



Diplomová práce

Tvorba reportů pro systém energetického managementu chytrých budov a sítí

Studijní program:

N2612 Elektrotechnika a informatika

Studijní obor:

Informační technologie

Autor práce:

Bc. Martin Vondráček

Vedoucí práce:

Ing. Jan Kraus, Ph.D.

Ústav mechatroniky a technické informatiky

Liberec 2023



Zadání diplomové práce

Tvorba reportů pro systém energetického managementu chytrých budov a sítí

<i>Jméno a příjmení:</i>	Bc. Martin Vondráček
<i>Osobní číslo:</i>	M19000160
<i>Studijní program:</i>	N2612 Elektrotechnika a informatika
<i>Studijní obor:</i>	Informační technologie
<i>Zadávací katedra:</i>	Ústav mechatroniky a technické informatiky
<i>Akademický rok:</i>	2021/2022

Zásady pro vypracování:

1. Seznamte se s obvyklými metodami zpracování a prezentace dat v systémech pro management spotřeby energií a pro vyhodnocování kvality elektrické energie.
2. Doplněte existující přenositelnou datovou vrstvu v prostředí NET5/6 o vámi navržené a správně implementované algoritmy pro automatizované zpracování vstupních dat měření s ohledem na vyhledávání anomálií a zpracování alarmových stavů v jednotlivých měřeních i v jejich složitějších kombinacích.
3. Výstupy realizujte formou přehledných reportů ve formátu pdf či v jiných vhodných a výslednou aplikaci důsledně otestujte i s ohledem na to, že na každé jednotlivé instanci může probíhat najednou větší množství analýz paralelně.
4. Výsledky stručně a přehledně shrňte v textové části práce a diskutujte možnosti dalšího rozšiřování tématu.

Rozsah grafických prací: dle potřeby dokumentace
Rozsah pracovní zprávy: 40–50 stran
Forma zpracování práce: tištěná/elektronická
Jazyk práce: Čeština

Seznam odborné literatury:

- [1] JOHN, J., et al. Project Haystack Data Standards. In: *Energy and Analytics*. River Publishers, 2020. p. 237-243.
- [2] GASSER, Jan, et al. Predictive energy management of residential buildings while self-reporting flexibility envelope. *Applied Energy*, 2021, 288: 116653.
- [3] ZHANG, Di; SHAH, Nilay; PAPAGEORGIU, Lazaros G. Efficient energy consumption and operation management in a smart building with a microgrid. *Energy Conversion and Management*, 2013, 74: 209-222.
- [4] LEE, P. K.; LAI, L. L.; CHAN, S. W. A practical approach of energy efficiency management reporting systems in micro-grid. In: *2011 IEEE Power and Energy Society General Meeting*. IEEE, 2011. p. 1-5.
- [5] NASRIN, Nahid, et al. Integration of data reporting and visualization of an Energy Management System for Buildings. 2020.

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

Datum zadání práce: 12. října 2021
Předpokládaný termín odevzdání: 22. května 2023

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

doc. Ing. Josef Černohorský, Ph.D.
vedoucí ústavu

V Liberci dne 12. října 2021

Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Jsem si vědom toho, že na mou diplomovou 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 diplomové práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li diplomovou 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.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má diplomová práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

Poděkování

Děkuji svému vedoucímu Janu Krausovi za pomoc a čas, který mi věnoval při vytváření této práce. Také chci poděkovat svým rodičům za hojnou psychickou i materiální podporu.

Tvorba reportů pro systém energetického managementu chytrých budov a sítí

Abstrakt

Tato práce se zabývá jednou z důležitých komponent systémů pro management elektrické energie: analýzou měření veličin elektrické energie a vizualizací důležitých informací ve formě vygenerovaných dokumentů (reportů). První část práce obsahuje popis základních nabízených funkcí systémů energetického managementu (EnMS). Funkce, které se nacházejí na mnohu procházených EnM platformách, jsou shrnuty v tabulce a některé zajímavé komponenty blíže popsány v textu. Následně je popsána knihovna STEP a knihovny DevExpress, které můj program využívá k vytváření, upravování a generování reportů.

Praktickou část tvoří dva projekty napsané v programovacím jazyce C# pomocí vývojářské platformy .NET7: ReportCore knihovna a EReporter aplikace. ReportCore knihovna implementuje načítání nastavení reportů, čtení dat a generování reportů. Při vytváření této knihovny jsem tedy navrhl množství různých reportů včetně reportů kvality elektrické energie, reportů spotřeby a produkce elektrické energie a dalších. Aby reporty zobrazovaly podstatné informace relevantní pro uživatele, implementoval jsem algoritmy pro analýzu dat, zpracování nastavení mých reportů, načítání i úpravu šablon mých reportů a další podstatné aspekty tohoto procesu.

EReporter je jednoduchá WinForms aplikace, která uživateli umožňuje definovat vstupní data, požadovaný report i jeho nastavení a následně vygenerovat report pomocí méj ReportCore knihovny. Report lze v EReporteru otevřít v DevExpress nástroji pro zobrazování reportů, popřípadě ho vyexportovat do pdf či jiných formátů. Dalším využitelným nástrojem je grafický editor pro vytváření a upravování šablon reportů. EReporter lze využívat i jako externí nástroj spustitelný jiným softwarem a jeho vstupy lze předat jako vstupní argumenty.

Oba realizované projekty, jejich návrh i implementace jsem v této zprávě vystihl pomocí textového popisu doplněného o grafické ilustrace a shrnul jsem i možnosti rozšíření mého programu a jeho perspektivy do budoucna.

Klíčová slova: Reporting, DevExpress, Kvalita elektrické energie, Monitoring elektrické energie, Analýza elektrické energie

Automated reports for the energy management system of smart buildings and smart grids

Abstract

This work deals with one of the important components of energy management systems: the analysis of electrical energy measurement quantities and the visualization of important information in the form of generated documents (reports). The first part of the work contains a description of the basic functions offered by energy management systems (EnMS). The functions of the EnM platforms are summarized in a table and some interesting components are described in more detail in the text. Subsequently, the STEP library and the DevExpress libraries, which my program uses for creating, modifying, and generating reports, are also described.

The practical part consists of two projects written in the C# programming language using the .NET7 development platform: the ReportCore library and the EReporter application. The ReportCore library implements the loading of report settings, data reading, and report generation. When creating this library, I designed a number of different reports, including reports on the quality of electrical energy, energy consumption, and others. To display relevant information for users, I implemented algorithms for data analysis, processing of report settings, loading and modification of report templates, and other important aspects of this process.

EReporter is a simple WinForms application that allows users to define input data, the desired report and its settings, and then generate a report using my ReportCore library. The report can be opened in the DevExpress report viewing tool in EReporter, or exported to PDF or other formats. Another useful tool is the editor for creating and modifying report templates. EReporter can also be used as an external tool executable by other software, and its inputs can be passed as input arguments.

In this report, I described both implemented projects, their design and implementation, using textual description supplemented by graphic illustrations, and also summarized the possibilities for extending my program and its prospects for the future.

Keywords: Reporting, DevExpress, Electric power quality, Electrical energy monitoring, Electrical energy analysis

Obsah

Seznam zkratk	11
1 Úvod	12
2 Systémy pro správu elektrické energie	14
2.1 Popis	14
2.2 Prozkoumané EnM platformy	16
2.3 Shrnutí	18
3 Využité knihovny a funkce	19
3.1 DevExpress	19
3.1.1 DevExpress.Reporting.Core	20
3.1.2 DevExpress.Win.Design, DevExpress.Win.Reporting	22
3.2 STEP	23
3.2.1 DataSource	24
3.2.2 Calculations	25
3.2.3 Resources	26
3.3 Shrnutí	26
4 Knihovna ReportCore: Implementace reportů a jejich nastavení	27
4.1 Třída ReportBase a její potomci	27
4.2 Dělení reportu na kapitoly	29
4.2.1 Groupování dat uvnitř kapitoly reportu	29
4.2.2 Dělení na kapitoly	30
4.2.3 Mapa dokumentu	31
4.3 Třída ReportSetting	31
4.4 Soubory s šablonami reportů a jejich nastavením	33
4.5 Metody generující reporty	35
4.6 Shrnutí	36
5 Realizované reporty	37
5.1 Reporty spotřeby a produkce elektrické energie	37
5.2 Reporty alarmů	39
5.3 Reporty výkonu	40
5.4 Reporty harmonických	42
5.5 Reporty kvality elektrické energie	43
5.6 Samostatné části reportu	45

5.7	Shrnutí	47
6	Implementovaná aplikace EReporter	49
6.1	Popis	49
6.2	Shrnutí	51
7	Závěr	53
	Použitá literatura	54
8	Přílohy	59
8.1	Prvky v existujících EnM systémech	59

Seznam obrázků

3.1	Základní stavební bloky projektu	19
3.2	ReportDesignTool s otevřenou šablonou reportu	23
3.3	ReportPrintTool s vygenerovaným reportem	24
4.1	Struktura repx souboru s šablonou reportu	34
4.2	Struktura repCfg souboru s nastavením reportu	34
5.1	Graf a část tabulky energy consumption reportu	38
5.2	Kruhový diagram spotřeby energie po přístrojích	39
5.3	Grafy průběhu alarmu	40
5.4	Tabulka změn hodnot jednoho alarmu	40
5.5	Křivka trvání zátěže z Load duration curve reportu	41
5.6	Tabulka a graf pro jednu veličinu v Demand reportu	42
5.7	DISDIP tabulka	46
5.8	Graf průběhu veličin	46
5.9	Histogram	47
5.10	Kvartilový graf	47
6.1	Form pro generování reportů nad daty v databázi	52
8.1	Energy Elephant [15]: Graf spotřeby a interaktivní menu	59
8.2	Grafana [11]: Nastavení alarmu	59
8.3	Energomonitor [16]: Nastavení alarmu	60
8.4	Watticks [18]: Porovnání očekávaného a skutečného průběhu	60
8.5	Watticks [18]: Výpočet úspor při nulové spotřebě v označený čas	61
8.6	Watticks [18]: Porovnávání tarifů	61
8.7	Monsol [17]: Porovnání spotřeby a vlastní produkce	61
8.8	Power Radar [7]: Porovnání spotřeby	62
8.9	Power Radar [7]: Sankeyho diagram	62
8.10	Energomonitor [16]: Interaktivní heatmapa a graf pro konkrétní den	63
8.11	Grafana [11]: Přidání poznámek	63
8.12	Watticks [18]: Porovnání tarifů	64
8.13	Navysis [8]: Export do různých formátů	64
8.14	Grafana [11]: Snapshot panelu	65
8.15	Grafana [11]: Embed	65
8.16	Grafana [11]: Sdílení odkazem	66

Seznam zkratek

EnM	Energy management (Management elektrické energie)
EnMS	Energy management systems (Systémy managementu e. energie)
WinForms	Windows Forms
PQ	Power Quality (Kvalita elektrické energie)
THD	Total harmonic distortion (Celkové harmonické zkreslení)
DISDIP	Distribution Voltage Dip (Distribuční ztráty napětí)
GUID	Globally unique identifier (Univerzální unikátní identifikátor)
HVAC	Heating, ventilation and air conditioning (Vytápění, ventilace a klimatizace)

1 Úvod

Systém správy elektrické energie se dá charakterizovat jako uskupení technologií sloužících ke sledování, analýze a regulaci elektrické energie a dalších veličin (např. teploty). Jedním z cílů je informovat zákazníka o spotřebě, popřípadě i produkci elektrické energie, o dodržování příslušných norem a o dalších relevantních skutečnostech. Právě tomuto důležitému aspektu EnMS, tedy analýze, zpracování a zobrazování naměřených dat, se bude tato práce věnovat.

Teoretická část se bude zabývat rešerší EnM systémů, nabízenými službami a jejich provedením. Jelikož lze očekávat, že platformy budou často obsahovat podobné funkce, byť v trochu jiném provedení, mým cílem bude identifikace a popis právě těchto nejčastěji nabízených funkcí. Vzhledem k zadání se zaměřím především na popis vizuálního zpracování reportů a jejich obsahu. Výsledkem by měla být přehledná zpráva, ze které půjde čerpat při návrhu a implementaci praktické části mé práce. V ideálním případě bude ze zprávy vyplývat, jaké veličiny se v reportech sledují, jaké informace jsou pro uživatele důležité, jak jsou výsledky analýzy v reportu prezentovány, a kdy jsou pro uživatele tyto informace důležité. Následně blíže popíši problematiku reportů, jejich strukturu a nastavení. Zde začnu zakomponovávat i popis knihoven a dalších nástrojů, které se na základě rešerše rozhodnu používat při realizaci praktické části diplomové práce. Tato kapitola se bude zabývat i vztahem mezi reportem a daty, jejich strukturou a dělením uvnitř reportů. Návrh řešení této problematiky by měl využívat zmíněné knihovny a nástroje.

Následovat bude implementace knihovny umožňující nastavení a generování reportů. Knihovna by měla obsahovat metody pro načítání dat, jejich analýzu a zobrazení výsledků (vše dle vstupního nastavení). Výstupem budou reporty ve formátu pdf. Knihovna by měla být multiplatformní a použitelná ve webových i desktopových aplikacích. Nakonec realizuji samostatně spustitelný program umožňující uživateli pracovat se zmíněnou knihovnou. Uživateli nabídnu možnost zvolit zdroj dat, zadat požadované nastavení reportu a nakonec report vygenerovat. Užitečnou funkcí, kterou bych do programu rád zakomponoval, je také možnost uživatele upravovat vzhled reportů, resp. jejich šablon, potenciálně i návrh nových šablon. Za plánovaný výsledek tedy považuji funkční program schopný generovat smysluplné reporty zabývající se kvalitou elektrické energie, hlášením anomálií a důležitých událostí, popř. dalšími podobnými jevy. Funkčnost a efektivitu mého kódu otestuji na reálných datech poskytnutých mým vedoucím.

Po realizaci opět všechny výsledky mého snažení stručně popíši v této zprávě. Na závěr moji práci zhodnotím a navrhnou potenciální rozšíření, úpravy a zdokonalení programu do budoucna. Výsledný program by měl být použitelný v praxi

jako součást systému energetického managementu schopného konkurovat ostatním produktům na trhu.

2 Systémy pro správu elektrické energie

2.1 Popis

Pojmem systém správy elektrické energie myslíme aktivní monitorování, kontrolu, optimalizaci, organizaci spotřeby elektrické energie za účelem minimalizace nákladů. Správa energie zahrnuje drobné kroky, jako je sledování měsíčních účtů za energie a vylepšení na úsporné žárovky. Může ale také znamenat rozsáhlejší zlepšení, jako je přidání izolace, instalace odrazivé střešní krytiny nebo vylepšení zařízení pro vytápění, větrání a klimatizaci (HVAC) pro optimalizaci energetického výkonu [1]. Typicky se jedná o monitorování spotřeby energie, ukládání naměřených dat, hledání příležitostí pro snížení spotřeby energií (pomocí analýzy naměřených dat), využití nalezených příležitostí pro implementaci úsporných opatření a sledování, zda tato opatření mají pozitivní efekt [2].

Podle Mezinárodní organizace pro normalizaci (ISO) zahrnuje systém pro správu energie vytvoření a implementaci politiky pro správu energie, stanovení dosažitelných cílů pro využití energie a navrhování akčních plánů pro jejich dosažení a měření pokroku. To může zahrnovat implementaci nových energeticky úsporných technologií, snižování plýtvání energií nebo zlepšování současných procesů za účelem snížení nákladů na energie [3]. ISO 50001 [4] je mezinárodní standard, který stanovuje požadavky na systém managementu energií. Je určen všem organizacím bez ohledu na velikost či odvětví. Je vhodný jak pro komerční sektor, tak pro vládní instituce a zařízení. ISO 50001 pomáhá mezinárodním organizacím prosazovat osvědčené postupy při managementu energií a posiluje tak energetické chování organizací. Standard poskytuje návody a rady v oblasti benchmarkingu, měření, dokumentování, psaní zpráv o zlepšení energetické účinnosti a plánovaném snížení emisí skleníkových plynů. ISO 50001 také může pomoci organizacím při hodnocení a výběru nových technologií, jež zvažují nakoupit za účelem snížení energetické náročnosti [5].

Rešerší EnM systému jsem se zabýval v mém diplomovém projektu [6], ze kterého v této kapitole vycházím. Obvykle nabízené funkce na EnMS platformách, se kterými jsem se setkal na internetu, jsem rozdělil do několika základních kategorií, které zde podrobněji popíši.

Základním kamenem EnMS systémů je schopnost měřit elektrické veličiny pomocí měřících přístrojů a uchovávat data v databázi či souborech. Vývojáři se snaží podporovat co nejvíce způsobů komunikace (Wi-Fi, Ethernet, celulární radiová síť, Modbus TCP, ...). Firmy někdy nabízí své vlastní hardwarové prvky, které měří energii, nebo z měřících přístrojů přijímají data a posílají je do centrálního úložiště.

Alternativně, import dat do systému může probíhat pouze softwarově (jako csv soubory, z SQL databáze, pdf soubory faktur za spotřebu energie, ...). S naměřenými a uloženými daty se pomocí různých softwarových nástrojů dále pracuje. Následuje popis softwarových funkcí, které EnMS platformy většinou nabízejí.

- **Vizualizace dat v dashboardech:** Hlavní část EnMS umožňující zobrazovat data se označuje jako „dashboard“. Jedná se o interaktivní panel tabulek a grafů s různými informacemi o průběhu a vlastnostech veličin realizovaný jako součást webové, desktopové či mobilní aplikace. Uživatel si vybírá časové období, přístroje, veličiny a areály, které chce v dashboardu zobrazit. Nabízí se mu různé interaktivní grafy, tabulky či jiné moduly k zobrazení. Poskytuje se možnost filtrování a popisování dat, analytické funkce a různé druhy vizualizace výsledků. Obsah dashboardu je často možné dynamicky měnit (viz obr. 8.1).
- **Alarmy a notifikace:** Uživatel má možnost nastavit si notifikace a alarmy upozorňující ho na výskyt zvláštních událostí či porušení pravidel, která si sám definoval (viz obr. 8.2 a 8.3). Cílem je co nejrychleji uživatele informovat o důležitých událostech způsobem, jenž si sám zvolí, aby mohl co nejdříve reagovat. Notifikace mohou mít formu emailu, SMS zprávy, zaslání upozornění do aplikací 3. stran (Slack, PagerDuty, ...) apod.
- **Benchmarking a predikce průběhu:** Na základě průběhu veličin elektrické energie v minulosti nebo pomocí veličin, které mají na spotřebu vliv (např. venkovní teplota), se předvídá průběh veličin v budoucnosti a vytváří se různé prediktivní modely. Tyto funkce mohou pomoci zlepšit kvalitu elektrické energie, snížit výdaje atd. Dále lze modelovat očekávané či požadované průběhy (baseline) či limity a později je porovnávat se skutečným průběhem elektrických veličin (viz obr. 8.4).
- **Archivace a validace dat, reportů, faktur:** Platformu lze používat jako centrální úložiště důležitých dokumentů: faktur, smluv, vygenerovaných reportů atd., které může uživatel procházet a porovnávat. Některé platformy umožňují provádět automatizovanou kontrolu obsahu těchto dokumentů, vytěžování faktur a zefektivnění procesu jejich zpracování.
- **Optimalizace financí** Platformy, které se zabývají správou financí, mohou kromě zpracování a validace faktur či ověření spočítaných výdajů využívat informace o tarifech a porovnat ceny u různých dodavatelů energie. Některé platformy umí doporučovat vhodné kroky ke zmenšení nákladů, upozorňují na slevy energie, předvídají spotřebu v následujícím měsíci podle různých indikátorů a snaží se uživateli nabídnout možnosti úspor (viz obr. 8.5).
- **Generování reportů:** Generování reportu může probíhat jednorázově nebo periodicky (za týden, za měsíc,...), kdy se po uplynutí zadaného časového úseku vygeneruje nová iterace reportu. Uživatel si report navrhuje sám, může si vybírat z již vytvořených šablon nebo může požádat firmu o vytvoření reportu

dle svých představ. Vývojáři se snaží uživateli nabídnout co největší škálu modulů, grafů a funkcí k snadnému vytvoření reportu dle jeho požadavků. Obsahem reportu je většinou shrnutí naměřených dat ke konkrétnímu okamžiku a zobrazení důležitých údajů o vlastnostech těchto dat. Reporty mohou být ve formátu pdf, html, docx, xlsx, csv, jako obrázek atd. Někdy je možné reporty distribuovat přes zadané emailové adresy nebo mohou být dostupné na dané platformě (na webu, v aplikaci, ...). Reporty často zůstávají uložené v centrálním úložišti, aby měl uživatel přístup ke všem iteracím reportu. Najdeme zde stejné grafické prvky jako v dashboardech: grafy, tabulky atd. Pro čtení reportu není potřebný přístup k dané platformě (k aplikaci či webové stránce). Z toho důvodu také nejsou interaktivní, dynamicky se nemění a obsahují shrnutí požadovaných informací vzhledem k času, kdy byly generovány.

2.2 Prozkoumané EnM platformy

Během rešerše jsem prošel velké množství různých EnM platform. Většina obsahuje nástroje sloužící k účelům, které jsem popsal v předešlé sekci. Tato sekce obsahuje seznam platform, jež jsem při rešerši procházel spolu s tabulkou nabízených služeb. Nabízené služby jsou u každé platformy trochu jiné, tudíž je tabulka zobecněna tak, aby vyjadřovala, zda platforma nabízí nějaké služby v základních kategoriích, které jsou popsány výše (zkráceně dashboardy, predikce, reporty, alarmy, archivace, finance).

V tabulce jsou obsaženy i platformy, které nemusí být používány pouze pro EnM, např. nástroje pro obecnou vizualizaci a analýzu dat jako Grafana či InfluxDB, databáze navržená pro ukládání, dotazování a vizualizaci časově závislých dat, nicméně všechny tyto platformy je možné využít v energetickém managementu. Během procházení těchto platform a seznamování se s jejich obsahem jsem získal povědomí o tom, které informace se uživateli zobrazují a jak. Základní funkcí je informování o spotřebě a produkci elektrické energie. Tyto informace se často zobrazují v čárových grafech jako křivka spotřeby a produkce (viz obr. 8.7) nebo v koláčových a sloupcových grafech jako suma spotřeby či produkce za určitý časový interval (viz obr. 8.1 a 8.8). Jelikož data často pocházejí z různých zdrojů (měřících přístrojů), dělí se často do několika kategorií, např. po místnostech, budovách, areálech atd. Spotřeba a produkce se tedy obvykle zobrazuje ve více grafech či datových prvcích grafu, aby bylo jasné, jaká je spotřeba v jednotlivých komponentách monitorovaného objektu (viz obr. 8.8). K zobrazení spotřeby po jednotlivých komponentách se někdy využívá Sankeyho diagram (viz obr. 8.9), který znázorňuje tok hodnot mezi různými kategoriemi [27].

Další způsob, kterým se zobrazená data v dashboardech a reportech dělí, je členění po časových úsecích (dny, týdny, ...). Graf, který se k těmto účelům někdy používá, je heatmapa: nástroj, jenž hodnoty ve zdroji dat vizualizuje pomocí barev [28]. Některé platformy zobrazují heatmap tabulku, jejíž buňky reprezentují jednotlivé časové úseky z celkového intervalu (např. dny z měsíce) a jejich obarvení indikuje spotřebu či jinou měřenou veličinu v tomto úseku (viz obr. 8.10). Uživatel si také

Platforma	Dashb.	Predikce	Reporting	Alarm	Archiving	Finance
PowerRadar[7]	ANO	ANO	ANO	ANO	ANO	NE
Navisys[8]	ANO	NE	ANO	ANO	NE	ANO
JLL[9]	ANO	ANO	ANO	ANO	NE	NE
InfluxDB[10]	ANO	NE	NE	ANO	NE	NE
Grafana[11]	ANO	NE	NE	ANO	NE	NE
Smarkia[12]	ANO	ANO	ANO	ANO	ANO	ANO
Electrex[13]	ANO	ANO	ANO	ANO	NE	ANO
Energis[14]	ANO	ANO	ANO	ANO	ANO	ANO
EnergyElephant[15]	ANO	ANO	NE	NE	NE	ANO
EnergoMonitor[16]	ANO	NE	ANO	ANO	NE	ANO
Monsol[17]	ANO	NE	ANO	ANO	NE	NE
Watticks[18]	ANO	ANO	ANO	ANO	NE	ANO
OPIT[19]	ANO	ANO	ANO	ANO	NE	ANO
Flowbox[20]	ANO	ANO	ANO	ANO	ANO	ANO
Elcontrol e. net[21]	ANO	NE	ANO	ANO	ANO	ANO
Janitza[22]	ANO	NE	ANO	ANO	NE	NE
SchneiderElectric[23]	ANO	NE	ANO	ANO	ANO	ANO
beedata[24]	ANO	NE	ANO	ANO	NE	NE
Entronix[25]	ANO	ANO	ANO	ANO	NE	ANO
Skyfoundry[26]	ANO	ANO	ANO	ANO	NE	NE

Tabulka 2.1: Prozkoukamé EnM platformy

často vybírá, jaký časový interval chce zobrazit a po jakých časových intervalech se mají jednotlivá zobrazovaná měření dělit (viz obr. 8.1).

Některé platformy mají možnost přidávat k informacím zobrazených v dashboardech různá metadata. Grafana nabízí možnost přidat textovou poznámku k jednomu časovému bodu či úseku v grafu (kliknout v grafu + zvolit přidání anotace). K anotaci lze přidat další metadata (tagy) (viz obr. 8.11). Platforma Watticks zase umožňuje přidávat poznámky a v nich tagovat uživatele (například zaměstnance stejné firmy, kteří mají k datům přístup). Tagovanému uživateli přijde upozornění na přidanou poznámku, ve které je zatagován. To slouží k usnadnění komunikace v týmu. Watticks také umožňuje pracovat s tarify a zvolit, v jakou dobu jsou aktivní. Následně můžeme porovnávat finanční výhodnost současného tarifu s alternativními tarify (viz obr. 8.6).

Zajímavou oblastí práce s dashboardy je sdílení panelů a dat z dashboardů. Pokud se nevyžaduje na straně příjemce přihlášení či podobná autentifikace, je možné sdílet všechna možná data, ať už je Dashboard nastaven jakkoliv. To je na některých platformách řešeno těmito způsoby: Za účelem sdílení lze panel exportovat do grafického formátu jako soubor png, jpg, svg,... (viz obr. 8.13). Tato funkce je přítomna na většině platform. Také je možné mít vyrenderovaný obrázek uložený na platformě a poskytovat k němu přístup pomocí odkazu. V nastavení lze zvolit, po jaké době se obrázky z platformy mažou. Sdílení Snapshotem představuje způsob interaktivního zobrazení, které ale poskytuje přístup pouze k vybraným datům. Nad data lze opět používat nástroje, které platforma nabízí, neposkytuje se však přístup

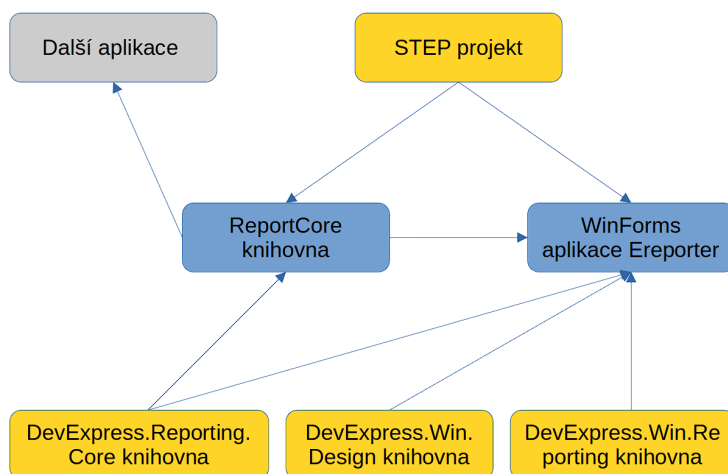
ke všem datům, nýbrž pouze výlučně k datům v panelu. Lze nastavit časový úsek, v němž snapshot existuje. Po vypršení zadaného času je snapshot vymazán a již ho nelze používat. Smazat snapshot můžeme i ručně (viz obr. 8.14). Přístup k panelům lze omezit. Od uživatele se může vyžadovat přihlášení nebo musí být dashboard nastaven jako veřejný. Jinak se příjemce k datům nedostane. Panely i dashboardy jako takové lze umožnit sdílet přes url. V případě panelu se sdílí odkaz na celý dashboard, ale sdílený panel je aktivní v módu celé obrazovky. Informace, který dashboard je sdílený (který bude v režimu celé obrazovky), časový úsek panelu atd., se předává jako parametr url. Nad daty lze používat nástroje, které platforma nabízí. Uživatel musí mít k danému panelu povolený přístup. Dashboard může být veřejný nebo lze vyžadovat přihlášení uživatele na platformu (viz obr. 8.16). Embed sdílení umožňuje pomocí html kódu přidat panel do libovolné webové stránky. Pokud není umožněn anonymní přístup, panel se zobrazí pouze, pokud je návštěvník webové stránky přihlášen na platformě, která tuto službu poskytuje (např. Grafana; viz obr. 8.15).

2.3 Shrnutí

V této kapitole jsem stručně shrnul poznatky získané během procházení různých EnM platforem. Popsal jsem základní funkce, jež tyto platformy nabízí, zmínil jsem některé zajímavé nástroje a prvky vyskytující se na některých platformách a seznámil jsem se s informacemi a daty, která jsou často zobrazována v dashboardech a reportech (resp. jejich grafických prvcích). V následující kapitole popíši funkce a nástroje, jež jsem se rozhodl využívat pro vytvoření praktické části této práce. Informace z této kapitoly mohou sloužit jako inspirace, co by využití technologie měly být schopné pomoci vytvářet.

3 Využití knihovny a funkce

Praktickou část mé práce jsem se rozhodl rozdělit do několika logických celků. Analyzování dat a generování reportů bude nezávislé na aplikaci s uživatelským rozhraním sloužícím pro zpracování vstupu od uživatele. Proto bude vytváření reportů implementováno jako .NET 7 knihovna, jíž bude možné začlenit do různých typů aplikací. Ve zbytku této práce budu tuto knihovnu označovat názvem „ReportCore“. Druhá část praktické práce obsahuje implementaci desktopové aplikace s uživatelským rozhraním, jejíž účelem bude přijímání vstupních instrukcí od uživatele a na jejich základě generovat pomocí ReportCore knihovny reporty. Tuto aplikaci jsem se rozhodl pojmenovat „EReporter“. V této kapitole popíši nástroje a knihovny využití k implementaci ReportCore a EReporteru.



Obrázek 3.1: Základní stavební bloky projektu
Využití nástroje žlutě, mnou implementované nástroje modře, potenciální další aplikace šedě

3.1 DevExpress

DevExpress [29] je softwarová firma zabývající se vývojem softwarových vývojářských nástrojů, které jsou ve většině případů určeny pro vývoj webových, desktopových a mobilních aplikací, vývoj komponent uživatelského rozhraní, správu dat atd. Mezi oblasti, jimiž se DevExpress zabývá, je i tzv. reporting, tedy vývoj

a práce s reporty. Jedním z hlavních DevExpress produktů je DXperience, všestranná sada nástrojů pro vývoj desktopových aplikací pro platformu .NET. DXperience zahrnuje mnoho užitečných nástrojů, jako jsou reporty, komponenty uživatelského rozhraní, nástroje pro spravování a analýzu dat a mnoho dalších. DXperience také nabízí nástroje pro vývoj webových aplikací, včetně komponent a nástrojů pro ASP.NET a Angular. Tyto nástroje usnadňují vývojářům vytváření webových aplikací, především jejich uživatelského rozhraní, a poskytují mnoho užitečných funkcí, které vývojář nemusí sám implementovat. Podobné produkty jsou nabízeny i pro vývoj mobilních aplikací na operační systémy Android. DevExpress produkty zabývající se reportingem jsou pro moji práci zvláště důležité. DevExpress Reporting nabízí mnoho prvků pro tvorbu reportů, včetně textových polí, tabulek, grafů. Vlastnosti těchto prvků jako jsou barvy, fonty, umístění, velikost a další lze snadno přizpůsobovat jak při návrhu, tak dynamicky za běhu aplikace. Užitečnou funkcí je možnost vytvářet subreporty, což jsou reporty vložené do jiného reportu. To umožňuje vývojářům vytvářet složité reporty s mnoha úrovněmi detailů. Reporting má také vestavěnou podporu pro tisk a export reportů do různých formátů. Uživatelé mohou rovněž interaktivně filtrovat, řadit a seskupovat data přímo v reportu. Škála nabízených DevExpress služeb je velmi bohatá, uživatelsky přívětivá a s přímočarou dokumentací online. Proto jsem se rozhodl k vývoji reportů využít právě tento produkt.

3.1.1 DevExpress.Reporting.Core

Pomocí uvedené knihovny jde vytvářet a upravovat reporty, přistupovat ke komponentům reportu (tabulky, grafy,...) a generovat reporty nad zvolenými daty do zvolených formátů (pdg, docx, xlsx,...) [30]. Základním kamenem je třída XtraReport [31] reprezentující instanci reportu. Struktura, obsah a chování reportu se definuje následujícími komponenty reportu:

Pásky definují základní strukturu reportu a reprezentují jednotlivé části reportu, do kterých lze umísťovat jednotlivé komponenty reportu. „Detail“ je páska vytištěná pro každý nevyfiltrovaný řádek ze zdrojové tabulky dat. „Header“ a „Footer“ pásky mohou sloužit jako zápatí a záhlaví celého reportu, podreportů či jednotlivých stránek dokumentu apod. Pásky jsou většinou vertikální (skládají se za sebe odshora dolů) ale některé pásky je možné řadit i vertikálně.

Ovládací prvky jsou grafické komponenty reportu. Mohou to být textová pole, tabulky, buňky tabulek, různé typy grafů, obrázky a tak dále. Ovládací prvky se umísťují do pásek a jejich umístění v reportu záleží právě na tom, ve které pásce se nachází.

Vlastnosti ovládacích prvků definují funkci a chování ovládacích prvků. Může se jednat o umístění, velikost, viditelnost, text, barvu, font, zdroj dat, informace ze zdroje dat, se kterými má prvek pracovat, odsazení textu, zarovnání a všechna další nastavení.

Řazení a groupování rozhoduje o tom, v jakém pořadí se budou data v reportu zobrazovat. Libovolnému sloupci ve zdroji dat můžeme přiřadit vzestupné či sestupné řazení a podle toho se budou řádky ze zdroje dat v reportu zobrazovat. Groupování slouží k seskupení dat dle hodnoty vybraného sloupce. Pro každé seskupení jde přidat pásku záhlaví a zápatí. Problematice groupování se budu podrobněji věnovat v dalších částech této práce. Seskupení i groupování jdou samozřejmě kombinovat, řetězit.

Parametry reportu jsou proměnné různých typů, jejichž nastavení může ovlivnit chování reportu. Může se jednat o boolean hodnotu nastavující viditelnost jednotlivých částí grafu, číselnou proměnnou použitou při výpočtech, string, jehož hodnota se zobrazí ve vybraném textovém poli atd.

Styly fungují jako definují souhrn nastavení grafických vlastností pro libovolný ovládací prvek. Může zde být nastaven font, velikost a vlastnosti písma, odsazení, ohrazení, barva textu, barva pozadí atd. Vytvořené styly jde pak přiřazovat jednotlivým ovládacím prvkům reportu.

Kalkulovaná pole jsou virtuální veličiny, které se počítají z polí obsažených ve zdroji dat, z parametrů reportu a z konstant. S těmito poli jde následně pracovat jako s kterýmkoliv jiným polem ze zdroje dat. Pokud chceme několik polí sčítat a pracovat se součtem jako s jednou proměnnou, vynásobit všechny prvky pole konstantou a podobně, je možné tyto operace provést právě pomocí kalkulovaných polí.

Výrazy jsou speciální vlastností ovládacích prvků, které umožňují zobrazit v ovládacích prvcích hodnoty ze zdroje dat. Tabulky, textová pole, grafy a další prvky mohou zobrazovat jednotlivé hodnoty vyjádřené ze zdroje dat. Výrazy ale také umožňují s hodnotami v polích výpočty, kombinace a agregace hodnot (maximum, minimum, suma, ...). Příkladem může být zobrazení průměrné hodnoty, minima a maxima vybraného pole z záhlaví reportu, zobrazení všech hodnot daného pole v tabulce, zobrazení histogramu v grafu atd. Všechny tyto operace a způsob jejich zobrazení v ovládacích prvcích reportu se definuje pomocí jejich výrazů.

Skripty jsou programátorem definované algoritmy navázané na eventy v reportu. Daný skript se provede ve chvíli, kdy nastane event, na který je skript navázán (např. při tisknutí dané komponenty během generování reportu). Skripty je možné psát v programovacích jazycích C#, Visual Basic a JScript .NET. Jelikož jsou skripty součástí reportu, při uložení šablony reportu do souboru jsou skripty v čitelné formě součástí tohoto souboru. To vyvolává spoustu otázek z pohledu bezpečnosti aplikace a proto se jejich použití nedoporučuje. Proto jsem se rozhodl skripty v mých reportech nevyužívat.

Instanci XtraReport třídy lze vytvářet a upravovat za běhu programu. Jednotlivé instance jde také spojovat do jednoho dokumentu. Po přiřazení dat lze zavolat metodu XtraReport.CreateDocument, která vygeneruje instanci PrintingSystemBase.Document třídy reprezentující dokument připravený k zobrazení a exportu.

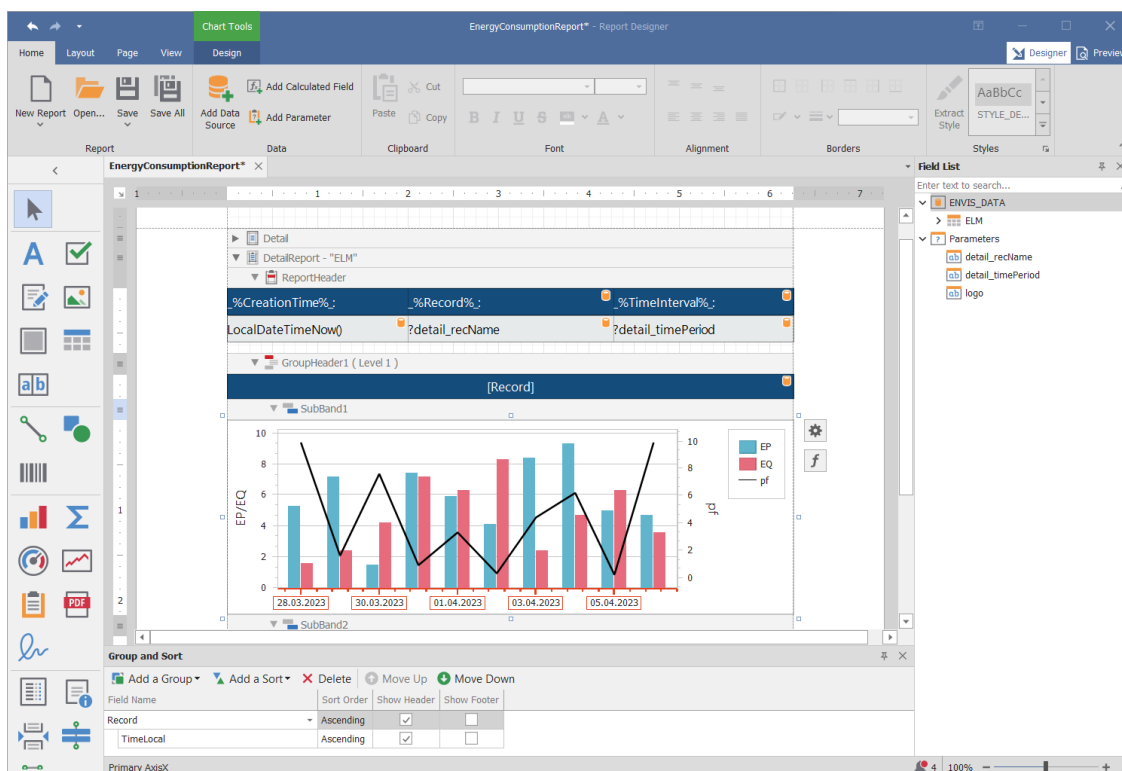
Tato instance je uložena uvnitř XtraReport instance, konkrétně v XtraReport.PrintingSystem.Document [32]. V tuto chvíli již je možné report vygenerovat a exportovat do zvoleného formátu. Podporované formáty jsou pdf, docx, rtf, image, csv, txt, xls, xlsx, hmtl, mht a ppt. S použitím MailKit knihovny lze reporty rovnou posílat jako email zprávu [33]. Strukturu vytvořené instance XtraReport třídy lze uložit do souboru xml s příponou repx. Repx soubor obsahuje informace o všech komponentách grafu spolu s jejich uspořádáním. Díky tomu ho lze využít pro uložení struktury reportu na disk a také při inicializaci XtraReport třídy. (Struktura a komponenty reportu jsou inicializovány dle informací v souboru.) Repx soubor tudíž funguje jako šablona reportu, která uchovává všechny informace o obsahu reportu. Repx šablony jsou důležité i při práci s nástroji z DevExpress.Win.Design a DevExpress.Win.Reporting knihoven. Jejich popis bude proto následovat v další sekci této práce (3.1.2). Kromě šablon reportů lze do souborů ukládat i další typy šablon, např. šablony stylů. Styly jde opět dynamicky měnit během programu, ukládat je do souborů, ze souborů načítat a přiřazovat reportům. DevExpress knihovny poskytují pro vytváření reportů velkou škálu různých nástrojů a funkcionalit. Ty prvky, které se rozhodnu využívat v mé knihovně, popíši spolu s jejich účelem a využitím podrobněji v kapitolách 4 a 5.

3.1.2 DevExpress.Win.Design, DevExpress.Win.Reporting

DevExpress.Win.Design a DevExpress.Win.Reporting poskytují nástroje pro vytváření uživatelského rozhraní WinForms aplikací. DevExpress poskytuje obdobné produkty pro další Microsoft technologie, např. ASP.NET nebo Wpf. Protože jsem se rozhodl pro technologii WinForms, budu nakonec využívat právě tyto dvě knihovny.

Jednou z hlavních funkcí aplikace EReporter bude vizualizace vygenerovaných reportů a přívětivé uživatelské rozhraní, kde bude mít uživatel možnost s reporty pracovat. Uživatel by měl mít možnost zvolit si, do jakého formátu chce vygenerovaný report exportovat, nicméně chtěl bych, aby si uživatel mohl vytvořený report prohlédnout ještě předtím, než se ho rozhodne uložit jako soubor libovolného typu. Dále by měl EReporter obsahovat nástroj, pomocí kterého bude uživatel moci šablony reportů (repx soubory) sám vytvářet a upravovat. Zmíněné DevExpress knihovny nabízí nástroje, které přesně pro tyto účely půjde využít.

ReportDesignTool [34] je nástroj pro editování reportů ve WinForms aplikacích. Při inicializaci se mu předává instance XtraReport třídy, jejíž struktura a obsah se následně graficky zobrazí v novém designer okně. Okno obsahuje řadu nástrojů, které lze využít pro manipulaci s obsahem reportu a nastavením téměř všech vlastností XtraReportu. Mezi nejdůležitější funkce patří přidávání, odstraňování a upravování pásek, popisků, tabulek, grafů a všech dalších komponent reportu, nastavení jejich vlastností (umístění, text, viditelnost, styl, ...), nastavení řazení dat a groupování reportu, nastavení záložek a mapy dokumentu, správa kalkulovaných polí, přidávání a upravování stylů, filtrace zdroje dat, nastavení parametrů stránky (velikost, okraje, orientace, ...), definování parametrů reportu, definování výrazů, přiřazování obsahu ovládacím prvkům reportu (konstanty, pole ze zdroje dat reportu, parametry, kalkulovaná pole, ...), atd. (viz obr. 3.2).

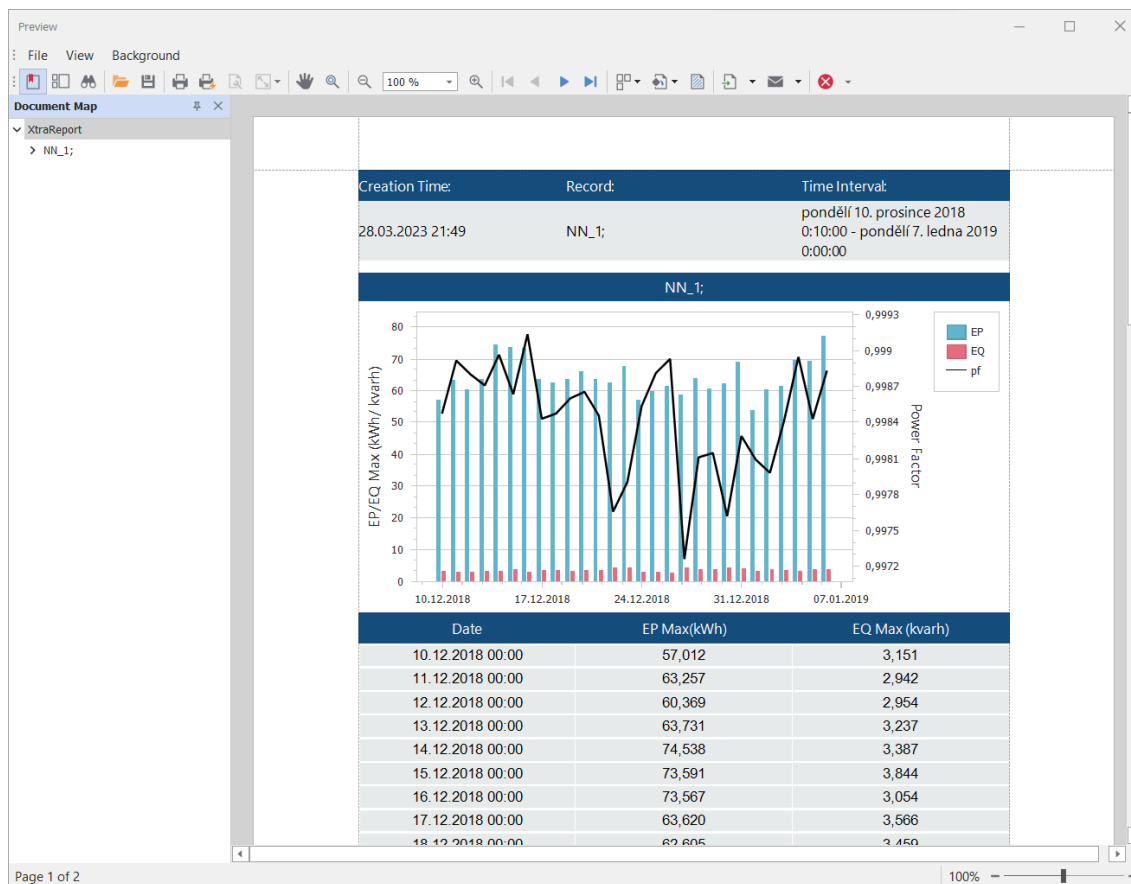


Obrázek 3.2: ReportDesignTool s otevřenou šablonou reportu

ReportPrintTool [35] na rozdíl od ReportDesignTool není určen pro úpravu reportů, nýbrž pro vygenerování reportu nad specifickými daty a zobrazení v grafickém okně. Po zobrazení vygenerovaného reportu nabízí ReportPrintTool možnost exportování dokumentu do různých formátů (pdf, html, xlsx, docx atd.), vytisknutí reportu, přiblížení/oddálení, hledání v zobrazeném dokumentu, změny viditelných parametrů reportu, vygenerování nového reportu po změně parametrů, změny základního nastavení jako barvy pozadí atd. (viz obr. 3.3).

3.2 STEP

Projekt STEP, jenž mi poskytl vedoucí mé diplomové práce, obsahuje soubor různých tříd pro framework .NET7, které budu moci využít pro načítání a analýzu dat nacházejících se buď v souboru nebo databázi. Zdroj dat je reprezentován třídami FileDataSource (soubor) a DBDataSource (databáze), jejichž metody mi umožňují čtení a zápis dat a různých informací ze souboru či databáze. Další knihovny, jež bude možné využívat jsou kalkulační a agregační knihovny, které lze použít pro výpočty a analýzu dat. Také se zde nachází pro moji aplikaci užitečné překlady slov a frází, v podobných aplikacích často používaných. Díky tomu budou mnou vytvořené nástroje přeložitelné do různých jazyků. Důležité funkce tohoto projektu blíže popíši v následujících částech této kapitoly. STEP projekt není množina knihoven, které mohu používat. Jelikož bude v budoucnu navržené řešení repor-



Obrázek 3.3: ReportPrintTool s vygenerovaným reportem

tingu úzce spojeno s ostatními STEP knihovnami, bylo mi dovoleno upravovat kód a kontribuovat k obsahu STEP projektu.

3.2.1 DataSource

Třídy FileDataSource a DBDataSource jsou potomky abstraktní třídy DataSource a reprezentují zdroj dat. Metoda DataSource.GetRows vrací instanci třídy RowCollection. Jak lze odvodit z názvu, jedná se o kolekci řádků obsahujících hodnoty zvolených veličin, čas měření, konfiguraci přístroje, který řádek naměřil a další informace. Při načítání dat je třeba specifikovat několik parametrů, jež definují, jaká data se mají načíst. Jedná se o následující položky:

Quantity je třída reprezentující veličinu z datasource, např. jednotlivá měření napětí (U1, U2, ...), proudů, jejich harmonických složek, výkonů atd. Při načítání dat je třeba definovat, pro které veličiny se má celý proces provádět. Při inicializaci instance třídy Quantity je třeba specifikovat název veličiny, archiv, v němž se veličina nachází, popř. i její jednotku.

DateRange je celkový časový interval, pro který se mají data načíst. Tuto třídu tvoří dvojice DateTime proměnných a pár pomocných metod.

Groups reprezentují přístroje, ze kterých se data mají načítat. Přístroje jsou uspořádány ve stromové struktuře. Úzel může být buď přístroj, nebo skupina přístrojů a každý přístroj či skupina může obsahovat další podskupiny a přístroje. Při načítání dat je třeba definovat, ze kterých přístrojů chceme data načítat pomocí jejich globálně jedinečného identifikátoru (GUID).

Archive definuje, z jakého archivu se data mají načítat. Každý přístroj může obsahovat různé archivy, např. archiv s veličinami spotřeby a produkce elektrické energie (rozdělené na jalovou a činnou), „Main“ archiv může obsahovat měření napětí, proudu, výkonu atd. Archiv specifikujeme proměnnou typu byte, protože každému archivu odpovídá právě jedna byte hodnota.

Parametry metody GetRows budou předávány dle nastavení reportu nebo podle potřeby konkrétního reportu. K načtení DBDataSource je třeba autorizace ze strany uživatele. V databázi mohou být uloženy jak samotná data, tak nastavení a cesta k šabloně reportu. V případě FileDataSource budou data, nastavení reportu a repx šablona uloženy ve třech různých souborech na disku.

3.2.2 Calculations

Kromě načítání dat a informací umožňují agregační a výpočetní třídy a metody nacházející se v knihovně Calculations provádět nad načítanými daty různé výpočty. Agregování a analyzování dat je samozřejmě možné provádět po načtení dat (a tento postup v některých případech využiji), z hlediska časové efektivity je ale většinou výhodnější provádět různé agregace již během načítání dat. Právě k tomu slouží algoritmy z třídy Aggregations, které jsou schopné pro zvolené části dat spočítat např. průměry, maxima, minima atd. již během prvotního načítání dat z DataSource. Ke specifikaci, jak se data mají agregovat, slouží následující typy tříd:

Selektory slouží k rozdělení dat na časové a datové intervaly, pro které se má agregace provést. Pokud chceme vypočítat denní průměry, selektor, který použijeme, bude rozdělovat data po dnech.

Agregátory slouží k definici samotné agregace. Pokud chceme počítat percentil dané veličiny, je třeba definovat algoritmus, podle kterého se tento percentil bude počítat. Výhodou nástrojů ze STEP projektu je, že vytvořené agregátory mohou provádět své výpočty paralelně.

Quantity je název třídy reprezentující veličinu z datasource, např. jednotlivá měření napětí (U1, U2, ...), proudů, jejich harmonických složek, výkonů atd. Při agregaci dat je třeba definovat, pro které veličiny se má celý proces provádět.

Tyto algoritmy budu doplňovat vlastními agregačními funkcemi, pokud bude třeba. V situacích, kdy to bude možné, budu využívat možnost paralelní agregace (více agregátorů bude pracovat nad načtenými daty zároveň), čímž by se měla zmenšit časová náročnost implementovaných algoritmů.

3.2.3 Resources

Třída Resources obsahuje překlady slov a frází, které se v podobných aplikacích často vyskytují. Jsou zde obsažena slova přeložená do angličtiny, němčiny, češtiny, polštiny a dalších jazyků. Tyto překlady jsou uloženy v tzv. souborech prostředků. Jedná se o soubory ve formátu xml s příponou resx. Každé slovo má unikátní jméno, ke kterému je v každém jazyce přiřazen jeden překlad tohoto slova. .NET k práci se soubory prostředků nabízí třídu ResourceManager, jež pro zvolený jazyk najde pro každé unikátní jméno odpovídající překlad. Pro usnadnění práce s překlady použijte ve Visual studiu doplněk ResxManager, populární nástroj pro práci se soubory prostředků. Pomocí něho lze zobrazit všechna slova a jejich překlady ze všech resx souborů v přehledné tabulce. V ní jde všechna slova a překlady rychle najít, číst, upravovat, mazat i přidávat.

3.3 Shrnutí

Firma DevExpress nabízí produkty pro vývoj desktopových aplikací i nástroje pro vytváření reporting služeb. Ke generování reportů využijte třídy a funkce z knihovny DevExpress.Reporting.Core, jejíž třída XtraReport je přesně pro tyto účely určena. Dále využijte nástroje z DevExpress knihoven DevExpress.Win.Design a DevExpress.Win.Reporting pro vývoj aplikace EReporter, která bude moci reporty vytvářet, upravovat, generovat, zobrazovat a exportovat do pdf či jiných formátů. Knihovny z projektu STEP budou sloužit pro načítání a analýzu dat z databáze či souboru a také pro doplnění EReporteru o překlady do různých jazyků. Za zmínku také stojí, že při psaní této práce jsem využil ChatGPT, rozsáhlý jazykový model schopný generovat smysluplné texty v různých jazycích, odpovídat na otázky a poskytovat různorodé informace. Využil jsem ho k ucelenému formulování popisů některých funkcí a nástrojů, kterými se moje práce zabývá. Zatímco tato kapitola se soustředila především na popis samotných komponent obsažených v mé aplikaci, způsob jejich využití bude popsán v následujících kapitolách.

4 Knihovna ReportCore: Implementace reportů a jejich nastavení

Tato kapitola se bude zabývat popisem mnou implementované .NET7 knihovny ReportCore, její strukturou, třídami, metodami a vlastnostmi. Zatímco v minulé kapitole jsem se zabýval nástroji, které jsem použil při realizaci mého programu, zde popíši, jak jsem je použil při realizaci knihovny, která umožňuje generovat reporty v závislosti na vstupních požadavcích. Než jsem začal implementovat konkrétní reporty a jejich obsah, bylo třeba specifikovat princip vytváření reportů a implementovat společnou infrastrukturu, která se bude ke generování reportů používat. Ačkoliv je v zadání uvedeno, že algoritmy se mají implementovat v prostředí NET5/6, po domluvě s vedoucím jsem se rozhodl k využití NET7, které bylo při vytváření programu nejnovější verzí .NET.

4.1 Třída ReportBase a její potomci

Výchozím objektem je abstraktní třída ReportBase. Ta obsahuje metody a vlastnosti, které se často využívají při generování některých (nebo všech) reportů. Vlastnost (property) Setting typu ReportSetting obsahující nastavení reportu je pro vygenerování všech reportů nezbytná. Proto obsahuje konstruktor třídy Report parametr typu ReportSetting, kterým se zmíněná vlastnost inicializuje. ReportSetting třída je blíže popsána v sekci 4.3. Následuje popis důležitých metod obsažených v ReportBase.

Inicializace a přeložení reportu: Proces generování reportu začíná inicializací instance XtraReport třídy pomocí šablony reportu uložené v repx souboru. Jak jsem již zmínil, vytvořené reporty jsou překládány do aktivního jazyka Windows pomocí STEP knihovny Resources (viz sekce 3.2.3). Nejvhodnější chvíle pro přeložení reportu je hned po inicializaci XtraReport třídy, z toho důvodu se obě akce provedou v jedné, mnou implementované metodě LoadAndTranslateReportLayout. Načtení obsahu reportu se provádí pomocí XtraReport metody LoadLayout. K překládání slov do libovolného jazyka se používá .NET třída ResourceManager [36], jež poskytuje přístup ke zdrojům na základě jazyka za běhu programu. Všechna slova, která se v reportu mají přeložit, budou ohraničena řetězci „- %“ a „%-“. Proces překládky probíhá tak, že vlastnost Text všech ovládacích polí, která se v inicializované XtraReport instanci nachází, jsou prohledána a mnou implementovaná metoda načte

všechna slova, která se mezi zmíněnými řetězci nachází. Pomocí třídy `ResourceManager` jsou tato slova přeložena do aktivního jazyka a překlady slov nahrazují původní část textu ovládacího pole včetně ohraničujících řetězců „%“ a „%_“. Načtený a přeložený `XtraReport` metoda `LoadAndTranslateReportLayout` vrací jako návratovou hodnotu, která se při procesu generování reportu dále využívá.

Nastavení formátu čísel a času v reportu: V reportu by se měly datumy a hodnoty veličin zobrazovat ve formátu definovaném v nastavení reportu. Zvolit formát datumů a čísel v reportu jde pomocí vlastnosti ovládacích polí `TextFormatString` (typu `string`). Tento úkol jsem se rozhodl vyřešit následujícím způsobem: Textová pole reportu, kde se má při generování reportu nastavit zvolený formát čísel, budou pojmenována vždy s předponou „num“ (např. „numTableCell6“). Jména textových polí zobrazujících datumy budou mít zase předponu „time“. Na začátku generování reportu se inicializované instance `XtraReport` prochází všechna textová pole a u těch, jejichž jména mají předponu „num“ či „time“, nastaví vlastnost `TextFormatString` dle zvoleného nastavení reportu. Tento proces je implementován v třídách `SetNumTableCells`, `SetTimeTableCells` a `SetNumAndTimeTableCells`.

Řetězení reportů: Jednotlivé instance třídy `XtraReport` je kvůli groupování často nutné spojit do jednoho dokumentu (viz sekce 4.2). Proto jsem implementoval metodu `MergeReports`, která zřetězí libovolné množství reportů připravených k exportu do jediného nového reportu. K tomu používá metodu `XtraReport.ModifyDocument` [37], pomocí níž se do reportu dají přidávat stránky z cizích dokumentů.

Zjištění přítomnosti veličin v DataSource: Jednou z důležitých informací je, které veličiny se ve zdroji dat vyskytují. Tato informace je užitečná při analýze dat, protože potřebujeme vědět, zda se veličina, kterou chceme načíst a analyzovat, ve zdroji dat skutečně nachází. Metoda `GetMissingQuantitiesAsText`, které jako parametr předáváme jména veličin, se kterými chceme pracovat, načte jména všech veličin v instanci třídy `DataSource` pomocí její metody `GetQuantities` a vrátí jména těch veličin, jež se ve zdroji dat nenacházejí. Tyto veličiny pak můžeme vynechat a o jejich absenci informovat v reportu.

Generování mapy dokumentu: Mapa dokumentu jde realizovat pomocí vlastnosti `Bookmark`, kterou mají všechny ovládací prvky `XtraReportu`. Jelikož se ale bude výsledný report vytvářet z několika zřetězených instancí `XtraReport`, je třeba mapu konečného dokumentu upravit pomocí mnou implementované metody. Detailní popis této metody je popsán v sekci 4.2.3.

Pro vygenerování reportu je třeba specifikovat, nad jakými daty se má report generovat. To se uskutečňuje pomocí vlastnosti třídy `XtraReport` zvané `DataSource`. Tato vlastnost typu `object` reprezentuje data, nad kterými se má report vytvořit. Ačkoliv je typ vlastnosti `DataSource object`, pro správné fungování programu je

třeba aby byl XtraReport.DataSource přiřazen objekt, který implementuje rozhraní IList, IList nebo IBindingList, alternativně je možné použít třídu DataSet.

DataSet je .NET třída obsažená v namespace System sloužící jako mezipaměť dat načtených ze zdroje dat v paměti. Je hlavní součástí architektury ADO.NET. DataSet se skládá z kolekce DataTable objektů, které můžete vzájemně propojit s DataRelation objekty. DataTable třída představuje jednu tabulku dat v paměti [38]. Při práci s DataTable je třeba definovat její sloupce (DataColumn) v DataColumnCollection vlastnosti. Následně jde do DataTable přidávat řádky (DataRow). Vzhledem k tomu, že manipulace s DataSet a DataTable třídami je velmi jednoduchá a nevyžaduje implementaci žádného rozhraní, rozhodl jsem se DataSet využívat jako zdroj dat pro XtraReport. Po načtení, agregaci a analýze dat výsledky uložím do mnou vytvořené DataSet struktury, kterou následně přiřadím XtraReport vlastnosti DataSource.

Z třídy ReportBase dědí třídy EnergyReport, PowerQualityReport, AlarmsReport, HarmonicsReport a GeneralReport, v nichž jsou obsaženy metody pro generování specifických reportů, popř. i další metody, které se při jejich generování využívají. Důvodem, proč jsem se rozhodl pro tuto strukturu kódu, byl fakt, že mít všechny metody, struktury a podtřídy v jedné třídě ReportBase se ukázalo být velmi nepřehledné.

4.2 Dělení reportu na kapitoly

Jedním ze základních pilířů přehledné prezentace dat v reportech je členění reportu na různé části. Reporty se nejčastěji člení podle časových úseků (např. po hodinách, dnech, týdnech, měsících, ...) a přístrojů, které data naměřily. K tomuto účelu používám dva způsoby dělení reportu. Ty jsou popsány v následující části tohoto textu.

4.2.1 Groupování dat uvnitř kapitoly reportu

Ve výsledných reportech musí být data často rozdělena na požadované úseky, tzv. „zgroupována“, a pro jednotlivé úseky zobrazit požadované informace. Příkladem může být report spotřeby elektrické energie, který zobrazuje celkovou spotřebu, maximum a průměr spotřeby pro každý měřící přístroj po měsících. Jak již bylo zmíněno v kapitole 3, groupování dat lze specifikovat uvnitř reportu pomocí GroupField, GroupHeaderBand a GroupFooterBand tříd. Třída GroupHeaderBand reprezentuje záhlaví skupiny a jde ji přidat do kolekce pásek náležících instanci třídy XtraReport. Pomocí vlastnosti GroupHeaderBand.GroupField lze specifikovat, dle jakého pole ve zdroji dat se má report groupovat. Pokud ve zdroji dat máme pole obsahující název přístroje, můžeme specifikovat, že data budou rozdělena do skupin (group) dle názvu přístroje apod. Pro každou skupinu se vytiskne záhlaví skupiny. Kromě záhlaví reprezentovaného pomocí GroupHeaderBand můžeme přidat skupině i zápatí pomocí třídy GroupFooterBand reprezentující pásku zápatí skupiny [39]. Připomínám, že groupování nemění nic na tom, že pro každý řádek ve zdroji

dat bude vytištěna jedna DetailBand páska. Groupování pouze uspořádává jejich pořadí do koherentních skupin. Tento způsob seskupování dat je využit téměř ve všech mnou implementovaných reportech.

4.2.2 Dělení na kapitoly

Uživatel by měl mít možnost volit, jak se data budou groupovat, pokaždé jinak. Report by měl být rozdělen na samostatné kapitoly dle toho, jak chce uživatel data seskupit. To je možné implementovat programově tak, že budeme za běhu programu upravovat instanci třídy XtraReport. Vzhledem k tomu, že tento postup by musel být do jisté míry implementován pro každý report jinak, bylo pro mně toto řešení velmi neatraktivní. Namísto toho jsem zvolil následující postup.

Mnou implementovaná třída IntervalBlock obsahuje list přístrojů, kde je každý přístroj reprezentován pomocí jeho GUID a časový interval reprezentovaný třídou DateRange. Tato třída také obsahuje statickou metodu GetIntervalBlocks s parametrem nastavení reportu v podobě třídy ReportSetting, která vrací list instancí IntervalBlock, kde každý prvek listu představuje datový rozsah jedné kapitoly. List přístrojů a časový rozsah v IntervalBlock tedy definují časové období (např. konkrétní den/týden/atd.) a přístroje, pro které se mají načíst data při vytváření jedné kapitoly reportu. Následně můžeme pro každou kapitolu načíst data do DataSetu, pro každý DataSet vygenerovat samostatný report a nakonec pomocí metody MergeReports popsané v sekci 4.1 vygenerované reporty zřetězit do jednoho reportu. Dá se tedy říct, že jsme vygenerovali několik reportů (každý nad různými daty) a následně jsme je spojili do jednoho dokumentu. Groupování je možné nastavit ve třídě ReportSetting pomocí vlastnosti GroupBy, kde lze zvolit jednu z následujících možností:

- **negroupovat:** Data se negroupují. Pro všechna data se vygeneruje jedna kapitola reportu.
- **dle zařízení:** Data se rozdělí na několik částí, kde každý DataSet obsahuje všechna měření pro jedno zařízení.
- **dle časové periody:** Každý DataSet obsahuje všechna měření ze všech měřících zařízení pro jednu časovou periodu (týden/měsíc/...).
- **nejdřív dle zařízení, potom dle časové periody:** Data se nejdřív rozdělí do podtabulek dle zařízení, tyto tabulky se následně znovu rozdělí dle časové periody.
- **nejdřív dle časové periody, potom dle zařízení:** Téměř stejný proces jako předtím jen s rozdílem, že nejdříve dochází k dělení podle času a až potom dle přístroje. (Což se projeví v řazení jednotlivých kapitol reportu.)

Hlavní výhoda tohoto způsobu dělení reportu tkví v tom, že ho lze velmi snadno zadat uživatelem. Upravovat repx šablonu není třeba. Hlavní nevýhodou je, že generujeme několik různých podreportů, které následně řetězíme a tudíž jednotlivé

podreporty nemají přístup k datům v jiných podreportech. To omezuje některé možnosti práce s daty, například u generování jednotné mapy dokumentu.

4.2.3 Mapa dokumentu

Všechny ovládací prvky umístěné v instanci XtraReport třídy mají vlastnosti Bookmark a BookmarkParent [40], pomocí kterých jde v reportu definovat mapu dokumentu. Bookmark neboli záložku jde vytvořit přiřazením string objektu zmíněné vlastnosti BookMark. Každé záložce lze zároveň přiřadit nadřazenou záložku tak, že vlastnosti BookmarkParent přiřadíme ovládací prvek, jež slouží jako záložka. Záložku tedy může reprezentovat libovolný string, ať už se jedná o konstantu, pole ze zdroje dat či jejich kombinaci vytvořenou pomocí výrazů (viz 3.1.1). U reportů, které pracují s groupováním, budou dvě nejvyšší záložky reprezentovat časovou periodu a potom zařízení (či skupinu zařízení), pro které byla kapitola reportu vygenerována.

Jelikož ale výsledné reporty vytváříme řetězením jednotlivých podreportů reprezentujících kapitoly reportu, v některých situacích může nastat stav, že výsledný report obsahuje duplikátní záložky. Například pokud groupujeme report nejdřív podle časové periody, potom podle zařízení a generujeme report pro tři zařízení, výsledný report bude obsahovat tři stejné záložky (záložky, jejichž text je informace o časovém rozsahu dané kapitoly). Proto jsem ve třídě ReportBase vytvořil metodu CreateDocumentMapForMergedReport, která projde strukturu vytvořeného dokumentu (tvořeného z několika podreportů) a duplicitní záložky spojí do jedné.

4.3 Třída ReportSetting

Třída ReportSetting představuje vstupní požadavky předávané třídě ReportBase (resp. jejím potomkům), jež obsahují všechny informace o tom, který report se má vytvořit a jaké má být jeho nastavení. Následuje výčet vlastností obsažených ve třídě ReportSetting.

- **Repx soubor (string):** Jedná se o vlastnost pojmenovanou RepxFile, která reprezentuje cestu k šabloně reportu uložené na disku. Pomocí této vlastnosti se tedy specifikuje zvolený report.
- **Agregační intervaly (uint):** AggMainInterval, AggPQInterval a AggELMInterval udávají, s jakým agregačním intervalem (v milisekundách) se data budou načítat ze zdroje dat. Jak již bylo zmíněno v sekci 3.2.1, data jsou ve zdroji rozdělena do různých archivů a pro každý archiv lze definovat vlastní agregační interval. Pro každý archiv se při čtení dat používá trochu jiná agregace, nicméně proces agregace při načítání dat je implementován ve STEP projektu, tudíž jsem se o správný typ agregace nemusel zajímat.
- **Formáty datumu a čísel (string):** V reportu lze měnit formát čísel (počet desetinných míst) a času. Ovládací prvky v reportu mají vlastnost TextFormatString, pomocí kterých se tyto formáty nastavují. Proto jsou v ReportSetting nastavitelné vlastnosti TimeFormat, MainFormat a ELMFormat, jež

reprezentují zvolený formát pro všechny číselné a časové veličiny. Během generování reportu se tyto formáty nastaví u všech potřebných ovládacích prvků v XtraReportu (viz 4.1).

- **Jednotky (string):** MainUnit a ELMUnit udávají, v jakých hodnotách se mají zobrazovat veličiny (kilo, Mega, ...). Načtené hodnoty veličin se podle tohoto nastavení vydělí správným dělitelem a informace o dané jednotce se nastaví do zvolených ovládacích prvků XtraReportu a zobrazí v reportu.
- **Seznam veličin (List<Quantity>):** Jedná se o list instancí třídy Quantity, která reprezentuje veličinu ze zdroje dat. (Bližší popis této třídy se nachází v sekci 3.2.1.) Tato vlastnost specifikuje, které veličiny má report obsahovat. Některé reporty mají jejich obsah pevně daný (co se veličin týče), u některých si může uživatel veličiny v reportu zvolit.
- **Generování reportu vs otevření v editoru (bool):** Tato bool vlastnost udává, zda se report má otevírat v ReportDesignTool, či zda se má vygenerovat a zobrazit ReportPrintTool. Důvod, proč jsem zavedl tento údaj, je následující: pokud nechceme report generovat, není důvod načítat, agregovat a analyzovat data. Při upravování šablony reportu v ReportDesignToolu stačí, aby funkce generující daný report inicializovala instanci XtraReport pomocí repx šablony a vrátila ji programu, který ji volal. Tím se program vyhne zbytečné a časově náročné práci s daty.
- **Nastavení časového intervalu:** Celkový časový interval lze zvolit několika způsoby, nejjednodušším je zadat dvěma daty definovaný časový úsek. Vzhledem k faktu, že ho nelze použít jako součást trvalého nastavení, protože se potom definovaný časový úsek nemění, je tento způsob vhodný pro jednorázové použití. Další možností je pracovat se všemi naměřenými daty. V tom případě bude celkový časový interval odpovídat časovému úseku, který pokrývají data v DataSource. Nejčastější typ je zvolení počtu časových period, jež mají celkový časový úsek tvořit. Pokud si jako časovou periodu zvolíme týden, můžeme definovat, kolik týdnů bude celkový časový úsek obsahovat. V tomto případě je třeba nastavit, od jakého okamžiku se časové periody budou načítat (ode dneška/od včerejška/od posledního záznamu v datech/od poslední úplné časové periody v datech/...). Při dělení dat podle času si vybíráme časovou periodu. Většinou se jedná o dělení po hodinách, dnech, týdnech, měsících či rocích. Metoda GetTotalDateRange vrací celkový časový interval a enumerační vlastnost TimeIntervalE definuje, po jakých časových intervalech se má report groupovat do kapitol (po dnech/týdnech/měsících/nijak/...).
- **Groupování (DataGrouping):** Třída DataGrouping definuje, jak se má report groupovat (viz sekce 4.2). Také je zde obsažena enumerační vlastnost TimeInterval, která udává, po jakých časových úsecích se má groupování provádět. (Tato vlastnost se bere v potaz pouze v případě, že se má groupování provádět podle času.) ReportSetting obsahuje vlastnost Grouping používanou právě ke specifikaci těchto údajů.

- **Ostatní nastavení:** Ačkoliv některé údaje obsažené v nastavení (převážně popsané v předešlých sekcích) jsou relevantní pro všechny reporty, většina nastavení bude určena pro konkrétní reporty. Nejčastěji se jedná o nastavení různých mezních hodnot a limitů, zvolení, jaké typy veličin mají být v reportu obsaženy a které komponenty se mají v reportu zobrazovat. (Report může mít volitelné komponenty, jež lze vynechat, např. histogramy v reportu statistické analýzy.) Všechna tato nastavení budou uložena ve vlastnosti `SettingDetails`, která je typu `SettingReport` (třída ze STEP projektu určená pro práci s nastavením reportů). Specifické nastavení bude vždy popsáno v sekcích zabývajících se konkrétními reporty v kapitole 5.

Instance třídy `ReportSetting` tedy udává, s jakým reportem se pracuje, zda se má generovat či zda se bude otevírat v nástroji `ReportDesignTool`, a všechna nastavení potřebná k jeho vygenerování. Při vytváření reportu se ty části nastavení, které nejsou pro daný report relevantní, budou ignorovat. Některé části jeho nastavení se mohou ukládat jako šablona nastavení do `repCfg` souboru (viz sekce 4.4), ale cesta k `repx` souboru, soubor s limity a zda se má report vygenerovat či předat editoru, bude zadán samostatně. Aplikace `EReporter` tedy bude muset umožňovat uživateli zadat všechna potřebná nastavení `ReportSetting` třídy a předat je knihovně `ReportCore`, která dle nastavení splní požadavky uživatele.

4.4 Soubory s šablonami reportů a jejich nastavením

V knihovně `ReportCore` budou umístěny i samotné šablony reportů. Jedná se o xml soubory s příponou `.repx`. Jejich obsah reprezentuje stromovou strukturu reportu, kde jsou definovány jednotlivé komponenty reportu: pásky, ovládací prvky, jejich vlastnosti atd. [41] Příkladem může být report obsahující mimo jiné hlavičku reportu (`Report Header`) s tabulkou (`table`), v níž jsou jednotlivé buňky tabulky (`table cells`).

Důležitou součástí šablony jsou parametry reportu, které mohou sloužit k nastavení chování reportu, viditelnosti jednotlivých komponent, mohou reprezentovat proměnné atd. Skrytý parametr „`reportName`“, jenž definují ve všech šablonách bude sloužit k identifikaci reportu a následném volání metody, která má tento report generovat. Zjednodušená ilustrace struktury reportu je zobrazena na obrázku 4.1. Nastavení reportu (třída `ReportSetting`, viz 4.3) obsahuje vlastnost `RepxFile` reprezentující cestu k šabloně reportu. Kromě šablony reportů obsahuje `ReportCore` i další zdroje, např. soubor obsahující nastavení reportů (opět v xml formátu), jehož obsah se využije jako výchozí nastavení, pokud si uživatel nspecifikuje jiné nastavení. Zjednodušená ilustrace struktury souboru je zobrazena na obr. 4.2.

```

<?xml version="1.0"?>
<SettingReport xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  .
  .
  .
  <statAnalysis>
    <Current>true</Current>
    <Other>true</Other>
    <Power>true</Power>
    <Voltage>true</Voltage>
    <HarmEnabled>true</HarmEnabled>
    <HarmEven>true</HarmEven>
  </statAnalysis>
  <energy>
    <SumGraphs>true</SumGraphs>
    <SumTables>true</SumTables>
    <Graphs>true</Graphs>
  </energy>
  .
  .
  .
</SettingReport>

```

Obrázek 4.1: Struktura repx souboru s šablonou reportu

```

<?xml version="1.0"?>
<SettingReport xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  .
  .
  .
  <statAnalysis>
    <Current>true</Current>
    <Other>true</Other>
    <Power>true</Power>
    <Voltage>true</Voltage>
    <HarmEnabled>true</HarmEnabled>
    <HarmEven>true</HarmEven>
  </statAnalysis>
  <energy>
    <SumGraphs>true</SumGraphs>
    <SumTables>true</SumTables>
    <Graphs>true</Graphs>
  </energy>
  .
  .
  .
</SettingReport>

```

Obrázek 4.2: Struktura repCfg souboru s nastavením reportu

4.5 Metody generující reporty

Nyní, když jsem popsal všechny důležité třídy, jejich metody a vlastnosti, které jsou ke generování reportů třeba, popíši typický průběh generování reportu. Všechny vytvořené reporty, mají přiřazenou metodu, kterou se daný report generuje. Tyto metody se nacházejí v některém z potomků třídy ReportBase. Metody pro reporty energií se nacházejí ve třídě EnergyReport, report alarmů se nacházejí ve třídě AlarmsReport atd. Pokud se pro zvolený report nenajde odpovídající generující metoda, ReportCore se pokusí report vygenerovat pomocí výchozí metody CreateDefaultReport ze třídy GeneralReports. Ačkoliv se různé reporty generují různými metodami, tyto metody mají většinou podobný průběh, protože základní procesy typu inicializace třídy XtraReport a DataSetu, překlad, načtení dat apod. se dělají vždy a v podobném sledu. Generování reportu má tedy následující obecný průběh:

1. Inicializace některého z potomků třídy ReportBase pomocí instance třídy ReportSetting.
2. Zavolání třídy GetXtraReport. Tato metoda je obsažena v ReportBase a dědí ji všichni její potomci.
3. Zkontroluje se validita nastavení reportu.
4. Uvnitř metody GetXtraReport se načte jméno reportu uložené v jeho šabloně a podle něho se zavolá metoda, kterou se report generuje.
5. Jakmile se program nachází ve správné generující metodě, začíná samotný proces vytváření reportu. Jako první se inicializuje instance třídy XtraReport pomocí šablony definované v ReportSetting.RepxFile. Obsah této šablony se následně uloží instance třídy MemoryStream.
6. Pomocí statické třídy IntervalBlock.GetIntervalBlocks se načte list instancí IntervalBlock, kde je uložen datový rozsah pro jednotlivé kapitoly reportu.
7. Následně se pro každý IntervalBlock generuje jedna kapitola reportu.
8. Generování každé kapitoly reportu začíná inicializací DataSet. Vytvoří se DataSet, kolekce tabulek dat (DataTable) a do každé tabulky se přidají sloupce (DataColumn) s odpovídajícím jménem a typem sloupce.
9. Následuje načítání dat z DataSource. a jejich analýza. K tomu se využívají selektory a agregátory ze STEP projektu či další metody pro to určené.
10. Výsledky analýzy se ukládají do vytvořeného DataSetu.
11. Po dokončení analýzy dat se inicializuje nová instance XtraReport pomocí MemoryStreamu vytvořeného na začátku celé metody. XtraReport instanci se následně přiřadí všechny potřebné parametry a zavolá se metoda XtraReport.CreateDocument.

12. Tento postup proběhne pro všechny prvky v listu `IntervalBlock`.
13. Pro každou kapitolu reportu je tedy vytvořena jedna instance třídy `XtraReport`. Všechny instance se zřetězí do jednoho dokumentu (nové `XtraReport` instance) pomocí metody `ReportBase.MergeReports`.
14. Výslednému dokumentu se vygeneruje mapa dokumentu.
15. Generování je u konce. Výsledný dokument vrací daná generující třída metodě `GetXtraReport`, která ho také použije jako návratovou proměnnou.

4.6 Shrnutí

Knihovna `ReportCore` popsaná v této kapitole tvoří samostatný multiplatformní blok využitelný v různých aplikacích, jehož hlavním účelem je generování reportů dle zvoleného nastavení. Třída `ReportSetting` reprezentuje konfiguraci reportu a jsou v ní obsaženy veškeré informace o nastavení a účelu reportu, který se má generovat: jeho šablona, požadované groupování, agregační intervaly, formáty datumů a čísel, seznamy veličin, jež mají být v reportu obsaženy, zvolení časového intervalu reportu a další nastavení.

Abstraktní třída `ReportBase` s vlastností `Setting` typu `ReportBase` obsahuje metody schopné nastavení načíst a zpracovat. Dále zahrnuje metody používané různými typy reportů jako inicializace a překlad reportu do vhodného jazyka, nastavení formátu datumů a čísel v reportu, řetězení reportů do jediného dokumentu, různé práce se zdrojem dat typu `DataSource`, generování mapy dokumentu atd. Potomci `ReportBase`, třídy `EnergyReport`, `AlarmsReport`, `HarmonicsReport`, `GeneralReport` a `PowerQualityReport` obsahují funkce, které umožňují pomocí metod vlastních nebo zděděných z jejich předka generovat reporty nad zvolenými daty dle nastavení reportu definovaného ve vlastnosti `Setting`.

`Report` reprezentuje instance třídy `XtraReport` pocházející z knihovny `DevExpress.Reporting.Core`, jejíž obsah definuje strukturu reportu. Při generování reportu je tato instance inicializována a upravena tak, aby odpovídala definovanému nastavení. Následně dochází k načtení dat, jejich analýze uložení výsledků do instance třídy `DataSet`, která slouží jako zdroj dat pro `XtraReport`. K práci s daty jsem využíval knihovny a funkce z projektu `STEP`, který jsem dále upravoval a doplňoval o vlastní kód za účelem agregace a analýzy dat. Jednotlivé instance `XtraReport` lze zřetězit dle zvoleného groupování do jediného dokumentu s ucelenou strukturou a mapou dokumentu. Výsledný dokument lze použít jako vstup pro nástroje `ReportDesignTool` a `ReportPrintTool`, které umožňují upravování šablony reportu, zobrazení vygenerovaného reportu a export reportu do pdf a jiných formátů. Dalšímu popisu `ReportCore` knihovny, konkrétně obsahu jednotlivých vytvořených reportů se věnuje následující kapitola.

5 Realizované reporty

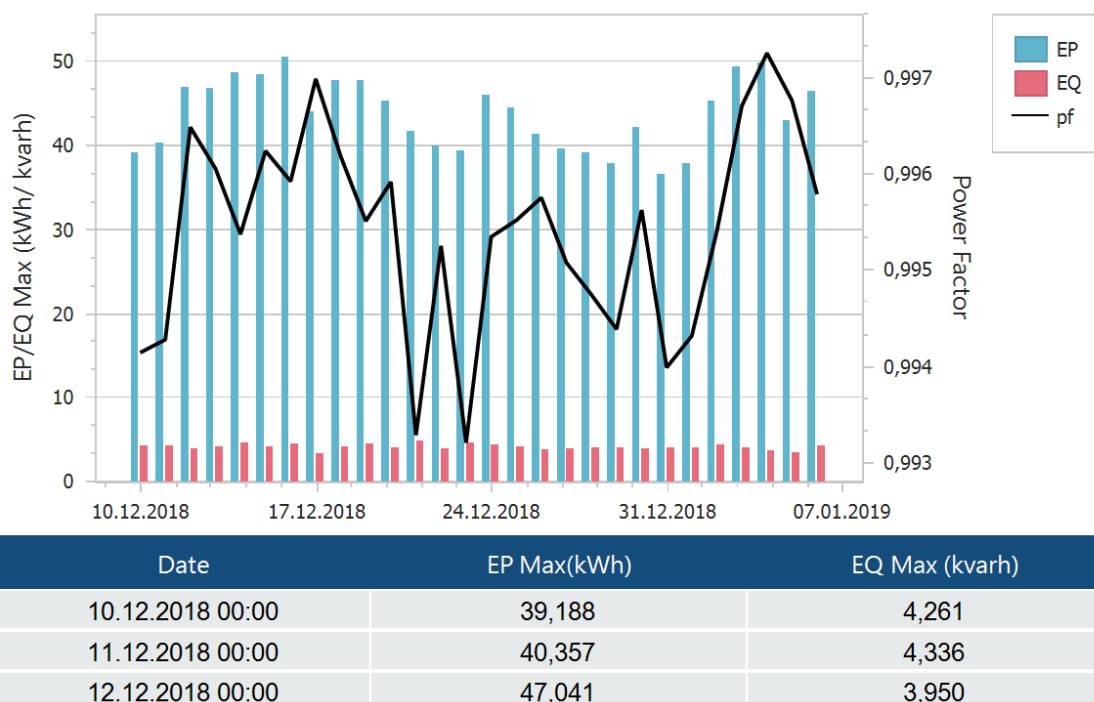
V minulé kapitole jsem popsal typický průběh generování reportů (načítání dat, vytváření XtraReportu, překlad reportu, ...). Zde se budu věnovat popisu obsahu a smyslu všech implementovaných reportů, které lze pomocí ReportCore knihovny vygenerovat. Některé grafy a tabulky jsou stejné nebo podobné ve více reportech a využívají stejné metody k analýze dat. Proto budu popisovat některé části reportu jako samostatné jednotky a uvedu, v jakých reportech jsou obsaženy.

5.1 Reporty spotřeby a produkce elektrické energie

Prvním typem vytvořených reportů, jsou reporty spotřeby a produkce elektrické energie. DataSource, jehož obsah se dělí na archivy, má všechny veličiny spotřeby a produkce uloženy v samostatném archivu s názvem „ELM“.

Jak jsem zjistil v mé rešerši o platformách pro management spotřeby a kvality elektrické energie, údaje o spotřebě a produkci e. energie jsou přítomné téměř na všech platformách. Většinou se zobrazuje suma, průměr, popř. minima a maxima spotřeby pro zvolené intervaly buď v tabulkách nebo grafech. Grafy mohou být sloupcové grafy se sumami, čárové grafy zobrazující průběh, Sankeyho diagramy zobrazující sumu spotřeby pro jednotlivá zařízení atd. Jelikož se jednalo o první reporty, které jsem vytvářel, rozhodl jsem se vytvořit větší množství jednoduchých reportů, jež budou mít podobný obsah a strukturu, nicméně budou poskytovat přístup k většině důležitých informací obsažených v reportech spotřeby na platformách, které jsem kvůli rešerši procházel. Tyto reporty nevyžadovaly rozšíření nastavení o vlastní nastavení a využívají nastavení určené pro většinu reportů jako groupování, agregační intervaly, vstupní veličiny apod. Po konzultaci s mým vedoucím jsem nakonec realizoval následující reporty:

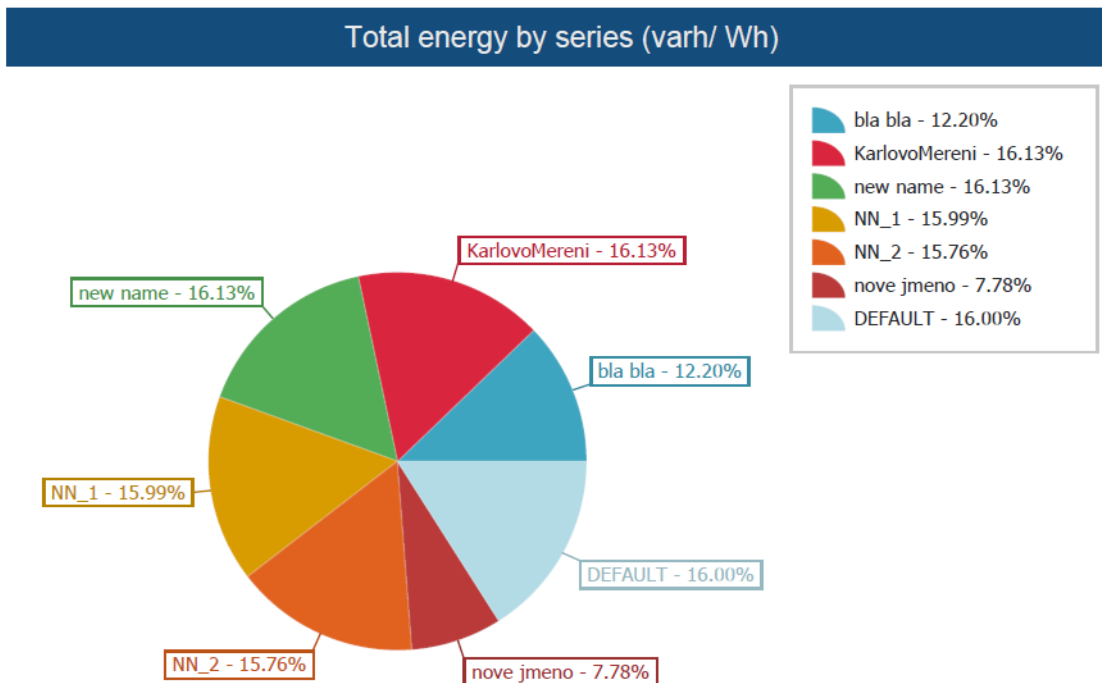
- 3Ep Report - Report obsahuje čárový graf zobrazující spotřebu energie za celkové období. Pod ním se nachází tabulka s maximální a minimální spotřebou po dnech.
- Consumption report - Obsahuje sloupcový graf s činnou a jalovou energií a čárový graf zobrazující účinník. Následuje tabulka spotřeby energie po dnech.
- Consumption vs production report - Report obsahuje sloupcový graf se spotřebou a produkcí e. energie po dnech a opět tabulku denní produkce a spotřeby.



Obrázek 5.1: Graf a část tabulky energy consumption reportu

- Summary report - Report s tabulkami zobrazujícími import a export reálné, induktivní a kapacitní energie rozdělené po tarifech. Podrobnější tabulky pro jednotlivé tarify jsou rozděleny i po fázích.
- Consumption summary report - Jedná se o koncepčně podobný report Summary reportu, který však obsahuje pouze informace o spotřebě. Tabulky opět zobrazují spotřebu reálné, induktivní a kapacitní energie rozdělené po tarifech a fázích.
- Energy by records report - Na rozdíl od předchozích reportů vyžaduje tento report jako vstupní informaci seznam veličin, které se mají analyzovat. Celková spotřeba se zobrazí v koláčovém grafu, jehož jednotlivé části reprezentují spotřebu zvolených měřících přístrojů v procentech z celkové spotřeby. (viz obr. 5.2) Následuje sloupcový graf, který po dnech zobrazuje spotřebu jednotlivých přístrojů. Nakonec je zde tabulka denní spotřeby pro jednotlivé přístroje.
- Energy and power overview - Tento report zobrazuje spotřebu a výkon elektrické energie ve sloupcovém grafu a tabulce.
- Energy const and consumption report - Na rozdíl od předchozích reportů se zde nevyužívá groupování standardním způsobem. Namísto spojování jednotlivých XtraReportů se zde dle zvoleného groupování nastavuje DevExpress ovládací prvek XRCrossTabReport, které umožňuje zobrazovat multi-dimenzionální data v tabulce. To znamená, že tabulka se rozšiřuje vertikálně

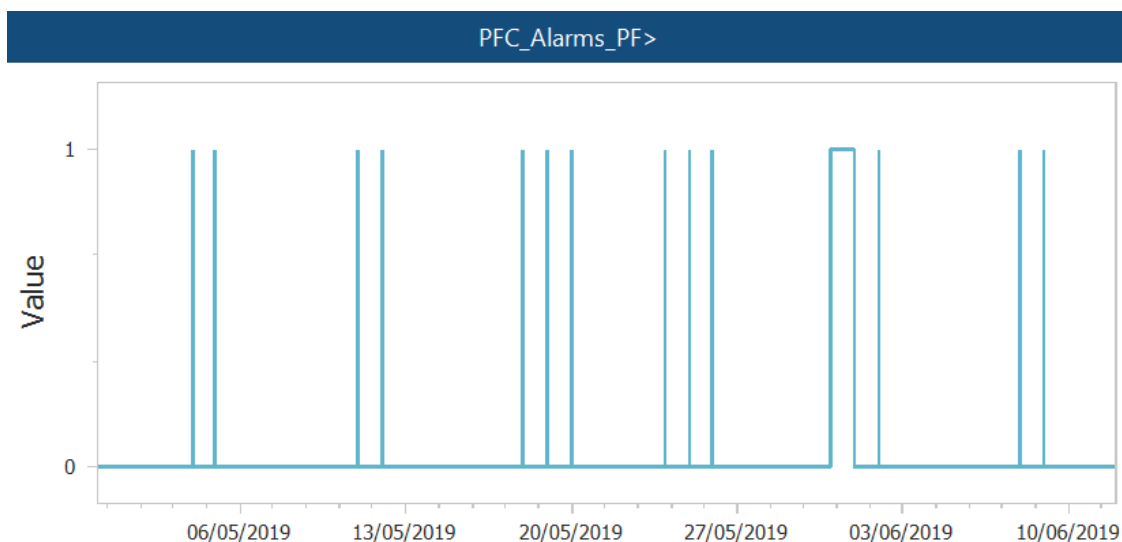
i horizontálně podle nastaveného groupování. Sloupce se skládají z jednotlivých časových intervalů, řádky pak zobrazují spotřebu a cenu spotřebované e. energie. Řádky jsou také zgroupovány po přístrojích. Na konci tabulky je zobrazena totální spotřeba a cena pro všechny přístroje.



Obrázek 5.2: Kruhový diagram spotřeby energie po přístrojích

5.2 Reporty alarmů

Alarmy jsou ve zdroji dat uloženy jako binární veličiny nabývající hodnot true nebo false (aktivován/neaktivován). Mohou signalizovat různé informace, např. zda veličina přesahuje určitou hodnotu, zda se hodnoty veličiny nachází v povoleném pásmu atd. Pro efektivní práci při generování reportů těchto veličin jsem vytvořil agregátor pojmenovaný NonRepeatingValuesExtractor, který při načítání veličiny ze zdroje dat ukládá pouze první a poslední zaznamenanou hodnotu a také hodnoty, jež mají jinou velikost než hodnota předešlá. Spolu s těmito hodnotami se ukládá i jejich příslušný čas. Díky tomu lze průběh těchto veličin zobrazit v grafu či tabulce bez ztráty jakékoliv informace přesto, že se graf generuje nad podstatně menším objemem dat. To vede k zefektivnění generování algoritmu. Podobně jako v případě reportů spotřeby a produkce elektrické energie, i zde jsem vytvořil jednoduché reporty nevyžadující žádné speciální nastavení. Jedinou neobvyklou specifikací je seznam veličin alarmů, jež mají být v reportu obsaženy. Pro tyto veličiny se následně zobrazuje tabulka dat, kde každý řádek reprezentuje změnu hodnoty: čas a informaci, zda došlo k aktivaci či deaktivaci alarmu. Veličiny se zobrazují také v grafu.



Obrázek 5.3: Grafy průběhu alarmu

Tam jsou na rozdíl od tabulky zobrazeny i jejich první a poslední hodnoty, aby graf pokrýval celý časový interval reportu. Výsledný report tedy pro zvolené veličiny informuje o alarmových stavech v jednotlivých měřeních. Nakonec jsem vytvořil dvě verze reportu alarmů, jeden s grafem a tabulkou, druhý pouze jako tabulku. Po konzultaci s vedoucím jsme dospěli k názoru, že další reporty alarmů není třeba vytvářet.

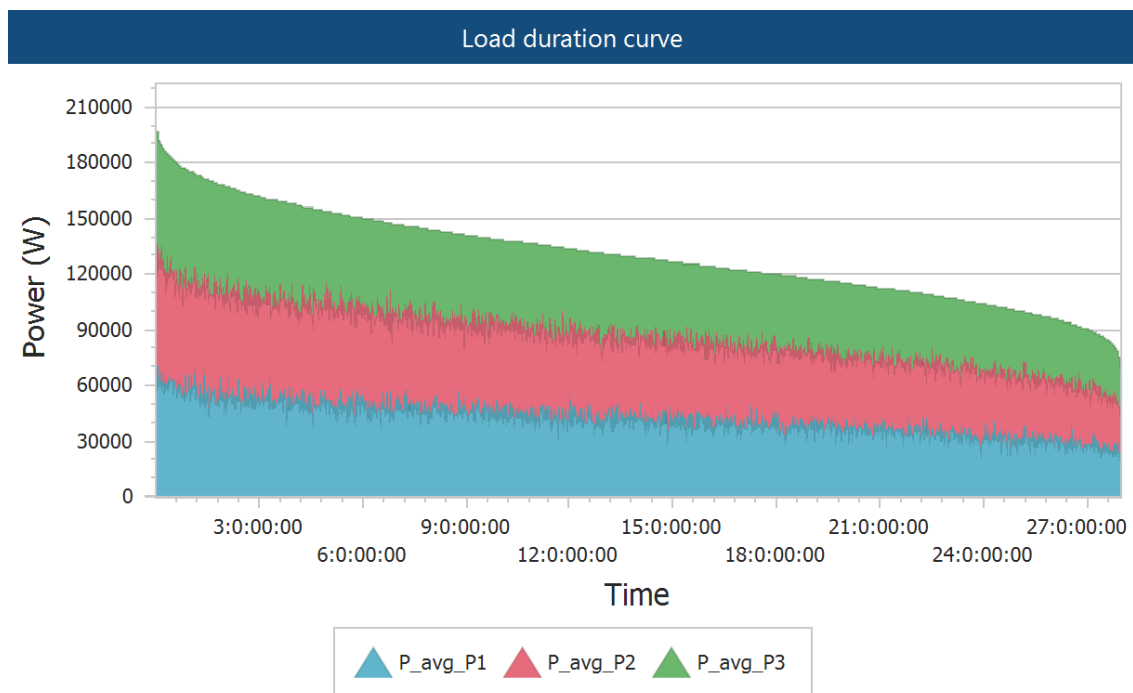
	Value	Time
<input checked="" type="checkbox"/>	True	04/05/2019 20:01:15
<input checked="" type="checkbox"/>	False	04/05/2019 20:01:30
<input checked="" type="checkbox"/>	True	04/05/2019 20:02:45
<input checked="" type="checkbox"/>	False	04/05/2019 20:03:00
<input checked="" type="checkbox"/>	True	04/05/2019 21:58:30
<input checked="" type="checkbox"/>	False	04/05/2019 21:58:45

Obrázek 5.4: Tabulka změn hodnot jednoho alarmu

5.3 Reporty výkonu

Prvním realizovaným reportem výkonu je tzv. Load duration curve report, jehož hlavní prvkem je křivka trvání zátěže, tedy křivka zátěže, jejíž hodnoty jsou sestupně seřazeny [42]. Křivku zátěže lze sestavit z jednoho nebo více měření. V případě jediného měření stačí naměřené hodnoty seřadit od nejvyšší po nejnižší hodnotu. Pokud se křivka má skládat z několika měření, hodnoty se musí uspořádat tak, aby byl sestupně seřazen součet měření, nikoliv jednotlivá měření. Zatěžovací křivky vybraných měření se v každém časovém bodě sečtou, tím vznikne jedna zatěžovací

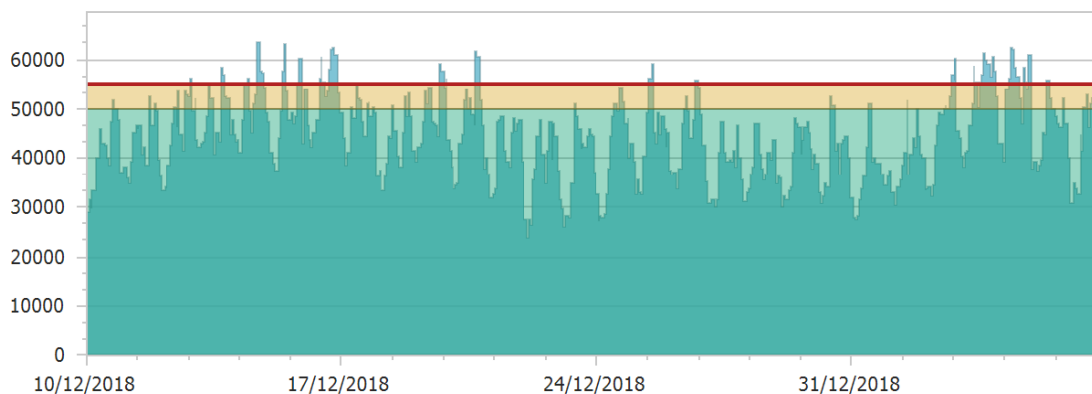
super-křivka a až ta se řadí od nejvyšší po nejnižší hodnotu. Jednotlivá měření tvořící super-křivku se uspořádají stejně. Správně sečtená a vytvořená křivka trvání zátěže tvoří jeden celek, nicméně jednotlivá měření tvořící křivku jsou barevně odlišena, aby mohl uživatel měření rozlišit a analyzovat. Pro jasnější popis přidávám ilustraci křivky trvání zátěže tvořené ze 3 měření (obr. 5.6). Realizovaná funkce je součástí reportu, který kromě samotné křivky trvání bude obsahovat i graf s klasickými zatěžovacími křivkami. Reportu je třeba specifikovat, jaké veličiny v něm mají být obsaženy.



Obrázek 5.5: Křivka trvání zátěže z Load duration curve reportu

Druhým implementovaným reportem výkonu je Demand report. V nastavení se specifikuje, které veličiny se budou analyzovat, v jakých jednotkách se mají zobrazovat a jaké jsou hodnoty jejich limitů. Každá veličina má přiřazený OK a WARNING limit, ty dělí hodnoty veličiny do tří pásem: OK, WARNING, ERROR. V reportu je následně pro každou veličinu zobrazena tabulka, v níž je u každého pásma specifikována maximální hodnota dané veličiny v daném pásmu. Počet naměřených hodnot, které se v pásmu nacházejí, celková doba, kdy se veličina v daném pásmu nacházela a suma energie (celková suma výkonu v daném pásmu). Hodnoty veličin výkonu se načítají pro zvolený agregační interval jako aritmetický průměr. Pod tabulkou se nachází také graf s průběhem veličiny a zvolenými limity. Díky tomu uživatel snadno vidí, které části průběhu přesahují či nepřesahují zvolené limity.

P_avg_P1(W)					
Type	Limit	Count	Duration	Worst	Energy [Wh]
Critical	∞	524	5d 11h	74221	31349566
Warning	± 55000	550	5d 17h	54964	28798922
OK	± 50000	2957	30d 19h	50000	117591020



Obrázek 5.6: Tabulka a graf pro jednu veličinu v Demand reportu

5.4 Reporty harmonických

Pro popis harmonických jsem vytvořil dva velmi podobné reporty. Uživatel si volí, které konkrétní harmonické budou v reportu obsaženy. Také musí zvolit soubor s uloženými limity harmonických. Soubor by měl být formátu csv. V prvním sloupci by mělo být číslo harmonické a ve druhém sloupci na odpovídajícím řádku hodnota limitu pro danou harmonickou. Jelikož se v reportu mohou objevovat meziharmonické a supraharmónické, soubor může obsahovat tři zmíněné dvousloupcové tabulky, každou pro různý typ harmonických. V první části reportu se vždy zobrazí tabulka, ve které je pro každou harmonickou vypsáný řádek s limitem, maximální i minimální dosaženou hodnotou harmonické a maximální hodnotou vyčíslenou procentuálně vzhledem k limitu. Pokud limit nebyl překročen, bude řádek označen jako OK a modře obarven. V případě, že hodnota harmonické limit překročí, označí se řádek jako NOT OK a obarví se červeně. Každá fáze má svojí tabulku harmonických a na konci tabulky bude vždy graf shrnující tabulku: Maximální hodnota bude vynesena sloupcově, limity budou vyneseny jako částečně průhledné, větší sloupce umístěné za maximy. Druhý report je velmi podobný, ale pro všechny fáze se zobrazuje společný graf.

Třetí report jsem nazval IEEE 519 report a jak napovídá jeho název, smyslem tohoto reportu bude kontrolovat, zda veličiny v naměřených datech splňují požadavky definované normou IEEE 519-2022. Tato norma se věnuje napěťovým vlněním a harmonickým zkreslením v elektrických systémech. Jejím hlavním cílem je minimalizovat vznik nežádoucích elektrických jevů, které mohou ovlivňovat chod elektrických zařízení v daném systému. Norma definuje maximální hodnoty harmonických zkreslení pro různé typy elektrických zařízení a určuje, jak tyto hodnoty

měřit a vyhodnocovat. Jsou zde definovány horní limity pro denní a týdenní hodnoty kvadratického průměru a časové intervaly, nad kterými se kvadratické průměry počítají. Výsledný report obsahuje několik tabulek, které informují čtenáře reportu o hodnotách jednotlivých harmonických, celkovém harmonickém zkreslení (THD) a zda nastaly nějaké anomálie (byla či nebyla pravidla definovaná v normě splněna, popř. které veličiny ji překročily a jak).

5.5 Reporty kvality elektrické energie

Reporty kvality elektrické energie (PQ reporty) zpravidla zobrazují, zda se hodnoty veličin jako napětí, frekvence, THD atd. pohybují v stanovených mezích, jaké množství hodnot se nacházelo mimo stanovené meze, zda nedošlo k alarmovým stavům a obecně informují uživatele o sledovaných měřeních.

Stanovené meze a vlastnosti, které mají veličiny dodržovat, jsou standardizovány různými normami. Jednou z důležitých norem, která se hodnotou elektrické energie zabývá, je norma ČSN EN 50160. Ta stanovuje technické požadavky a kvalitativní parametry pro napětí v elektrických distribučních sítích. Obsahuje specifikace pro napětí a jeho fluktuace, harmonické zkreslení a nesymetričnost napětí. Smyslem této normy je zajistit, aby napětí v distribučních sítích a odpovídalo předepsaným standardům a bylo stabilní. Norma definuje limity pro různé parametry napětí, aby se minimalizovaly rušivé vlivy na elektrická zařízení a aby se zaručila bezpečnost provozu elektrických sítí. Hlavním cílem normy ČSN EN 50160 je zabezpečit spolehlivé fungování elektrických sítí a minimalizovat riziko poruch a výpadků elektrického napájení [43]. Dalšími sledovanými normami jsou ČSN EN 61000-2-2 [44], ČSN EN 61000-2-4 [45] a ČSN EN 61000-2-12 [46]. Tyto normy se zabývají rušením šířeným ve vedení a stanovují kompatibilní úroveň těchto rušení. Jádrem většiny PQ reportů je hlavní tabulka výsledků. Jednotlivé řádky hlavní tabulky vystihují výsledky analýzy jedné veličiny. Pro každou veličinu je stanoven horní, popř. i dolní limit, který by neměl být překročen. Pro uživatele je důležité vědět, zda byl překročen a v jakém množství, tedy kolik procent naměřených hodnot se nachází mimo limity. Veličina může mít také nastavenou toleranci- určité procento hodnot se tedy může vyskytovat mimo limity. U každé veličiny je tedy třeba rozhodnout, zda zadané podmínky splňuje nebo ne. Seznam realizovaných reportů je následující:

- Compliance report a EN 50160 report - Smyslem těchto reportů je kontrolovat, zda naměřené hodnoty elektrické energie splňují požadavky na kvalitu elektrické energie dodávané z veřejných distribučních sítí, které jsou nařízené normou ČSN EN 50160. Ta stanovuje technické požadavky a kvalitativní parametry pro napětí v elektrických distribučních sítích. Obsahuje specifikace pro napětí a jeho fluktuace, harmonické zkreslení a nesymetričnost napětí. Smyslem této normy je zajistit, aby napětí v distribučních sítích odpovídalo předepsaným standardům a bylo stabilní. Norma definuje limity pro různé parametry napětí, aby se minimalizovaly rušivé vlivy na elektrická zařízení a aby se zaručila bezpečnost provozu elektrických sítí. Hlavním cílem normy ČSN

EN 50160 je zabezpečit spolehlivé fungování elektrických sítí a minimalizovat riziko poruch a výpadků elektrického napájení.

V Compliance reportu je každý řádek rozdělen na sloupce pro jednotlivé fáze a pro všechny veličiny se zobrazuje vypočítaný horní a dolní percentil veličiny zadaný normou. Také se zobrazuje, zda veličina překročila stanovené limity.

Obsah hlavní tabulky EN 50160 reportu je podobný hlavní tabulce Compliance reportu, v případě EN 50160 reportu se však namísto percentilů zobrazuje pro každou veličinu počet naměřených hodnot v daném intervalu. Uživatel si může zvolit, že se v reportu budou zobrazovat pouze kapitoly, ve kterých není splněn některý předepsaný parametr. Opět lze zvolit, zda bude v reportu DISDIP tabulka. Dalšími volitelnými prvky jsou grafy průběhu veličin (viz obr. 5.8), histogramy (viz obr. 5.9) a kvantilové grafy (viz obr. 5.10) pro každý typ veličiny.

Uživatel si pro oba reporty může přednastavené limity upravit a vytvořit si svoji interní normy tak, že si zadá v nastavení vlastní limity pro jednotlivé veličiny. Dále je možné nastavit, zda se mají zobrazit tabulky harmonických, popř. zobrazit pouze ty harmonické, které nesplnily zadané parametry (tedy přesahují limity ve větším množství, než dovoluje stanovená tolerance). Dále si lze zvolit, zda bude report obsahovat DISDIP tabulku (viz obr. 5.7), zda má být DISDIP zobrazena pouze, když některé veličiny nesplnily předepsané parametry nebo zda nemá být v reportu vůbec přítomna.

- EMC-LF report - Report kontrolující, zda naměřené hodnoty elektrické energie splňují požadavky elektromagnetickou kompatibilitu, která je předepsaná zmíněnými normami ČSN EN 61000-2-2, ČSN EN 61000-2-4 a ČSN EN 61000-2-12. Pro jednotlivé typy veličin se opět vyhodnocuje, zda splňují horní a dolní limit a pro jaké množství naměřených hodnot se vyhodnocuje výsledek. Report může obsahovat grafy průběhu veličin, histogramy a kvantilové grafy. Volitelná je i možnost vyfiltrovat kapitoly, ve kterých nedošlo k překročení žádného limitu a zda má report obsahovat shrnutí a tabulku harmonických.
- Statistics report - Obsah tohoto reportu je následující. Pro každou veličinu se hledá minimální a maximální hodnota spolu s časem, kdy tyto hodnoty nastaly. Dále se počítá průměrná hodnota. Tyto hodnoty se následně prezentují v hlavní tabulce, která je opět vertikálně rozdělena po fázích. Lze zvolit, které veličiny se mají v reportu vyskytovat (napětí, proud, výkon, harmonické, meziharmonické, supraharmónické, ...). Volitelnou součástí jsou opět grafy průběhu veličin, histogramy a kvantilové grafy.
- PQ-Events report - Jedná se o report obsahující pouze tabulky, v nichž shrnuje výskyt významných událostí jako podpětí a přepětí. U těchto událostí nás zajímá, jak moc se hodnota veličiny odchýlila od nominální hodnoty, kdy a na jaké fázi událost nastala, jak dlouho tato událost trvala a kolik podobných událostí se v daném období vyskytlo. Uživatel si může nastavit, zda bude report obsahovat třífázové měření, jednofázové měření či obojí. Dále si může

zvolit, jaké typy událostí budou zpracovány: pokles napětí, přepětí, přerušení napětí, chyba napájení. Také může zvolit délku časového intervalu, který musí událost splnit, aby se započítala. Výsledný report obsahuje dle zvoleného nastavení tabulky zobrazující počet událostí rozdělených dle typu, časového intervalu a velikosti vůči nominálnímu napětí. Při hodnotě větší než nula se políčko v tabulce obarví červeně.

V případě EN 50160 reportu a EMC-LF reportu je na začátku dokumentu shrnutí všech kapitol reportu. Celý report se na kapitoly dělí dle zvoleného groupování (viz sekce 4.2). Zde je pro každou kapitolu pouze uvedeno, zda měření splňují stanovené podmínky či nikoliv. Podrobnější stav je potom uveden v hlavních tabulkách.

Jedním z typů veličin, kterými se tyto reporty zabývají, jsou harmonické. Ty se zobrazují ve speciální tabulce rozdělené na tři části: sudé harmonické, liché harmonické dělitelné třemi a sudé harmonické nedělitelné třemi. V tabulce se vždy nachází číslo harmonické, její limit (zadaný normou, popř. vstupem od uživatele) a opět počet hodnot nesplňujících limit (vyjádřených procentuálně).

5.6 Samostatné části reportu

Některé části reportů se opakují v různých reportech. Proto jsem pro jejich analýzu a generování jejich DataSetu vytvořil samostatné metody v třídě ReportBase, tudíž je mohou využívat všechny metody generující libovolné reporty.

DISDIP tabulka

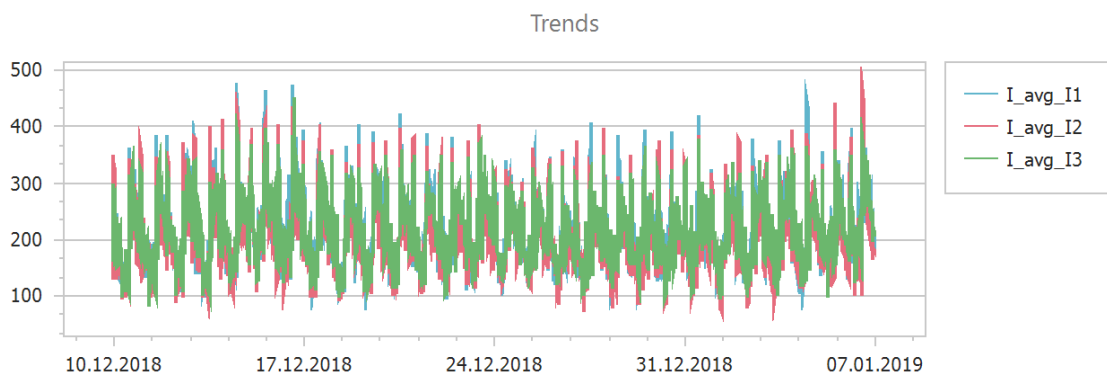
Jedná se o tabulku, ve které jsou vyneseny počty událostí, kdy se hodnota napětí odchýlila od nominální hodnoty o více než 10 %. Jednotlivé řádky jsou úrovně odchylky (80-90 %, 70-80 %, ... od nominální hodnoty) a jednotlivé sloupce jsou úrovně trvání události (100-200 ms, 200-500 ms, ...). Pokud se v dané buňce tabulky nachází hodnota větší než nula, bude obarvena červeně. V tabulce se tedy dozvíme, jaké anomálie se v měření napětí nacházejí. Uživatel si může zvolit, jaký řetězec znaků se bude v tabulce zobrazovat, pokud bude v dané kategorii nula událostí. Tato tabulka je volitelnou součástí Compliance reportu a EN-50160 reportu.

Průběhy veličin

Jedním z nejčastěji používaných grafů v reportech je jednoduchý graf průběhu veličin, jehož osa x reprezentuje čas a osa y hodnotu dané veličiny. Proto jsem implementoval metodu pro načtení dat a vytvoření DataSetu, který obsahuje názvy veličin, jejich hodnoty s časy a název měřícího přístroje. Pomocí těchto dat lze v libovolném reportu zobrazit průběh veličin v ovládacím prvku XRChart.

DISDIP table:									
	10ms <= t < 100ms	100ms <= t < 200ms	200ms <= t < 500ms	500ms <= t < 1s	1s <= t < 3	3s <= t < 20s	20s <= t < 1min	1min <= t < 3min	t < 3min
140<U<=160	--	--	--	--	--	--	--	--	--
120<U<=140	--	--	--	--	--	--	--	--	--
115<U<=120	--	--	--	--	--	--	--	--	--
110<U<=115	--	--	--	--	--	--	--	--	--
085<=U<090	2	--	--	--	--	--	--	--	--
070<=U<085	--	--	--	--	--	--	--	--	--
040<=U<070	--	--	--	--	--	--	--	--	--
005<=U<040	--	--	--	--	--	--	--	--	--

Obrázek 5.7: DISDIP tabulka



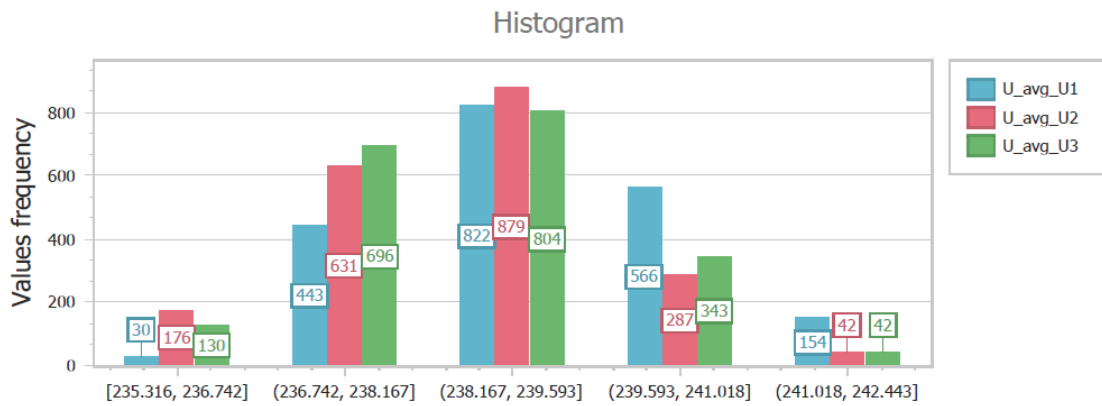
Obrázek 5.8: Graf průběhu veličin

Histogram

Pojem histogram označuje grafickou reprezentaci dat ve sloupcovém grafu, který informuje o rozložení hodnot v dané množině dat. Na ose x se zobrazuje množina hodnot, na ose y zase frekvence, s jakou se tyto hodnoty v množině dat vyskytují. Histogram tedy ukazuje, kolik hodnot spadá do zvolených intervalů a tudíž indikuje rozložení hodnot v množině [47]. DevExpress grafy mají mód histogram, který z průběhu veličin vygeneruje histogram skládající se z nastaveného množství sloupců. Díky tomu nemusí můj program histogram kalkulovat. Stačí reportu předat načtené průběhy dat a nastavit ovládací prvek XRChart do režimu histogram.

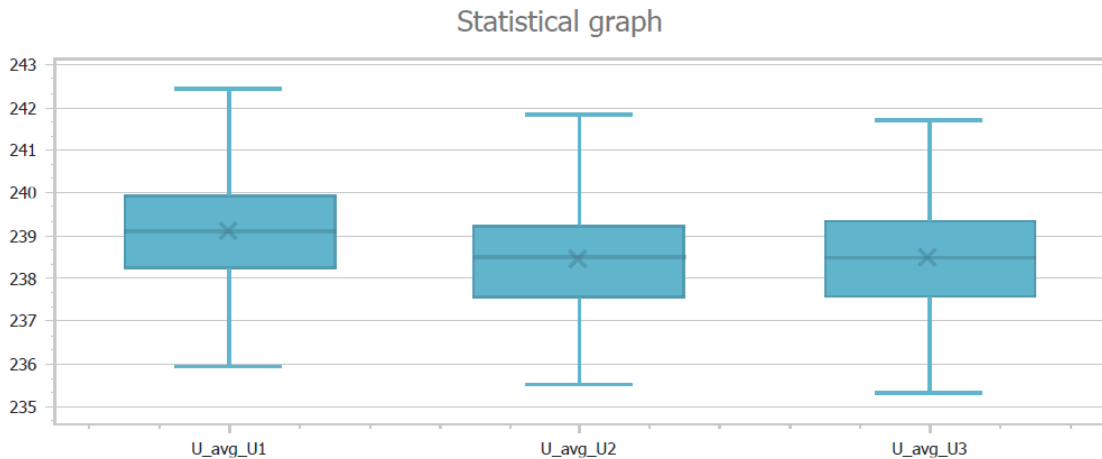
Kvartilový graf

Kvartilový graf, někdy označovaný jako BoxPlot či krabicový graf, graficky znázorňuje rozdělení dat do kvartilů, zobrazuje průměr, medián a odlehlé hodnoty. Z krabic mohou vést vertikální čáry nazývané vousy. Tyto čáry označují proměnlivost mimo horní a dolní kvartily a jakýkoliv bod mimo tyto čáry, neboli vousy, je považovaný za odlehlou hodnotu [48]. Výpočet jednotlivých částí kvartilového grafu a uložení



Obrázek 5.9: Histogram

do DataSetu ve vhodné formě jsem implementoval v mém kódu. XRchart je možné nastavit do módu boxplot, pomocí kterého lze v tomto ovládacím prvku kvartilový graf zobrazit.



Obrázek 5.10: Kvartilový graf

Grafy průběhu, histogramy a kvartilové grafy jsou volitelnou částí Compliance reportu, EN 50160 reportu a Statistics reportu. Pokud si uživatel zvolí, že daný typ grafu má být součástí reportu, proloží se jimi hlavní tabulka. Ve výsledku report vypadá tak, že v každé kapitole je pro každý typ veličiny (např. napětí) krátká tabulka, po které následují grafy této veličiny. Každý typ veličiny je v odděleném grafu, nicméně průběhy veličin pro všechny fáze se zobrazí společně. Report se tedy dělí po typu veličin.

5.7 Shrnutí

V této kapitole jsem popsal obsah všech reportů, které lze pomocí ReportCore knihovny generovat. Implementované reporty informují uživatele o široké škále

důležitých údajů. Reporty elektrické energie zobrazují sumy, průměry, minima a maxima pro aktivní a reaktivní produkci či spotřebu e. energie. Reporty alarmů jsou zaměřeny na binární veličiny alarmů. Popisují, zda podmínka daného alarmu (překročení stanovených limitních hodnot) byla v daném časovém okamžiku porušena či ne. Reporty výkonu informují o vlastnostech zátěže: Load duration curve report zobrazuje křivku trvání zátěže, která mimo jiné přehledně ukazuje dobu trvání pro jednotlivé hodnoty zatížení. Demand report zase dělí hodnoty zátěže do uživatelem definovaných pásem (OK, WARNING, ERROR pásmo) a sděluje, jakou dobu se daná veličina v jakém pásmu pohybovala, kolik hodnot se v pásmu nacházelo, maximální hodnotu v daném pásmu a celkovou energii. Reporty harmonických umožňují vybrat harmonické složky napětí a proudu, které mají být v reportu a zvolit jejich limity. Tabulky a grafy reportů potom obsahují informace o tom, zda byly limity překročeny. Report IEEE 519 zase kontroluje, zda se v měřeních harmonických složek napětí a THD vyskytovaly anomálie definované normou IEEE 519-2022. Reporty kvality elektrické energie opět kontrolují, zda naměřená data splňují různé české i mezinárodní normy, kontrolují, zda veličiny splňují uživatelem nastavená pravidla či nepřekračují nastavené limity a informují uživatele o vlastnostech veličin. Tabulky obsahují důležité informace jako hodnoty maxim, minim, průměrů, zvolených percentilů atd. Dále se zde mohou nacházet grafy s průběhy veličin, histogramy, kvartilové grafy a tabulky s počtem nalezených přepětí a podpětí.

Obsah reportu, jeho struktura, viditelnost jednotlivých částí reportu, parametry načítání dat, formát čísel a času a další konfigurace lze měnit pomocí nastavení reportu, dle kterého reporty dynamicky mění své vlastnosti.

Výsledná množina reportů se skládá z reportů spotřeby a produkce elektrické energie, reportů alarmů, reportů výkonu, reportů harmonických a nejobsáhlejší skupiny - reporty kvality elektrické energie. K jejich vytvoření a testování jsem použil moji aplikaci EReporter popsanou v následující kapitole, jež umožňuje uživateli všechny popsané reporty generovat.

6 Implementovaná aplikace EReporter

6.1 Popis

Pro generování reportů pomocí ReportCore knihovny jsem implementoval jednoduchou WinForms aplikaci EReporter, která slouží jednak jako samostatně spustitelná aplikace, ve které může uživatel zadat všechny potřebné vstupy, ale také jako externí nástroj spustitelný jiným programem, jemuž se všechny potřebné informace a pokyny zadávají jako vstupní argumenty aplikace. WinForms je velmi rozšířená technologie určená pro snadný vývoj desktopových aplikací. WinForms umožňuje vývojářům vytvářet uživatelská rozhraní pomocí různých ovládacích prvků, jako jsou tlačítka, textová pole, seznamy a další [49].

K vytvoření aplikace jsem využil DevExpress knihovny DevExpress.Win.Design a DevExpress.Win.Reporting, které mimo jiné poskytují alternativu ke standardním Winforms nástrojům (label, button, list atd.). Hlavním důvodem byla možnost využití skinů, jež DevExpress nabízí [50]. Aplikaci lze měnit aktivní skin, který využívají všechny DevExpress prvky v aplikaci. Díky tomu má EReporter ucelený a konzistentní vzhled, jenž lze snadno měnit uživatelem. Jelikož je EReporter využíván jako doplněk další aplikace, která DevExpress skiny také využívá, bylo možné použít skin sjednotit. Aktivní skin ve zmíněné aplikaci je uložen v databázi. Pokud se EReporter připojí k databázi, ve které je uložena informace o aktivním skinu, tento skin se aktivuje i v EReportu. Tudíž lze vzhled obou aplikací sjednotit a z velké části odpadá nutnost zabývat se designem, výběrem barev a nastavením fontů v mé aplikaci. Informace o aktivovaném skinu se ukládá do registru. Ve chvíli, kdy informace o skinu v databázi není, nebo se s databází vůbec nepromluže, nastavuje se aktivní skin právě podle informací v registru.

Jedna z knihoven projektu STEP, ke které jsem měl přístup, je knihovna Resources obsahující překlady často využívaných slov a frází v podobných projektech. Díky tomu jsem mohl implementovat překlad EReporteru do angličtiny, češtiny, němčiny a některých dalších jazyků. Po spuštění se všechny texty v aplikaci určené k přeložení načtou a přeloží do aktivního jazyka Windows pomocí třídy ResourceManager.

Nejdřív popíši, jak funguje EReporter v módu samostatně spustitelného programu. Naměřená data mohou být uložena buď v souboru nebo v databázi. Pro přístup k databázi je třeba autorizace ze strany uživatele (jméno, heslo), pro práci s souborem stačí specifikovat jeho lokaci na disku. Uživatelské nastavení se může opět nacházet v databázi (stejně jako data) nebo v souboru.

Po specifikaci zdroje dat se uživateli zobrazí stromová struktura dat v DevExpress prvku Tree list. Uživatel si může zvolit „uzly“, ze kterých chce generovat report. Uzel může být buď přístroj, nebo skupina přístrojů. Každý přístroj či skupina může obsahovat další podskupiny a přístroje. Při zvolení uzlu se předpokládá, že uživatel chce generovat report i pro všechny potomky tohoto uzlu.

Dále lze zvolit samotný report. V instalačním adresáři EReporteru se nachází složka se všemi dostupnými repx šablonami reportů, které je možné vybrat ke generování v rozbalovacím listu. Alternativně jde i procházet disk a zvolit libovolný repx soubor ručně.

Kromě reportu je třeba zvolit i jeho nastavení. Stejně jako report, i nastavení reportu je uloženo v xml souboru. (V tomto případě je jeho příponou .repCfg.) EReporter obsahuje repCfg soubor s výchozím nastavením pro všechny reporty, který si uživatel může otevřít a nastavit v DevExpress Property Grid prvku. I zde je ale uživateli umožněno najít si soubor s nastavením na disku. V případě, že chceme vygenerovat report, který vyžaduje další dynamická nastavení, např. zvolení měření, která mají být v reportu obsažena, otevře se další Form, v němž lze zvolit měření dle libosti.

Nakonec má uživatel možnost zvolit, jakou akci chce provést: buď editovat report v ReportDesignTool nebo ho otevřít v ReportPrintTool. ReportDesignTool nevyžaduje zadání šablony- pokud není žádná šablona zvolena, otevře se v ReportEditoru prázdný report nad zvolenými daty. STEP knihovna také obsahuje VerifyPluginLicense třídu, pomocí které lze podmiňovat používání ReportDesignToolu vlastnictvím licence. V případě ReportPrintTool je třeba report zvolit.

Po zvolení požadované akce dochází ke generování reportu pomocí knihovny ReportCore, jejíž třídám/metodám se předávají informace z uživatelem zadaného nastavení. Generování reportu se provádí v paralelním vlákně, aby se dalo s EReporterem pracovat i během tohoto procesu.

Z předešlého popisu je zřejmé, že pro vygenerování reportu musí uživatel zadat velké množství vstupních parametrů. To může být při častém používání aplikace poněkud frustrující a časově náročné. Proto jsem se rozhodl většinu vstupních informací ukládat, aby je bylo možné při dalším spuštění načíst a vstup předvyplnit. Tyto informace lze ukládat do databáze či souboru, ale oba tyto způsoby nemusí vyhovovat všem uživatelům. Proto jsem se nakonec rozhodl ukládat vstup do registrů. Server, jméno databáze, uživatel, heslo, nastavení i zvolený report se po ukončení aplikace zapíše do zvolených registrů a při spuštění aplikace se opět načtou. Díky tomu si EReporter pamatuje poslední zvolené vstupní parametry, což může uživateli práci s aplikací značně ulehčit.

Vzhledem k tomu, že v době, kdy jsem tuto práci psal, ještě nebyla třída FileDataSource (reprezentující soubor jako zdroj dat) v STEP knihovně plně implementovaná, je zatím uživateli umožněno zvolit pouze databázi (DBDataSource) jako zdroj vstupních dat. Až bude FileDataSource hotová, půjde ji snadno používat také, protože FileDataSource i DBDataSource dědí z abstraktní třídy DataSource a ke čtení dat budou používat stejné metody.

Jak jsem již zmínil, EReporter se také dá používat jako externí nástroj, kterému

se při spuštění předají všechny informace a instrukce jako vstupní parametry aplikace. Všechny vstupy se nejdříve zvalidují. Ve chvíli, kdy pracujeme s databází, se specifikuje název serveru, název databáze, uživatel a uživatelské heslo. S ceo souborem jako zdrojem dat stačí zadat jeho umístění na disku. Pro skupiny a přístroje, které mají být v reportu obsaženy, se předává jejich GUID. Po připojení k databázi se prochází stromová struktura tvořená přístroji a skupinami přístrojů a pro všechny zvolené uzly se hledají i jejich případné poduzly. Xml soubor s nastavením reportu může být opět buď na disku (specifikujeme cestu k němu) nebo v databázi (uvádíme jeho GUID). Lze specifikovat i způsob výstupu: buď může být vygenerovaný report zobrazen v ReportPrintTool a dále může být exportován do pdf a dalších formátů, nebo ho lze otevřít a upravovat v ReportDesignTool.

Pokud jsou všechny potřebné vstupní informace validně zadány, zobrazí se Wait Form, DevExpress prvek indikující, že probíhá generování reportu. Report se generuje na pozadí v jiném vlákne aby WaitForm zůstal responzivní a uživatel mohl generování zrušit. Pokud nedojde ke zrušení, otevře se výsledný report buď v ReportPrintTool nebo v ReportDesignTool. Po zavření kteréhokoliv z obou nástrojů se EReporter ukončí. Pokud je některá část vstupu neplatná (špatné uživatelské údaje, ...) nebo při generování dojde k chybě, program o tom informuje uživatele a ukončí se. I zde se na začátku tohoto procesu nastavuje aktivní skin tak, jak to bylo popsáno na začátku kapitoly.

EReporter a jeho nástroje jsem použil během vytváření šablon reportů a testování generování reportů. Všechny reporty jsem vyzkoušel vygenerovat nad různými daty poskytnutými vedoucím mé práce. Jednalo se o data jak uložená v souboru, tak v databázi, ke kterým může mít přístup více uživatelů. ReportCore knihovna byla použita a vyzkoušena i na webové aplikaci vytvořené na platformě angular implementované kolegou Pavlem Glosem, který její použití popsal ve své diplomové práci.

6.2 Shrnutí

Implementovaná aplikace EReporter slouží k vytváření, upravování a především generování reportů. Uživatel specifikuje zdroj dat (databázi nebo soubor) a přístroje, ze kterých se má report generovat. Dále si zvolí report který chce vygenerovat a jeho nastavení. K vytvoření reportu se využívá moje knihovna ReportCore. Ta je popsána v předešlé kapitole. EReporter se dá používat jako samostatně spustitelná aplikace i jako externí nástroj spustitelný jiným programem, kde se potřebné vstupní informace EReporteru předávají jako parametry.

Report se po vygenerování v paralelním vlákne zobrazí v DevExpress nástroji ReportPrintTool, odkud ho lze exportovat do pdf, xlsx, html a dalších formátů. Šablona reportu se také dá otevřít v dalším DevExpress nástroji ReportDesignTool, kde se může obsah reportu upravovat a měnit, popř. lze vytvářet i úplně nové šablony reportů.

Aplikace využívá knihovnu Resources z projektu STEP k překladu aplikace do různých jazyků. K vytvoření aplikace jsem využil nástroje z balíčku DevExpress

(XtraForm, WaitForm, label, button, tree list, combo box atd.), které umožňují používání DevExpress skinů k nastavení konzistentního vzhledu aplikace. Pro usnadnění práce se při ukončení aplikace poslední nastavení EReporteru ukládá do registrů a při opětovném spuštění se vstupy načtou a předvyplní. Vytvořená aplikace tedy plní všechny požadované funkce a pro potřeby této práce a její současná forma je dostačující.

EReporter

Server Name:

Database Name:

User Name:

Password:

ENVIS User Name:

ENVIS User Password:

C:\defaultConfig.repCfg

Select Report:

Select Groups

Group	Type	Object	Record
▷ ▾ DEFAULT	Directory	Object	
▷ 1612	Directory	Object	
▷ 2_11_22	Directory	Object	
▷ 6691	Directory	Object	
▽ 7541	Directory	Object	
NN_3	Directory	Object	

Obrázek 6.1: Form pro generování reportů nad daty v databázi

7 Závěr

Diplomová práce se zabývá problematikou reportingu v systémech managementu elektrické energie. Kapitola 2 obsahuje rešerši EnM systémů, jejich obsahu a často nabízených funkcí. Za tímto účelem jsem prošel velké množství různých internetových platforem, popsal je a definoval účel, který reporty na těchto platformách plní.

Pro implementaci mého programu jsem si vybral (a v kapitole 3 popsal) nástroje, jež byly k tomuto účelu použity, konkrétně knihovny od firmy DevExpress. Jejich funkce určené k vytváření reportů a desktopových aplikací jsem popsal a nastínil, k jakému účelu slouží a jak je budu využívat. Dále jsem se seznámil s knihovnamí ze projektu STEP umožňujícími práci s daty uloženými v databázi nebo souboru.

Praktickou část diplomové práce jsem se rozhodl rozdělit na dvě samostatné části: knihovnu ReportCore umožňující generování reportů na základě zvoleného nastavení a WinForms aplikaci EReporter.

ReportCore knihovna popsaná v kapitole 4 využívá DevExpress nástroje pro vytváření reportů, upravování jejich obsahu za běhu programu, analýzu a zpracování dat, generování dokumentu nad daty a exportování do souboru. Můj program všechny tyto funkce využívá k vytváření a prezentování reportů uživateli dle zvoleného nastavení. Pomocí STEP nástrojů se data načítají ze zdroje DataSource, agregují a analyzují. Agregace probíhá pomocí mnou implementovaných agregátorů, jež mohou pracovat paralelně. Výsledky se použijí při generování reportu nad těmito daty. Jsou zde obsaženy reporty spotřeby a produkce elektrické energie, alarmů, výkonu, harmonických složek napětí a proudu a reporty kvality elektrické energie. Jejich podrobný popis obsahuje kapitola 5.

V kapitole 6 jsem popsal aplikaci EReporter. Ta umožňuje uživateli zvolit report, nastavení reportu a zdroj dat buď v podobě souboru nebo databáze. Po zadání všech těchto vstupů je možné report vygenerovat. Ten se uživateli zobrazí pomocí vizualizačního DevExpress nástroje ReportDesignTool. Odtud ho jde vyexportovat do pdf, docx, xlsx a dalších formátů. Nástroj ReportDesignTool zase slouží k vytváření a upravování šablon reportu. Po vytvoření aplikace jsem ji využil k otestování funkčnosti ReportCore funkcí a generování reportů.

Vytvořené programy plní požadavky definované v zadání této diplomové práce a budou využívány i v praxi. Jednou z oblastí, kde je možné ReportCore dále rozvíjet, je optimalizace a větší integrace s nástroji ze STEP projektu, kterého by měla být tato knihovna součástí.

Oblastí, kterou jsem se v mé práci chtěl věnovat více, bylo testování aplikace.

Ačkoliv byla ReportCore knihovna a EReporter testovány nad množstvím reálných dat uložených v souborech i databázi, bylo by dobré v budoucnu pro všechny metody těchto programů napsat vhodné unit testy. Dále bude určitě třeba zakomponovat do programu logování, aby se při dalším používání mých programů odhalilo co největší množství chyb a neošetřených situací, díky čemuž půjde tyto programy zdokonalovat.

Použitá literatura

- [1] *What is Energy Management?* — IBM. URL: <https://www.ibm.com/topics/energy-management> [cit. 08. 05. 2023].
- [2] *The What, Why, and How of Energy Management.* URL: <https://www.energylens.com/articles/energy-management> [cit. 08. 05. 2023].
- [3] Charles Authur. *What is an energy management system?* — UNIDO. URL: <https://www.unido.org/stories/what-energy-management-system> [cit. 08. 05. 2023].
- [4] *ISO - ISO 50001 — Energy management.* ISO. 20. říj. 2021. URL: <https://www.iso.org/iso-50001-energy-management.html> [cit. 08. 05. 2023].
- [5] Nexavision. *ČSN EN ISO 50001:2019.* URL: https://www.qes-cert.cz/en-iso-50001?utm_source=AdWords [cit. 08. 05. 2023].
- [6] Martin Vondráček. „Techniky zpracování a vizualizace dat v systémech energetického managementu chytrých budov“ (2020). [cit. 18. 04. 2023].
- [7] *Introduction to PowerRadar - Centrica Business Solutions Knowledge Base.* URL: <https://support.powerradar.energy/en/article/introduction-to-powerradar> [cit. 15. 10. 2022].
- [8] *NAVISYS - Systém monitoringu a měření spotřeby energií.* NAVISYS. URL: <https://www.mereni-spotreby.cz/> [cit. 15. 10. 2022].
- [9] *Smart Building Solutions.* URL: <https://www.us.jll.com/en/solutions/smart-buildings> [cit. 15. 10. 2022].
- [10] *InfluxDB Time Series Data Platform.* InfluxData. URL: <https://www.influxdata.com/home/> [cit. 18. 10. 2022].
- [11] *Grafana: The open observability platform.* Grafana Labs. URL: <https://grafana.com/> [cit. 24. 10. 2022].
- [12] *Smarkia.* URL: <https://en.smarkia.com/> [cit. 18. 10. 2022].
- [13] *Home page.* Electrex - the energy saving technology. URL: <https://www.electrex.it/en/> [cit. 21. 10. 2022].
- [14] *Home.* Energis. URL: <https://energis.cloud/en/> [cit. 27. 10. 2022].
- [15] *Energy management systems and sustainability software - EnergyElephant.* URL: <https://energyelephant.com/> [cit. 27. 10. 2022].

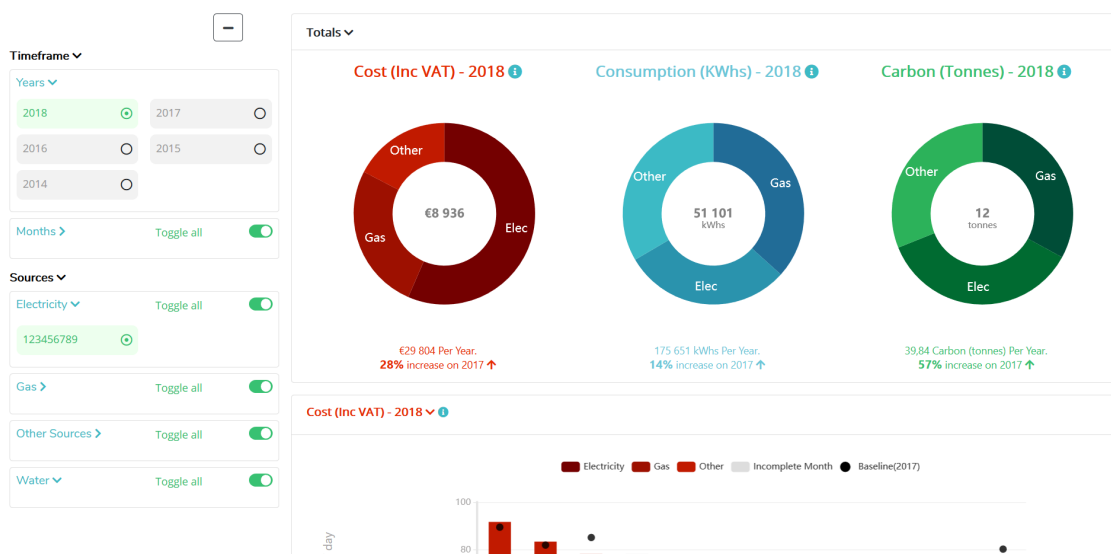
- [16] *Energomonitor - Insight into your house*. Energomonitor. URL: <https://www.energomonitor.com/> [cit. 27. 10. 2022].
- [17] *Monitoring Solutions 2023. ESC Spectrum — Industry Leaders in Air Emissions Compliance*. Monitoring Solutions. URL: <https://monsol.com/> [cit. 27. 10. 2022].
- [18] *Energy Management & Monitoring System Dashboard: Cloud Based Software*. Wattics. URL: <https://www.wattics.com/dashboard/> [cit. 30. 10. 2022].
- [19] *OPIT Solutions AG*. URL: <https://www.opit.ch/> [cit. 30. 10. 2022].
- [20] *Flowbox – Řešení pro digitalizaci průmyslu, budov a měst*. Flowbox. URL: <https://www.flowbox.com/> [cit. 02. 11. 2022].
- [21] *Energy Studio Cloud – Elcontrol*. URL: <http://www.elcontrol-energy.net/divisioni/software-e-reti/energy-studio-cloud/> [cit. 03. 11. 2022].
- [22] *Power quality - Home*. URL: <https://www.janitza.com/power-quality.html> [cit. 03. 11. 2022].
- [23] *Power Management solutions*. URL: <https://www.se.com/ww/en/work/solutions/power-management/> [cit. 10. 11. 2022].
- [24] *Home*. beedata. URL: <https://beedataanalytics.com/> [cit. 02. 11. 2022].
- [25] *Entronix Energy Management*. URL: <https://entronix.io/> [cit. 13. 11. 2022].
- [26] *Home – SkyFoundry*. URL: <https://skyfoundry.com/> [cit. 05. 11. 2022].
- [27] *Sankey Diagram — Charts*. Google Developers. URL: <https://developers.google.com/chart/interactive/docs/gallery/sankey> [cit. 19. 04. 2023].
- [28] *What is a Heatmap? + How to Create, Analyze & Use Heatmaps — FullStory*. URL: <https://www.fullstory.com/heatmap/> [cit. 19. 04. 2023].
- [29] *Developer Express Inc. About Us: Our Mission and Contacts — DevExpress*. URL: <https://www.devexpress.com/aboutus/> [cit. 10. 03. 2023].
- [30] *Export Reports — Reporting — DevExpress Documentation*. URL: <https://docs.devexpress.com/XtraReports/1302/detailed-guide-to-devexpress-reporting/store-and-distribute-reports/export-reports> [cit. 18. 04. 2023].
- [31] *XtraReport Class — Reporting — DevExpress Documentation*. URL: <https://docs.devexpress.com/XtraReports/DevExpress.XtraReports.UI.XtraReport> [cit. 10. 03. 2023].
- [32] *PrintingSystemBase.Document Property — Cross-Platform Class Library — DevExpress Documentation*. URL: <https://docs.devexpress.com/CoreLibraries/DevExpress.XtraPrinting.PrintingSystemBase.Document> [cit. 10. 03. 2023].
- [33] Jeffrey Stedfast. *MailKit*. original-date: 2013-12-07T04:07:06Z. 18. dub. 2023. URL: <https://github.com/jstedfast/MailKit> [cit. 18. 04. 2023].
- [34] *ReportDesignTool Class — Reporting — DevExpress Documentation*. URL: <https://docs.devexpress.com/XtraReports/DevExpress.XtraReports.UI.ReportDesignTool> [cit. 10. 03. 2023].

- [35] *ReportPrintTool Class* — *WinForms Controls* — *DevExpress Documentation*. URL: <https://docs.devexpress.com/WindowsForms/DevExpress.XtraReports.UI.ReportPrintTool> [cit. 10. 03. 2023].
- [36] dotnet-bot. *ResourceManager Class (System.Resources)*. URL: <https://learn.microsoft.com/en-us/dotnet/api/system.resources.resourcemanager?view=net-8.0> [cit. 10. 03. 2023].
- [37] *XtraReport.ModifyDocument(Action;IDocumentModifier;) Method* — *Reporting* — *DevExpress Documentation*. URL: [https://docs.devexpress.com/XtraReports/DevExpress.XtraReports.UI.XtraReport.ModifyDocument\(System.Action-DevExpress.XtraReports.IDocumentModifier-\)](https://docs.devexpress.com/XtraReports/DevExpress.XtraReports.UI.XtraReport.ModifyDocument(System.Action-DevExpress.XtraReports.IDocumentModifier-)) [cit. 18. 04. 2023].
- [38] dotnet-bot. *DataSet Class (System.Data)*. URL: <https://learn.microsoft.com/en-us/dotnet/api/system.data.dataset?view=net-8.0> [cit. 18. 04. 2023].
- [39] *Group and Sort the Report Data* — *Reporting* — *DevExpress Documentation*. URL: <https://docs.devexpress.com/XtraReports/1298/detailed-guide-to-devexpress-reporting/shape-report-data/group-and-sort/group-and-sort-a-reports-data> [cit. 18. 04. 2023].
- [40] *Add Bookmarks and a Document Map* — *Reporting* — *DevExpress Documentation*. URL: <https://docs.devexpress.com/XtraReports/2603/detailed-guide-to-devexpress-reporting/add-navigation/add-bookmarks-and-a-document-map> [cit. 18. 04. 2023].
- [41] *Store Report Layouts* — *Reporting* — *DevExpress Documentation*. URL: <https://docs.devexpress.com/XtraReports/2592/detailed-guide-to-devexpress-reporting/store-and-distribute-reports/store-report-layouts-and-documents/store-report-layouts> [cit. 18. 04. 2023].
- [42] *Load Duration Curve in Power Station*. Section: Power System. 29. zář. 2022. URL: <https://studyelectrical.com/2022/09/load-duration-curve-in-power-station.html> [cit. 18. 04. 2023].
- [43] *ČSN EN 50160 ED.3 (330122) Charakteristiky napětí elektrické energie dodávané z veřejných distribučních sítí*. Standard. ČAS, 2011. [cit. 18. 04. 2023].
- [44] *ČSN EN 61000-2-2 (333432) Elektromagnetická kompatibilita (EMC) - Část 2-2: Prostředí - Kompatibilní úrovně pro nízkofrekvenční rušení šířené vedením a signály ve veřejných rozvodných sítích nízkého napětí*. Standard. ČAS, 2004. [cit. 18. 04. 2023].
- [45] *ČSN EN 61000-2-4 ED.2 (333432) Elektromagnetická kompatibilita (EMC) - Část 2-4: Prostředí - Kompatibilní úrovně pro nízkofrekvenční rušení šířené vedením v průmyslových závodech*. Standard. ČAS, 2003. [cit. 18. 04. 2023].
- [46] *ČSN EN 61000-2-12 (333432) Elektromagnetická kompatibilita (EMC) - Část 2-12: Prostředí - Kompatibilní úrovně pro nízkofrekvenční rušení šířené vedením a signály ve veřejných rozvodných sítích vysokého napětí*. Standard. ČAS, 2004. [cit. 18. 04. 2023].

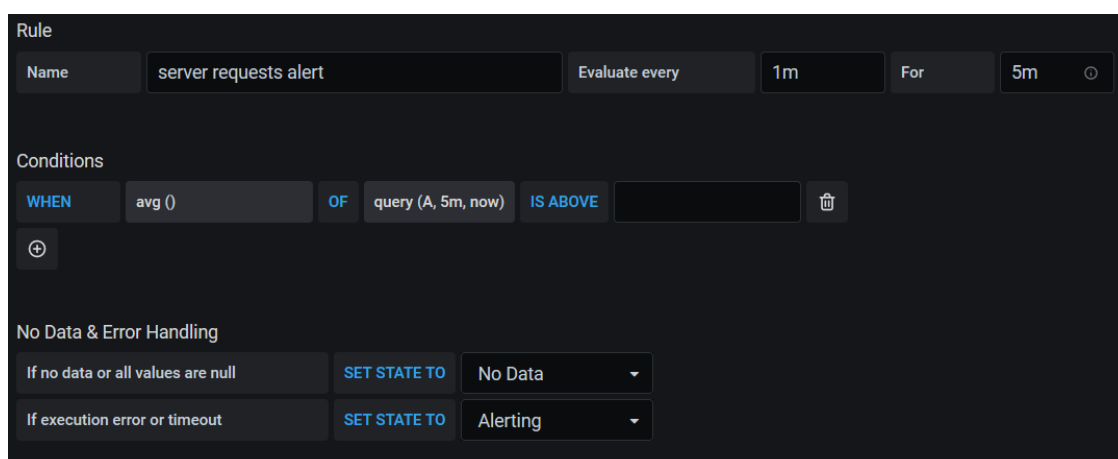
- [47] *Numeracy, Maths and Statistics - Academic Skills Kit*. URL: <https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/data-presentation/histograms.html> [cit. 18.04.2023].
- [48] *Box plot review (article)*. Khan Academy. URL: <https://www.khanacademy.org/math/statistics-probability/summarizing-quantitative-data/box-whisker-plots/a/box-plot-review> [cit. 18.04.2023].
- [49] adegeo. *What is Windows Forms - Windows Forms .NET*. 14. ún. 2023. URL: <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/overview/> [cit. 18.04.2023].
- [50] *Skins — WinForms Controls — DevExpress Documentation*. URL: <https://docs.devexpress.com/WindowsForms/2399/build-an-application/skins> [cit. 18.04.2023].

8 Přílohy

8.1 Prvky v existujících EnM systémech



Obrázek 8.1: Energy Elephant [15]: Graf spotřeby a interaktivní menu



Obrázek 8.2: Grafana [11]: Nastavení alarmu

Pojmenujte své upozornění*

Kdy chcete dostat upozornění? [?](#)

Odběrné místo* Demo

Senzor* Hlavní odběr, fáze 2 (W)

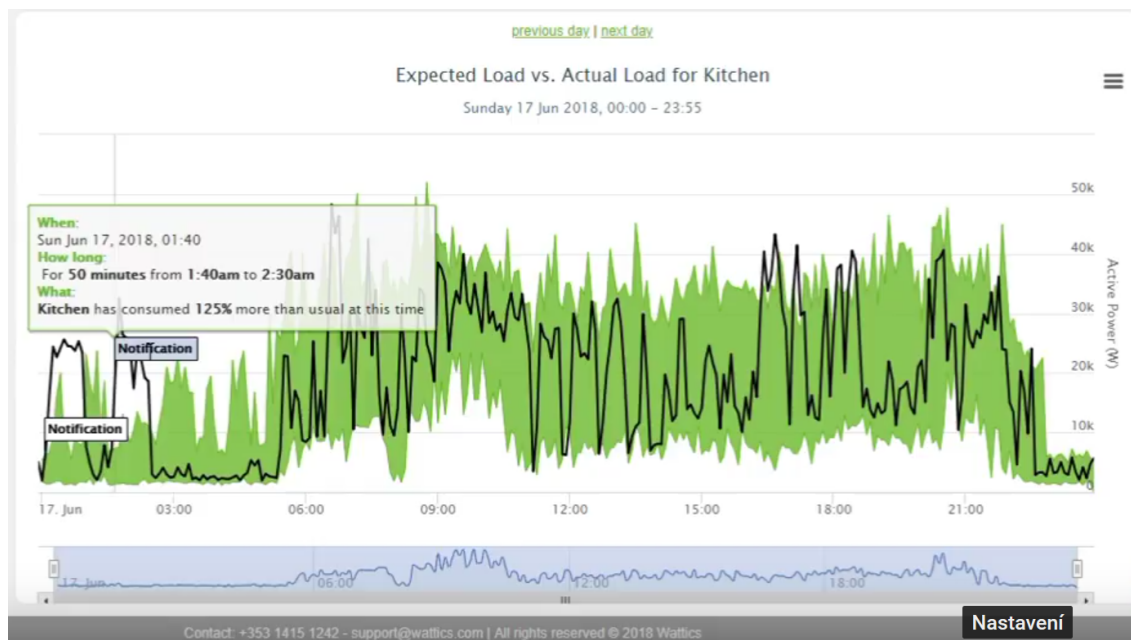
Operátor* >=

Hodnota*

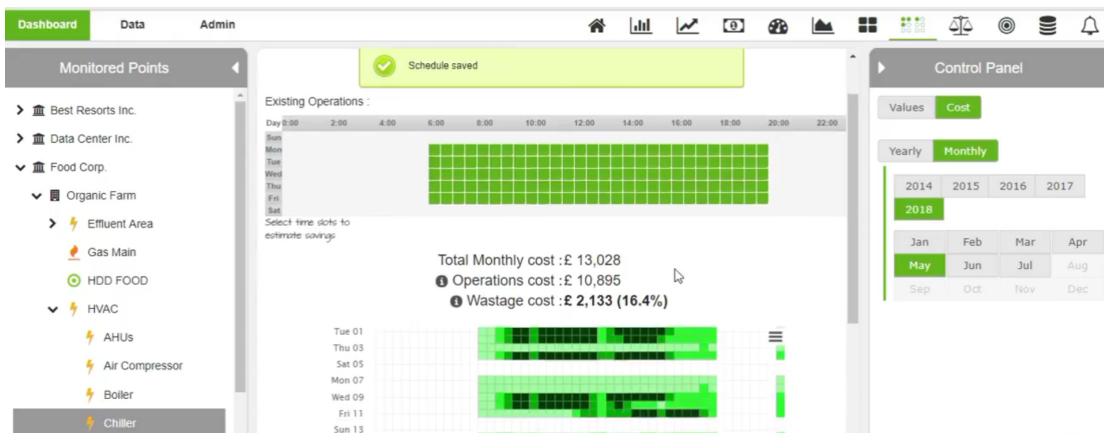
Alespoň* 15 minut

Pondělí

Obrázek 8.3: Energomonitor [16]: Nastavení alarmu



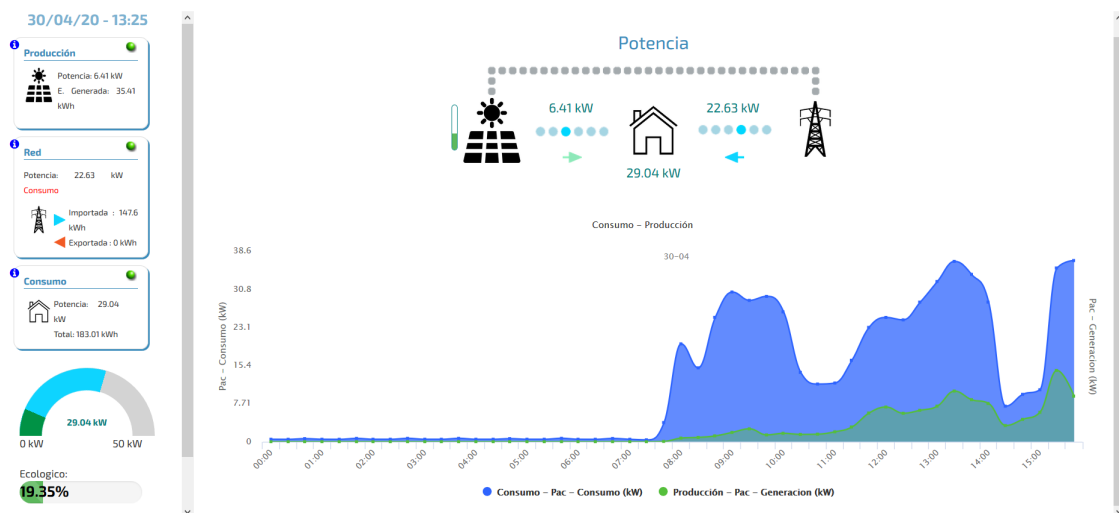
Obrázek 8.4: Watticks [18]: Porovnání očekávaného a skutečného průběhu



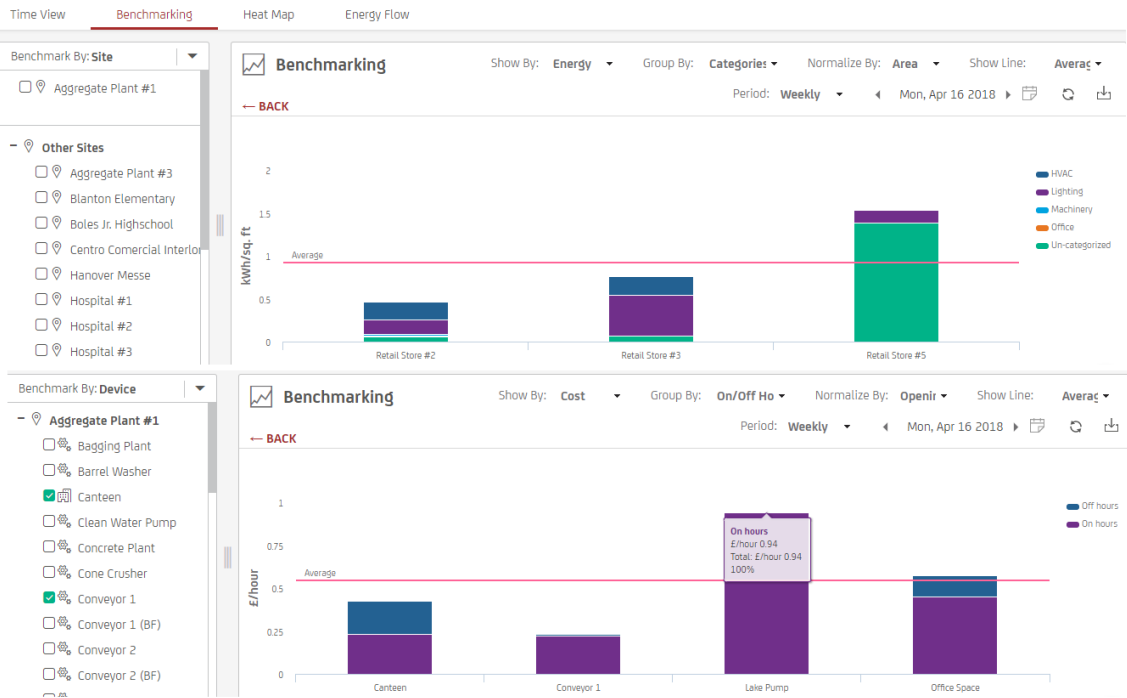
Obrázek 8.5: Watticks [18]: Výpočet úspor při nulové spotřebě v označený čas



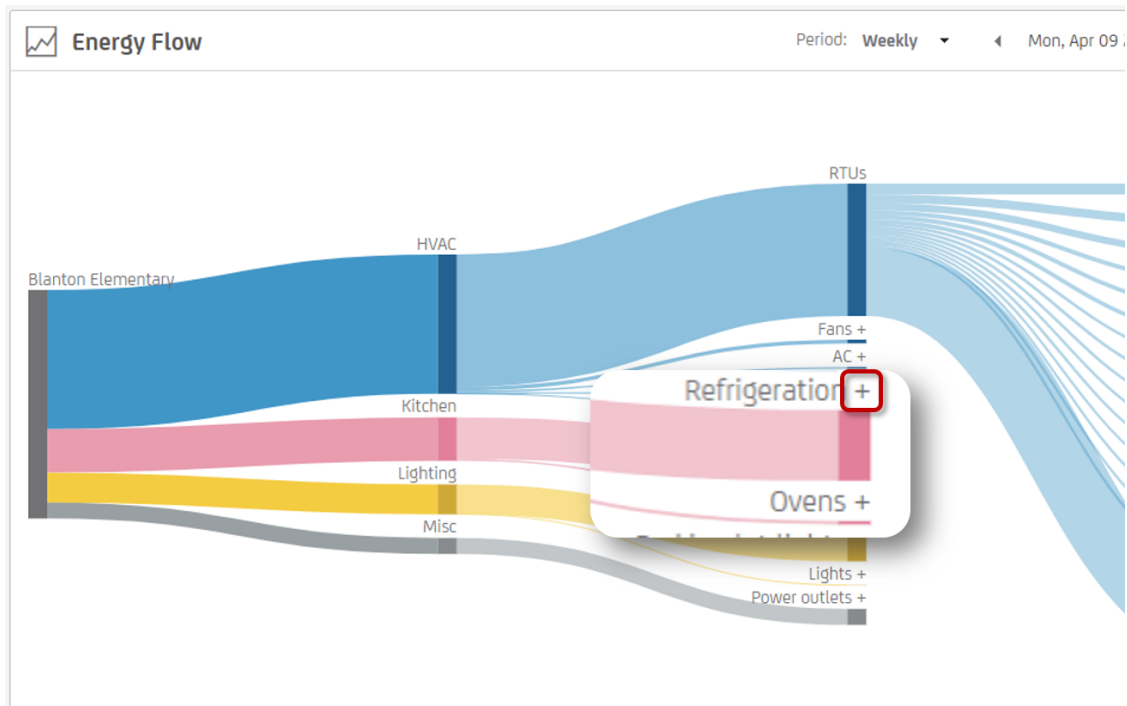
Obrázek 8.6: Watticks [18]: Porovnání tarifů



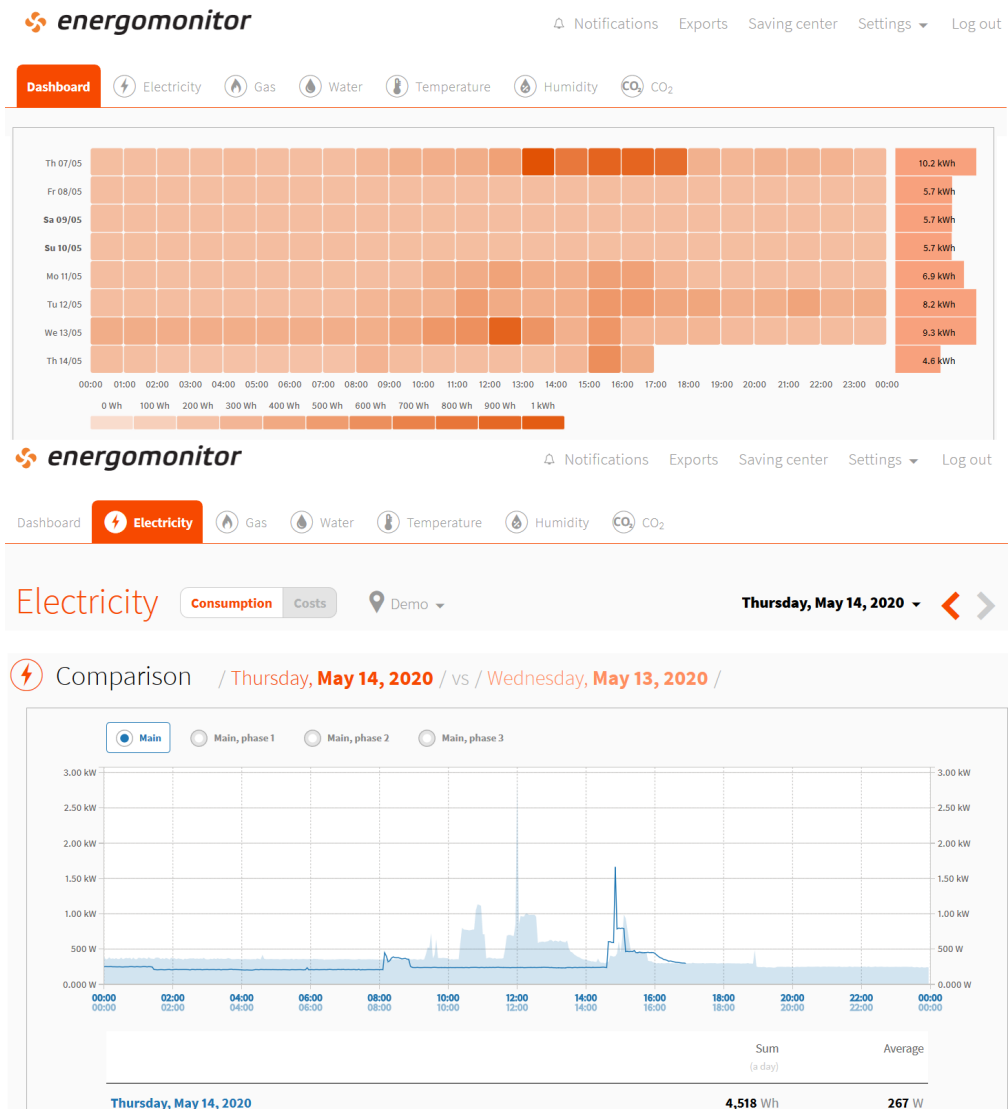
Obrázek 8.7: Monsol [17]: Porovnání spotřeby a vlastní produkce



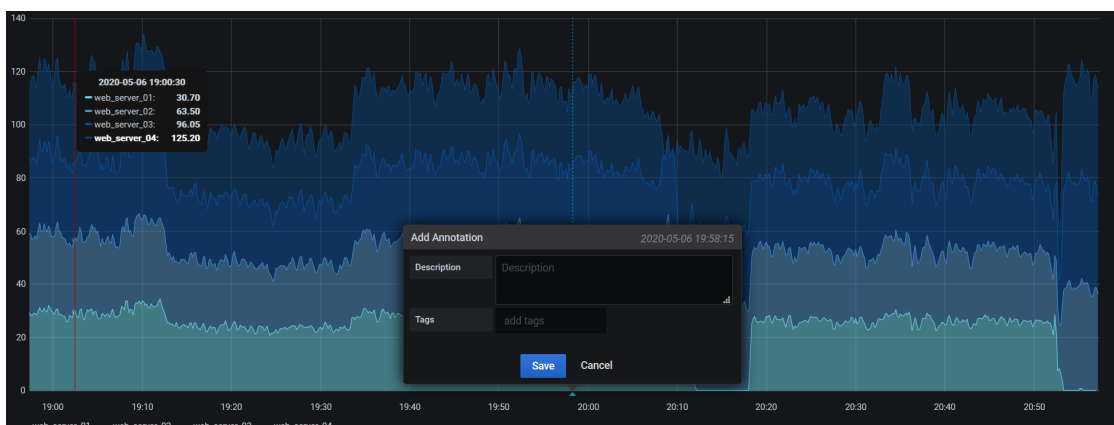
Obrázek 8.8: Power Radar [7]: Porovnání spotřeby



Obrázek 8.9: Power Radar [7]: Sankeyho diagram



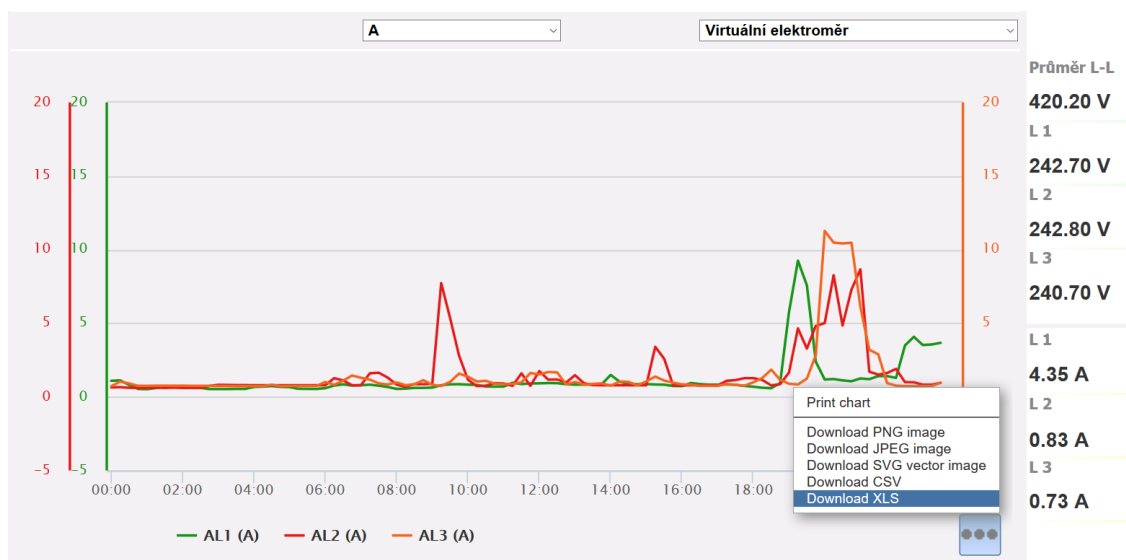
Obrázek 8.10: Energomonitor [16]: Interaktivní heatmapa a graf pro konkrétní den



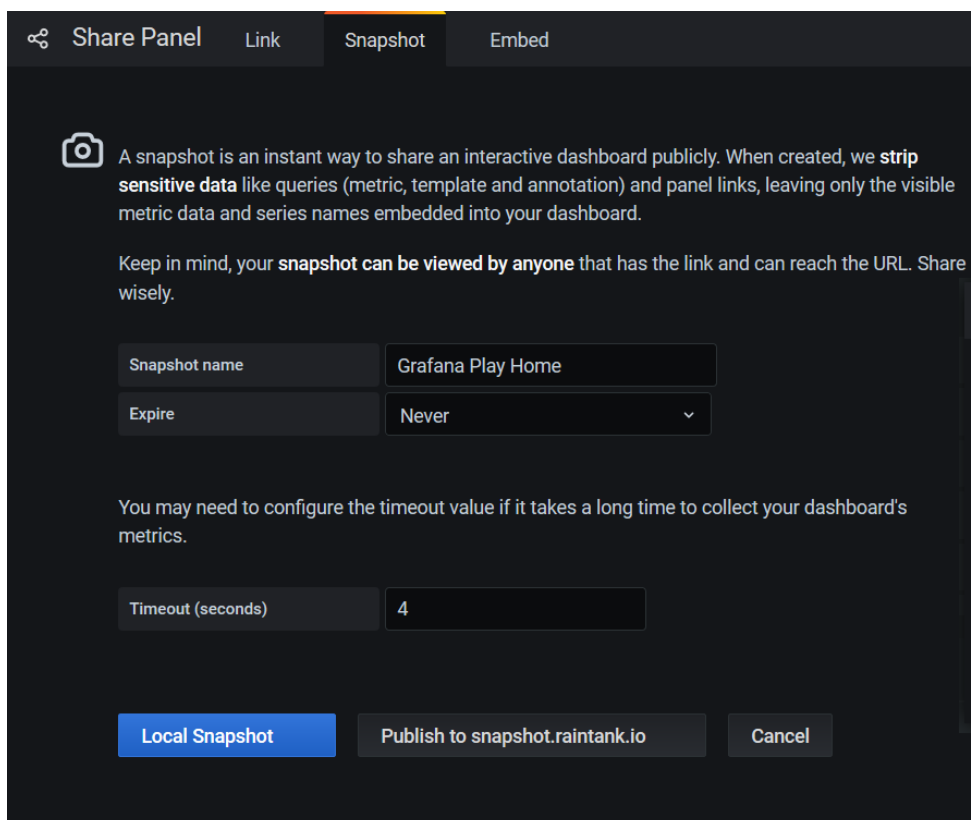
Obrázek 8.11: Grafana [11]: Přidání poznámek



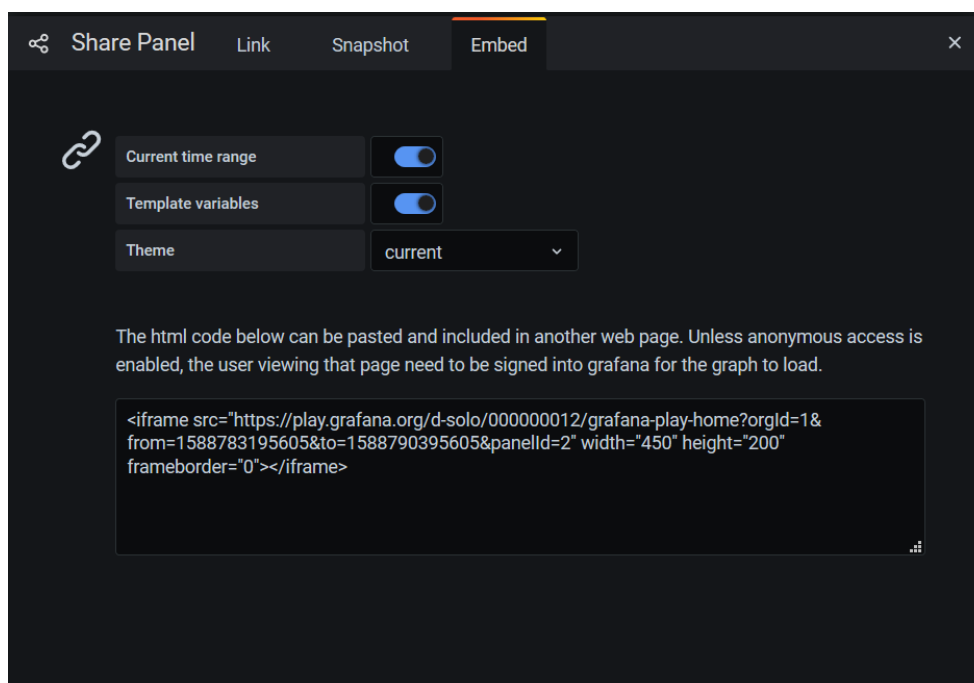
Obrázek 8.12: Watticks [18]: Porovnání tarifů



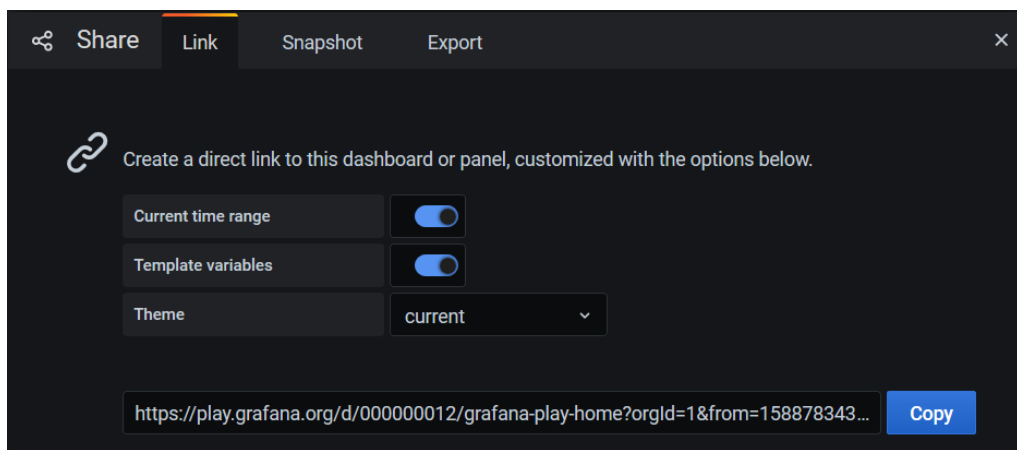
Obrázek 8.13: Navysis [8]: Export do různých formátů



Obrázek 8.14: Grafana [11]: Snapshot panelu



Obrázek 8.15: Grafana [11]: Embed



Obrázek 8.16: Grafana [11]: Sdílení odkazem