



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

Vizualizace a tvorba analýz velkých objemů dat měření spotřeby elektrické energie

Bakalářská práce

Studijní program: B2646 – Informační technologie
Studijní obor: 1802R007 – Informační technologie
Autor práce: **Jan Kálecký**
Vedoucí práce: Ing. Jan Kraus, Ph.D.



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Jméno a příjmení: **Jan Kálecký**
Název práce: **Vizualizace a tvorba analýz velkých objemů dat měření spotřeby elektrické energie**
Zadávající katedra: **Ústav mechatroniky a technické informatiky**
Vedoucí práce: **Ing. Jan Kraus Ph.D.**
Rozsah práce: **30—40 stran**

Zásady pro vypracování:

1. Seznamte se s nejčastěji používanými technikami pro vyhodnocování a grafickou prezentaci výsledků v běžných systémech pro správu spotřeby elektrické energie, nalezené postupy důkladně zanalyzujte a přehledně sepište.
2. V prostředí .NET vyberte vhodné nástroje pro efektivní, přehledné a uživatelsky přívětivé zobrazení posloupností hodnot archivovaných veličin a nebo vytvořte vlastní.
3. Navrhněte koncepci architektury jednoduché aplikace pro vyhodnocování archivních dat a v ní demonstруйте správnou funkci Vámi navržených či upravených komponent.
4. V závěru shrňte dosažené výsledky a diskutujte možnosti dalšího rozvoje vytvořených komponent či jejich nasazení ve větším systému.

Seznam odborné literatury:

- [1] KRAUS, Jan a Martin BLÍŽKOVSKÝ. Uživatelská příručka aplikace ENVIS v. 1.2 [online]. 2015. [cit. 2015-1-08]. 1.2. Dostupné z: <http://www.kmb.cz/>
- [2] MENEZES, A. C., et al. Estimating the energy consumption and power demand of small power equipment in office buildings. *Energy and Buildings*, 2014, 75: 199-209.
- [3] MAGOULÈS, Frédéric; ZHAO, Hai-Xiang. *Data mining and machine learning in building energy analysis*. John Wiley & Sons, 2016.

V Liberci dne

.....
Ing. Jan Kraus Ph.D.

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 (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

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

Bakalářskou práci jsem vypracoval samostatně 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.

28. 4. 2019

Jan Kálecký

Vizualizace a tvorba analýz velkých objemů dat měření spotřeby elektrické energie

Abstrakt

Obsah práce tvoří analýza existujících systémů pro správu spotřeby elektrické energie a z nich odvozené nejčastěji používané techniky pro grafickou prezentaci dat. Práce zkoumá nástroje, kterými lze docílit uživatelsky přívětivé vizualizace. Všechny poznatky uplatňuje při návrhu jednoduché webové aplikace pro zobrazení a analýzu archivních dat a při její implementaci. Dále popisuje funkce vytvořeného systému a knihovny nebo komponenty, které program využívá. Závěrečné kapitoly hovoří o problémech navrženého systému a jeho případném zdokonalení.

Klíčová slova

vizualizace dat, analýza dat, spotřeba elektrické energie, návrh webové aplikace, systém správy energie

Visualisation and big data analysis from a typical energy management system

Abstract

This thesis analyzes existing management systems for the consumption of electrical energy and its most frequently used techniques for graphical data presentation. Furthermore, it explores tools which provide favourable visualisation to the user. All of the findings are used to design and implement a simple web application for displaying and analysing archive data. Chosen tool libraries as well as key system functions are being described. Final chapters speak about imperfection and drawbacks of the designed system and its potential refinement.

Key words

data visualisation, big data analysis, electrical energy consumption, web application design, energy management system

Obsah

Seznam obrázků.....	8
Úvod	9
1 Správa spotřeby elektrické energie	10
1.1 Dostupná řešení a systémy.....	10
1.1.1 Energomonitor.....	10
1.1.2 Monsol	11
1.1.3 e3m	11
1.1.4 FLOWBOX.....	12
1.1.5 UBC Energy & Water Services	13
1.1.6 ENVIS	13
1.1.7 Mervis	14
1.2 Shrnutí využívaných technik a funkcí systémů.....	15
1.2.1 Zobrazení dat za zvolený časový interval	15
1.2.2 Přehled nejdůležitějších informací	16
1.2.3 Agregace a normalizace dat	16
1.2.4 Srovnání s historickými daty.....	17
1.2.5 Sankeyův diagram	17
1.2.6 Heat mapa	17
1.2.7 Upozornění a reporty	18
1.2.8 Export dat.....	19
2 Vybrané nástroje pro vizualizaci v prostředí .NET.....	20
2.1 Nástroje pro desktop.....	20
2.1.1 Nevron Open Vision.....	20
2.1.2 teeChart.....	21
2.1.3 LiveCharts.....	21
2.1.4 OxyPlot	21
2.2 Nástroje pro ASP.NET	22
2.2.1 D3.js.....	22
2.2.2 Plotly.js	22
2.2.3 Highcharts	22
2.2.4 Echarts	22
2.2.5 Chartist.js	23
2.2.6 Grafana a InfluxDB	23
3 Návrh a řešení aplikace	25
3.1 Návrh architektury	25

3.2 Zvolené nástroje	26
3.3 Datová vrstva	28
3.4 Serverová část aplikace	29
3.4.1 Model a databázový kontext	29
3.4.2 Pohledy	30
3.4.3 Controller	30
3.5 Klientská část aplikace	31
3.5.1 Výběr dat.....	31
3.5.2 Způsob zobrazení	33
3.5.3 Agregace a porovnávání	34
3.5.4 Interakce s grafem a další možnosti úpravy	36
3.6 Problematické úlohy	37
3.7 Doplnující funkce systému	38
Závěr	40
Literatura	41

Seznam obrázků

Obrázek 1: Nástěnka v systému Energomonitor.....	10
Obrázek 2: Sankeyův diagram rozložení výkonu v elektrárně (e3m).....	11
Obrázek 3: Náhled čtvrt hodinového maxima v systému FLOWBOX.....	12
Obrázek 4: Histogram z rozšíření pro program ENVIS [12].....	14
Obrázek 5: Přehled v systému Mervis	15
Obrázek 6: Alarm v systému Mervis	18
Obrázek 7: Možnosti exportování dat (UBC SkySpark).....	19
Obrázek 8: Příklad ukazatele z nástroje NOV pro kritické stavy [14]	20
Obrázek 9: Různé typy grafů v jednom zobrazení.....	21
Obrázek 10: Nástěnka ve ukázkové aplikaci systému Grafana	24
Obrázek 11: Architektura MVC	25
Obrázek 12: Možnosti výběru v aplikaci ASP.NET Core MVC	26
Obrázek 13: Kalendář z nástroje Air Datepicker	27
Obrázek 14: Základní pohled aplikace.....	31
Obrázek 15: Možnosti výběru z monitorovaných míst a veličin	32
Obrázek 16: Volba kalendáře a zobrazení posledních hodnot.....	32
Obrázek 17: Sloupcové grafy vertikální skládaný (vlevo) a horizontální sdružený	33
Obrázek 18: Krabicové diagramy – srovnání dvou sérií a dílčí hodnoty	34
Obrázek 19: Zobrazení v podobě heat mapy za měsíc únor	34
Obrázek 20: Nastavení vizualizace – typ grafu, agregace a její funkce, porovnávání	35
Obrázek 21: Porovnání s historickými hodnotami.....	36
Obrázek 22: Přiblížení v grafu výběrem oblasti myší.....	36
Obrázek 23: Upravené barevné spektrum heat mapy v Chart Studiu	37
Obrázek 24: Stav dnešních hodnot, cena spotřeby a porovnání (Energomonitor).....	39

Úvod

Využívání elektrické energie je v domácnostech i v průmyslové výrobě zcela běžnou věcí. Jednou z hlavních překážek pro její efektivní užívání je absence systémů pro správu spotřeby. To může být způsobeno neexistencí takových systémů, jejich nedokonalostí, nebo špatnou dostupností. Kvalitní systém přitom může přispět k optimalizaci provozu a snížení nákladů díky analýze využití energie v průběhu sledovaného časového období, a to např. během jednotlivých dnů a týdnů, nebo zobrazením kritických hodnot a odchylek od dlouhodobého průměru apod.

Cílem této bakalářské práce je navrhnout aplikaci v prostředí .NET, která graficky přehledně vyhodnocuje a prezentuje poskytnutá data. Aplikace se nestará o zaznamenávání nových hodnot, ale pracuje pouze s archivními veličinami, které získává z databáze. Navržený systém využívá nejčastějších technik pro vizualizaci a analýzu dat měření spotřeby elektrické energie, s nimiž seznamuje na základě informací z již fungujících a dostupných systémů. Vizualizaci dat v aplikaci zprostředkují nástroje vybrané ze širší skupiny otestovaných knihoven a komponent. Pro demonstraci správného chodu systému jsou používána data z aplikace UBC Energy & Water Services od Univerzity Britské Kolumbie.

První kapitola se týká systémů správy elektrické energie a způsobů zpracování dat, vizualizace a analýzy. Uvádí konkrétní aplikace, jejich funkce a případně používané nástroje. Z nabytých poznatků o těchto systémech kapitola podává stručný přehled o nejčastěji využívaných technikách pro grafickou prezentaci informací uživateli. Další část je věnovaná vybraným komponentám a knihovnám, které zjednodušují proces zobrazování archivovaných dat, a to jak pro desktopové programy, tak pro webové aplikace. Poslední kapitola a její podkapitoly se věnují postupu při návrhu a implementaci jednoduché webové aplikace, na níž jsou demonstrovány způsoby vyhodnocování archivních dat skrze upravený nástroj. V práci jsou zmíněny i zjištěné problémy a dále funkce, které by systému prospěly, ale aplikace je z popsáných důvodů neobsahuje.

1 Správa spotřeby elektrické energie

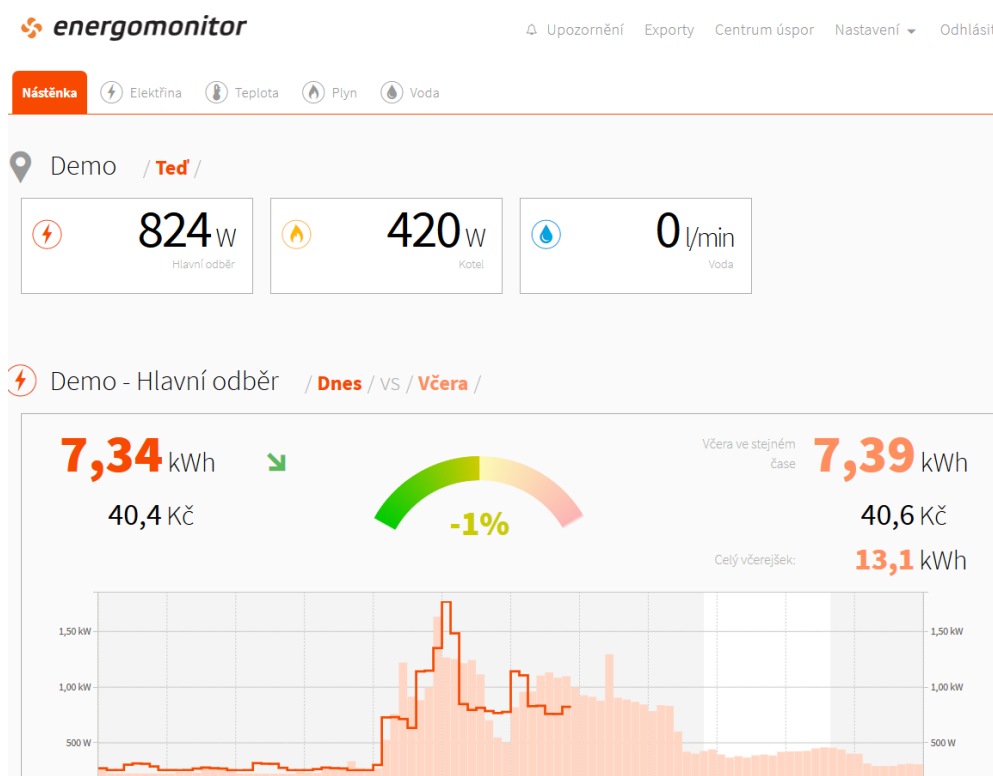
Kapitola se zaměřuje na přehledný popis současných systémů, které analyzuje, a poskytuje přehled o nejčastěji používaných technikách pro vizualizaci a grafickou prezentaci výsledků. Cílem kapitoly je dále o systémech získat poznatky, které budou sloužit pro tvorbu vlastní aplikace.

1.1 Dostupná řešení a systémy

V následujících podkapitolách jsou popsány vybrané systémy pro správu spotřeby elektrické energie, a to jak české, tak zahraniční. Vše probíhá na základě zkoumání jejich dostupných demo verzí, které by měly dostatečně posloužit k analýze klíčových funkcí jednotlivých systémů.

1.1.1 Energomonitor

Energomonitor je česká společnost, která se zabývá vývojem a distribucí produktů pro přehled a kontrolu spotřeby energií do domácností a firem. Systém Energomonitor předává informace o spotřebě a nákladech na energie v reálném čase, kdy se data ukládají každou jednu minutu. Uživatel si může nastavit upozornění na problémy, kontrolovat data v týdenních nebo



Obrázek 1: Nástěnka v systému Energomonitor

měsíčních reportech. Přehled lze zobrazit na libovolné platformě. Mezi technologiemi, které využívá, je např. PostgreSQL, InfluxDB nebo HiveMQ [8].

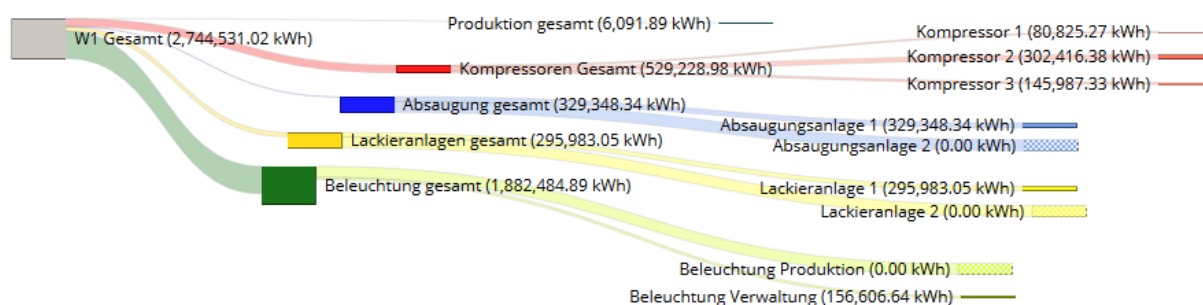
Aplikace se skládá z několika částí, v nástěnce můžeme nalézt shrnutí všech informací (viz *Obrázek 1*), tedy kolik činí momentální elektrický výkon, srovnání s předchozím dnem včetně procentuálního nárůstu, heat mapu pro srovnání celého týdne, nebo celkovou cenu za poslední měsíc apod. V záložkách pro jednotlivé druhy sledovaných energií se nachází detail spotřeby nebo nákladů za vybrané období. V sekci upozornění jsou uchovány všechny zprávy dle nastavených alarmů. Systém poskytuje i možnost exportu dat z libovolného období do formátu XLS nebo CSV. Data lze agregovat po patnácti minutách, jedné hodině, jednom dni, nebo po jednom měsíci. Aplikace dále zahrnuje tipy pro snížení spotřeby ve chvíli, kdy ve sledovaném objektu není nikdo přítomen.

1.1.2 Monsol

Monsol je španělská společnost, která se zabývá monitorováním solárních a fotovoltaických panelů a elektráren. Jejich monitorovací systém nabízí údaje o elektrickém výkonu a elektrické energii pro jednotlivá sledovaná místa za dané období, a to v reálném čase. Vizualizaci provádí prostřednictvím javascriptového nástroje HighCharts. Systém využívá zejména spojnicové a sloupcové grafy, ke kterým přikládá tabulku se zobrazovanými hodnotami a dále např. procentuální odchylku od průměrné hodnoty. V systému dále nalezneme nastavení upozornění, generování reportů, export dat do XLS, export grafu a mapu všech sledovaných míst s aktuálním výkonem.

1.1.3 e3m

S cílem efektivního využití energií vznikl systém e3m, který sbírá a verifikuje data, následně je vyhodnocuje a snaží se pomoci řešit problémy s energiemi. Systém mohou využít jednotlivé objekty jako např. administrativní budovy, bazény, ale také celé obchodní řetězce, průmyslové závody nebo elektrárny [9].



Obrázek 2: Sankeyův diagram rozložení výkonu v elektrárně (e3m)

V demo verzi nalezneme několik monitorovaných míst, mezi nimi zejména elektrické elektrárny v Německu, o nichž na úvodní straně najdeme aktuální informace o elektrickém výkonu, srovnání měsíčních hodnot za poslední rok v podobě sloupcového grafu, porovnání jednotlivých roků ve spojnicovém grafu nebo údaje o nákladech. Každé z míst lze zobrazit v detailnějším pohledu, kde se navíc objevují koláčové diagramy pro zobrazení procentuální části z celkové výroby. Pomocí Sankeyova tzv. bilančního diagramu (viz *Obrázek 2*) se dozvídáme, z jakých složek je dosaženo celkového elektrického výkonu. V přehledu pro konkrétní místo e3m uvádí vedle historických hodnot také prognózu na následující týden. Údaje za posledních 365 dnů lze vyčíst z heat mapy – kde na ose X jsou vyneseny jednotlivé dny, osa Y rozděluje dny do intervalů délky zhruba patnácti minut, barva údaje v daný moment určuje jeho konkrétní hodnotu na intervalu barevného prostoru heat mapy.

1.1.4 FLOWBOX

FLOWBOX nabízí kontrolu smysluplného toku všech druhů energií v různých objektech dle [10] s cílem zvýšit efektivitu a snížit náklady. Na webu nenalezneme demo ukázkou systému,



Obrázek 3: Náhled čtvrt hodinového maxima v systému FLOWBOX

ale nahrazují ji případové studie dle způsobu jeho využití. Například pro výrobní podnik jsou instalovány senzory pro měření a výsledná integrace technologií umožňuje monitoring, analýzu, predikci řízení a optimalizaci. Na *Obrázek 3* vidíme vyhodnocení čtvrt hodinového maxima, osa Y je rozdělena na činný a jalový odběr, níže ve sloupcovém grafu vidíme srovnání čtvrt hodinových maxim během jednoho dne.

1.1.5 UBC Energy & Water Services

Univerzita Britské Kolumbie (UBC) poskytuje data o využití energií v budovách jejího kampusu. Vizualizaci a analýzu zprostředkovává platforma SkySpark, která dle [13] pomáhá nalézt vše potřebné z různých smart systémů. Systém UBC poskytuje čtyři různé aplikace SkySparku, mezi nimi *Buildings*, *Energy*, *Historian* a *Weather*.

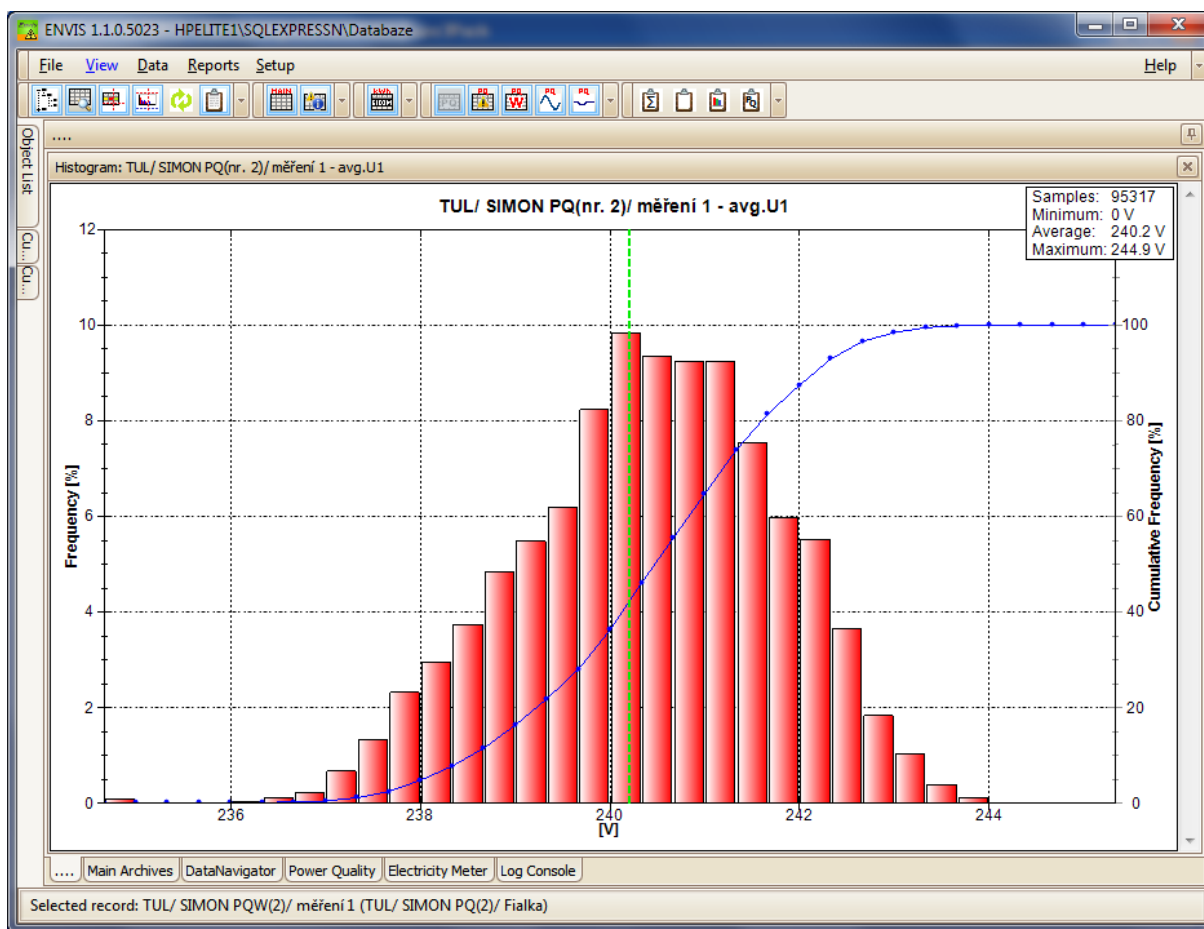
Aplikace *Buildings* udává kompletní údaje o jednotlivých monitorovaných budovách, např. výměr plochy budovy, EUI (energy utilization intensity, nebo-li míra využití energie v závislosti na ploše) jednotlivých místností, graf ročních EUI, graf vytížení elektrické a termální sítě (pro zvolené období udává dobu v hodinách, po kterou síť nabývala hodnoty vytížení, např. za poslední rok dosahovala hodnoty 29 kW po celkovou dobu 1020 hodin). V aplikaci *Energy* lze sledovat data ze zvolených míst za vybraný časový interval, a to v rámci dnů, týdnů, měsíců, kvartálů nebo v intervalu několika let. Nastavení módu vizualizace nabízí volbu mezi spojnicovým, sloupcovým a koláčovým grafem, výběr sledované veličiny, agregaci dle libovolného intervalu (např. den, týden, kvartál) dle agregační funkce (např. průměr, maximum, median). Data lze normalizovat dle plochy sledované budovy nebo dle venkovní teploty a srovnat je s údaji za minulý měsíc, rok apod. Porovnávat můžeme i jednotlivé dny ve spojnicovém grafu nebo za pomoci heat mapy. Další funkcí systému je srovnání všech vybraných budov v přehledné tabulce, která v řádcích zobrazuje budovy, ve sloupcích pak sledované veličiny. Další aplikací systému je *Historian*, který sleduje všechny měřené součásti budovy, např. elektroměr, teploměr vzduchu a vody, a to za vybrané časové období. Aplikace *Weather* zahrnuje přehled počasí pro vybraný den, včetně prognózy a náhledu historických dnů. Celý systém podporuje export zvolených dat do mnoha formátů, včetně CSV, PDF nebo SVG a právě pro široký záběr dat z různých míst po dlouhou dobu jsem zvolil UBC jako zdroj pro testování vlastní aplikace.

1.1.6 ENVIS

Měřicí přístroje firmy KMB systems s.r.o. spolu s aplikací ENVIS nabízí řešení správy elektrické energie pomocí vyhodnocování naměřených dat o spotřebě energie nebo kvalitě

napětí [10]. Systém nabízí interaktivní vizualizaci vybraných dat, jejich analýzu, vytváří reporty a poskytuje také export dat [1].

Pro rozšíření funkcionality aplikace je možné nainstalovat některý z dostupných zásuvných modulů, mezi nimi například histogram rozšiřující možnosti vizualizace (viz Obrázek 4) nebo EOnlineGadgets pro miniaplikace na ploše operačního systému (OS) [11]. Na rozdíl od předešlých systémů, ENVIS je desktopovou aplikací určenou pro OS Windows.



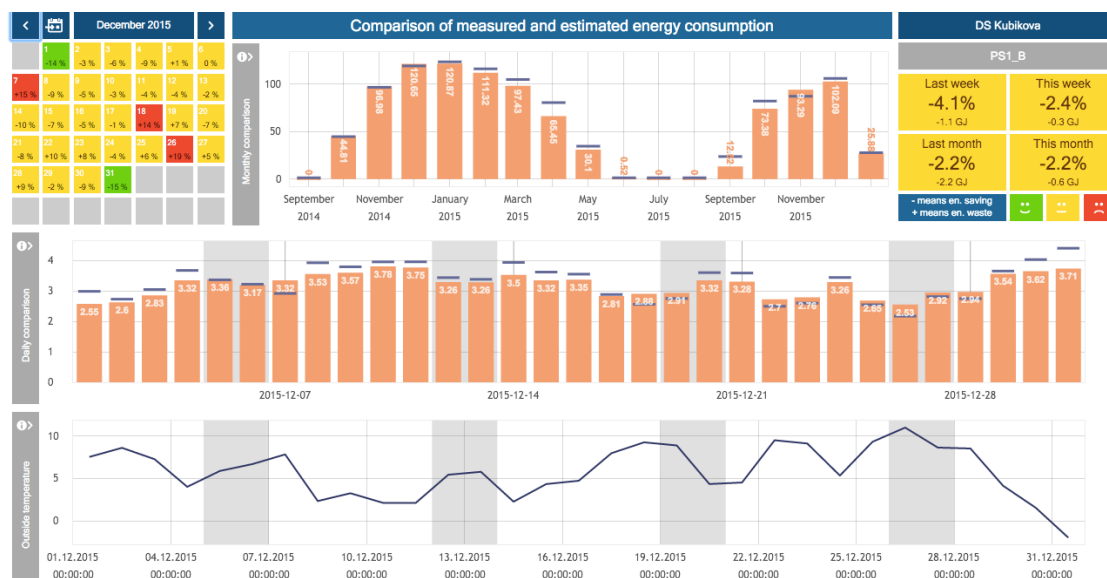
Obrázek 4: Histogram z rozšíření pro program ENVIS [12]

1.1.7 Mervis

Cloudová služba Mervis zajišťuje komplexní přehled nad provozovanými technologiemi, mezi nimi přehled o spotřebě elektrické energie. Prezентuje se přívětivým uživatelským rozhraním, nabízí pokročilé vyhodnocování dat, reporty, snadný přístup z různých zařízení.

V demo aplikaci nalezneme tabulkové pohledy pro zobrazení všech datových bodů v projektu, schematický pohled, který znázorňuje např. elektrickou síť ve vybraném objektu a udává aktuální stavové hodnoty na příslušných uzlech systému. Po zvolení záložky *Grafy* lze

pozorovat data pro vybraný časový interval pomocí spojnicového a sloupcového grafu, nebo v heat mapě. Pro kritické nebo jinak zajímavé hodnoty můžeme nastavit alarmy a sledovat historii alarmů i různých událostí. Na *Obrázek 5* vidíme přehled pro konkrétní projekt, který obsahuje kalendář s denními odchylkami od průměrného dnu v měsíci, srovnání naměřené a odhadované spotřeby elektrické energie a dále rozdíly mezi předchozím a stávajícím týdnem nebo měsícem.



Obrázek 5: Přehled v systému Mervis

1.2 Shrnutí využívaných technik a funkcí systémů

Z výše uvedených systémů získáváme poznatky o tom, jak vypadají již existující systémy pro správu elektrické energie. Některé z nich jsou pojaty obecněji a umožňují správu nejen elektrické energie, ale mj. také například tepelné energie. Ačkoliv se v těchto případech liší zkoumaná veličina nebo způsob odběru, styl zobrazování spotřeby nebo nákladů je velmi podobný jako u samotné elektrické energie. Do následujících kapitol jsou zařazeny nejhojněji využívané techniky vizualizace a analýzy dat.

1.2.1 Zobrazení dat za zvolený časový interval

Nejjednodušším a nejvhodnějším způsobem zobrazení, které všechny ze zvolených systémů využívají, je vyobrazení posloupnosti hodnot do grafů, a to nejčastěji spojnicových nebo sloupcových. Hodnoty může být též vhodné poskytnout ve formě tabulky, ale při větším počtu informací tento způsob prezentace ztrácí na přehlednosti. Systémy uživateli poskytují

jednoduchý výběr z monitorovaných míst a veličin, zpravidla je možné sledovat více údajů najednou a v grafech jsou odlišeny jednotlivými datovými sériemi.

Co se týče výběru intervalu, ve kterém chceme zvolená data zobrazit, častou funkcí systémů bývá poskytnutí kalendáře, ze kterého si uživatel vybere časový úsek dle libosti, a to v řádu hodin, dnů, týdnů, měsíců, kvartálů, nebo celého roku. V některých systémech se může jednat i o intervaly kratší nebo dále umožňují cíleně vynechat víkendové hodnoty apod. Vedle výběru od kdy do kdy jsou praktické i přepínače jako např. předchozí týden, tento týden, včera, minulý měsíc apod. pro zjednodušení procesu výběru, tedy aby uživateli stačil jeden klik myši místo složitějšího hledání správného data.

1.2.2 Přehled nejdůležitějších informací

Častý prvek systémů tvoří přehled nejdůležitějších informací např. o stavu a vytížení sítě, o denním průměru vůči předchozím dnům nebo o nákladech za spotřebovanou energii, a to vše z toho důvodu, aby uživatel měl vše podstatné přehledně na jednom místě a nemusel požadované informace vždy znovu vyhledávat. Některé systémy uvádí i predikci dat, již lze docílit mj. modelem *bottom-up* [2]. V přehledu se dále může objevit např. EUI, upozornění na překročené hodnoty nebo nadměrné vytížení sítě a další.

1.2.3 Agregace a normalizace dat

Při archivaci naměřených hodnot je důležité vědět, jak často data ukládáme, zda tak činíme každou milisekundu, každých deset sekund, nebo se jedná o minutové intervaly apod. V prvním případě se již při zobrazení jedné hodiny dostáváme na počet 3 600 000 různých hodnot, což je zcela neakceptovatelné, nereálné a zbytečné, protože takové zobrazení by bylo nepřehledné a na monitoru není možné od sebe rozeznat tolik informací na pouhých několika pixelech. Z toho důvodu data často agregujeme, jednodušeji řečeno jsou sdružována do skupinek po několika hodnotách a to podle předepsané funkce, která nejlépe vystihne podstatu zobrazovaných informací. Pokud bude zobrazovanou veličinou elektrická energie, vhodnou agregační funkcí může být suma, ale také např. průměr, median nebo maximum. Volba funkce tedy závisí na sledované veličině, ale také na úhlu pohledu a smyslu informace. Data můžeme agregovat již při archivaci do databázového systému, v případě webové aplikace lze agregovat data po jejich získání z databáze (DB) na serverové straně, nebo na straně klienta, ale nelze jednoznačně říci, který ze způsobů je nejvhodnější. Kvalitní systém by měl data automaticky agregovat, aby zbytečně nezatěžoval uživatelský systém a zobrazení nebylo zpožděné nebo trhané.

Normalizace dat je dalším způsobem, jak upravit vstupní hodnoty. Např. pokud monitorujeme rodinný dům a elektrické vytápění tvoří hlavní složku spotřeby elektrické energie, můžeme jednotlivé hodnoty podělit celkovým výměrem vytápěné plochy a získáme tak lepší obraz o tom, jakou spotřebu máme v porovnání s ostatními objekty a z toho možná i zjistit vysokou tepelnou ztrátu objektu apod. Normalizovat můžeme také pomocí venkovní teploty apod.

1.2.4 Srovnání s historickými daty

V případě dlouhodobé spotřeby elektrické energie je časté srovnání hodnot ve vybraném časovém intervalu s hodnotami např. z minulého týdne, měsíce nebo roku. Pokud data zobrazujeme pomocí spojnicového grafu, můžeme odlišit datové série např. vykreslením jedné z nich jako přerušovanou linii. Pro přehled uživatele se zdá jako přívětivé také zobrazení procentuální odchylky průměru (nebo dalších funkcí) např. pro konkrétní den (týden, měsíc) od minulého dne (týdne, měsíce).

1.2.5 Sankeyův diagram

Sankeyův diagram často slouží k vizualizaci toků v síti, zobrazení procesů apod. Podává informace o vztazích a transformacích mezi uzly. Představuje orientovaný ohodnocený graf, ve kterém pro všechny uzly platí, že suma vstupních váh je rovna váze výstupní [4].

Pro využití v systému správy elektrické energie lze diagramu využít např. pro přehlednou vizualizaci spotřeby energie v síti několika budov – získáváme informace o celkové spotřebě v síti, spotřebě jednotlivých budov a o vzájemném propojení, ale také o dílčích spotřebách z měřených spotřebičů uvnitř objektu.

1.2.6 Heat mapa

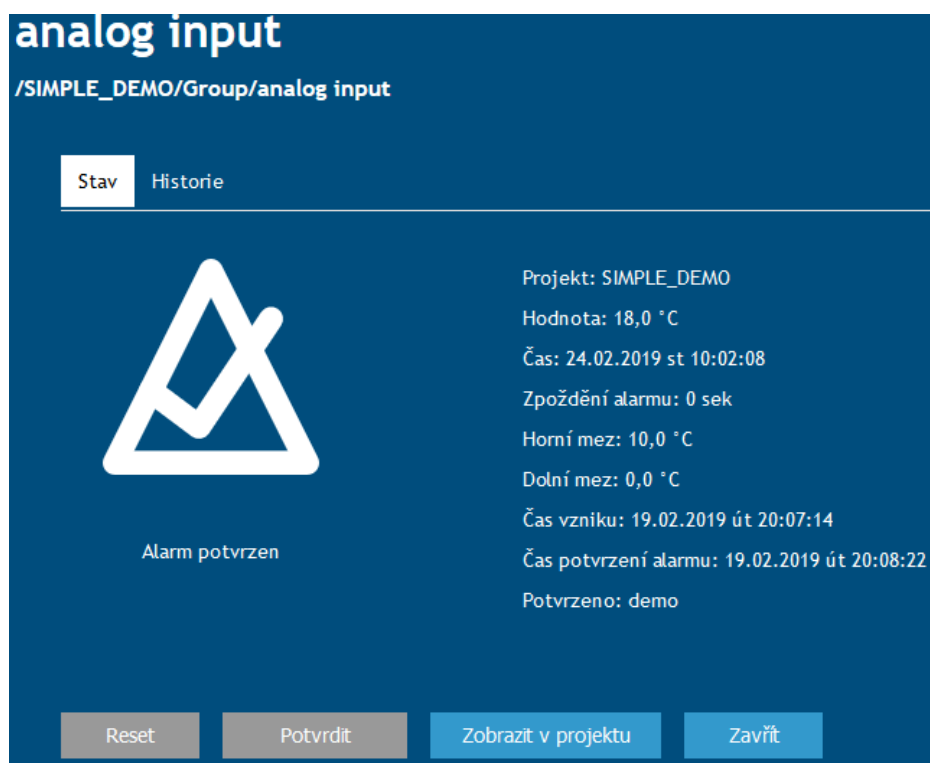
Jednou z efektivních metod zobrazení dat, která se mění v čase, je zobrazení heat mapy. Ve většině případů si vystačíme se spojnicovými, případně sloupcovými grafy, ale pokud je cílem efektivně znázornit data např. za několik po sobě jdoucích dnů a nechceme data více agregovat, heat mapa přehledněji zhustí více informací než-li graf spojnicový.

Pro sestrojení takového grafu potřebujeme znát hodnoty dat (mohou již být agregovány, např. po 15 minutách) v průběhu zvoleného časového období. Následně interval rozdělíme podle libovolného časového úseku – nejčastěji to mohou být dny (týdny, měsíce), které se často znázorňují na vodorovné ose. Svislá osa pak slouží pro rozdělení dílčích dnů (týdnů, měsíců) na menší časové úseky, např. hodiny (dny, pokud pracujeme s týdny nebo měsíci) apod. Prostor k vizualizaci hodnot je rozdělen do mřížky a hodnoty, jež v grafu chceme zobrazit,

představují barevně odlišené čtyřúhelníky, a to dle zvoleného barevného spektra [5]. Pro nižší hodnoty často využíváme spíše studenějších barev, naopak vyšší hodnoty zobrazujeme zpravidla teplejší barvou, odtud také název teplotní mapa.

1.2.7 Upozornění a reporty

Upozornění na krizové stavy, chyby v systému nebo ve sledovaném prostředí atd. poslouží v okamžik, kdy po systému chceme, aby poskytoval informace sám o sobě, tedy aby uživatel nemusel pravidelně kontrolovat stav monitorovaného objektu osobně. Příklad upozornění je znázorněn na *Obrázek 6* ze systému Mervis (viz kapitola 1.1.7). Říká nám o důvodu upozornění, místě alarmu, o času vzniku a zda došlo k potvrzení (jedná se o historické upozornění, které již uživatel systému viděl) apod. V případě vzniku upozornění na chybový stav apod. můžeme uživateli dát vědět různými způsoby – např. e-mailem, SMS, nebo přímo v systému. Aplikace může archivovat upozornění a jejich stavy a umožnit uživateli náhled do historie.



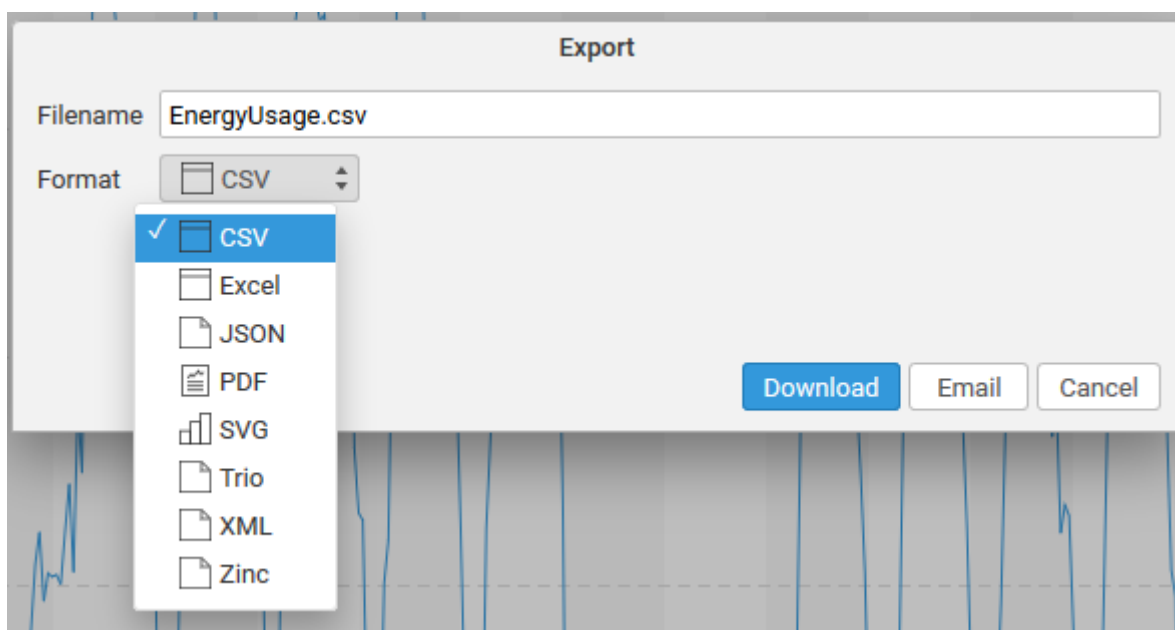
Obrázek 6: Alarm v systému Mervis

Reporty podávají strukturované informace o monitorovaném prostředí a jeho stavech. Může jím být např. týdenní zpráva o spotřebě elektrické energie a vzniklých alarmech, ale kromě tabulkových hodnot mohou obsahovat i exporty grafů. Systémy často nabízejí generování

reportů manuálně, nebo automatické, pravidelné zasílání reportů. Pro tvorbu reportů lze využít mj. SQL Server Reporting Services [6].

1.2.8 Export dat

Systémy správy elektrické energie poskytují možnost exportu dat do různých formátů, mezi nimi např. CSV, XLS, JSON, XML, ale také exporty grafů. Pokud je graf vykreslován pomocí vektorové grafiky, lze jej získat ve formátu SVG, v opačném případě pouze v rastrových formátech, jako je PNG nebo JPG.



Obrázek 7: Možnosti exportování dat (UBC)

2 Vybrané nástroje pro vizualizaci v prostředí .NET

Pro návrh a implementaci klíčových funkcí systému bylo potřeba využít vhodných nástrojů pro efektivní a přehledné zobrazení archivovaných dat, protože vizualizace dat za pomoci grafů je nejjenodušším způsobem jak interpretovat data a pochopit je dokáže každý [3]. Prvními prozkoumanými možnostmi jsou knihovny a komponenty pro WinForms a WPF, kterých je pomálu a často jsou placené. Volně dostupné nástroje nedosahují takových kvalit nebo nemají dostačující a přehlednou dokumentaci. Druhou oblastí bádání po nástrojích se staly javascriptové knihovny pro ASP.NET MVC aplikaci. Existují desítky javascriptových knihoven pro tvorbu grafů, často bývají open source a jejich funkcionality rozhodně není pozadu oproti placeným desktopovým knihovnám.

2.1 Nástroje pro desktop

2.1.1 Nevron Open Vision

Nevron Open Vision (NOV) je vývojová platforma pro .NET framework, která předkládá nástroje uživatelského rozhraní (UI) a komponenty pro tvorbu multiplatformních aplikací. Web [14] jej připodobňuje modernímu prohlížeči, který nelimituje HTML5 a navíc přidává další funkce. Můžeme využít kvalitních UI kontrol, mřížek, editoru formátovaného textu, grafů, ukazatelů (viz *Obrázek 8*), diagramů apod. Pro otestování lze využít 60denní trial verze, jinak je aplikace zplacněná.

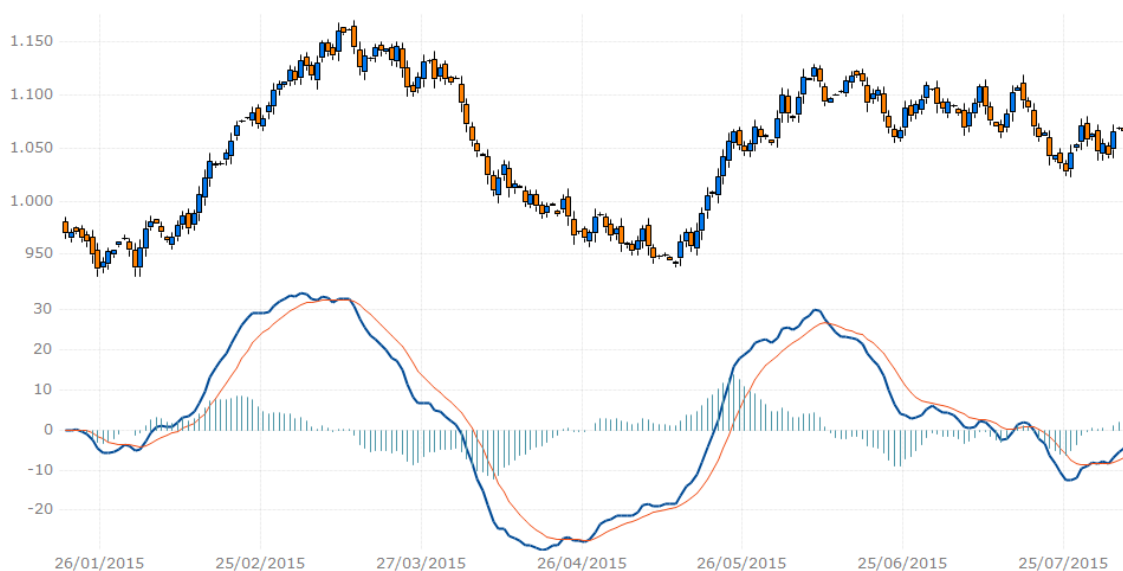


Obrázek 8: Příklad ukazatele z nástroje NOV pro kritické stavy [14]

Mezi podporované typy grafů mimo ty standardní patří krabicový graf, stromová mapa, heat mapa, vektorový graf a další. Osy lze libovolně škálovat a stylovat, můžeme se pohybovat po grafu a přibližovat jej, porovnávat hodnoty, graf lze přesouvat a dokovat. Systém podporuje export grafu např. do formátů PNG, TIFF nebo PDF.

2.1.2 teeChart

Komponenta pro vizualizaci dat ve více jak šedesáti různých grafech, ukazatele, mapy statistické a finanční grafy (viz *Obrázek 9*), jenž lze editovat pomocí přiložených nástrojů. Nástroj lze využít pro WinForms i WPF, ale také pro ASP.NET aplikaci.



Obrázek 9: Různé typy grafů v jednom zobrazení

2.1.3 LiveCharts

LiveCharts je open source knihovna sloužící pro vizualizaci dat v jednoduché, flexibilní formě a zároveň podporuje široké spektrum grafů. V základu je sice zdarma, ale nabízí také rozšířenou formu *LiveCharts.Geared*, která dokáže zobrazit až 100 000 hodnot bez výkonnostních potíží. Použít ji lze pro WPF a WinForms aplikaci, ale podpora přibývá i pro programy na platformě UWP (Universal Windows Platform), které lze spustit mj. i na mobilních přístrojích [15].

2.1.4 OxyPlot

Knihovna OxyPlot nabízí multiplatformní řešení pro tvorbu grafů v prostředí .NET [16], a to např. pro WPF, Windows Phone, Xamarin apod. V základu lze využít základních typů grafů a exportu do PNG, SVG a PDF, ale i přes jeho dostupnost oproti jiným nástrojům je funkčně značně limitován.

2.2 Nástroje pro ASP.NET

2.2.1 D3.js

Pravděpodobně nejpoužívanější javascriptovou knihovnou pro vizualizaci dat ve webových prohlížečích je D3.js. Umožňuje z pole hodnot vygenerovat HTML tabulku nebo přehledné a interaktivní SVG grafy, a to velmi efektivně a rychle [17]. Podporuje animace a přechody mezi typy grafů, desítky různých způsobů vizualizace včetně kartogramů, kalendářového zobrazení, map, Voroného diagramu atd.

2.2.2 Plotly.js

Nástroj postavený částečně na D3 nabídne zhruba 20 typů grafů, včetně 3D grafů, statistických zobrazení a SVG map. Poskytuje základní možnosti interakce – posouvání po libovolné ose, přiblížení, oddálení, výběr dat myší, porovnání datových bodů po přjetí myší apod. Součástí jsou i heat mapy, krabicové diagramy, Sankeyův diagram a další. Podporuje animace, tvorbu událostí (např. po kliknutí na datový bod, přiblížení aj.), filtry, agregace a další transformace. Již existující graf lze aktualizovat s novými daty, lze plynule přecházet z jednoho typu grafu na druhý, měnit styl grafu a jeho dat. Vedle javascriptové verze existuje ještě verze pro programovací jazyky Python, R, MATLAB a další.

2.2.3 Highcharts

Nástroj hojně využívaný systémy pro správu spotřeby elektrické energie nabídne desítky typů grafů, včetně zobrazení jako je barometr a jemu podobná měřidla, hodiny, heat mapa, histogram, Eulerův a Vennův diagram a další. Pro všechny své kvality se jedná o zpoplatněnou knihovnu.

2.2.4 Echarts

Open source knihovna, která dle [18] zvládne vyrenderovat až miliony hodnot. Přístup k datům a nastavení stylu vizualizace je zcela oddělená pro jednodušší úpravu zobrazení. Grafy jsou zcela responzivní, není tedy problém cokoliv zobrazit na libovolném přístroji. Při vizualizaci lze kombinovat vícero typů grafů, např. série zobrazena spojnicovým grafem proložena sloupcovým grafem s hodnotami druhé série. Knihovna dokáže zobrazit také 3D zpracování zeměkoule atd.

2.2.5 Chartist.js

Vcelku jednoduchý nástroj pro vizualizaci dat ve formě SVG nabízí vizuálně pěkně zpracované grafy, responzivní design a podporu ve všech standardních prohlížečích dnešní doby. Pro knihovnu lze získat vícero rozšíření, které přidávají další funkcionality. Samotná knihovna také umožňuje grafy animovat, ale neobsahuje ani zdaleka tolik typů grafů, co předešlé nástroje.

2.2.6 Grafana a InfluxDB

Grafana je platforma umožňující vizualizaci dat z libovolných zdrojů, protože nativně podporuje široké spektrum databázových systémů a data z nich lze sloučit do jednotného formátu. Poskytuje široké spektrum typů upravitelných grafů, které můžeme poskládat na nástěnku dle vlastní volby (viz *Obrázek 10*). V jedné nástěnce tak uchováme vícero různých zobrazení těch samých archivovaných dat, která lze aktualizovat najednou, a v systému můžeme uložit libovolný počet nástěnek – tedy přehledů vybraných dat. Grafy můžeme obohatit o metadata tak, že vybranou část grafu popíšeme, nebo k ni přidáme informativní tagy. Grafana dále obsahuje nástroj pro vytváření upozornění a notifikuje uživatele v případě, že byl alarm aktivován. Celý nástroj je open source a je možno jej využít na jakékoliv platformě [19].



Obrázek 10: Nástěnka v ukázkové aplikaci systému Grafana

Častým zdrojem dat při využití této platformy bývá databáze časových řad InfluxDB, kterou Grafana nativně podporuje a záznamy lze jednoduše získat prostřednictvím dotazovacího systému, kde si uživatel výběrem vygeneruje výsledný dotaz zasílaný databázi. Hodnoty z této DB lze také agregovat nebo je filtrovat podle vlastního kritéria.

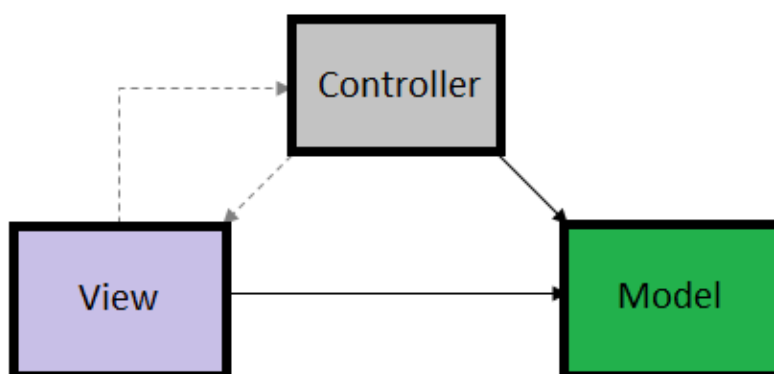
3 Návrh a řešení aplikace

Následující kapitola se zaměřuje na postup při návrhu aplikace v rámci .NET, která splňuje zadání – jedná se o systém pro vyhodnocování velkého objemu archivních dat spotřeby elektrické energie, a to za použití nástroje pro přehlednou a uživatelsky přívětivou vizualizaci a následnou analýzu zobrazovaných veličin. Jednotlivé podkapitoly informují o zvolených nástrojích a přístupu k návrhu systému, následně o koncepčním návrhu a implementaci jeho datové, logické a prezentační části.

3.1 Návrh architektury

Myšlenka návrhu systému je následující:

- Archivovaná data spotřeby elektrické energie, jež jsou velkého objemu, ukládáme v databázovém systému ve vhodné formě
- Systém bude webovou aplikací s architekturou MVC (viz *Obrázek 11*), z DB získá požadovaná data, serverová část aplikace je zpracuje a odešle klientské (prezentační) části
- V prezentační vrstvě s použitím javascriptových knihoven se s daty bude nakládat dle vůle uživatele prostřednictvím předdefinovaných funkcí a grafického rozhraní (GUI)

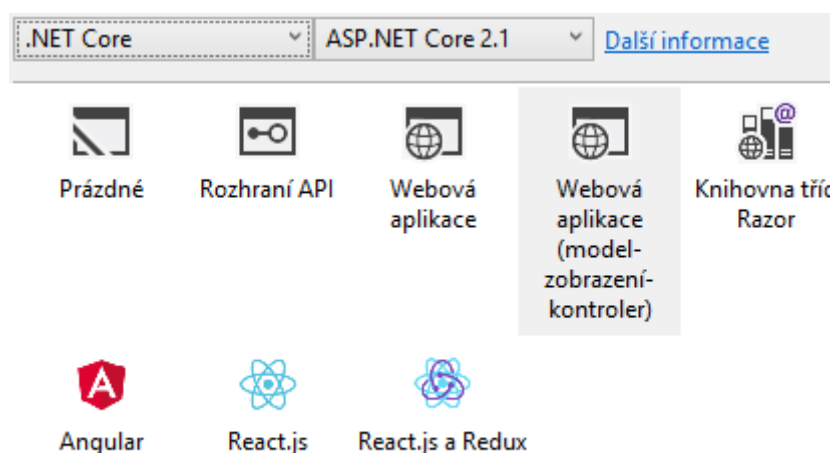


Obrázek 11: Architektura MVC

3.2 Zvolené nástroje

V následujících odstavcích popisují zvolené nástroje, mezi které patří následující: ASP.NET Core MVC, Plotly.js, MySQL, Bootstrap, jQuery. S implementací systému pomáhá vývojové prostředí Visual Studio 2017 (VS), s datovou vrstvou balíček XAMPP.

Pro dílčí části systému bylo zapotřebí zvolit vhodné nástroje pro zjednodušení práce při tvorbě systému. Zadání obsahuje požadavek, aby aplikace fungovala v prostředí .NET. To je docíleno využitím ASP.NET platformy pro tvorbu webových aplikací. Při implementaci návrhu ve vývojovém prostředí Visual Studio jsem měl na výběr mezi dvěma typy aplikací a zvolil jsem multiplatformní řešení ASP.NET Core. Po vytvoření nového projektu VS ještě nabízí již předvytvořené šablony pro různé architektury (viz *Obrázek 12*), mezi nimi zvolená MVC.



Obrázek 12: Možnosti výběru v aplikaci ASP.NET Core MVC

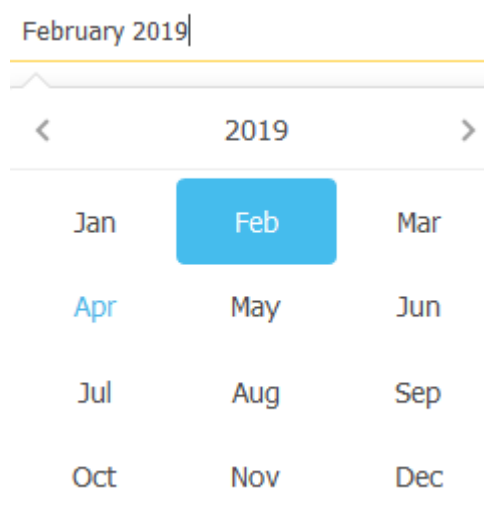
Výběr konkrétního DB systému (DBS) zadání práce nijak neomezuje a nelze jednoznačně určit jaký DBS se pro účel této aplikace nejvíce hodí. Data pro testování aplikace, získaná z *UBC Energy & Water Services* od *Univerzity Britské Kolumbie*, byla stažena ve formátu CSV. Tento způsob uložení je podporovaný řadou DBS a lze z něho přímo nainportovat data do DB. Pro vlastní potřeby jsem zvolil systém MySQL, který je dostupný spolu s Apache HTTP serverem v balíčku XAMPP.

Nejdůležitější část práce obsahuje komponenty pro grafickou prezentaci dat, a proto jejich výběr byl podstatně složitější úlohou. Při zkoumání nejen v práci uvedených nástrojů docházím k závěru, že javascriptová knihovna Plotly.js (viz kapitola 2.2.2) k účelům práce

zcela dostačuje a zároveň vzhledově vypadá nejpřívětivěji, snad až na grafické zpracování knihovny Chartist (viz kapitola 2.2.5), který ale nedosahuje takových kvalit a neobsahuje všechny potřebné funkce.

ASP.NET aplikace již obsahuje i sadu nástrojů pro zjednodušení procesu tvorby webové aplikace Bootstrap, jejíž stylování HTML elementů využívám pro přehledný a alespoň částečně responzivní design aplikace. Dalším velmi hojně využitým rozšířením aplikace se stala knihovna jQuery, která je také součástí ASP.NET MVC aplikace a s níž lze lépe ovládat vztahy mezi javascriptovým kódem a HTML, mezi klientskou a serverovou stranou aplikace zjednodušuje použití technologie AJAX.

Pro přehledné zobrazení kalendáře a tedy pohodlný výběr časového intervalu, ze kterého může uživatel chtít vidět data, systém používá knihovnu Air Datepicker. Kalendář lze přenastavit dle libosti a uživatel tak může volit jednotlivá data, interval v rámci dnů, měsíců, nebo let (viz *Obrázek 13*). Podobných komponent existuje širší množství, ale tato mj. podporuje i libovolný jazyk kalendáře.



Obrázek 13: Kalendář z nástroje Air Datepicker

O všechno ostatní se stará vlastní kód – na serverové části v jazyce C# a na té klientské javascript (JS), který také upravuje způsob jakým jsou data vizualizovaná již zmíněnou knihovnou Plotly.js. Pohledy zasílané uživateli jsou vytvořeny v čistém HTML bez použití šablonovacích systémů, neboť hlavní část aplikace tvoří pouze jedna stránka. ASP.NET aplikace navíc používá syntaxi Razor. Ta umožňuje vkládání C# kódu pomocí znaku @ a příponou tohoto formátu je *.cshtml*. Kromě Bootstrapu využívám i vlastní CSS stylování.

3.3 Datová vrstva

Využitým DBS při tvorbě aplikace se stal systém MySQL. Pro potřeby vizualizace archivovaných veličin vytvářím dvě tabulky s následující strukturou:

- *id* – identifikační číslo záznamu pro lepší průchod databází; jeho hodnota je nastavena automaticky, a to inkrementálně
- *date* – údaj o datu a čase dle mezinárodní formy ISO 8601, který nesmí nabývat nulové hodnoty
- další sloupce uvádí názvy monitorovaných míst a pro ně příslušné hodnoty ve formě desetinných čísel se zvolenou přesností (pro testování aplikace byl zvolen formát deseti platných číslic před desetinnou čárkou a pěti za ní, v reálném případě by záleželo na měřené veličině a požadované přesnosti)

Tabulky obsahují údaje vždy pouze pro jednu sledovanou veličinu, pro testovaná data obsahující veličiny elektrické energie a elektrického výkonu. Například údaje o datu tak mohou být redundantní, ale pro práci s těmito získanými daty se model zdál jako nejvhodnější řešení pro jednoduché dotazování na vybraná místa a naměřené veličiny. Práce se zaměřuje zejména na vyhodnocování archivovaných hodnot, pro které se tento model stal zcela dostačujícím. Příklad testovacích dat uvádí *Tabulka 1*, v které vidíme, že každé monitorované místo může mít různou přesnost měření elektrické energie a hodnoty byly archivovány každých 15 minut.

Tabulka 1: Příklad archivovaných dat pro testování aplikace

Datum	Cunningham	Library	Gym	Longhouse
2019-01-01 00:00:00	24,828125	7,5	6	9,5
2019-01-01 00:15:00	28,34375	7,25	6	9,5
2019-01-01 00:30:00	29,015625	7,25	5,75	9,5
2019-01-01 00:45:00	28,90625	7	5,25	10
2019-01-01 01:00:00	29,1875	7,25	5,25	9,5

3.4 Serverová část aplikace

Systém vyžaduje kvalitní propojení klientské části aplikace, kde se provádí vizualizace a analýza dat, s datovou vrstvou, odkud získává potřebná data a odesílá je ve formátu JSON. Celou funkci zastává serverová část aplikace. Ta dle architektury MVC obsahuje modely, pohledy a jejich prostředníky, tzv. *controllery*.

3.4.1 Model a databázový kontext

Dle navržené datové vrstvy aplikace obsahuje model *ElModel*, obsahující údaj o konkrétním datu a slovník pro uložení hodnot z jednotlivých míst. Po dotazování DB se jednotlivé řádky zapisují jako instance modelu do kolekce *List*. Ještě před předáním dat pohledu dochází k jejich serializaci do formátu JSON pro lepší manipulaci s nimi na klientské části aplikace. Nově vygenerovaná aplikace VS navíc obsahuje model *ErrorViewModel*, ale mimo to systém nevyžaduje žádné další modely.

Funkci pro naplnění modelu konkrétními daty, zastává kontextová třída *ChartContext*. Obsahuje *string* hodnotu pro připojení k databázi (ta je uložena v nastavení aplikace v souboru *appsettings.json* a obsahuje údaje o názvu databáze, jejím umístění a přihlašovacích údajích), vytváří instanci pro připojení a definuje metody, které využívá hlavní „kontroler“ pro správu dat. Mezi metody a funkce patří:

- *GetLast300Records* – z DB vrací posledních 300 uložených hodnot. Aplikace ji využívá při prvním spuštění, ale možné využití obdobné funkce je např. pro zobrazení hodnot v reálném čase, po mírné úpravě může případně vracet údaje pro poslední den.
- *GetColumnNames* – o požadované tabulce získá jména jejích sloupců. Tímto způsobem zjišťujeme, jaká monitorovaná místa se v DB nachází.
- *GetRecordsSince* a *GetRecordsBetween* – parametry funkcí jsou časové údaje ve formátu *string*, od kterých chceme získat informace až po dnešní datum, nebo po datum předané druhým parametrem. Další parametry tvoří pole sledovaných míst a *string* hodnota veličiny, tedy např. elektrický výkon nebo elektrická energie. Obě metody převádí datum a čas do správného formátu, následně zkonstruují SQL dotaz, předávají jej následující funkci a výsledek vrací *controlleru*.
- *GetRecordsByQuery* – jejím parametrem je SQL dotaz a seznam vybraných sloupců (výchozí nastavení obsahuje všechny sloupce). Funkce vytvoří novou kolekci modelů,

připojí se k DB a zasílá SQL dotaz. Po přijetí odpovědi řádek po řádku naplňuje kolekci a vrací ji.

3.4.2 Pohledy

Aplikace v podstatě obsahuje pouze jeden jediný pohled ve formě nástěnky. V něm probíhá vizualizace dat dle volby uživatele, obsahuje proto řadu ovládacích prvků GUI v podobě rozbalovacích seznamů, kalendáře, tlačítek, zaškrťovacích polí a přepínačů. Kromě nastavení tedy obsahuje také oblast pro vykreslování grafu a za pomoci vlastního javascriptového kódu lze přidat libovolné množství těchto bloků ve dvou různých rozměrech, nebo je odebrat pomocí příslušných tlačítek.

ASP.NET Core MVC aplikace poskytuje řadu již vygenerovaných pohledů, kterých lze v aplikaci využít. Kromě celého HTML dokumentu můžeme vytvářet také částečné pohledy a ty pak integrovat do větších částí dokumentu apod. Příkladem využití tohoto systému v aplikaci je sdílený pohled *_TimeButtons.cshtml*, v němž nalezneme skupinky tlačítek, která umožňují získat data např. za poslední rok, poslední den, hodinu apod. V případě rozšiřování aplikace (např. přidání zobrazení do samostatné tabulky atd.) by se nastavení vizualizace, tedy HTML ovládací prvky, mohlo rozkouskovat do menších sdílených souborů a ty následně využít dle potřeby v různých pohledech.

3.4.3 Controller

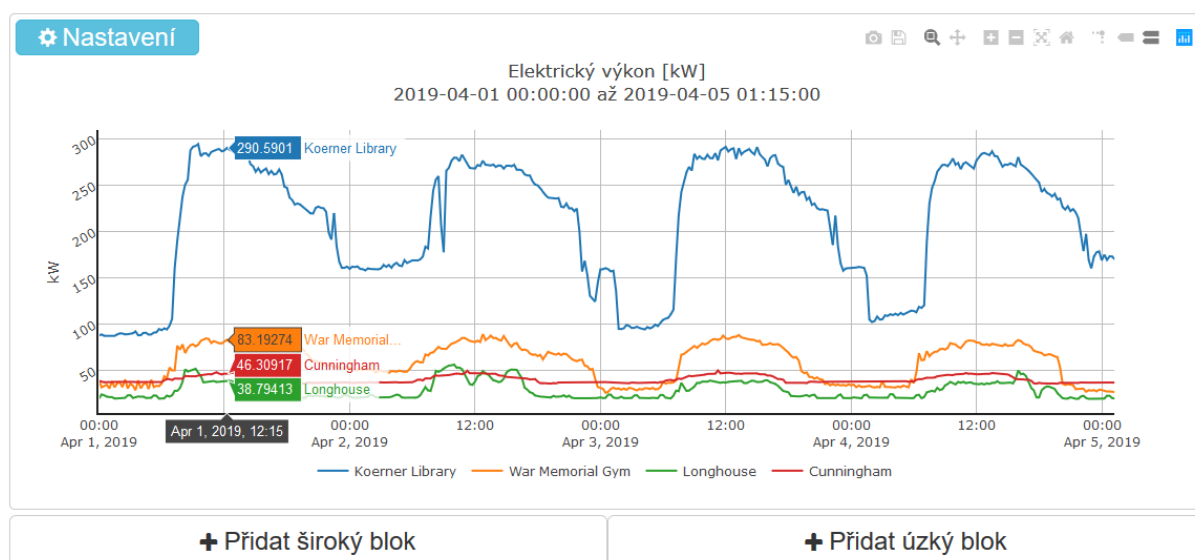
Důležitým bodem serverové části aplikace je řízení toku dat mezi datovou vrstvou a vrstvou prezentační. To provádí „kontrolery“ k tomu určené. *HomeController* poskytuje klientovi pohled *Home*, tedy domovskou stránku aplikace s posledními daty z modelu, nebo pohledy pro chybové stavy (ty ale aplikace neřeší, protože to není podstatou této práce). Vedle něho *ChartController* řídí asynchronní volání z klientské strany aplikace a vrací požadovaná data.

V *ChartControlleru* nalezneme metody, které odráží funkcionalitu databázového kontextu, jehož úkol uvádí kapitola 3.4.1. Z toho důvodu existují *DataColumnNames*, *DataSince* a *DataBetween*. Můžeme je volat pomocí dotazovací metody GET protokolu HTTP, např. *GET /Chart/DataColumnNames*. „Kontroler“, po tom co získá data z modelu, provádí serializaci do formátu JSON.

3.5 Klientská část aplikace

Prezentační vrstvu představuje klientská část aplikace, tedy HTML dokument, který s pomocí javascriptu a dalších nástrojů umožňuje vizualizaci archivovaných dat dle uživatelské volby nastavení jednotlivých grafů.

Při prvním načtení aplikace uživatel uvidí jeden blok (viz *Obrázek 14*), který obsahuje graf s 300 posledními hodnotami (což lze jednoduše přenastavit na serverové části aplikace) a jeho nastavení. V nejnižší části programu nalezne dvě tlačítka – pro přidání dalších bloků (malého, ty mohou být dva vedle sebe, a velkého), do kterých může zobrazit zcela odlišný interval hodnot v naprosto odlišné formě zobrazení. Následující podkapitoly se zaměřují na způsob výběru dat, způsob zobrazení, funkce agregace a porovnávání.



Obrázek 14: Základní pohled aplikace

3.5.1 Výběr dat

Velkou část z nastavení zobrazení zabírá výběr dat. Uživatel dostává na výběr ze všech dostupných míst formou zaškrtnutí polí (v případě vyššího počtu monitorovaných objektů by bylo vhodnější využít výběrového seznamu, což pro testování nebylo nutné), následně volí mezi přepínači zobrazovaných veličin (viz *Obrázek 15*).

Data

☒ Koerner Library

☒ War Memorial Gym

☒ Longhouse

☒ Cunningham

Veličina

☒ Elektrický výkon

☐ Elektrická energie

Potvrdit výběr dat a času

Obrázek 15: Možnosti výběru z monitorovaných míst a veličin

Druhou důležitou složkou je časový interval, za který chce uživatel data zobrazit (viz *Obrázek 16*). Zde může zvolit, jakou formou tak učinit – zda provede selekci konkrétních dnů (měsíců, nebo roků) v kalendáři (viz *Obrázek 13*), nebo kliknutím na jedno z tlačítek dat za poslední hodinu, den, měsíc apod. Pro testování systému byl za aktuální čas považován poslední časový údaj v databázi. Pro odeslání požadavku při výběru z kalendáře slouží potvrzovací tlačítko, které za pomoci jQuery a AJAX odesílá dotazovací metodou POST požadavek na server, kde jej převezme *ChartController*. Po příchodu odpovědi získáváme data ve formě JSON a AJAX metoda data ukládá a aktualizuje s nimi graf.

Kalendář

Dny

2019-04-01 až 2019-04-05

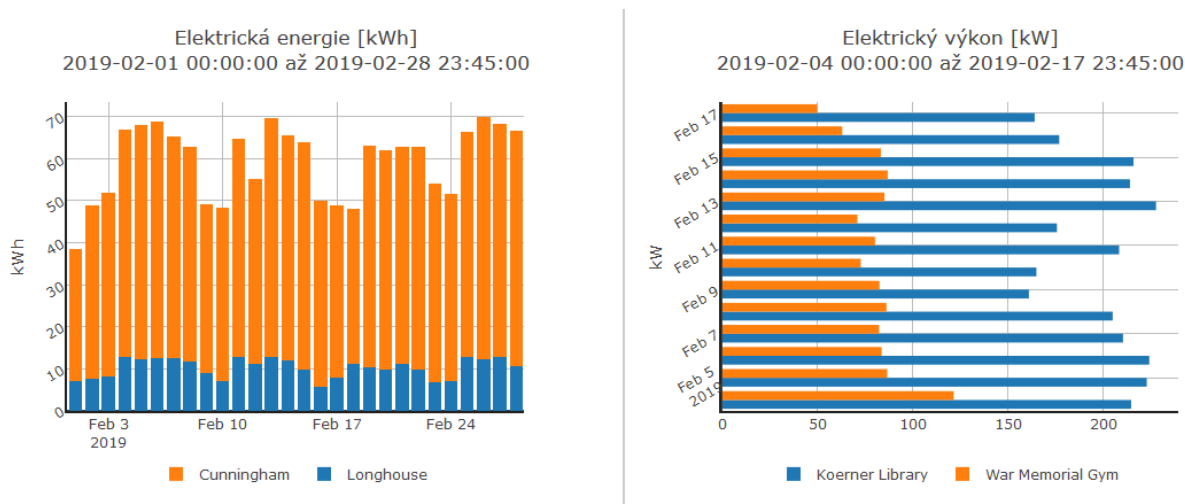
Data za poslední:

3 Roky	2 Roky	Rok
6 Měsíc	3 Měsíce	Měsíc
7 Dnů	2 Dny	Den
12 Hodin	6 Hodin	Hodinu
30 Minut	15 Minut	5 Minut

Obrázek 16: Volba kalendáře a zobrazení posledních hodnot

3.5.2 Způsob zobrazení

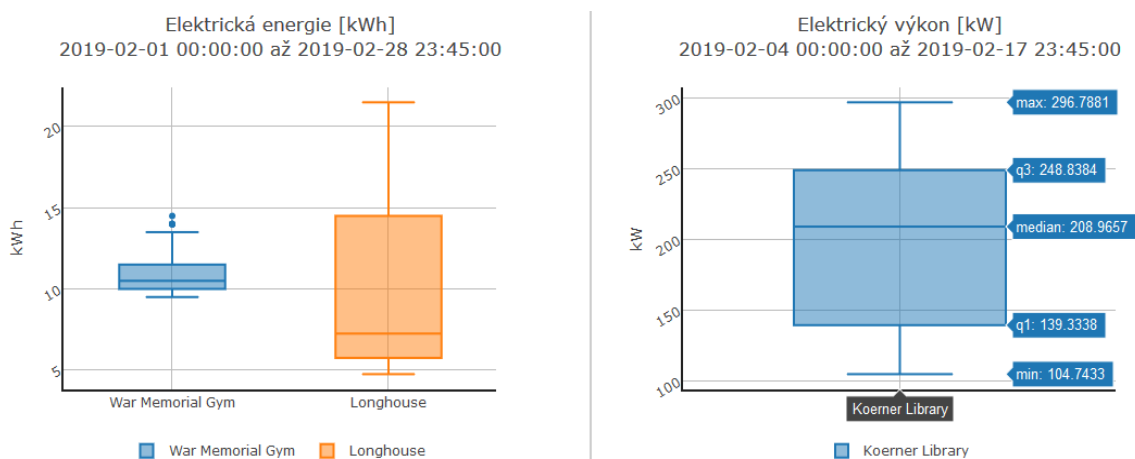
Ve třetím sloupci funkcí v nastavení grafu nalezneme mj. také přepínání typu grafu (viz *Obrázek 20*). Aplikace dává na výběr mezi dvěma spojnicovými grafy, vertikálními i horizontálními sloupcovými grafy, krabicovým grafem, heat mapou a liniovým grafem s výplní.



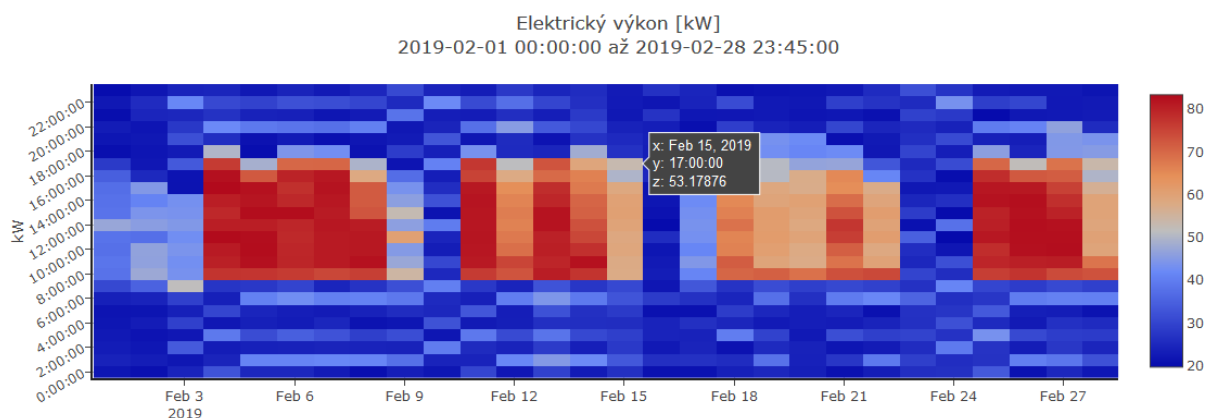
Obrázek 17: Sloupcové grafy vertikální skládaný (vlevo) a horizontální sdružený

Spojnicový graf má dvě nastavení – datové body mohou a nemusí být označeny puntíkem (pro velké množství hodnot se zdá příhodnějším body nevyznačovat, naopak pro nižší počet hodnot to může být přínosné). Sloupcové grafy mohou být sdružené – to znamená, že pro jednu hodnotu na časové ose program zobrazuje hodnoty ze všech datových sérií vedle sebe – nebo skládané ze všech sérií do jednoho sloupce. Sloupcový typ zobrazení navíc můžeme zobrazit vodorovně i horizontálně (viz *Obrázek 17*).

Další zajímavou variantu tvoří statistické zobrazení krabicového grafu, nebo-li *boxplotu*. Z jeho částí vypočítáme minima a maxima (a případně odlehlá pozorování), median a hranice prvního a třetího kvartálu (viz *Obrázek 18*). Při výběru typu heat mapy dochází k transformaci dat (viz kapitola 1.2.6) a v grafice se lehce můžeme zorientovat a zjistit naměřené hodnoty v určitou chvíli, ale také získáváme přehled o průběhu jednotlivých dnů. Tímto způsobem lze data jednoduše analyzovat v průběhu času a porovnávat je (viz *Obrázek 19*).



Obrázek 18: Krabicové diagramy – srovnání dvou sérií a dílčí hodnoty



Obrázek 19: Zobrazení v podobě heat mapy za měsíc únor

Použitý nástroj umožňuje vizualizaci řadou dalších typů zobrazení, mezi nimi např. i trojrozměrné grafy nebo mapy. Pokud by systém vyžadoval některé z takových typů grafů a diagramů, lze jej o tyto způsoby grafické prezentace snadno rozšířit.

3.5.3 Agregace a porovnávání

Systém dále nabízí nastavení agregace dat a jejich porovnávání s historickými hodnotami (viz *Obrázek 20*). V případě agregace je k dispozici možnost data shlukovat dle minutových intervalů (funkce sloužila pro prvotní data, která se aktualizovala několikrát do minuty, aktuální rozestup hodnot je 15 minut), po hodinách, nebo po dnech. Ideální by byl i stav agregace po patnácti minutách, protože ve spojení se spotřebou elektrické energie se často hovoří o tzv. čtvrt hodinovém maximu – to udává spotřebu nebo elektrický výkon během 15 minut a snahou je, aby tento údaj byl co nejnižší z důvodu jeho běžné regulace. Proto inteligentní systémy umí v situaci překročení maxima odpojovat a připojovat spotřebiče s

nižší zátěží [7]. Volbu čtvrt hodinové agregace ale v systému nenalezneme, neboť hodnoty jsou již v databázi uloženy po patnácti minutách. Uživatel dostává možnost volby i v případě funkce, kterou se agregace provádí. Tou může být ve výchozím nastavení průměr, dále pak minimum, maximum, median, nebo suma. Systém lze v budoucnu obohatit o další časové jednotky nebo agregační funkce.

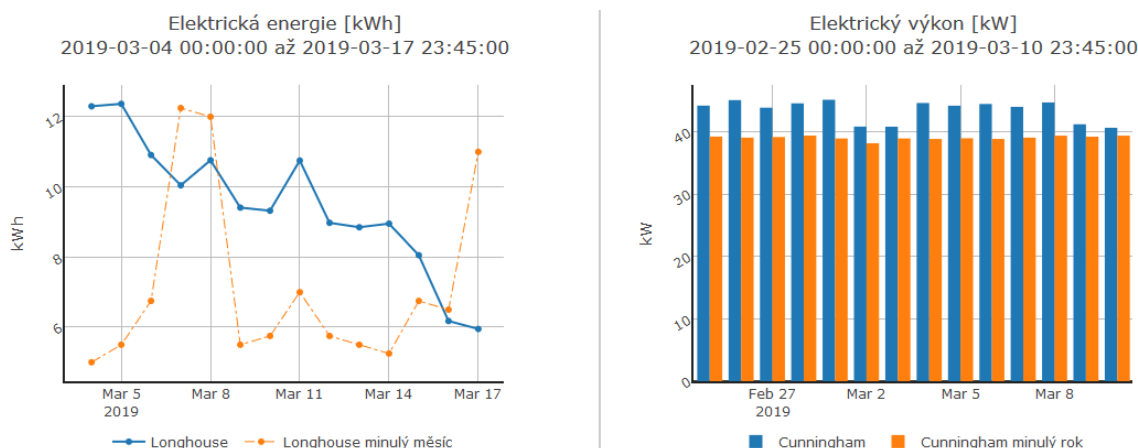
Transformace v případě dnů se provádí zhruba takto: postupně procházíme všechny časové intervaly a uchováваме si údaj o posledním dni, který ještě není zatříděn do finální kolekce agregovaných dat. Následně dochází k porovnání aktuálního času a pokud se srovnávané dny neliší (jedná se o stejný rok, měsíc a den, naopak hodiny a kratší časové úseky se mohou lišit), hodnota v prohlížený okamžik se zařadí do pole hodnot, na němž provádíme sdružení. Jakmile se dostaneme k hodnotě s odlišným datem, zmiňované pole se odešle funkci pro přepočet hodnot dle zvoleného nastavení a návracený údaj přidáváme mezi data určená k zobrazení. Pole pro agregaci je vynulováno a proces pokračuje až do chvíle, kdy nezbývají žádné další hodnoty. Uvnitř funkce pro seskupování hodnot se provádí přepočet hodnot dle uživatelem zvoleného systému. Aplikace využívá standardních metod pro výpočet průměru, medianu, sumy a volby minima a maxima.



Typ grafu
Line
Agregace dat
Žádná
Agregační funkce
Průměr
Porovnání
Neporovnávat

Obrázek 20: Nastavení vizualizace – typ grafu, agregace a její funkce, porovnávání

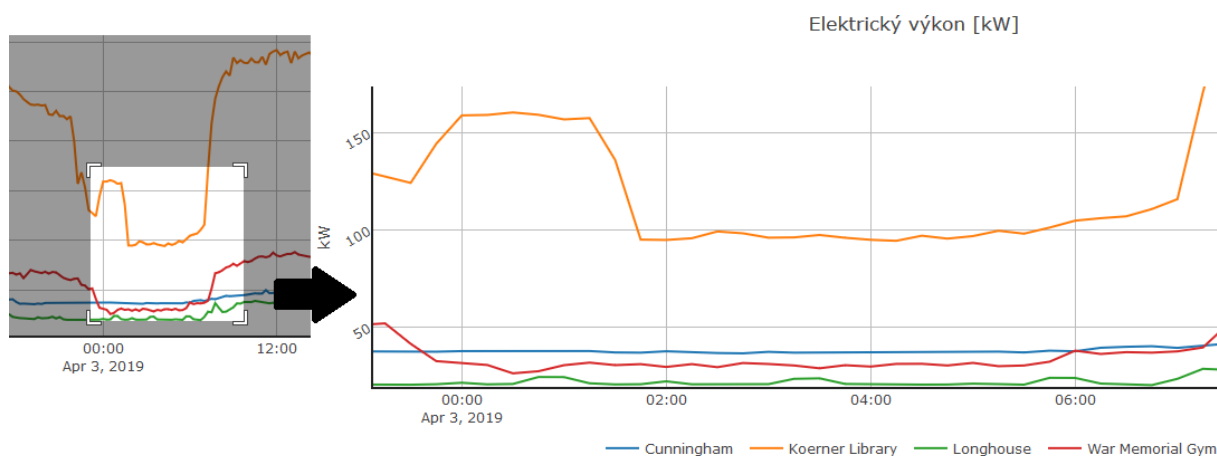
Poslední možností nastavení zobrazení tvoří porovnání aktuálně zvolených dat s daty předchozími. Chování je následující: uživatel vybere data v určitém intervalu a má možnost zvolit porovnání vůči předchozímu měsíci (to znamená, že interval nové série je posunut zpět o 30 dnů) nebo roku. Pro přehlednost systém historická data zobrazuje v podobě přerušované čáry (viz *Obrázek 21*) při zvoleném typu spojnicový graf.



Obrázek 21: Porovnání s historickými hodnotami

3.5.4 Interakce s grafem a další možnosti úpravy

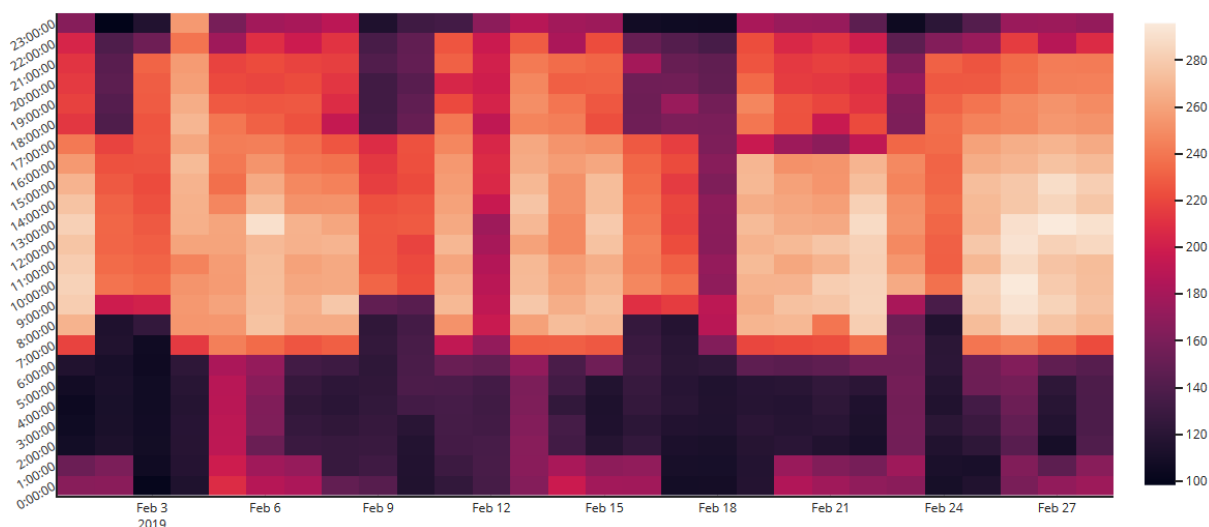
Použitý nástroj Plotly.js (viz kapitola 2.2.2) poskytuje několik možností, jak se po grafu pohybovat, jakým způsobem porovnávat hodnoty, na které zrovna míříme kurzorem myši apod. Po grafu se lze posouvat libovolně, a to posunutím po jednotlivých osách, nebo po obou zároveň. Diagram můžeme přibližovat a oddalovat tlačítka v pravém horním rohu, ale druhým způsobem je výběr oblasti myši. Dle pohybu v určitém směru dosahujeme přiblížení grafu v dílčích osách, nebo v obou naráz (viz *Obrázek 22*).



Obrázek 22: Přiblížení v grafu výběrem oblasti myši

V případě, že uživatel požaduje zobrazená data ve formátu tabulky, nebo vyžaduje změny zobrazení, využije tlačítka *Edit in Chart Studio* v pravé horní části grafu, které ho přesměruje do editoru Plotly. Zde může pro další potřeby graf upravovat, včetně změny stylu (viz *Obrázek 23*), typu grafu, úpravy legendy a os, barev jednotlivých sérií apod. Registrace do systému umožňuje export dat a grafu, nicméně hodnoty lze z tabulky, v níž můžeme i upravit

záhlaví, přidat jednotky apod., zkopírovat a uložit např. ve formátu XLS. Export grafu do obrázku provedeme přímo v aplikaci, a to tlačítkem *Download plot as png*.



Obrázek 23: Upravené barevné spektrum heat mapy v Chart Studiu

3.6 Problematické úlohy

V průběhu řešení návrhu a jeho implementace do webové aplikace bylo nutné řešit několik problémových úloh a nepodařilo se ve všech případech dojít k jejich správnému řešení. Snahou a cílem práce bylo vytvořit jednoduchý fungující systém, který bychom mohli využít pro správu již monitorovaných objektů, a to nejlépe pro vícero z nich najednou. Problémem je fakt, že dnes každé běžně dostupné řešení pro sběr informací o monitorovaných objektech funguje značně jinak – liší se formát dat, perioda jejich ukládání apod. Aplikace, jenž vznikla dle zadání práce, funguje dle očekávání na testovaných datech, ale pokud by měla běžet i s daty od nějakého zákazníka, bylo by nutné adaptovat celý systém, počínaje datovou vrstvou a způsobem zobrazování nastavení konče.

Druhý značný problém způsobila snaha minimalizovat počet aktualizací webové stránky (ačkoliv hlavní využitou architekturu tvoří MVC aplikace) za pomoci asynchronního HTTP požadavku při změně výběru dat. Ačkoliv systém s tímto způsobem aktualizace hodnot funguje, všechna volání po nových datech probíhají uvnitř jediné funkce. Překážkou se stává výběr dat u různých typů grafu – heat mapu musíme nastavit složitěji než spojnicový graf. Zároveň ale nebylo možné tyto případy jednoduše odlišit v několika různých funkcích, protože celý systém a každá jeho změna se projeví na více místech.

Vlastním požadavkem je i souvislý přechod mezi typy grafů. Přechody sice vypadají pěkně, o to více pokud jsou jejich součástí i animace. Spojnicový graf poměrně jednoduše lze transformovat do sloupcového grafu apod., ale problém nastává např. při změně horizontálního sloupcového grafu do spojnícového, kdy je potřeba přehodit osy nástrojem Plotly, což ale nelze vždy provést, protože spojnícový typ grafu oproti sloupcovému nemá nastavení horizontálního zobrazení a místo toho je nutností manuálně prohodit uložené hodnoty, což v situaci, kdy pracujeme s velkým objemem dat, jako je tomu u této aplikace, může být zdlouhavé a náročné na výpočetní výkon, který navíc nemáme u uživatele vždy zaručený. Další atypický přechod vytváří např. změna z heat mapy na krabicový diagram apod. – heat mapa nenávratně transformuje data do tří rozměrů, a proto systém opakuje poslední AJAX volání serveru, což ústí v přepisování dat ve chvíli, kdy by to nemělo být zcela nutné. V případě potíží se zobrazením je tak nejlepším řešením nový výběr dat.

I agregace dat způsobuje problém. Pokud jsou data sdružená např. podle dnů, nelze se lehce vrátit zpět k původnímu stavu bez agregace. Jediným viditelným řešením je uložení původních dat, což způsobuje jejich redundanci a nadměrné zatížení systému uživatele. Aplikace ale vykonává stejné řešení, jako při změně typu, tedy opětovné načítání dat ze serveru.

Za nedokonalý návrh lze považovat i systém správy a transformace dat, která je prováděna výhradně na klientské straně aplikace v jazyce JS. To sice není značným omezením pro stávající systém a testovaná data, ale v případě vyššího počtu monitorovaných objektů a častějšího ukládání hodnot se to problémem může stát. Po uvedení celého systému do chodu se nabízí minimálně dvě vypozaované možnosti. První by bylo využití volání AJAX ještě v širší formě – pokud již zobrazuji nějaká data, ale chci je např. agregovat, server dostane požadavek obsahující nejen časové údaje a záznam o tom, z jakého místa nebo objektu data chceme, ale také o tom, co s nimi chceme provádět. Veškeré funkce by tak implementoval server a chod aplikace tak omezí jedině kvalita serveru a jeho stabilita. Druhou možnost představuje opustit asynchronní volání a naplno využít potenciálu architektury MVC. Obdobně jako u první z možností by veškerou funkcionalitu týkající se dat a jejich struktury zajistila serverová část aplikace, nikoliv ta klientská.

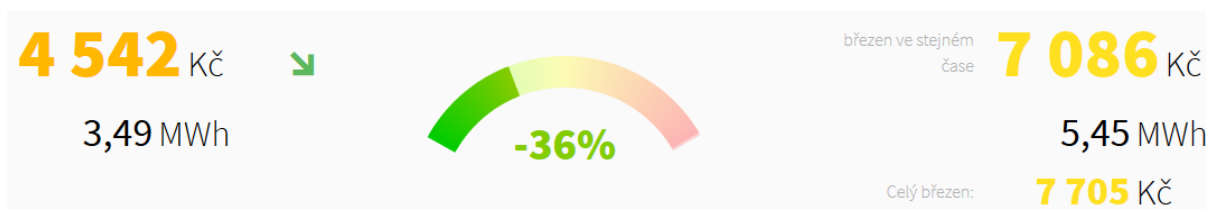
3.7 Doplnující funkce systému

Kapitola se zaměřuje na vytyčení funkcí, které nebylo možné do aplikace zahrnout nebo se přímo netýkaly zadání práce, ale bylo by možné a vhodné je v budoucnosti přidat. Mezi

chybějící funkcionalitu lze zařadit možnost tvorby různých upozornění nebo alarmů. Důvod absence této funkce v systému je zejména fakt, že systém vyhodnocuje pouze archivovaná data, přitom největší potenciál a využití upozornění vidím zejména pro hodnoty získané v reálném čase. Knihovna Plotly podporuje jednoduché filtrování zobrazených informací, ale nejedná se o výrazný nástroj a filtrování se ani ve zkoumaných systémech (viz kapitola 1.1) běžně neobjevuje.

Naproti tomu vytváření reportů představuje běžně využívanou metodu pro prezentaci stavů sledovaného objektu za uplynulý interval – např. měsíc, týden. Protože se nejedná o funkci, kterou by zprostředkoval vizualizační nástroj a „reporting“ nebyl požadovanou vlastností systému a téma reportování řeší mnohé jiné práce, aplikace jej neobsahuje. Ze vzniklého systému by reporty mohly využít získávání dat z DBS a jejich agregaci a dále různé možnosti grafické prezentace prostřednictvím grafů a diagramů.

Pokud by systém sloužil vícero uživatelům a monitoroval větší počet míst, aplikace by nutně vyžadovala správy uživatelů a jejich práv. Ve většině případů se jedná o neveřejná data a zákazník si nepřeje, aby kdokoliv cizí nahlížel na jeho data, případně je odcizil nebo dokonce zneužil. S tím souvisí i celkové zabezpečení systému – včetně ochrany proti SQL útokům, podvržení hesel atd, kterou vzniklý systém vůbec neřeší z důvodu nepodstatnosti pro účel této práce.



Obrázek 24: Stav dnešních hodnot, cena spotřeby a porovnání (Energomonitor)

V situaci, kdy data systém zaznamenává v reálném čase, lze za vhodné považovat zobrazení údajů o aktuálním stavu ve vztahu informace – hodnota, nebo např. změny vůči předchozím týdnům (měsícům apod.), cena spotřebované energie za jednotku času, a to obdobným způsobem jako na Obrázek 24. Ve vlastní aplikaci podobný přehled nelze vytvořit, neboť neznáme kontext testovaných dat – např. cenu spotřeby. Jediným způsobem jak porovnat data vůči historickým hodnotám v aplikaci, je nastavení porovnávání zvoleného intervalu s minulým měsícem (nebo rokem) jak je popsáno v kapitole 3.5.3. Pro zjištění celkové denní spotřeby zvolíme požadovaný den, libovolný typ grafu a denní agregaci.

Závěr

Hlavního cíle – navržení koncepce architektury jednoduché aplikace a její implementace pro demonstraci funkce vyhodnocování archivních dat – práce dosahuje prostřednictvím vlastní webové aplikace, v níž lze archivovaná data vizualizovat různými způsoby. Grafickou prezentaci provádí prostřednictvím základních typů grafů a jejich variací, statistickým zobrazením v podobě krabicového diagramu, nebo za pomoci heat mapy, která poskytuje lepší přehled o průběhu a změnách ve spotřebě elektrické energie během zvoleného intervalu. Díky využitým a poupraveným nástrojům se tak děje v přehledné formě, které porozumí i laický uživatel. Z jednotlivých grafů a diagramů si uživatel může poskládat libovolný počet zobrazení vedle sebe a snadno tak porovná hodnoty z mnoha monitorovaných míst nebo různých časových období.

Z bádání po vhodných nástrojích pro grafickou prezentaci dat a po existujících systémech, které zajišťují pohodlnou správu spotřeby elektrické energie, práce konstruuje a podává shrnutí nejčastěji využívaných technik analýzy dat a jejich vizualizaci. Vedle obyčejného zobrazení dat ze zvoleného časového intervalu je tvoří agregace a normalizace dat (např. výměrem vytápěného objektu), srovnání s historickými daty, tvorba upozornění a alarmů z důvodu varování o kritických a jinak nestandardních stavech, poskytování týdenních nebo měsíčních reportů, ale také zobrazení toku v síti prostřednictvím Sankeyova diagramu apod.

V průběhu vzniku práce nastaly problémy s návrhem a implementací klíčových funkcí mj. kvůli snaze vytvořit všeobecný systém, a právě potížím se věnuje kapitola 3.6. Volba dat pro testování aplikace nebyla zcela šťastná, protože z nich samotných nelze získat potřebný kontext. Z toho důvodu aplikace neposkytne přehled o ceně spotřeby apod. Dále nebylo možné některé ze zmíněných technik uplatnit, neboť se vážou na hodnoty aktualizované v reálném čase (např. současný výkon nebo upozornění na kritické stavy). Na absenci dalších funkcí se zaměřuje kapitola 3.7. V případě nasazení aplikace do většího systému je nutné ji upravit dle již zavedených konvencí nasbíraných dat, a to zejména v datové vrstvě, a nebo digitalizované informace uzpůsobit stávajícímu systému.

Aplikaci je možné nadále rozšiřovat o další funkce. Vzhledem k reálnému využití systému větším počtem uživatelů přibývá nutnost implementace správy uživatelů a jejich práv. Po napojení systému na průběžně monitorované objekty jej lze obohatit o stavové přehledy nebo o funkci tvorby reportů a jejich zasílání e-mailem apod. Ve stávajícím stavu systém umožňuje dostatečné vyhodnocování archivních dat dle zadání práce.

Literatura

- [1] KRAUS, Jan a Martin BLÍŽKOVSKÝ. Uživatelská příručka aplikace ENVIS v. 1.1 [online]. 2013. [cit. 2019-04-05]. 1.1. Dostupné z: <http://www.kmb.cz/>
- [2] MENEZES, A.C., A. CRIPPS, R.A. BUSWELL, J. WRIGHT a D. BOUCLAGHEM. Estimating the energy consumption and power demand of small power equipment in office buildings. *Energy and Buildings* [online]. 2014, **75**, 199-209 [cit. 2019-04-28]. DOI: 10.1016/j.enbuild.2014.02.011. ISSN 03787788. Dostupné z: <https://linkinghub.elsevier.com/retrieve/pii/S0378778814001224>
- [3] SATO, Ichinari, DIPTARAMA a Ayumi SHINOHARA. Visualization and Analysis of Electrical Energy Consumption in Laboratories. In: *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)* [online]. IEEE, 2016, 2016, s. 509-512 [cit. 2019-04-09]. DOI: 10.1109/IIAI-AAI.2016.194. ISBN 978-1-4673-8985-3. Dostupné z: <http://ieeexplore.ieee.org/document/7557664/>
- [4] RIEHMANN, P., M. HANFLER a B. FROEHLICH. Interactive Sankey diagrams. In: *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005* [online]. IEEE, 2005, s. 233-240 [cit. 2019-04-12]. DOI: 10.1109/INFVIS.2005.1532152. ISBN 0-7803-9464-X. Dostupné z: <http://ieeexplore.ieee.org/document/1532152/>
- [5] KUMATANI, Satsuki, Takayuki ITOH, Yousuke MOTOHASHI, Keisuke UMEZU a Masahiro TAKATSUKA. Time-Varying Data Visualization Using Clustered Heatmap and Dual Scatterplots. In: *2016 20th International Conference Information Visualisation (IV)* [online]. IEEE, 2016, 2016, s. 63-68 [cit. 2019-04-28]. DOI: 10.1109/IV.2016.50. ISBN 978-1-4673-8942-6. Dostupné z: <http://ieeexplore.ieee.org/document/7557905/>
- [6] POLÍVKA, Pavel. *Návrh a implementace datové a aplikační vrstvy databáze pro vzdálený přístup, webové aplikace a reporting*. Liberec, 2013. Diplomová práce. Technická Univerzita v Liberci. Vedoucí práce Jan Kraus.
- [7] RINDT, Zdeněk. *Vizualizace dat spotřeby elektrické energie prostřednictvím webové aplikace*. Liberec, 2014. Bakalářská práce. Technická Univerzita v Liberci. Vedoucí práce Jan Kraus.
- [8] Energomonitor - Insight into your house. *Energomonitor* [online]. energomonitor, 2019 [cit. 2019-02-14]. Dostupné z: <https://www.energomonitor.com/cz/>
- [9] Active Energy Management Overview. *E3m* [online]. emation, 2019 [cit. 2019-02-14]. Dostupné z: http://www.e3m.ch/en/active_energy_management/overview/
- [10] FLOWBOX [online]. FLOWBOX, 2019 [cit. 2019-04-28]. Dostupné z: http://www.e3m.ch/en/active_energy_management/overview/
- [11] ENVIS ver. 1.2. *KMB Systems* [online]. Liberec: KMB Systems, 2011 [cit. 2019-03-15]. Dostupné z: <http://www.kmb.cz/index.php/cs/aplikace/envis>
- [12] Rozšíření pro ENVIS. *KMB Systems* [online]. Liberec: KMB Systems, 2011 [cit. 2019-03-15]. Dostupné z: <http://www.kmb.cz/index.php/cs/aplikace/rozsireni-pro-envis>

- [13] Product. *SkyFoundry* [online]. SkyFoundry, 2019 [cit. 2019-03-25]. Dostupné z: <https://skyfoundry.com/product>
- [14] Nevron Open Vision for .NET (NOV). *Nevron* [online]. SkyFoundry, 2019 [cit. 2019-04-10]. Dostupné z: <https://www.nevron.com/products-open-vision.aspx>
- [15] *LiveCharts* [online]. LiveCharts, 2019 [cit. 2019-04-10]. Dostupné z: <https://lvcharts.net/>
- [16] *OxyPlot* [online]. OxyPlot, 2019 [cit. 2019-04-10]. Dostupné z: <http://www.oxyplot.org/>
- [17] *D3.js – Data Driven Documents* [online]. Mike Bostock, 2019 [cit. 2019-04-11]. Dostupné z: <https://d3js.org/>
- [18] *ECharts* [online]. Apache Software Foundation, 2018 [cit. 2019-04-11]. Dostupné z: <https://ecomfe.github.io/echarts-doc/public/en/index.html>
- [19] *Grafana - The open platform for analytics and monitoring* [online]. Grafana Labs, 2019 [cit. 2019-04-17]. Dostupné z: <https://grafana.com/>