna React a také knihovna Material UI, na které je postaven design portálu. Server online portálu je realizován v systému Node.js; data z PQ monitorů jsou zparsována na základě vlastních algoritmů a jsou uložena do nerelačního modelu databáze MongoDB. Funkce analýzy dat z PQ monitorů je zprostředkována formou grafů vytvořených přes knihovnu ApexChart.js. Online portál je implementovaný a otestovaný na datech reálně naměřených PQ monitorem. V závěrečné části práce je uvedeno shrnutí cílů stanovených na začátku práce, vyhodnocení dosažených cílů, a také možnosti rozšíření online portálu.

**Klíčová slova:** PQ monitor, skladování a analýza dat PQ monitoru, online portál, Node.js, React, MongoDB, REST API.

## Abstract

The aim of this thesis is to describe the development of an on-line portal which provides users and system administrators with a convenient interface with features developed to analyse and share data measured by Power Quality monitors. The chapter Research examines the electric power quality, existing methods for the solution of problems related to electric power quality and the development of an effective data storage system for data obtained from PQ monitors. The last part of the chapter Research focuses on the development of an online portal which is described in detail in the chapter Online Portal. This chapter explores the development of an interface for an average user and system administrator, as well as the technology necessary for the development. The client part was programmed with the programming language TypeScript, React JavaScript library and library Material UI, which was used for portal development. Node.js was used to implement the backend part of the portal. Data obtained from PQ monitors were processed on the basis of algorithms and stored in the non-relational database MongoDB. Data were analysed in the form of graphs created in the library ApexChart.js. The online portal was implemented and tested on real data measured by a Power Quality monitor. The last chapter of the thesis summarises and evaluates the results achieved and discusses the possible expansion of the online portal.

**Keywords:** PQ Monitor, storing and analyzing PQ monitor data, online portal, Node.js, React, MongoDB, Rest API.



# Obsah

|  |           |
|--|-----------|
| Seznam zkratek . . . . .   | 14        |
| <b>1 Úvod</b>  | <b>15</b> |
| <b>2 Rešerše</b>   | <b>16</b> |
| 2.1 Problematika . . . . .   | 16        |
| 2.2 Struktury dat v archivu PQ monitoru . . . . .                        | 17        |
| 2.2.1 PQ monitor . . . . .   | 17        |
| 2.2.2 Struktury dat PQ monitoru . . . . .                                | 17        |
| 2.2.3 Metody zpracování dat PQ monitoru . . . . .                        | 18        |
| 2.2.4 Návrh efektivního datového úložiště pro data PQ monitoru . . . . . | 18        |
| 2.3 Analýza existujících řešení . . . . .                                | 21        |
| 2.3.1 Wattics.com . . . . .  | 22        |
| 2.3.2 Izenda.com . . . . .   | 23        |
| 2.3.3 Energycloud . . . . .  | 24        |
| 2.4 Návrh online portálu a motivace k bakalářské práci . . . . .         | 26        |
| <b>3 Online portál</b>   | <b>29</b> |
| 3.1 Popis implementovaných rozhraní . . . . .                            | 29        |
| 3.1.1 Dashboard . . . . .  | 29        |
| 3.1.2 Comparison . . . . .   | 30        |
| 3.1.3 Main . . . . .   | 30        |
| 3.1.4 PQ Main . . . . .  | 31        |
| 3.1.5 Electricity meter . . . . .  | 31        |
| 3.1.6 Energy cost . . . . .  | 32        |
| 3.1.7 Device manager . . . . .   | 32        |
| 3.1.8 Logs . . . . .   | 32        |
| 3.2 Klient . . . . .   | 33        |

|          |  |           |
|----------|--|-----------|
| 3.2.1    | React, Redux, StoryBook a TypeScript . . . . . | 34        |
| 3.2.2    | Material UI . . . . .                          | 36        |
| 3.2.3    | ApexChart . . . . .                            | 37        |
| 3.3      | Server . . . . .                               | 37        |
| 3.3.1    | Manipulace s daty PQ monitoru . . . . .        | 37        |
| 3.4      | Testování online portálu . . . . .             | 42        |
| <b>4</b> | <b>Závěr</b>                                   | <b>44</b> |
|          | <b>Literatura</b>                              | <b>45</b> |
| <b>A</b> | <b>Výsledky testování online portálu</b>       | <b>47</b> |

## Seznam obrázků

|      |  |    |
|------|--|----|
| 2.1  | Vzorek pro vyhodnocení proudu, fázového a sdruženého napětí . . . .                                      | 18 |
| 2.2  | Wattics.com: funkce Trend Analytics . . . . .  | 22 |
| 2.3  | Wattics.com: funkce exportu dat . . . . .  | 23 |
| 2.4  | Izenda.com: příklad uživatelského rozhraní . . . . .   | 24 |
| 2.5  | Izenda.com: příklad uživatelského rozhraní . . . . .   | 25 |
| 2.6  | Funkce sledování spotřeby elektrické energie v reálném čase aplikace<br>Energycloud . . . . .            | 25 |
| 2.7  | Funkce sledování spotřeby elektrické energie za období několika měsíců<br>aplikace Energycloud . . . . . | 26 |
| 2.8  | Funkce analýzy všech spotřebičů, které uživatel v domácnosti používá<br>aplikace Energycloud . . . . .   | 26 |
| 2.9  | Schéma návrhu online portálu . . . . .   | 27 |
| 3.1  | Funkce Dashboard online portálu . . . . .  | 30 |
| 3.2  | Funkce Dashboard online portálu . . . . .  | 31 |
| 3.3  | Funkce „Comparison“ online portálu . . . . .   | 32 |
| 3.4  | Funkce „Main“ online portálu . . . . .   | 33 |
| 3.5  | Funkce „PQ Main“ online portálu . . . . .  | 34 |
| 3.6  | Funkce „Electricity meter“ online portálu . . . . .  | 35 |
| 3.7  | Funkce „Energy cost“ online portálu . . . . .  | 36 |
| 3.8  | Funkce „Device manager“ online portálu . . . . .   | 37 |
| 3.9  | Funkce „Logs“ online portálu . . . . .   | 38 |
| 3.10 | Schéma předávání dat mezi komponentami React . . . . .   | 39 |
| 3.11 | Lokální úložiště Redux . . . . .   | 40 |
| 3.12 | Příklad použití StoryBook . . . . .  | 41 |
| 3.13 | Model databáze MongoDB . . . . .   | 42 |

|     |  |    |
|-----|--|----|
| A.1 | Testování PM2: 100 HTTP požadavků . . . . .  | 47 |
| A.2 | Testování Artillery.io: 100 HTTP požadavků . . . . .                                     | 47 |
| A.3 | Testování PM2: 500 HTTP požadavků . . . . .  | 47 |
| A.4 | Testování Artillery.io: 500 HTTP požadavků . . . . .                                     | 48 |
| A.5 | Testování PM2: 100 HTTP požadavků při spuštění více instalací Node.js                    | 48 |
| A.6 | Testování Artillery.io: 100 HTTP požadavků při spuštění více instalací Node.js . . . . . | 48 |
| A.7 | Testování PM2: 500 HTTP požadavků při spuštění více instalací Node.js                    | 49 |
| A.8 | Testování Artillery.io: 500 HTTP požadavků při spuštění více instalací Node.js . . . . . | 49 |

## Seznam tabulek

|     |  |    |
|-----|--|----|
| 2.1 | Výsledky testů rychlosti pro databáze MongoDB a PostgreSQL . . . | 21 |
|-----|--|----|

## Seznam zkratek

|               |   |
|---------------|---|
| <b>PQ</b>     | (Power Quality) - kvalita elektrické energie popisuje parametry elektrické energie, které zabraňují krátké životnosti a poruchám elektrických systémů |
| <b>ČSN</b>    | české technické normy   |
| <b>PQ-A</b>   | analýzátor kvality elektrické energie třídy A   |
| <b>CSV</b>    | (Comma-Separated Values) - souborový formát používaný pro manipulaci s tabulkovými daty   |
| <b>USB</b>    | (Universal Serial Bus) - sériová sběrnice určená k propojení periférií s počítačem  |
| <b>UI</b>     | (User interface) - rozhraní, pomocí kterého lidé ovlivňují chování programů, softwarů apod.   |
| <b>RS-485</b> | standard síťové komunikace  |
| <b>REST</b>   | (Representational State Transfer) - architektura komunikace mezi klientem a serverem  |
| <b>SQL</b>    | (Structured Query Language) - programovací jazyk používaný v relačních databázích   |
| <b>NoSQL</b>  | (not only SQL) - datový koncept, ve kterém je struktura úložiště a manipulace s daty odlišná od modelu relačních databází                             |
| <b>JSON</b>   | (JavaScript Object Notation) - formát určený k přenosu dat a pro lepší pochopitelnost založený na JavaScriptu   |
| <b>kWh</b>    | (Kilowatt hour) - mezinárodní energetická jednotka, která vyjadřuje množství dodané a spotřebované energie  |

# 1 Úvod

Webových služeb nabízejících zjednodušení přístupu k informacím o spotřebované nebo dodané elektrické energii je dnes poměrně hodně. Jsou to aplikace určené pro kontrolu tarifu spotřeby elektrické energie, kontrolu účtů za dodávky elektřiny nebo pro analýzu kvality elektrické energie a komunikaci s odborníky v této oblasti.

Tato bakalářská práce si klade dva konkrétní cíle: první - analytický a druhý - praktický, oba vzájemně propojené. Analytický cíl spočívá v obeznámení se se současnou problematikou kvality elektrické energie a s již existujícími způsoby řešení zjištěných problémů. Na základě splnění analytického cíle vzniká cíl praktický. Tím je návrh a implementace vlastního efektivního řešení k eliminaci zjištěných problémů.

Metoda řešení analytického cíle je seznámení se s definicí kvality elektrické energie. Ta popisuje parametry elektrické energie, a pokud je nebudeme brát v úvahu, můžeme dojít u elektrických systémů ke zkresleným výsledkům. Další metodou řešení analytického cíle je průzkum existujících řešení problémů s kvalitou elektrické energie. Parametry elektrické energie je nutné změřit a zanalyzovat. K tomu lze využít speciální zařízení na měření parametrů a také desktopové nebo webové aplikace.

Pro realizaci praktického cíle, tj. vývoj systému, je potřeba zvolit vhodné technologie a navrhnout efektivní datové úložiště, protože se jedná se o práci s velkým množstvím dat. Při implementaci online portálu je potřeba se soustředit na funkci analýzy a sdílení dat naměřených PQ monitorem se vzdálenými klienty. Důležitou částí realizace online portálu je také implementace jednoduchého klienta za účelem otestování portálu.

## 2 Rešerše

Tato kapitola popisuje problémy, spojené s kvalitou elektrické energie, definuje přístroje, které měří, vyhodnocují a ukládají hodnoty elektrických veličin, popisuje strukturu naměřených dat a metody jejich zpracování a přináší také návrh efektivního datového úložiště pro tyto data. Dále tato kapitola analyzuje existující webové aplikace pro ukládání a zpracovávání naměřených hodnot elektrických veličin. V závěru kapitoly je uveden návrh vlastního řešení, tj. návrh online portálu pro kvalitativní analýzu hodnot elektrické energie. Zmíněna je také moje motivace k napsání bakalářské práce na dané téma.

### 2.1 Problematika

Elektrické systémy se postupně stávají stále složitějšími. S využitím moderních výkonných elektronických zařízení je v dnešní době kontrola systému efektivnější, pohodlnější a spolehlivější. Nicméně používání těchto zařízení dostalo energetický systém do vážných problémů v souvislosti s kvalitou. Od té doby co používání inteligentních elektronických zařízení vzrostlo v každé životní sféře člověka, požadavky na kvalitu výkonu z důvodu jejich dlouhé životnosti se staly pro uživatele hlavním kritériem. Právě proto se kvalita elektrické energie stala předmětem zájmu energetických společností, konečných uživatelů i výrobců. Kvalita elektrické energie popisuje parametry elektrické energie, které zabrání zkrácení životnosti a poruchám elektrických systémů.[1]

Důvodem nízké kvality elektrické energie jsou obvykle poruchy elektrického vedení, jako jsou impulzy, stupně, pokles napětí, nerovnováha proudu a napětí. The International Electro-technical Commission a Standard Institute of Electrical and Electronics Engineers zahrnují ztrátu rovnováhy jako zdroj zhoršení kvality elektrické energie.[1] V České republice podléhá dodávka elektrické energie normě ČSN EN 50160.



## 2.2 Struktury dat v archivu PQ monitoru

### 2.2.1 PQ monitor

Na trhu existují různé přístroje, které zaznamenávají, vyhodnocují a ukládají do vnitřní paměti hodnoty elektrických veličin. Takovými přístroji jsou monitory PQ (Power Quality), které řeší problematiku spotřeby a kvality elektrické energie. Monitory PQ jsou přístroje určené pro zaznamenávání, monitoring a taky pro uchovávání dat o kvalitě elektrické energie do vnitřní paměti. Výsledná data takových přístrojů umožňují sledovat dynamiku zlepšení nebo zhoršení kvality elektrické energie a také minimalizovat poruchy a snížit ztráty.[1]

Pro analýzu struktury a metod zpracování dat kvality elektrické energie v této bakalářské práci byl zvolen analyzátor kvality ARTIQ 144 PQ-A, který nabízí velmi přesné měření hodnot elektrické energie a také velký výběr rozhraní vzdálené komunikace. Tento analyzátor se používá pro měření, hodnocení a záznam spotřeby a kvality elektrické energie.[2]

### 2.2.2 Struktury dat PQ monitoru

Analyzátor ukládá do vnitřní paměti naměřené hodnoty spotřeby a kvality elektrické energie. To jsou hodnoty elektrických veličin frekvence, napětí, harmonických a meziharmonických proudů, flikru, nesymetrie, činných a jalových výkonů, energií a ukazatelů kvality elektrické energie. Analyzátor také sčítá minimální, průměrné a maximální hodnoty a pravidelné odečty stavu elektroměru. Prostřednictvím webových nebo desktopových aplikací se generují soubory těchto dat ve formátu csv. Příklady takových souborů jsou soubor ElectricityMeter.csv, Main.csv, PQMain.csv a Log.csv.[2]

Soubor ElectricityMeter.csv obsahuje data činné a jalové energie. Do souboru se zapisují jednofázové až třífázové, kladné a záporné hodnoty činné a jalové energie. Kladné hodnoty popisují spotřebovanou energii, záporné naopak energii vyrobenou nebo dodanou. Do souboru PQMain.csv se dále zapisují informace pro vyhodnocení kvality elektrické energie, to jsou hodnoty veličin frekvence, napětí, proudu, činného výkonu, jalového výkonu, celkového harmonického zkreslení, atd. Soubor Log.csv obsahuje záznam informací o činnosti analyzátoru.

Analyzátor provádí přenos dat prostřednictvím USB, 2x RS-485 a prostřednictvím sítě Ethernet. Přenáší data pro další analýz do desktopových aplikací. ARTIQ 144

PQ-A podporuje standardní zabudovaný nativní webserver. Tím pádem lze sledovat všechny hodnoty a běh činnosti ARTIQ 144 PQ-A online pomocí webového prohlížeče. Při zadání příslušné IP adresy do webového prohlížeče se zobrazí webová stránka s informacemi z přístroje.[2]

### 2.2.3 Metody zpracování dat PQ monitoru

Zpracování dat analyzátozem zahrnuje měření, vyhodnocování, ukládání dat do vnitřní paměti analyzátozem a následný přenos dat do počítače. Měření dat analyzátozem probíhá prostřednictvím vyhodnocení frekvence, vzorkováním proudových a napěťových signálů a také vyhodnocováním sebraných dat. Například pro vyhodnocení proudu, fázového a sdruženého napětí se používá vzorek na obrázku 2.1, který obsahuje  $i$  - index vzorků,  $n$  - počet vzorků za periodu měření,  $U_{1i}$ ,  $U_{2i}$ ,  $I_{1i}$  - jednotlivé vzorky napětí a proudu.[2]

$$U_1 = \sqrt{\frac{1}{n} \sum_{i=1}^n U_{1i}^2}, U_{12} = \sqrt{\frac{1}{n} \sum_{i=1}^n (U_{1i} - U_{2i})^2}, I_1 = \sqrt{\frac{1}{n} \sum_{i=1}^n I_{1i}^2}$$

Obrázek 2.1: Vzorek pro vyhodnocení proudu, fázového a sdruženého napětí

### 2.2.4 Návrh efektivního datového úložiště pro data PQ monitoru

Cílem této bakalářské práce je navrhnout a implementovat online portál pro skladování a analýzu dat PQ monitoru. Jednou z důležitých částí návrhu a implementace online portálu je výběr a návrh efektivního datového úložiště pro tato data. Ve světě databázových technologií existují dvě hlavní oblasti - relační a nerelační databáze. Rozdíl mezi SQL a NoSQL spočívá v tom, jakou mají strukturu, jaké datové typy podporují a jakým způsobem uchovávají data.

Relační databáze uchovávají strukturovaná data, která reprezentují objekty reálného světa, například informace o osobě nebo obsah nákupního košíku. Tato data jsou setříděná do skupin tabulek, jejichž formát je specifikován ve fázi navrhování datového úložiště. Relační systémy řízení báze dat používají jako dotazovací jazyk SQL. Příkladem takových systémů je Oracle Database, MySQL, PostgreSQL, MS-SQL atp.[3]

NoSQL podporují nerelační datový model, který umožňuje rychlou a efektivní práci s daty. Ukládají data ve formě klíč/hodnota, a mohou také reprezentovat data jako dokumentové nebo grafové databáze. Příkladem NoSQL systému řízení báze dat jsou MongoDB, CouchDB, DocumentDB, apod.. Takové systémy se používají v případě, že relační systémy řízení báze dat nestačí nebo projekt vyžaduje podmínky, kterým relační systém řízení báze dat nevyhovuje.[4]

Důvodem k použití relačních systémů báze dat je realizace projektu kde struktura dat se během vývoje příliš nemění a projekt vyžaduje technologie, na které se lze spolehnout, což znamená velké zkušenosti vývojářů a technickou podporu.[3]

Důvodem k použití nerelačních systémů báze dat je uchovávání velkého množství nestrukturovaných dat. NoSQL neomezuje uživatele v použití několika datových typů, navíc v případě potřeby může uživatel v průběhu práce datové typy přidávat. Pokud požadavky na data jsou nejisté nebo se vyvíjejí společně s projektem a jedněmi z hlavních požadavků na databázi je škálovatelnost tak nerelační systém báze dat je dobrou volbou.[4]

Na základě analýzy relační a nerelační databáze a požadavků k této bakalářské práci lze říct, že nerelační systémy řízení báze dat vyhovují lépe, protože požadavky na funkce online portálu nejsou jisté, škálovatelnost je taktéž důležitá jde o skladování a práci s velkým množstvím nestrukturovaných dat PQ monitoru. Pro lepší návrh efektivního datového úložiště dále porovnávám relační a nerelační systémy řízení báze dat na příkladu PostgreSQL a MongoDB.

PostgreSQL je databázový systém s otevřeným zdrojovým kódem. PostgreSQL ukládá data do tabulek a pro přístup k databázi používá strukturovaný dotazovací jazyk (SQL). V PostgreSQL je potřeba na základě požadavků uživatelů ještě před implementací databáze definovat její strukturu a nastavit pravidla pro řízení relace mezi daty v tabulkách. Vzájemně související informace mohou být uloženy v samostatných tabulkách, ale mohou být spojeny pomocí relací. Jakékoliv změny ve struktuře však můžou vyřadit databáze z provozu nebo snížit výkon aplikace. [5]

MongoDB patří k NoSQL databázím a ukládá data jako jsou dokumenty. Ty umožňují snadno reprezentovat hierarchii vztahů a jednoduše tak uchovávání pole a další složitější datové typy.[5]

Podle mého názoru jsou pro výběr vhodnějšího systému řízení báze dat v této bakalářské práci důležitá následující kritéria: škálovatelnost, rychlost zpracování dat, dostupnost, rychlost vývoje a flexibilní změna struktury. Proto podle těchto kritérií porovnávám MongoDB a PostgreSQL.

PostgreSQL nenabízí žádné původní mechanismy, jak rozšířit databázi na víc než jeden server nebo způsob, jak zajistit její neustálou dostupnost. Obě tato omezení potlačují vývoj moderních, škálovatelných a globálně dostupných aplikací a také omezují použití cloudových architektur.[5]

Aplikace, které byly v minulosti navrženy tak, aby sloužily konečným uživatelům jsou nyní dodávány jako webové a cloudové služby, které musí být dostupné neustále, z mnoha různých zařízení na různých kanálech a globálně přístupné milionům uživatelů na celém světě. Aby byly splněny tyto požadavky, je MongoDB postavena na distribuované architektuře, kterou lze provozovat v rámci i napříč geograficky rozptýlených datových center a cloudových úložišť, což poskytuje dostupnost a škálovatelnost, kterou PostgreSQL postrádá.[5]

Práce s daty jakými jsou flexibilní JSON dokumenty, avšak ne ve formě pevných řádků a sloupců, pomáhá vývojářům zrychlit vývoj softwaru. Do dokumentu JSON programátor může přidávat nové atributy, aniž by musel měnit strukturu databáze. V horším případě by to způsobilo výpadek, v tom lepším výrazné snížení výkonnosti v relačních databázích. Pokud jsou všechna data jedné entity uchována v jednom dokumentu, namísto ukládání do několika relačních tabulek, databáze čte a zapisuje data na jednom místě. Když jsou všechna data dané entity na jednom místě, usnadňuje to vývojářům porozumět výkonu dotazů a optimalizovat je.[5]

Z těchto důvodů PostgreSQL a další relační databáze začaly podporovat datový typ JSON. Nicméně pouhé přidání datového typu JSON produktivitu vývojářů nezlepší. Dotazování a manipulace s obsahem dokumentu JSON vyžaduje použití specifických SQL funkcí pro přístup k hodnotám, které nejsou většině vývojářů známé.[5]

Pro praktické ověření rychlosti zpracování dat databáze PostgreSQL verze 11.5 a MongoDB verze 4.04 bylo provedeno testování lokálně v systému Node.js s použitím hardwaru: CPU Intel Core i5 2,3 GHz a RAM 8 GB LPDDR3 2133 MHz. Testování proběhlo na příkladu přidání, hledání a mazání 100, 1000 a 10000 řádků dat ve formátu name: David Novák, e-mail: david.novak@seznam.cz. V tabulce 2.1 jsou uvedeny zprůměrované výsledky 10 testů pro databáze MongoDB a PostgreSQL. Výsledky testů ukazují, že databáze PostgreSQL je rychlejší, což se nejvíc projevuje při zpracování velkého množství dat.

Tabulka 2.1: Výsledky testů rychlosti pro databáze MongoDB a PostgreSQL

|                        | MongoDB | PostgreSQL |
|------------------------|---------|------------|
| insert 100 rows[ms]:   | 19.55   | 5.84       |
| select 100 rows[ms]:   | 6.42    | 3.36       |
| delete 100 rows[ms]:   | 2.78    | 1.73       |
| insert 1000 rows[ms]:  | 49.76   | 33.98      |
| select 1000 rows[ms]:  | 33.68   | 20.85      |
| delete 1000 rows[ms]:  | 14.09   | 3.82       |
| insert 10000 rows[ms]: | 152.77  | 188.91     |
| select 10000 rows[ms]: | 107.76  | 68.15      |
| delete 10000 rows[ms]: | 97.91   | 5.11       |

I když databáze PostgreSQL zpracovává data rychleji, MongoDB nabízí flexibilnější nástroje pro rychlejší vývoj a škálovatelnost. Na základě porovnání systémů řízení báze dat byl zvolen vhodnější systém, kterým se stal MongoDB. Lze říct, že nelineární systémy řízení báze dat v této bakalářské práci vyhovují lépe, protože zaprvé na začátku nejsou jisté požadavky na funkce online portálu, zadruhé je důležitá škálovatelnost, zatřetí jde o skladování a práci s velkým množstvím nestrukturovaných dat PQ monitoru. Detailnější popis implementace databáze pro online portál popíšu v kapitole „Online portál“ této bakalářské práce.

## 2.3 Analýza existujících řešení

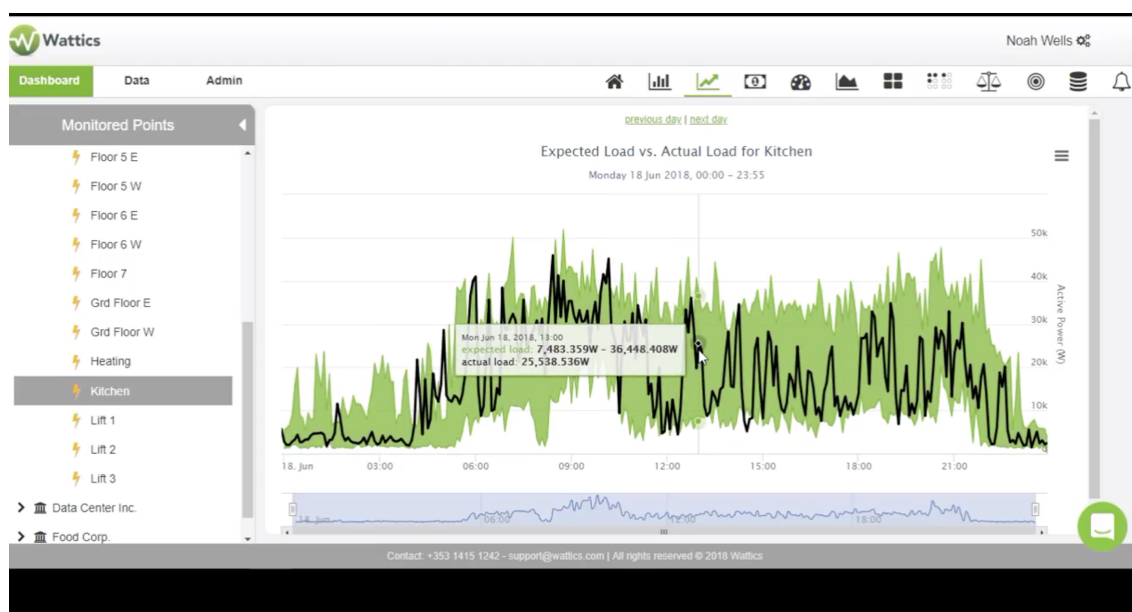
V dnešní době existuje na trhu velké množství aplikací na skladování a analýzu dat elektrické energie. Řada z nich jsou systémy, které umožňují provést komplexní analýzu dat formou grafů nebo diagramů. Pro analýzu existujících webových aplikací jsem si vybral aplikace wattics.com, izenda.com a vphrase.com, které podrobněji

popíšu v následující podkapitole.

### 2.3.1 Wattics.com

Wattics.com umí vytvořit prognózu možných úspor elektrické energie, analyzuje tarify plateb za elektrickou energii a hodnotí energetickou náročnost, to všechno provádí online a na jednom místě. Aplikace slouží jako nástroj pro manažery a podnikatele, kteří dělají rozhodnutí v rámci ušetření nákladů na spotřebu energie uvnitř firmy. Wattics.com poskytuje celou řadu funkcí pro zlepšování energetických služeb. Jednou z takových funkcí je Trend Analytics, která uživatelům umožňuje porovnávat spotřebu energie s očekávanými hodnotami a upozorňuje je na možnou úsporu energie.[6]

Na obrázku 2.2 je příklad dat o spotřebě elektrické energie ve formě grafu za období 30 dnů. Zelená barva představuje prognózu očekávaných hodnot spotřeby elektrické energie a černá barva představuje její reálné hodnoty. Pokud není reálná hodnota v rozsahu očekávaných hodnot, znamená to, že spotřeba energie je neobvyklá a aplikace to sama nahlásí uživatelům prostřednictvím emailu nebo sms.[6]



Obrázek 2.2: Wattics.com: funkce Trend Analytics

Další užitečnou funkcí pro uživatele této webové aplikace je analýza kvality elektrické energie, která uživatelům nabízí export dat ve formátu PNG, JPEG, PDF,

CSV a SVG z webové aplikace Wattics.com (obrázek 2.3). Analýzu těchto dat aplikace Wattics.com zobrazuje uživatelům prostřednictvím grafů a schémat.[6]



Obrázek 2.3: Wattics.com: funkce exportu dat

Podle mého názoru je konkurenční výhodou této aplikace to, že systém poskytuje synchronizaci dat solárních systémů pro zlepšení viditelnosti a kontroly sluneční energie. Uživatelé této aplikace se dělí na správce systému a běžné uživatele a každý z nich má určité rozhraní a pravomoce. Tato funkce řeší bezpečnost přístupu uživatelů k důležitým datům.[6]

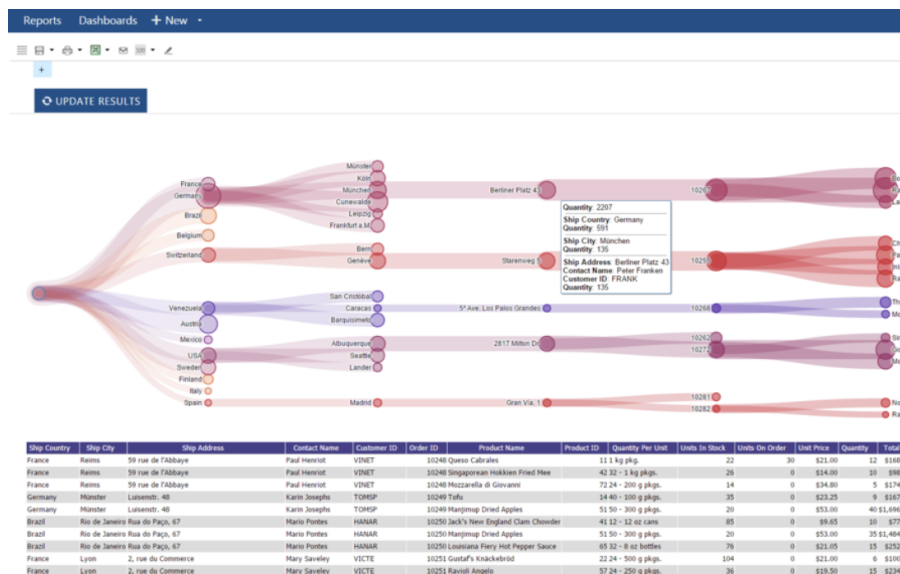
### 2.3.2 Izenda.com

Izenda.com je dalším řešením pro analýzu spotřeby elektrické energie na trhu webových aplikací. Tato aplikace mě zaujala především tím, že nabízí možnost integrace své webové aplikace do webových aplikací uživatelů.

To znamená, že všechny funkce aplikace Izenda.com se vkládají přímo do aplikace uživatelů a aplikace Izenda se připojuje k serverům SQL, Oracle a MySQL a k dalším zdrojům dat aplikace uživatelů. Uživatelé tím pádem získávají snadno použitelný nástroj, který se přizpůsobí jejich požadavkům. Tento nástroj se dokáže plně přizpůsobit vzhledu, navigaci a zabezpečení dat uživatelské aplikace.[7]

Na obrázcích 2.4 a 2.5 jsou příklady různých uživatelských rozhraní webové aplikace Izenda.

Důležitou uživatelskou funkcí tohoto systému je možnost vytvářet a přizpůsobovat



Obrázek 2.4: Izenda.com: příklad uživatelského rozhraní

uživatelské rozhraní webové aplikace přímo v prohlížeči a schopnost personalizovat, navrhovat a prohlížet uživatelské rozhraní v reálném čase. Uživatel si může zvolit vhodné komponenty na základě svých požadavků. Aplikace také umožňuje analyzovaná data stáhnout do formátu PDF, Word nebo Excel. Pro lepší pochopení a analýzu zobrazuje aplikace Izenda uživatelská data ve formě grafů, tabulek a schémat.[7]

Izenda nabízí plnou verzi své aplikace formou ročního předplatného, které uživatel platí na měsíční bázi. Tato verze obsahuje veškeré funkce platformy a poskytuje licence pro neomezený počet uživatelů. Pokud si uživatel chce Izendu pouze vyzkoušet, může si bezplatně stáhnout její demoverzi, ve které jsou však některé funkce omezené. Hlavní konkurenční výhodou této aplikace je to, že je flexibilní a uživatelé ji mohou používat k vytváření dynamických vizuálních analýz elektrické energie. Každý uživatel si tak může pomoci této aplikace vytvořit vlastní analytický nástroj dle svých požadavků. Podle mého názoru patří Izenda k lepším webovým řešením pro analýzu dat, které jsou na trhu dostupné.[7]

### 2.3.3 Energycloud

Další webovou aplikací, kterou jsem se inspiroval je Energycloud.

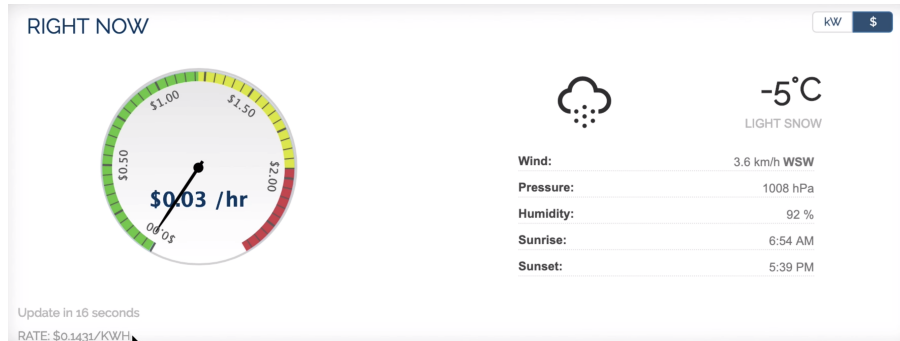
Tato webová aplikace slouží pro kontrolu a analýzu spotřeby elektrické energie v domácnosti. Hlavním cílem Energycloudu je snaha ušetřit uživatelům peníze





Obrázek 2.5: Izenda.com: příklad uživatelského rozhraní

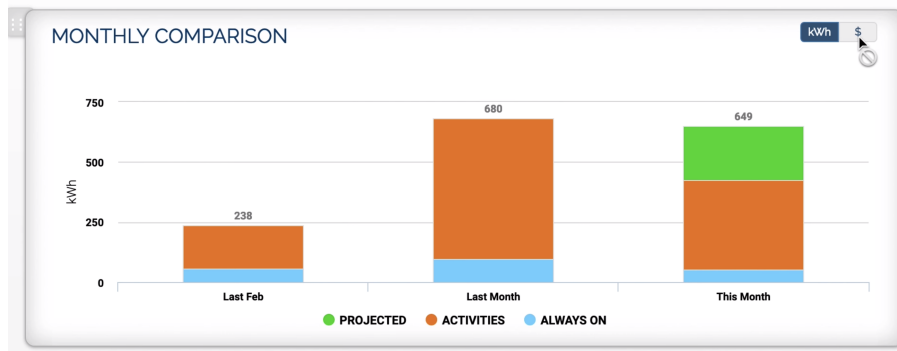
za spotřebu elektrické energie. Jednou z užitečných funkcí této webové aplikace je funkce sledování spotřeby elektrické energie (obrázek 2.6) v reálném čase v celém domě. Uživatelům se hodnoty spotřeby elektrické energie zobrazují v dolarech za hodinu na základě aktuálních tarifů nebo v kilowatthodinách.[8]



Obrázek 2.6: Funkce sledování spotřeby elektrické energie v reálném čase aplikace Energycloud

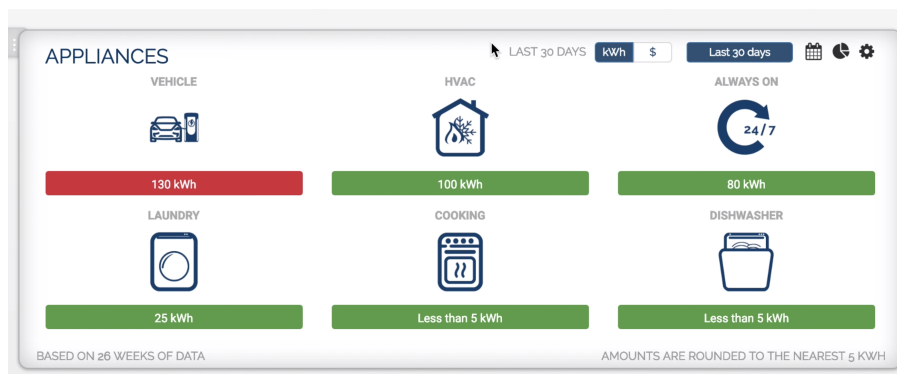
Na dalším obrázku 2.7 je ukázka hodnot spotřeby elektrické energie za období několika měsíců a uživatel zde může sledovat, jaké byly jeho hodnoty spotřeby v průběhu každého měsíce.[8]

Hlavní konkurenční výhodou Energycloudu je analýza všech spotřebičů, které uživatel v domácnosti používá (obrázek 2.8), jako jsou kuchyňská zařízení, pračka, myčka, auto. atp. Hodnoty spotřeby elektrické energie těchto spotřebičů se také



Obrázek 2.7: Funkce sledování spotřeby elektrické energie za období několika měsíců aplikace Energycloud

zobrazují v kilowatthodinách a v dolarech a uživatel má možnost nastavit si určitý časový interval jako je den, týden nebo měsíc. Tato funkce tedy dokáže zjistit jaké zařízení má největší spotřebu a uživatel tak ví, kde může ušetřit peníze.[8]



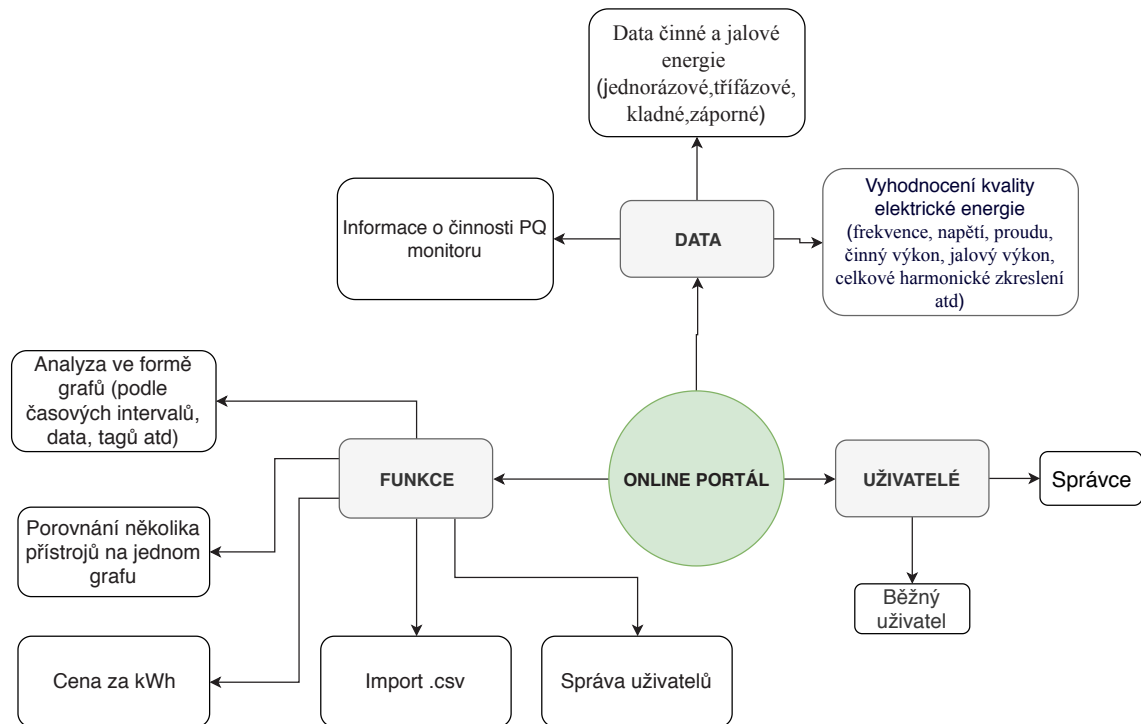
Obrázek 2.8: Funkce analýzy všech spotřebičů, které uživatel v domácnosti používá aplikace Energycloud

## 2.4 Návrh online portálu a motivace k bakalářské práci

V dnešní době, kdy se používání elektrických přístrojů a inteligentních zařízení těší vysoké popularitě, je z důvodu udržení dlouhé životnosti přístrojů důležitá analýza a kontrola kvality elektrické energie. Proto existuje hodně tak zvaných Power Quality monitorů, které slouží k měření, hodnocení a ukládání hodnot elektrické energie. Existuje také hodně webových aplikací vyvinutých k analyzování naměřených dat PQ monitorů. Na základě analýzy problematiky elektrické energie, porovnání a

zkoumání funkcí několika existujících webových aplikací, a také analýzy struktury dat PQ monitorů ukázané na příkladu analyzátoru ARTIQ 144 PQ-A, popisuje tato podkapitola návrh vlastního řešení online portálu pro skladování a analýzu dat PQ monitoru.

Podle schématu na obrázku 2.9 je online portál prakticky použitelný pro dva typy uživatelů: pro správce a běžného uživatele. Každý z nich má možnost používat určitou sadu funkcí daného online portálu.



Obrázek 2.9: Schéma návrhu online portálu

Na základě analýzy struktury dat PQ monitoru bude online portál analyzovat data činné a jalové energie, pak data vyhodnocení kvality elektrické energie, například frekvence, napětí, proudu, činný výkon, jalový výkon, celkové harmonické zkreslení, atp., a taky informace o činnosti PQ monitoru.

Online portál bude umožňovat analýzu dat ve formě grafů, kde si uživatel může zvolit PQ monitor, tagy, data (od kdy a do kdy analyzovat), časový interval (například data za dobu 15 nebo 30 minut) atd. Další funkce, kterou bude nabízet online portál je porovnání dat několika PQ monitorů na jednom grafu. Při zadání ceny za 1 kWh bude online portál počítat náklady spotřeby elektrické energie za určitý časový interval. Portál bude nabízet import naměřených dat PQ monitoru ve formátu csv.

Implementace a detailnější popis funkcí online portálu jsou popsány v praktické části této bakalářské práce. Mojí motivací k napsání této bakalářské práce bylo dokázat na příkladu vlastního řešení možnost implementace jednoduchého, praktického a snadno rozšiřitelného online portálu. V současné době existuje hodně firem, které se zabývají monitorováním kvality elektrické energie. Tyto firmy většinou používají pro analýzu spotřeby elektrické energie desktopové programy. Takovým programem je například Envis, který se musí instalovat na každém počítači ve firmě zvlášť a přístup k analyzovaným datům tak lidé mají pouze v kanceláři, kde je program nainstalovaný. Tento problém může pomoci vyřešit online portál pro monitorování spotřeb elektrické energie, ke kterému má uživatel přístup odkudkoliv, pokud je připojený k internetu.

## 3 Online portál

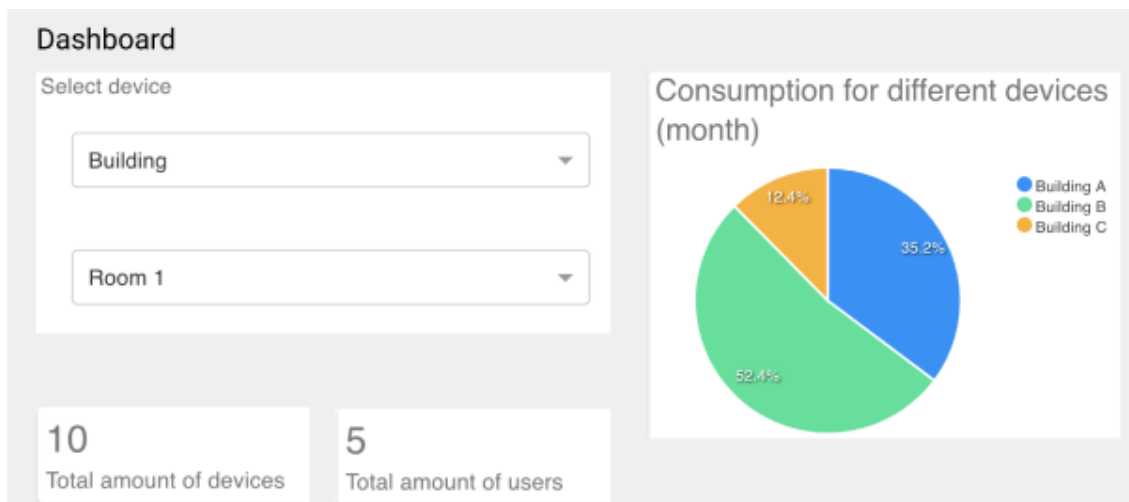
Tato kapitola detailně popisuje implementované rozhraní pro budoucí uživatele online portálu. Jak už bylo zmíněno v návrhu funkcí online portálu, systém vymezuje dva typy uživatelů - běžného uživatele a správce systému. Běžný uživatel má k dispozici „Dashboard“ - řídicí panel, ve kterém se na jednom místě integrují důležité informace ze všech stránek online portálu. Dále online portál nabízí rozhraní „Comparison“, ve kterém dochází k porovnání a analýze naměřených hodnot dvou nebo více PQ monitorů. Dalšími zajímavými funkcemi rozhraní, které má k dispozici běžný uživatel, jsou „Electricity meter“, „PQ Main“, „Main“ a „Logs“, které slouží k analýze formou grafů a tabulek naměřených hodnot frekvence, napětí, proudu, činného výkonu, jalového výkonu, celkového harmonického zkreslení a informací o činnosti PQ monitoru. Rozhraní „Energy cost“ nabízí běžnému uživateli kalkulačku, pomocí které si může spočítat cenu spotřebované elektrické energie. Správce systému má k dispozici stejné analytické rozhraní a navíc také nástroj „Device manager“ s funkcí importu csv souboru do online portálu. Dále popisují podrobně funkce těchto rozhraní a vysvětlují způsoby a technologie jejich implementace.

### 3.1 Popis implementovaných rozhraní

#### 3.1.1 Dashboard

Běžný uživatel i správce systému má možnost si zvolit pomocí řídicího panelu lokaci PQ monitoru, zařízení a hodnoty elektrické energie, jež PQ monitor měří. V řídicím panelu se pak zobrazí analýza zvoleného zařízení. Na obrázcích 3.1 a 3.2 je znázorněn řídicí panel uživatele, který obsahuje krátkou informaci o něm, tj. jeho jméno, příjmení a emailovou adresu, a také analýzu naměřených hodnot elektrické energie zvoleného zařízení.

Po zvolení lokace PQ monitoru a zařízení v ostatních rozhráních online portálu,



Obrázek 3.1: Funkce Dashboard online portálu

jako jsou „Electricity meter“, „PQ Main“, „Main“, „Energy cost“ a „Logs“, se zobrazí analýza naměřených hodnot zvoleného zařízení, což poskytuje uživateli pohodlnější práci s online portálem a nemusí znovu vybírat zařízení.

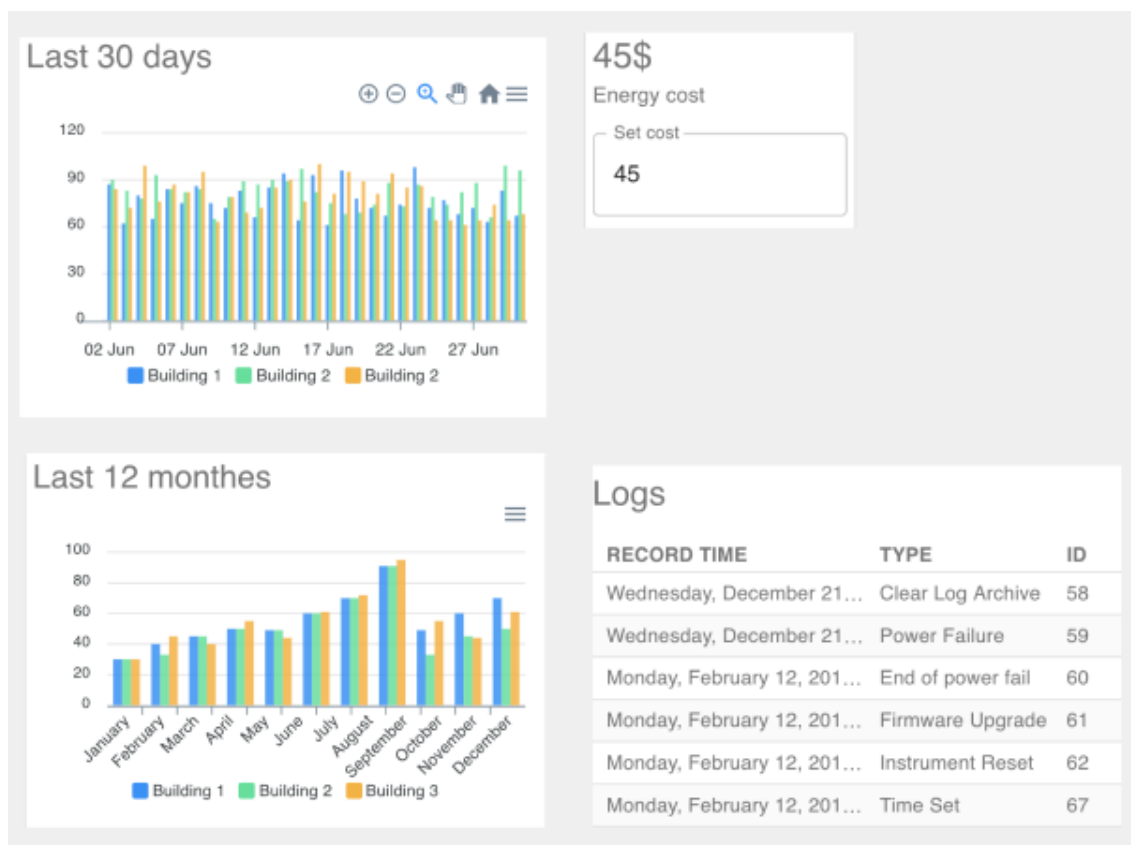
### 3.1.2 Comparison

Toto rozhraní online portálu umožňuje porovnávat naměřené hodnoty dvou a více PQ monitorů v jednom grafu.

Na obrázku 3.3 je vidět implementované rozhraní „Comparison“, kde má běžný uživatel nebo správce možnost porovnat naměřené hodnoty dvou nebo více PQ monitorů prostřednictvím volby časového intervalu naměřených dat (odkdy dokdy), intervalu vykreslení hodnot grafu, což jsou intervaly 15 min., 30 min., 1 hodina a 1 den, a tagu hodnot, který chce v grafu analyzovat. Uživatel může do grafu pro analýzu přidávat libovolný počet dalších zařízení stisknutím tlačítka „+“.

### 3.1.3 Main

Pomocí rozhraní „Main“ (obrázek 3.4) může uživatel analyzovat střední, maximální a minimální hodnoty veličin frekvence, napětí, proudu, činného výkonu, jalového výkonu, celkového harmonického zkreslení atd.



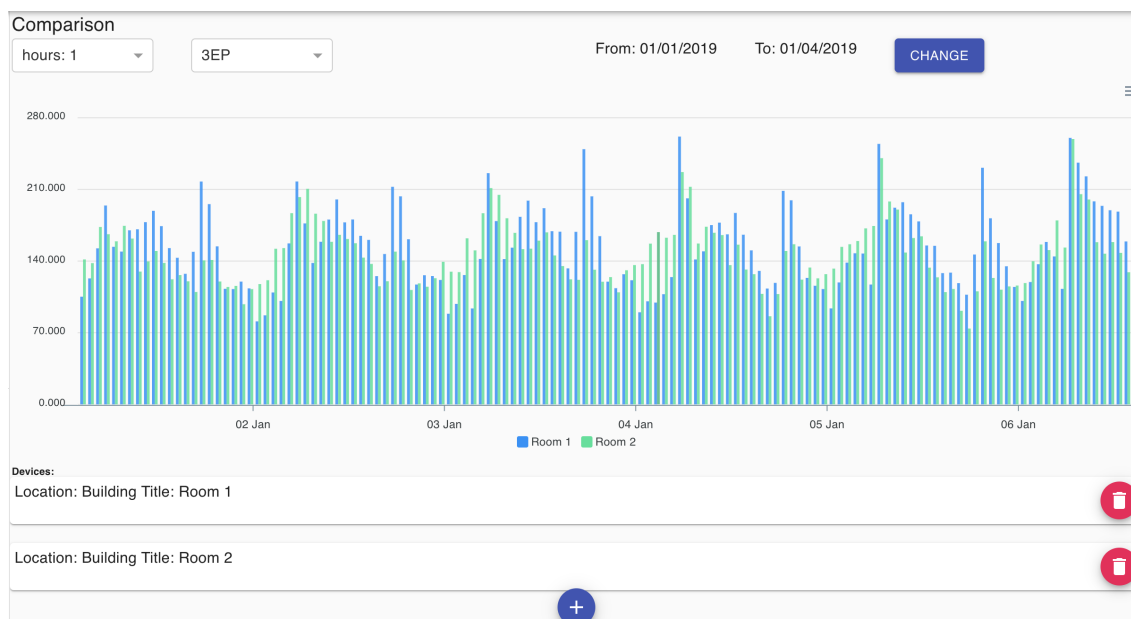
Obrázek 3.2: Funkce Dashboard online portálu

### 3.1.4 PQ Main

Pomocí rozhraní „PQ Main“ (obrázek 3.5) může uživatel analyzovat hodnoty napětí a frekvence z archivu PQ Main. Uživatel si zvolí firmu, PQ monitor, tag, časový interval měření a vykreslení hodnot (15 min., 30 min., 1 hod. nebo 1 den) a podle těchto hodnot se mu potom vykreslí graf.

### 3.1.5 Electricity meter

V rozhraní „Electricity meter“ jsou nastavené defaultní hodnoty lokace PQ monitoru a zařízení, které uživatel zvolil v „Dashboardu“, lze také vybrat hodnoty nové. Běžný uživatel nebo správce systému mají možnost vybrat časový interval hodnot, které chtějí zobrazit v grafu (naměřené hodnoty odkdy dokdy), a dále interval hodnot pro vykreslení grafu, tj. 15 min., 30 min., 1 hodina nebo 1 den. Na obrázku 3.6 je uveden graf naměřených hodnot z archivu „Electricity meter“, což jsou data jednofázové až třífázové, kladné i záporné hodnoty činné a jalové energie.



Obrázek 3.3: Funkce „Comparison“ online portálu

### 3.1.6 Energy cost

Pomocí „Energy cost“ (obrázek 3.7) má uživatel možnost sledovat v grafu tendence změn ceny spotřebované elektrické energie. Uživatel zadává lokace, zařízení a časové intervaly a také, což je důležité, cenu za 1 kWh. Na základě těchto dat spočítá online portál cenu za spotřebu elektrické energie a zobrazí ji v grafu.

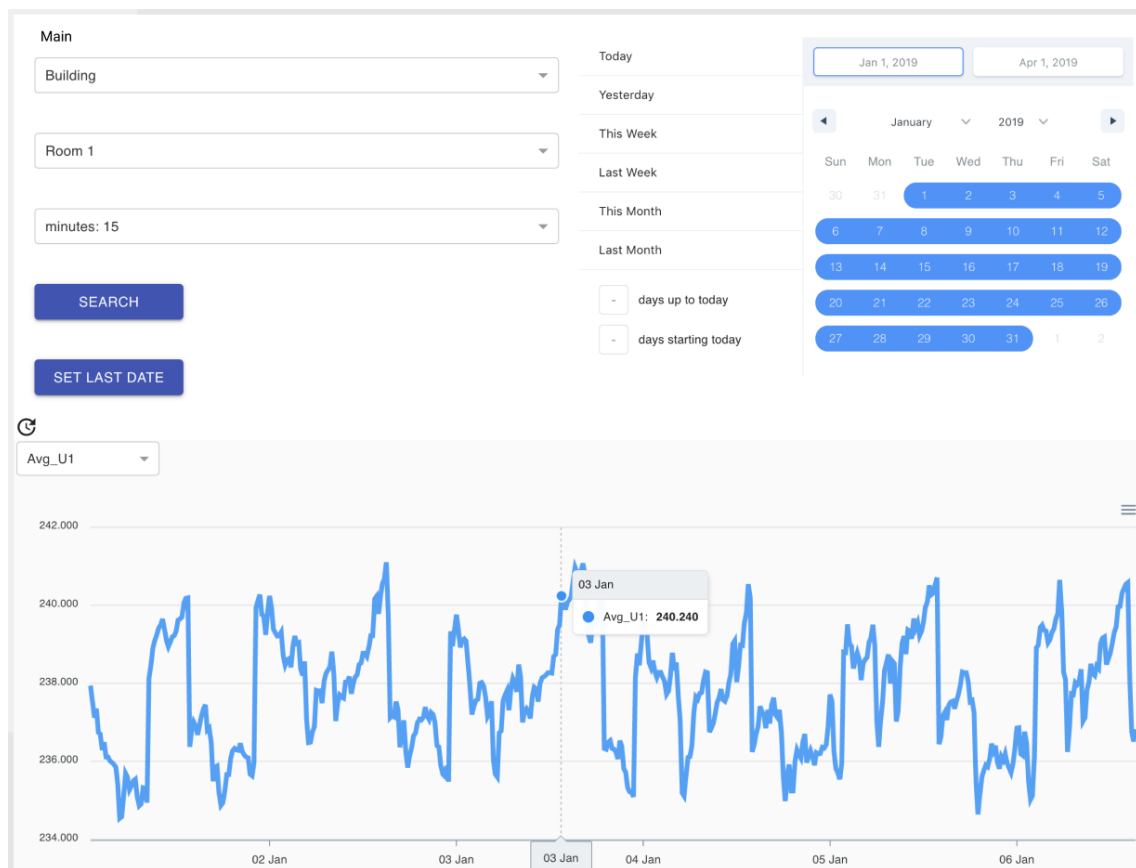
### 3.1.7 Device manager

„Device manager“ (obrázek 3.8) neboli správa zařízení je rozhraní určené pro správce systému a obsahuje funkce importu a mazání csv souborů. Rozhraní umožňuje nahrát do databáze online portálu určité typy csv souborů. Je přitom důležité vybrat typ souboru: electricity meter, logs, pq main nebo main, a také vybrat, zda-li má soubor header, tedy informaci obsaženou v hlavičce.

### 3.1.8 Logs

Rozhraní „Logs“ je implementované jak pro běžného uživatele, tak i pro správce systému. Stránka „Logs“ seznamuje uživatele s informacemi o činnosti PQ monitorů. Tabulka (obrázek 3.9) obsahuje hlášení chyb nebo varování v souvislosti s činnostmi PQ monitorů. Zvolením lokace PQ monitoru, zařízení, časového intervalu



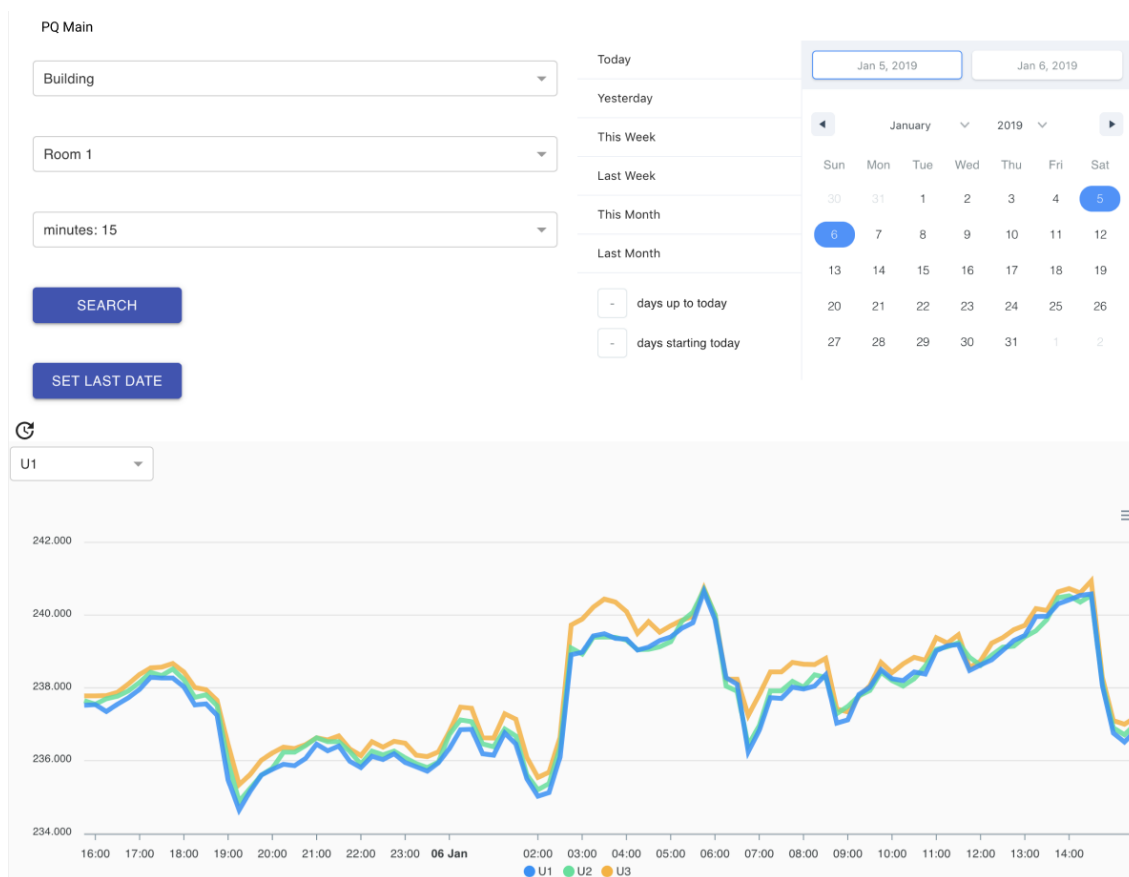


Obrázek 3.4: Funkce „Main“ online portálu

naměřených hodnot (odkdy dokdy) a také intervalu vykreslení hodnot – 15 min., 30 min., 1 hod. nebo 1 den – se v tabulce zobrazí datum, čas, typ a ID hlášení.

## 3.2 Klient

Klient online portálu je implementován v programovacím jazyce TypeScript s využitím JavaScriptové knihovny React, která umožňuje rozdělit UI na komponenty, které lze do sebe skládat a používat je na různých místech aplikace. Pro efektivnější práci s komponenty React byly použity nástroje Storybook a Redux. Design online portálu je realizován prostřednictvím knihovny Material UI a analýza dat PQ monitoru se zobrazuje pomocí knihovny ApexChart, která je vyvinuta pro vytvoření grafu a diagramu.

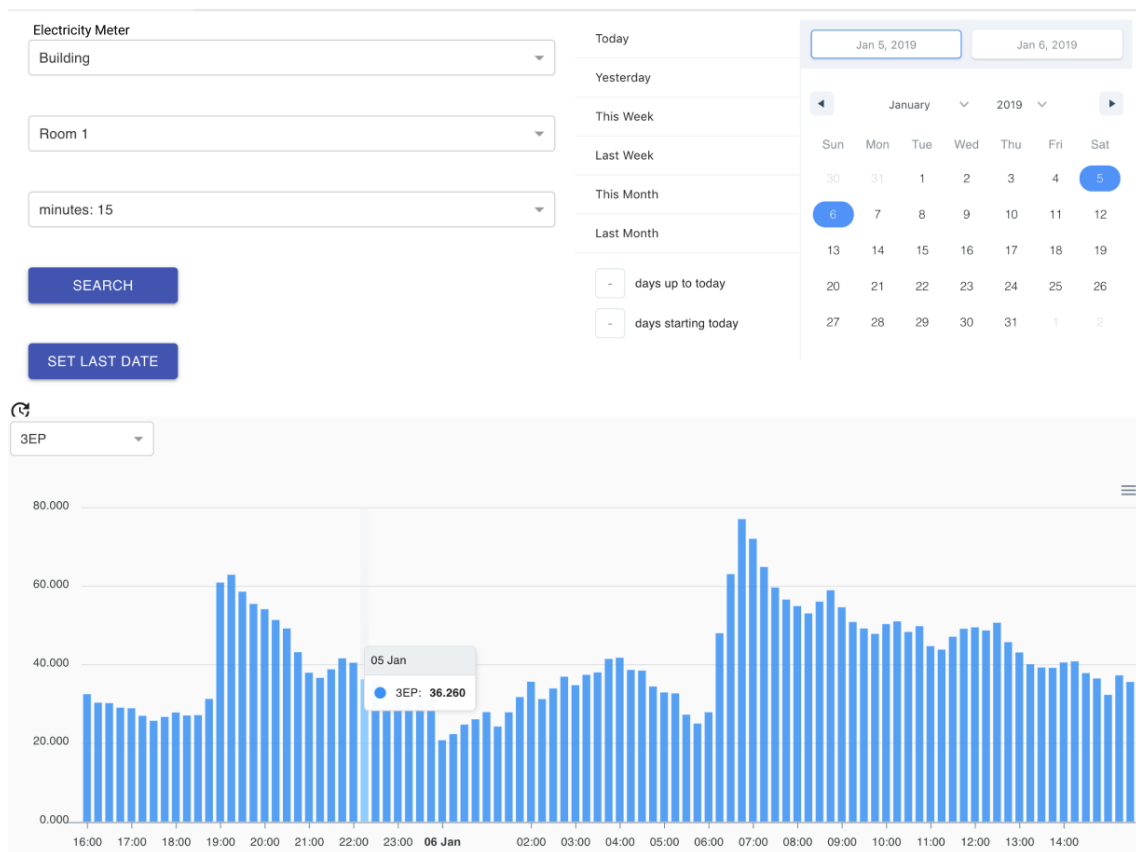


Obrázek 3.5: Funkce „PQ Main“ online portálu

### 3.2.1 React, Redux, StoryBook a TypeScript

Při realizaci online portálu byly zvolené technologie TypeScript, React, StoryBook a Redux. TypeScript je skriptovací programovací jazyk, který se kompiluje do JavaScriptu a podporuje statické typování. Statické typování umožňuje dopodrobna popsat vlastnosti a metody objektů a tříd, a tím vylučuje potřebu kontroly všech argumentů metod nebo funkcí, což činí kód napsaný v TypeScriptu jednodušším a čitelnějším. TypeScript je nadstavbou JavaScriptu, a to v praxi znamená, že podporuje všechny knihovny, frameworky a nástroje pro JavaScript.[9]

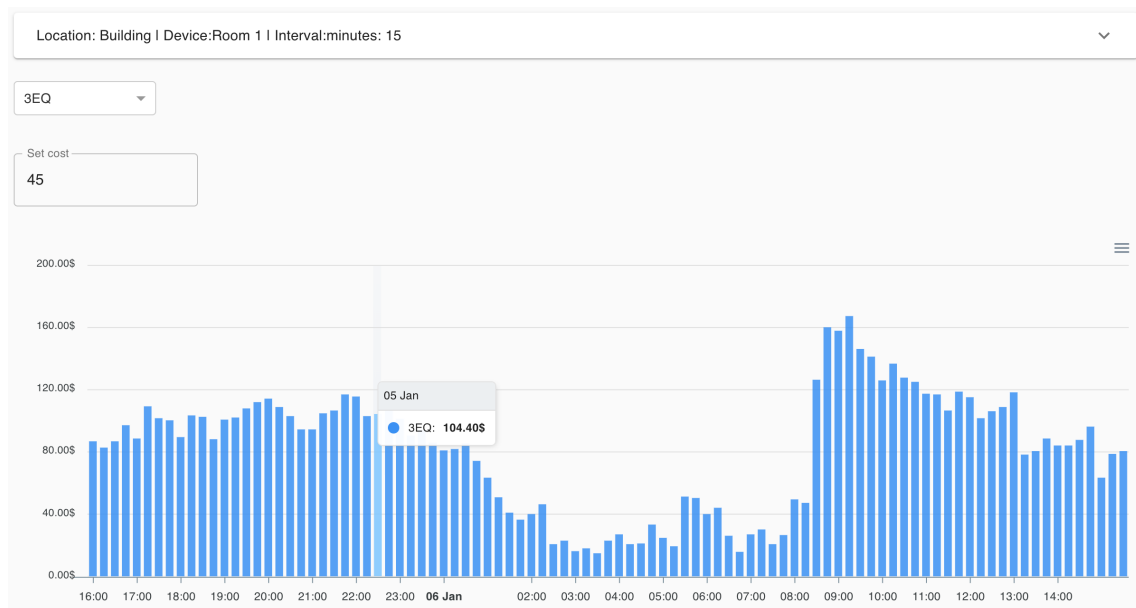
React je JavaScriptovou knihovnou založenou na komponentách, které usnadňují realizaci UI. Jak už jsem uváděl, React komponenty lze skládat do sebe a používat na různých místech aplikace nebo i v jiných aplikacích. Pokud chce uživatel během použití aplikace provést nějakou změnu, např. nahrát soubor nebo vygenerovat graf, knihovna React aktualizuje pouze tu část aplikace, která byla změněna a neprovede aktualizaci celé stránky nebo aplikace.[10]



Obrázek 3.6: Funkce „Electricity meter“ online portálu

Problém, který se objevil během vývoje klientské části online portálu a ze kterého se nakonec stal hlavní důvod k použití JavaScriptové knihovny Redux, je práce s daty aktuálního stavu aplikace. Například komponenta Electricity meter se skládá z dalších komponent menu, filtrů a grafu. Tyto komponenty mají svá vlastní izolovaná data, která je potřeba navzájem předávat pro další realizace funkcí (např. funkce vykreslení grafu podle zvolených filtrů). Problém bych mohl vyřešit přenosem všech dat do hlavní komponenty Electricity meter, to by ale vygenerovalo další problémy.

React komponenty tvoří stromovou strukturu, v níž existuje komponenta rodičovská a komponenty potomků. Předávání dat mezi komponentami React je zpravidla shora dolů (obrázek 3.10), tedy od rodičovské komponenty ke komponentě potomka. Problém ovšem spočívá v tom, že data mezi komponentami je potřeba předávat přes všechny úrovně stromu a zachovat přitom veškerou logiku v jediné hlavní komponentě. Tento problém vyřešila knihovna Redux, která poskytuje mezivrstvu - tzv. lokální úložiště pro data komponent (obrázek 3.11) a zároveň poskytuje snadný způsob předání těchto dat do komponenty, která je potřebuje a nedochází k předání



Obrázek 3.7: Funkce „Energy cost“ online portálu

dat zbytečně přes celý strom.[10]

Existuje mnoho různých nástrojů pro pohodlnější práci s React komponentami, např. StoryBook, který jsem v této práci využil. StoryBook je prostředí pro vývoj React komponent, který umožňuje vyvíjet organizovaně a efektivně.[11]

StoryBook (obrázek 3.12) běží mimo hlavní aplikace, a proto jsem během vývoje měl možnost vytvářet React komponenty izolovaně, což eliminuje starosti spojené se specifickými závislostmi a podmínkami v aplikaci.

### 3.2.2 Material UI

Při návrhu designu online portálu jsem zvolil knihovnu Material UI. Material UI je knihovnou s otevřeným kódem od společnosti Google, která poskytuje sadu komponent uživatelského rozhraní a která pomohla vytvářet React aplikace ve stylu Material Design.[12]

Při implementaci online portálu jsem využil i další Material UI rozhraní: navigaci, tabulky, tlačítka, rozhraní pro filtry apod. Pro realizaci designu Material UI byly komponenty online portálu přidány do speciální komponenty Material UI - CssBaseline. Po přidání komponenty online portálu zdědily od knihovny Material UI barvu pozadí Material Design, styl písma Roboto a velikost písma 16px - výchozí velikost písma prohlížeče, apod.

The image shows a web interface titled "Device manager" with two main panels. The left panel, "Remove device", has a sub-header "Choose device to delete:" and a dropdown menu currently showing "Room 2". Below the dropdown is a red button labeled "DELETE". The right panel, "Import csv:", has two input fields: "Title\*" with the value "TV" and "Location\*" with the value "Room 3". Below these is a dropdown menu showing "electricity\_meter". A checkbox labeled "Import file contain header" is checked. At the bottom of the right panel are two buttons: "UPLOAD" and "SUBMIT" with a cloud upload icon.

Obrázek 3.8: Funkce „Device manager“ online portálu

### 3.2.3 ApexChart

Základní funkcí online portálu je analýza dat naměřených PQ monitorem. Tato funkce je implementována pomocí knihovny ApexChart. Knihovna nabízí interaktivní vizualizace webových aplikací s použitím různých typů grafů. Použití ApexChart knihovny v online portálu bylo jednoduché, hlavními důvody pro volbu knihovny byly: snadnější zoomování hodnot v grafu, možnost umístění informačních poznámek do grafu, funkce panoramování dat z grafu. To je pro tuto práci důležité, neboť zpracovává velký objem dat naměřených PQ monitorem.

## 3.3 Server

Server online portálu běží v softwarovém systému Node.js s využitím frameworku Express.js. Datové úložiště pro data PQ monitoru je navrženo a implementováno jako nerelační model databáze MongoDB. Pro parsování dat PQ monitoru byly použity vlastní algoritmy. Použití těchto technologií pro implementaci serveru online portálu popisují podrobně v následujících podkapitolách.

### 3.3.1 Manipulace s daty PQ monitoru

V rešerši k této bakalářské práci byl zmíněn návrh efektivního datového úložiště pro data naměřená PQ monitorem. Na základě porovnání relačního a nerelačního modelu databáze byl zvolen vhodnější systém - MongoDB - neboť na začátku vývoje online

The screenshot shows a web interface for viewing logs. On the left, there are two dropdown menus: 'Building' (set to 'Building') and 'Room 2' (set to 'Room 2'). Below them is a blue 'SEARCH' button. To the right, there is a date selection area with 'Today', 'Yesterday', 'This Week', 'Last Week', 'This Month', and 'Last Month' options. A calendar for April 2019 is displayed, with the 1st highlighted. Below the calendar are two radio button options: 'days up to today' and 'days starting today'. At the bottom, a table titled 'Logs' displays the following data:

| RECORD TIME              | TYPE                 | ID   |
|--------------------------|----------------------|------|
| 2019-01-01T00:29:50.000Z | Time Synchronization | 1939 |
| 2019-01-01T01:44:50.000Z | Time Synchronization | 1940 |
| 2019-01-01T02:59:50.000Z | Time Synchronization | 1941 |
| 2019-01-01T04:29:51.000Z | Time Synchronization | 1942 |
| 2019-01-01T05:44:50.000Z | Time Synchronization | 1943 |
| 2019-01-01T07:14:50.000Z | Time Synchronization | 1944 |

Obrázek 3.9: Funkce „Logs“ online portálu

portálu nebyly pevně dané požadavky na jeho funkce a navíc se jedná o ukládání a práci s velkým množstvím nestrukturovaných dat naměřených PQ monitorem.

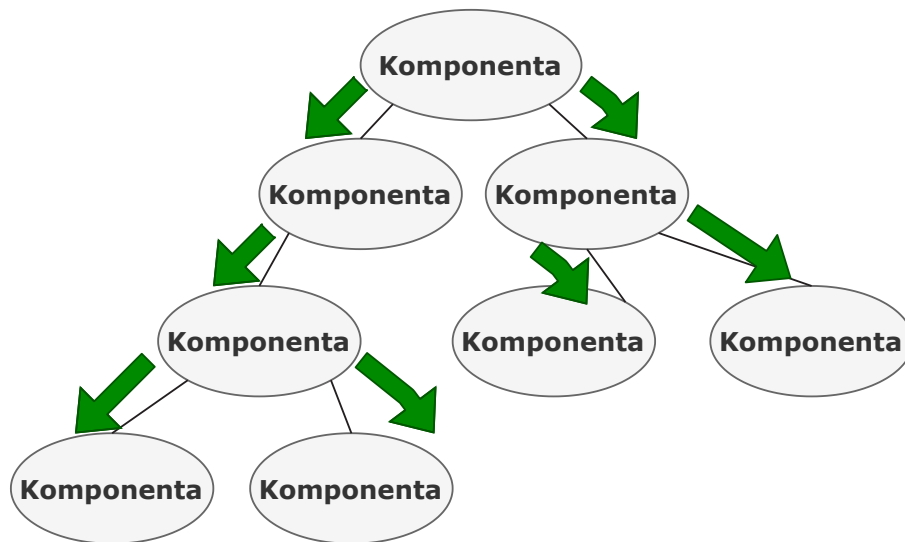
Základem databáze MongoDB je dokument, který má unikátní id a skládá se z párů klíč - hodnota. Dokument také může obsahovat odkaz na dokumenty z jiných kolekcí. Databáze MongoDB se skládá z kolekcí, které mají seznam vlastních dokumentů a každý dokument je prvkem reálného světa. Strukturu dokumentu popisuje zpravidla schéma kolekce.[13]

Na obrázku 3.13 je implementován model databáze MongoDB, který obsahuje pět kolekcí: archive, device, chartData, logs a user.

Schéma kolekce „user“ se skládá z atributů: id, jméno, příjmení, emailová adresa, heslo a role. Každý dokument z této kolekce „user“ má svůj unikátní klíč, což je id, podle kterého se dá vyhledat určitý dokument. Každý dokument z této kolekce má také atribut „role“, který dělí uživatele dle typu (běžný uživatel nebo správce systému).

Schéma kolekce „archive“ obsahuje atributy title a tags, tedy název archivu PQ monitoru a tag naměřených hodnot. V databázi se může uchovávat např. dokument s hodnotami archiv - PQMain a tag - f (f je frekvence).

Další schéma kolekce „device“ je implementované za účelem uchování informací



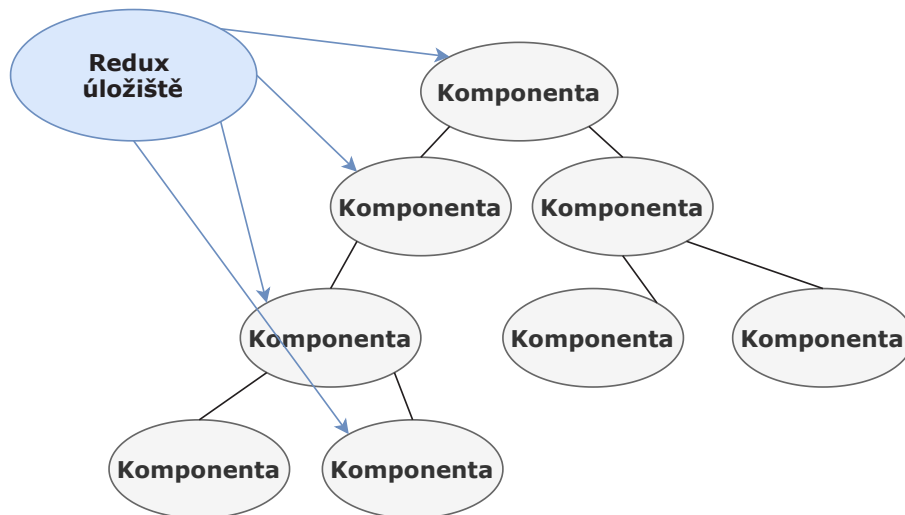
Obrázek 3.10: Schéma předávání dat mezi komponentami React

o zařízeních. Atribut kolekce title popisuje název zařízení, model popisuje model zařízení a location popisuje místo, kde se zařízení nachází.

Kolekce „chartData“ byla implementována za účelem uchování zparsovaných dat z PQ monitorů dle časového intervalu (15 min., 30 min., 1 hod., 1 den) pro další vykreslení hodnot do grafu. Atributy této kolekce jsou device, data, archive, recordTime a interval. Atribut recordTime uchovává datum a čas měření hodnot PQ monitorem. Atribut interval ukazuje, v jakém intervalu byla data zparsována – 15 min., 30 min., 1 hod. nebo 1 den.

Posledním schématem kolekce je „logs“, který byl implementovaný, aby uchovával informace ze souboru Logs.csv. To jsou informace o činnosti PQ monitoru - hlášení chyb a varování. Schéma má následující atributy: type - typ hlášení, id - identifikační klíč hlášení, device - zařízení, jehož se hlášení týká a recordTim - čas hlášení.

Cílem této bakalářské práce byla realizace online portálu pro skladování a analýzu dat naměřených PQ monitorem. Proto byla použita data reálně naměřená PQ monitorem. K dispozici jsem měl archivy dat z PQ monitoru: ElectricityMeter.csv, Main.csv, PQMain.csv a Log.csv, které obsahují velmi vysoký počet hodnot a nebylo možné vykreslovat všechny naměřené hodnoty z archivů do grafů nebo tabulek. Proto bylo hlavním cílem parsování těchto dat zprůměrovat naměřené hodnoty do intervalů 15 min., 30 min., 1 hod. a 1 dne. Parsování archivů jsem realizoval pomocí vlastních algoritmů. Logika parsování těchto dat je implementovaná v souboru index.ts, který posílá data v závislosti na jejich typu do dalších parserů, tj. do souborů



Obrázek 3.11: Lokální úložiště Redux

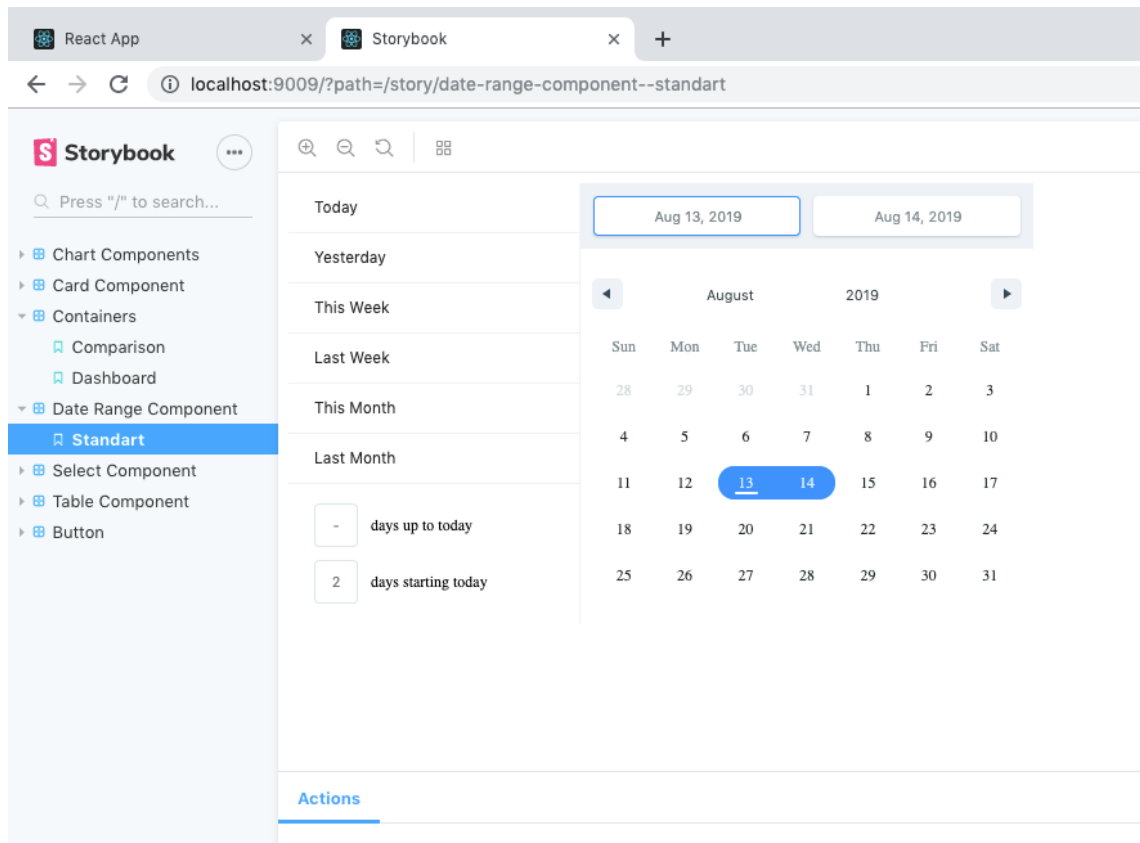
electricityMeterParser.ts, mainParser.ts, pqMainParser.ts a logsParser.ts. Soubor ElectricityMeter.csv obsahuje stále rostoucí hodnoty jednofázové, třífázové, kladné i záporné hodnoty činné a jalové energie, kde kladné hodnoty popisují spotřebovanou energii, záporné naopak energii vyrobenou nebo dodanou. Data bylo potřeba jednak zparsovat podle tagu hodnot (to jsou tagy 3EP, 3EQ, EP1, EQ1, EP1+, EP1-, EP2, EQ2, EP2+, EP2-, EP3, EQ3, EP3+, EP3-) a také zprůměrovat do intervalů 15 min., 30 min., 1 hod. a 1 dne.

Funkce electricityMeterParser v souboru electricityMeterParser.ts je určena pro parsování dat podle tagů a podle intervalů. Data v souboru electricityMeter.csv naměřená v intervalu 15 min. ve formátu: úterý 22. února 2018; 2:30:00 PM, úterý 22. února 2018; 2:45:00 PM, úterý 22. února 2018; 3:00:00 PM atd.

Při parsování dat v intervalu 15 min. funkce načítá každou hodnotu ze souboru. Při parsování dat v intervalu 30 min. načítá funkce ze souboru hodnoty naměřené každých 30 minut (2:30:00 PM, 3:00:00 PM, 3:30:00 PM atd.). Při parsování dat v intervalu 1 hod. funkce načítá hodnoty naměřené každou hodinu (2:30:00 PM, 3:30:00 PM, 4:30:00 PM atd.) a při parsování dat v intervalu 1 den načítá funkce hodnoty naměřené každou půlnoc (úterý 22. února 12:00:00 AM, středa 23. února 12:00:00 AM atd.). Všechna zparsovaná data se pak ukládají do kolekce chartData v databázi MongoDB.

Soubor PQMain.csv obsahuje rostoucí i klesající hodnoty naměřené PQ monitorem, které poskytují informace pro vyhodnocení kvality elektrické energie, tj. hodnoty veličin frekvence, napětí, proudu, činného výkonu, jalového výkonu, celkového



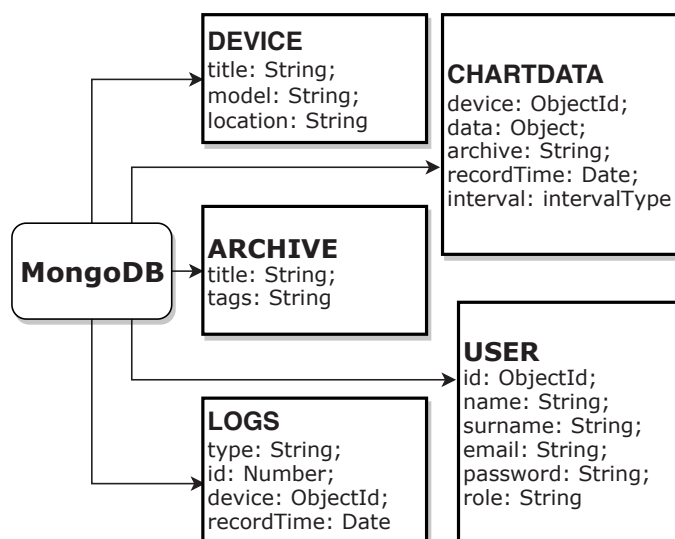


Obrázek 3.12: Příklad použití StoryBook

harmonického zkreslení atd.

Funkce `pqMainParser` v souboru `pqMainParser.ts` zprůměruje všechny hodnoty naměřené za 15 min., 30 min., 1 hod., 1 den. Funkce má proměnné `minDate` a `maxDate`, což jsou hodnoty začátku a konce každých naměřených 15 min., 30 min., 1 hod. nebo 1 dne. Pomocí těchto proměnných funkce od `minDate` do `maxDate` poskládá naměřené hodnoty a součet potom podělí jejich počtem. Výsledkem funkce budou hodnoty zprůměrované do intervalů a tyto hodnoty se pak uloží do kolekce databáze `chartData`.

Soubor `Logs.csv` obsahuje pouze hlášení chyb a varování týkající se činnosti PQ monitoru. Funkce `logsParser` zparsovává informace podle `data`, `typu` a `id` hlášení. Tyto informace se pak uloží do kolekce „logs“ databáze MongoDB.



Obrázek 3.13: Model databáze MongoDB

### 3.4 Testování online portálu

Jedním z důležitých kroků vývoje každého softwaru je testování. To softwaru zjišťuje informace o výkonnosti a rychlosti práce systému. Hlavními kritérii pro testování online portálu jsou kolik dat je portál schopný zpracovat najednou, kolik uživatelů může současně používat online portál a jak dlouho trvá zpracovávání takových informací a kolik paměti procesoru zabírá.

Pro otestování backendu online portálu byl zvolen PM2. PM2 je manažer monitorování operací pro Node.js, který umožňuje sledovat výkonnost CPU a MEM během posílání velkého množství požadavků na backend online portálu.[14]

Dalším nástrojem, který byl použit pro testování výkonnosti online portálu je Artillery.io. Tento nástroj umožňuje otestovat výkonnost aplikace pomocí HTTP, což znamená, že během každých 60. vteřin se posílalo 100 nebo 500 požadavků do API serverové části online portálu.[15]

Prostřednictvím Artillery.io lze sledovat počet virtuálních uživatelů, kteří byli vytvořeni během testování - „Scenarios launched“, počet uživatelů, kteří splnili úlohu testování - „Scenarios completed“, počet HTTP požadavků a odpovědí - „Requests completed“, průměrný počet HTTP požadavků, které byly provedené za dobu testování - „RPS sent“, hodnotu „Request latency“, která popisuje rychlost všech odpovědí na požadavky HTTP a hodnotu „Codes“, která popisuje stavové kódy HTTP odpovědí.[15]

Dále jsou uvedeny výsledky testování backendu online portálu prostřednictvím Artillery.io, který posílá každou vteřinu 100 (obrázky v příloze A.1, A.2, A.5, A.6) nebo 500 (obrázky v příloze A.3, A.4, A.7, A.8) HTTP požadavků do API portálu. Backend online portálu je realizován v softwarovém systému Node.js, kde jsou všechny požadavky obsluhované asynchronně v jednom vláknu, aby nedošlo k blokadě po dobu zpracování požadavku. Testy na obrázcích v příloze A.1, A.2, A.3, A.4 byly provedené právě v prostředí, kde Node.js zpracovává požadavky asynchronně v jednom vlákne.

Pro využití více jader procesoru je potřeba spustit více instalací Node.js, z nichž každá musí poslouchat na jiném portu.[16]

Další testy na obrázcích v příloze A.5, A.6, A.7, A.8 jsou realizované při spuštění více instancí Node.js, které poslouchají na různých portech. To vede k odlišným výsledným hodnotám testu Node.js a je zřejmé, že zatížení MEM a CPU je menší, protože balancování operací je rozděleno mezi několik instancí.

Na obrázcích A.2 a A.4 je vidět, že počet vytvořených uživatelů a uživatelů, kteří splnili úlohu testování pro 100 HTTP požadavků je stejný – 6000, ale pro 500 požadavků je počet uživatelů, kteří dokončili úlohu testování menší, než počet všech vytvořených uživatelů. Během testování 500 HTTP požadavků systém nahlásil 23 chyby, ale pro testování na 100 požadavcích systém pouze vypsal pro všechny požadavky stavový kód 200 - „OK“, což je standardní odpověď pro úspěšný HTTP požadavek.

## 4 Závěr

Na začátku této práce byl stanoven analytický cíl, kterým je seznámení se s problematikou kvality elektrické energie, s typickými strukturami dat v archivu PQ monitoru a s dostupnými metodami jejich odečtu, přenosu a ukládání.

Pro realizaci analytického cíle v rešerši k této bakalářské práci v podkapitole 2.1 je uvedena definice a popis parametrů kvality elektrické energie, pak v podkapitole 2.2 je uvedena analýza struktury dat PQ monitoru, metod jejich odečtu, přenosu a ukládání. Dále v podkapitole 2.3 je uvedena analýza webových aplikací Wattics.com, Izenda.com a Energycloud.

Praktickým cílem této práce bylo navrhnout a implementovat online portál a využít vhodné technologie pro vývoj webových služeb s funkcemi pro analýzu a sdílení dat se vzdálenými klienty.

Realizace praktického cíle je uvedena v kapitole 3. Zvolil jsem vhodné technologie pro vývoj portálu tak, aby co nejlépe odpovídaly navrženým funkcím. Pro analýzu dat PQ monitoru jsou implementována rozhraní - „Comparison“, „Electricity meter“, „PQ Main“, „Main“, „Energy cost“ a „Logs“, které jsou popsány v podkapitole 3.1. Online portál obsahuje rozhraní pro dva typy uživatelů - pro běžného uživatele a pro správce systému, což umožňuje otestování všech funkcí systému.

Dalším krokem bude zlepšit online portál a přidat funkci analýzy dat kvality elektrické energie a hlášení všech provozních událostí v reálném čase.

## Literatura

- [1] S. Chattopadhyay, M. Mitra a S. Sengupta. *Electric Power Quality*. Springer Dordrecht Heidelberg London New York, 2011.
- [2] KMB. *ARTIQ 144 PQ-A*. URL: <http://www.kmb.cz/index.php/cs/analyzatory-kvality/analyzator-kvality-50160-artiq-144>. (cit. 2019-20-08).
- [3] B. Forta. *Sams Teach Yourself SQL in 10 Minutes, Fourth Editiona*. 2013. ISBN: 9780672336072.
- [4] G. Harrison. *Next Generation Databases NoSQL, NewSQL, and Big Data*. 2015. ISBN: 9781484213308.
- [5] MongoDB. *Comparing PostgreSQL and MongoDB*. URL: <https://www.mongodb.com/compare/mongodb-postgresql>. (cit. 2019-20-08).
- [6] Wattics. *Energy management solutions aligned with your business*. URL: <https://www.wattics.com/turnkey-solutions/>. (cit. 2019-20-08).
- [7] Izenda. *Enhance your application with innovative reporting and analytics*. URL: <https://www.izenda.com/solutions/>. (cit. 2019-20-08).
- [8] BLUELINE. *Energycloud for energy providers and their customers*. URL: <https://www.bluelineinnovations.com/for-energy-providers>. (cit. 2019-20-08).
- [9] N. Rozentals. *Mastering TypeScript - Second Edition*. Packt Publishing Ltd., 2017. ISBN: 9781786468710.
- [10] A. Mardan a J. Sonmez. *React Quickly PAINLESS WEB APPS WITH REACT, JSX, REDUX, AND GRAPHQL*. Manning Publications Co., 2017. ISBN: 9781617293344.
- [11] StoryBook. *StoryBook Documentation*. URL: <https://storybook.js.org/docs/basics/introduction/>. (cit. 2019-20-08).

- [12] MaterialUI. *Material UI*. URL: <https://material-ui.com/>. (cit. 2019-20-08).
- [13] MongoDB. *Documentation*. URL: <https://docs.mongodb.com/>. (cit. 2019-20-08).
- [14] PM2. *PM2 Documentation*. URL: <http://pm2.keymetrics.io/docs/usage/pm2-doc-single-page/>. (cit. 2019-20-08).
- [15] Artillery.io. *Artillery Docs*. URL: <https://artillery.io/docs/>. (cit. 2019-20-08).
- [16] A. Young, B. Meck, M. Cantelon, T. Oxley a M. Harter. *Node.js in Action 1st*. Manning Publications Co., 2013. ISBN: 9781617290572.

## A Výsledky testování online portálu

| App name | id | version | mode | pid   | status | restart | uptime | cpu | mem      | user     | watching |
|----------|----|---------|------|-------|--------|---------|--------|-----|----------|----------|----------|
| index    | 0  | 1.0.0   | fork | 50729 | online | 0       | 51s    | 18% | 103.4 MB | kushchov | disabled |

Obrázek A.1: Testování PM2: 100 HTTP požadavků

```
All virtual users finished
Summary report @ 20:52:39(+0200) 2019-08-24
Scenarios launched: 6000
Scenarios completed: 6000
Requests completed: 6000
RPS sent: 99.21
Request latency:
  min: 1.4
  max: 95.3
  median: 2
  p95: 4.9
  p99: 24.4
Scenario counts:
  0: 6000 (100%)
Codes:
  200: 6000
```

Obrázek A.2: Testování Artillery.io: 100 HTTP požadavků

| App name | id | version | mode | pid   | status | restart | uptime | cpu | mem      | user     | watching |
|----------|----|---------|------|-------|--------|---------|--------|-----|----------|----------|----------|
| index    | 0  | 1.0.0   | fork | 56603 | online | 0       | 69s    | 35% | 274.3 MB | kushchov | disabled |

Obrázek A.3: Testování PM2: 500 HTTP požadavků

```

All virtual users finished
Summary report @ 21:02:18(+0200) 2019-08-24
Scenarios launched: 30000
Scenarios completed: 29977
Requests completed: 29977
RPS sent: 424.75
Request latency:
  min: 1.4
  max: 16133.4
  median: 21.5
  p95: 8246.2
  p99: 15671.8
Scenario counts:
  0: 30000 (100%)
Codes:
  200: 29977
Errors:
  ECONNRESET: 23

```

Obrázek A.4: Testování Artillery.io: 500 HTTP požadavků

| App name | id | version | mode    | pid   | status | restart | uptime | cpu  | mem     | user     | watching |
|----------|----|---------|---------|-------|--------|---------|--------|------|---------|----------|----------|
| index    | 0  | 1.0.0   | cluster | 52337 | online | 0       | 60s    | 4.2% | 96.1 MB | kushchov | disabled |
| index    | 1  | 1.0.0   | cluster | 52338 | online | 0       | 59s    | 4.2% | 94.9 MB | kushchov | disabled |
| index    | 2  | 1.0.0   | cluster | 52339 | online | 0       | 59s    | 3.8% | 93.6 MB | kushchov | disabled |
| index    | 3  | 1.0.0   | cluster | 52344 | online | 0       | 59s    | 5.4% | 65.8 MB | kushchov | disabled |

Obrázek A.5: Testování PM2: 100 HTTP požadavků při spuštění více instalací Node.js

```

All virtual users finished
Summary report @ 20:55:28(+0200) 2019-08-24
Scenarios launched: 6000
Scenarios completed: 6000
Requests completed: 6000
RPS sent: 99.21
Request latency:
  min: 1.8
  max: 124.8
  median: 2.8
  p95: 7.3
  p99: 26
Scenario counts:
  0: 6000 (100%)
Codes:
  200: 6000

```

Obrázek A.6: Testování Artillery.io: 100 HTTP požadavků při spuštění více instalací Node.js



| App name | id | version | mode    | pid   | status | restart | uptime | cpu   | mem      | user     | watching |
|----------|----|---------|---------|-------|--------|---------|--------|-------|----------|----------|----------|
| index    | 0  | 1.0.0   | cluster | 52337 | online | 0       | 3m     | 13.5% | 163.5 MB | kushchov | disabled |
| index    | 1  | 1.0.0   | cluster | 52338 | online | 0       | 3m     | 15.5% | 102.5 MB | kushchov | disabled |
| index    | 2  | 1.0.0   | cluster | 52339 | online | 0       | 3m     | 13.5% | 102.1 MB | kushchov | disabled |
| index    | 3  | 1.0.0   | cluster | 52344 | online | 0       | 3m     | 17%   | 100.6 MB | kushchov | disabled |

Obrázek A.7: Testování PM2: 500 HTTP požadavků při spuštění více instalací Node.js

```

All virtual users finished
Summary report @ 20:58:04(+0200) 2019-08-24
  Scenarios launched: 30000
  Scenarios completed: 30000
  Requests completed: 30000
  RPS sent: 367.47
  Request latency:
    min: 3.7
    max: 475.8
    median: 12.4
    p95: 45.2
    p99: 153.7
  Scenario counts:
    0: 30000 (100%)
  Codes:
    200: 30000

```

Obrázek A.8: Testování Artillery.io: 500 HTTP požadavků při spuštění více instalací Node.js