Вінницький національний технічний університет
(повне найменування вищого навчального закладу)
Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))
Кафедра автоматизації та інтелектуальних інформаційних технологій
(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до магістерської кваліфікаційної роботи
на тему Розробка автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах

Виконав: студентка 2 курсу, групи 1АКІТ-21м
спеціальності
151 – Автоматизація та комп'ютерно-інтегровані технології
(шифр і назва напряму підготовки, спеціальності)

Рудич Є. О.
(прізвище та ініціали)
Керівник Кулик Я. А.
(прізвище та ініціали)
Рецензент Ковалюк О.О.
(прізвище та ініціали)

Вінниця ВНТУ 2022

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації

Кафедра автоматизації та інтелектуальних інформаційних технологій

Освітньо-кваліфікаційний рівень - магістр

Напрям підготовки 151 – Автоматизація та комп'ютерно-інтегровані технології

ЗАТВЕРДЖУЮ

Завідувач кафедри АІІВ

О.В. Бісікало

"____"_____2022 року

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Рудич Єлизаветі Олександрівній

(прізвище, ім'я, по батькові)

1. Тема магістерської кваліфікаційної роботи Розробка автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах

керівник магістерської кваліфікаційної роботи

доцент Кулик Ярослав Анатолійович ,

(прізвище, ім'я, по батькові)

затверджені наказом вищого навчального закладу від

"__"_____2022 року №

2. Строк подання студентом магістерської кваліфікаційної роботи "__"_____2022 року

3. Вихідні дані до магістерської кваліфікаційної роботи

Мови голосового інтерфейсу – українська, російська, англійська,

максимальна кількість команд – 100,

максимальний час перекладу команди – 2 с,

призначення голосового інтерфейсу для проведення тестування – керування програмами ОС Windows.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Вступ, науково-технічне обґрунтування розробки моделей та методів автоматизованих систем контролю та обліку енергоресурсів , розробка математичних моделей споживання електроенергії в домогосподарствах, розробка програмного забезпечення споживання електроенергії в домогосподарствах, економічна частина.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) UML–діаграма класів сутностей системи, UML–діаграма основних функціональних класів системи, UML–діаграма репозиторію InfluxDbFindStore, UML–діаграма репозиторію InfluxDbStore, UML-діаграма тестів класу DateServiceTest.

6. Консультанти розділів магістерської кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	доцент кафедри АІІВ Кулик Я.А.		
2	доцент кафедри АІІВ Кулик Я.А.		
3	доцент кафедри АІІВ Кулик Я.А.		
4	професор кафедри ЕПОВ Козловський В.О.		

7. Дата видачі завдання "___"_____2022 року

Календарний план

№	Назва етапів роботи	Строк виконання етапів роботи	Примітка
1	Науково-технічне обґрунтування розробки моделей та методів автоматизованих систем контролю та обліку енергоресурсів		
2	Розробка математичних моделей споживання електроенергії в домогосподарствах		
3	Розробка програмного забезпечення споживання електроенергії в домогосподарствах		
4	Економічна частина		
5	Оформлення пояснювальної записки, графічного матеріалу і презентації		
6	Захист МКР		

Студентка Рудич Є.О. _____
(прізвище та ініціали) (підпис)

Керівник Кулик Я. А. _____
(прізвище та ініціали) (підпис)

АНОТАЦІЯ

У даній дипломній роботі розглянуто питання, пов’язані з розробкою автоматизованої системи аналізу споживання електроенергії в домогосподарствах.

Запропонований підхід дозволив спроектувати та розробити програмний засіб для оцінки та аналізу кількості споживаної енергії, які за рахунок ефективної організації процесу дозволяє аналізувати витраченні ресурси та контролювати фінансові затрати.

Також робота містить загальний огляд засобів та інструментів створення автоматизованої системи, аналіз їх переваг та недоліків та економічне обґрунтування доцільності даної розробки.

ABSTRACT

This thesis examines the issue related to the development of automated systems for analyzing electricity consumption in households.

The proposed approach makes it possible to design and develop a software tool for estimating and analyzing the amount of energy consumed, which, due to the effective organization of the process, allows for the analysis of spent resources and control of financial costs.

The work also contains a general overview of the means and tools for creating an automated system, an analysis of their advantages and disadvantages, and an economic justification of the feasibility of this development.

ЗМІСТ

ЗМІСТ 6

ВСТУП 9

1 НАУКОВО-ТЕХНІЧНЕ ОБГРУНТУВАННЯ РОЗРОБКИ МОДЕЛЕЙ ТА МЕТОДІВ АВТОМАТИЗОВАНИХ СИСТЕМ КОНТРОЛЮ ТА ОБЛІКУ ЕНЕРГОРЕСУРСІВ 12

1.1 Основні поняття та визначення 12

1.2 Варіанти організації та побудови АСКОЕ на прикладі систем обліку електроенергії 15

1.3 Програмне забезпечення АСКОЕ 21

1.4 Зарубіжний досвід впровадження автоматизованої системи контролю та обліку електроенергії 23

1.5 Впровадження автоматизованої системи контролю та обліку електроенергії в Україні 26

1.6 Постановка задачі 28

Висновки по розділу 29

2 РОЗРОБКА МАТЕМАТИЧНИХ МОДЕЛЕЙ СПОЖИВАННЯ ЕЛЕКТРОЕНЕРГІЇ В ДОМОГОСПОДАРСТВАХ 30

2.1 Розрахунок споживання 30

2.2 Відображення витрат у реальному часі 31

2.3 Обмеження споживання 32

2.4 Формування вимог до розроблюваної системи 35

Висновки до розділу 36

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СПОЖИВАННЯ ЕЛЕКТРОЕНЕРГІЇ В ДОМОГОСПОДАРСТВАХ 37

3.1 Основні функціональні вимоги програмного забезпечення 37
3.2 Структура сучасної системи управління 37
3.3 Відкриті стандарти комунікації 38
3.4 Серіалізація даних 42
3.5 Збереження даних 43
3.6 Вибір фрейморка для реалізації серверної частини.....44
3.7 Опис середовища системи.....44
3.8 Серверна обробка даних.....46
3.9 Розробка UML–діаграми класів сутностей системи 47
3.10 Розробка UML–діаграми основних функціональних класів системи 47
3.12 Інтерфейс користувача 48
Висновки по розділу 49
4. ЕКОНОМІЧНИЙ РОЗДІЛ.....50
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 68
ДОДАТКИ.....72
Додаток А (обов’язковий). 73
Додаток Б (обов’язковий).....77
Додаток В (обов’язковий). 82
Додаток Г (обов‘язковий).....83

ВСТУП

Актуальність даної роботи полягає в тому, що аналіз споживання електроенергії в домогосподарствах дає можливість ефективного використання ресурсів та слідкування за вартістю та обсягом споживання електроенергії, що неможливо швидко виконати вручну. Все більшої популярності у світі набуває модель розумного споживання, яка тим самим мінімізує надлишок вироблених і неспожитих домогосподарствами ресурсів.

Поточне зростання кількості електроприладів, які використовуються у домогосподарствах, викликає збільшення енергоресурсів, і тим самим збільшення їх вартості. В умовах, які створилися, стає актуальним чіткий облік спожитих ресурсів та підвищення контролю за їх використанням, а також зручність обліку та підрахунку, прийняття ефективних рішень, щодо раціоналізації споживання і попередження несанкціонованого відбору або витоків ресурсів.

Ефективним методом виконання зазначеного завдання є створення системи, яка відображає у реальному часу поточне споживання ресурсів (електроенергіїї, теплової енергіїї, води). Наявність цієї інформації дозволяє приймати рішення стосовно регулювання споживання енергоносіїв і тим самим зменшити витрати на їх споживання.

При побудові розумної енергосистеми проводиться заміна аналогових механічних лічильників на цифрові лічильники, які записують споживання у реальному часі. Часто ця технологія називається передова вимірювальна інфраструктура, оскільки лічильники самі по собі не є корисними, і повинні встановлюватись разом з комунікаційною інфраструктурою для передачі даних. Передова вимірювальна інфраструктура може надати канал зв'язку між електростанціями з однієї сторони і кінцевими споживачами у домогосподарствах і виробництвах з іншої. Ці пристрої кінцевих споживачів можуть включати розумні розетки та інші пристрої, здатні взаємодіяти з розумною енергосистемою, такі як водонагрівачі та термостати. У залежності

від програми постачальника можуть бути сповіщені споживачі, або пристрої можуть вимикатись, або їх налаштування можуть автоматично змінюватись в залежності від часу піку споживання.

Метою даної роботи є підвищення швидкості збору інформації про споживання електроенергії шляхом розробки системи обліку та аналізу споживання електроенергії в домогосподарствах на основі збору та аналізу інформації.

Об'єктом дослідження є процес споживання електроенергії в домогосподарствах.

Предметом дослідження даної роботи є методи обліку та контролю електричної енергії.

Для досягнення поставленої мети необхідно розв'язати наступні задачі:

– провести аналіз існуючих методів збору та обробки інформації щодо споживання електроенергії в домогосподарствах;

– побудувати модель процесу автоматизованої програмної системи, яка буде розроблятися;

– розробити програмне забезпечення, що на основі побудованої моделі, що реалізує систему обліку та аналізу споживання електроенергії в домогосподарствах;

– створити тестові сценарії для тестування системи;

Методи дослідження. В процесі дослідження застосовуються загальні аналітичні методи комп'ютерних наук, методи розробки автоматизованих систем, методи теорії алгоритмів, методи оптимізації.

Основний іноваційний результат роботи **-** розробка, оптимізація та застосування сучасних підходів, методів та алгоритмів для ефективної організації автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах.

Практичним значенням даної роботи є розроблене програмне забезпечення та алгоритм, що дозволить проводити облік споживання електроенергії в домогосподарствах.

Апробація. Дана робота була апробована у доповіді на МН-2023 конференції Вінницького національного технічного університету у 2022 році. *[1]*

1 НАУКОВО-ТЕХНІЧНЕ ОБГРУНТУВАННЯ РОЗРОБКИ МОДЕЛЕЙ ТА МЕТОДІВ АВТОМАТИЗОВАНИХ СИСТЕМ КОНТРОЛЮ ТА ОБЛІКУ ЕНЕРГОРЕСУРСІВ

1.1 Основні поняття та визначення

Автоматизована система контролю та обліку енергоресурсів (АСКОЕ) – система електронних програмно-технічних засобів для автоматизованого, у реальному масштабі часу дистанційного вимірювання, збору, передачі, обробки, відображення та документування процесу вироблення, передачі або споживання енергоресурсів за заданою множиною просторово розподілених точок їх вимірювання, що належать енергетичним об'єктам суб'єкта енергосистеми або споживача.[2]

Пристрій збору та передачі даних (ПЗПД) – мікропроцесорний пристрій (контролер) для запиту та прийому даних вимірювання та обліку від групи електролічильників за цифровими або іншим інтерфейсам, обробки отриманих даних, передачі їх канал зв'язку на верхній рівень АСКОЕ, а також зворотної передачі в електролічильники службових даних.[3]

Засоби обліку електроенергії – технічні засоби, до яких відносяться вимірювальні трансформатори струму та напруги, електролічильники та спеціалізовані системи обліку (Контролери).

Індукційний лічильник – лічильник з електромеханічним принципом вимірювання та відображення значень даних виміру.

Електронний лічильник – лічильник для вимірювання кількості та/або якості електроенергії та потужності з електронними схемами вимірювання та відображення даних вимірювання.[4]

За способом застосування розрізняють АСКОЕ трьох типів:

– АСКОЕ широкого застосування, що розробляються для серійного виробництва у вигляді закінчених виробів;

– АСКОЕ вузького застосування, що розробляються для одиничного або виготовлення, що повторюється дрібними партіями;

– АСКОЕ цільового застосування, що проектуються для певних об'єктів (груп однорідних об'єктів) і створюються як закінчений виріб безпосередньо на об'єкті експлуатації шляхом комплектації з компонентів серійного чи одиничного виготовлення та відповідного монтажу та налагодження, що здійснюються відповідно до проектної документації. [5]

За принципом організації існуючі АСКОЕ поділяють на три типи: локальні, регіональні та національні.

Локальна АСКОЕ містить:

– лічильники електричної енергії (ЕЕ) та потужності;

– пристрій збору та передачі даних ПЗПД (телесуматори, мультиплексори, концентратори тощо);

– сервер опитування ПЗПД і сервер бази даних (БД), що є ЕОМ, з'єднану з ПЗПД або лічильниками ЕЕ. На ЕОМ встановлюється спеціалізоване програмне забезпечення (ПЗ), здатне приймати дані від ПЗПД та зберігати їх у базі даних;

– робочі місця технологів чи енергетиків – ЕОМ, підключені до локальної обчислювальної мережі (ЛОМ) підприємства, в яких знаходяться сервер опитування ПЗПД та сервер БД. Сервер опитування ПЗПД та сервер БД утворюють вузол локальної АСКОЕ. Можлива також організація віддалених робочих місць.

У тих випадках, коли необхідно організувати збір та обробку даних від кількох локальних АСКОЕ, створюється регіональна АСКУЕ, що є багаторівневою системою. Верхні рівні цієї системи утворені вузлами, з'єднаними між собою лініями зв'язку, що містять відповідну каналоутворюючу апаратуру.

До нижнього рівня належать локальні АСКОЕ, від яких надходить інформація про споживання електричної енергії.

Національна АСКОЕ будується на тих же принципах, що й районна АСКОЕ, але охоплює територію всієї держави.[6]

За способом опитування лічильників ЕЕ АСКОЕ поділяють на сильно зв'язкові та слабо зв'язкові системи.

У сильно зв'язних (синхронних) системах опитування лічильників виконується циклічно із заданою частотою (зазвичай один раз за 1 або 3 хв). Такий принцип опитування характерний для АСКОЕ, містить лічильники ЕЕ з частотно-імпульсним виходом без запам'ятовування інформації.

Слабо зв'язні (асинхронні) системи будуються на базі лічильників ЕЕ з цифровим інтерфейсом та можливістю зберігання даних у них. Опитування лічильників може здійснюватися як циклічно, так і з ініціативи лічильників чи ПЗПД.

Складні багаторівневі АСКОЕ можуть містити як сильнозв'язкові, так і слабозв'язкові підсистеми, тобто комбіновані.

За функціональним призначенням розрізняють АСКОЕ для комерційного та технічного обліку ЕЕ. Багатофункціональні АСКОЕ здійснюють обидва види обліку одночасно.

У свою чергу, АСКОЕ поділяють на активні та пасивні.

Пасивні АСКОЕ орієнтовані на витяг квитанцій про оплату ЕЕ і не мають можливості оперативного управління режимами енергоспоживання. Активні АСКОЕ вирішують це завдання. [7]

На рис. 1.1 наведена типова структурна схема цифрової АСКОЕ.

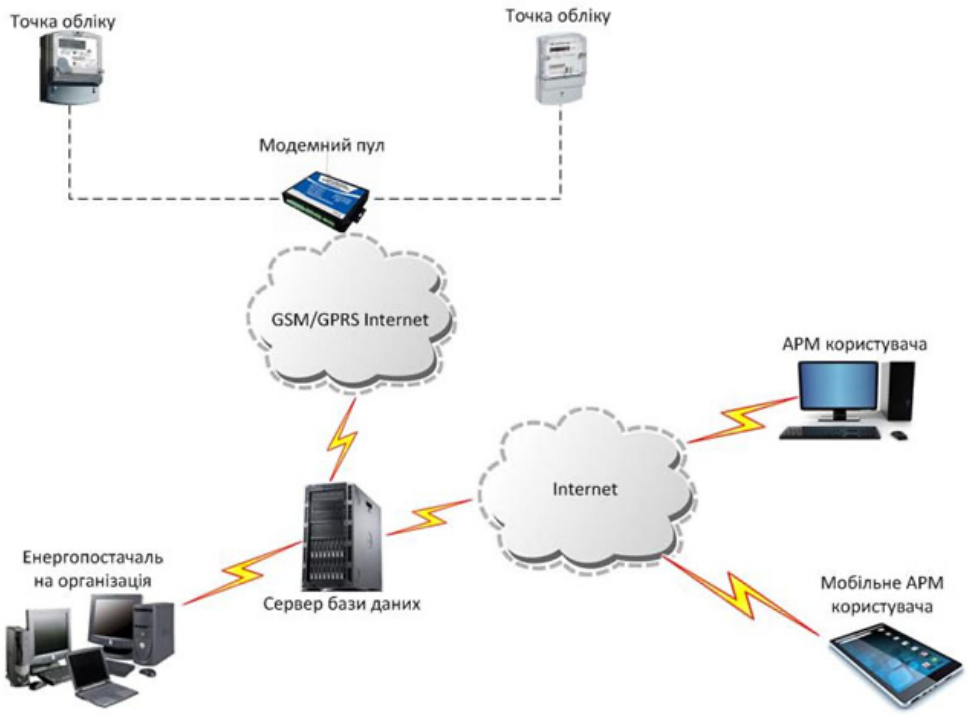

Рис. 1.1 – Структурна схема АСКОЕ

1.2 Варіанти організації та побудови АСКОЕ на прикладі систем обліку електроенергії

Організація АСКОЕ із проведенням опитування лічильників через оптичний порт. Це найпростіший варіант організації АСКОЕ. Лічильники не поєднані між собою. Між лічильниками та центром збору даних немає зв'язку. Усі лічильники опитуються послідовно при обході лічильників оператором. Опитування проводиться через оптичний порт за допомогою розміщеної на портативному комп'ютері програми, яка формує файл результатів опитування. На комп'ютері центру збору даних необхідні програмні модулі, що формують файл-завдання на опитування та завантажують інформацію в основну БД. Синхронізація часу лічильників з часом переносного комп'ютера відбувається у процесі опитування. Синхронізація часу переносного комп'ютера з часом центр збору даних проводиться в момент прийому файлів завдань на опитування лічильників. Для максимальної економії коштів на

створення АСКУЭ у цьому варіанті роль центру збору даних можна покласти на переносний комп'ютер. Недоліками цього способу організації АСКУЕ є велика трудомісткість збору даних з лічильників та неможливість використання в системі дешевих індукційних чи електронних лічильників з імпульсним виходом.

Організація АСКУЕ з проведенням опитування лічильників через оптичний порт дозволяє вирішувати такі завдання:

– точне вимірювання параметрів постачання/споживання;

– комерційний та технічний облік енергоресурсів по підприємству, його інфраструктурним елементам (котельня та об'єкти жилкомпобуту, цеху, підрозділи, субабоненти);

– контроль енергоспоживання за точками та об'єктами обліку в заданих часових інтервалах (30 хвилин, зони, зміни, доба, декади, місяці, квартали та роки) щодо заданих лімітів та технологічних обмежень потужності;

– обробка даних та формування звітів з обліку електроенергії;

– діагностика повноти даних;

– опис електричних з'єднань об'єктів та їх характеристик;

– діагностика лічильників;

– підтримка єдиного системного часу.[8]

Організацію опитування лічильників через оптичний порт можна спостерігати на рисунку 1.2.

Рис. 1.2 – Організація опитування лічильників через оптичний порт

Організація АСКОЕ з проведенням опитування лічильників переносним комп'ютером через перетворювач інтерфейсів, мультиплексор чи модем.

Між лічильниками та центром збору даних немає постійного зв'язку. ПЗПД виконує роль комунікаційного сервера. Комп'ютер центру збору даних повинен містити програмні модулі, що формують файл-завдання на опитування та завантажують інформацію у основну БД. Синхронізація часу лічильників з часом переносного комп'ютера відбувається у процесі опитування. Синхронізація часу переносного комп'ютера з часом центр збору даних проводиться в момент прийому файлів завдань на опитування лічильників. Виділений комп'ютер для центру збору даних у цьому варіанті також може бути відсутнім, його роль може виконувати переносний комп'ютер.

Організація АСКОЕ з проведенням опитування лічильників переносним комп'ютером через перетворювач інтерфейсів, мультиплексор чи модем дозволяє вирішувати такі завдання:

– точне вимірювання параметрів постачання/споживання;

– комерційний та технічний облік енергоресурсів по підприємству, його інфраструктурним елементам (котельня та об'єкти жилкомпобуту, цеху, підрозділи, субабоненти);

– контроль енергоспоживання за точками та об'єктами обліку в заданих часових інтервалах (30 хвилин, зони, зміни, доба, декади, місяці, квартали та роки) щодо заданих лімітів та технологічних обмежень потужності;

– обробка даних та формування звітів з обліку електроенергії;

– діагностика повноти даних;

– опис електричних з'єднань об'єктів та їх характеристик;

– діагностика лічильників;

– підтримка єдиного системного часу. [9]

Організації опитування лічильників персональним комп'ютером через перетворювач інтерфейсів, мультиплексор чи модем можна побачити на рисунку 1.3.

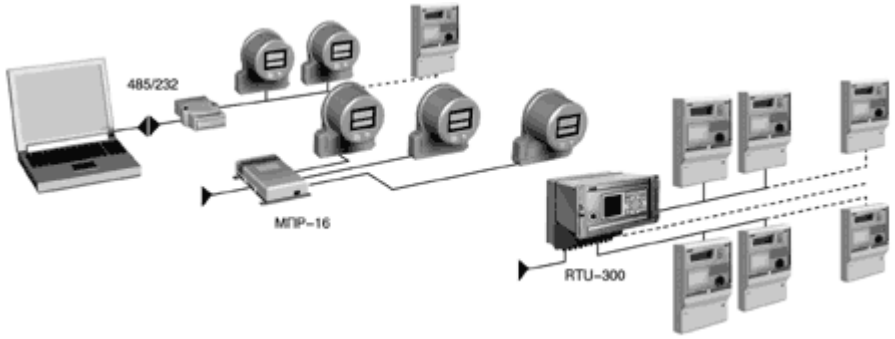

Рис. 1.3 – Організація опитування лічильників персональним комп'ютером через перетворювач інтерфейсів, мультиплексор чи модем

Організація АСКОЕ із проведенням автоматичного опитування лічильників локальним центром збору та обробки даних. Лічильники постійно пов'язані з центром збору даних прямими каналами зв'язку та опитуються відповідно до заданого розкладу опитування, зображено на рисунку 1.4.

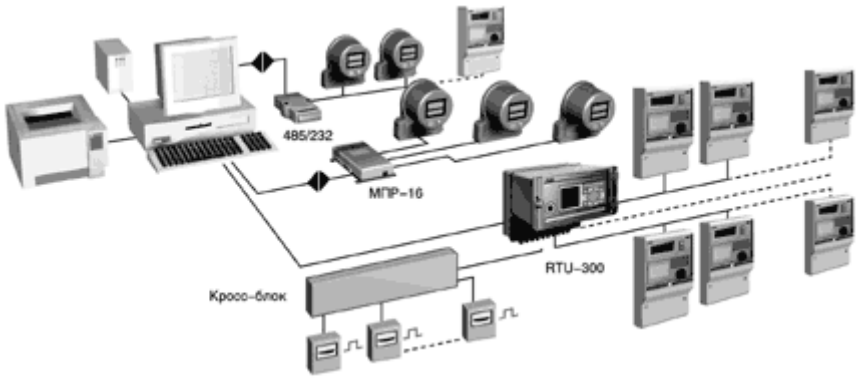

Рис. 1.4 – Організація автоматичного опитування лічильників локальним центром збору та обробки даних

Первинна інформація з лічильників записується у БД. Синхронізація часу лічильників відбувається в процесі опитування зі часом комп'ютера центру збору даних.

Організація АСКОЕ із проведенням автоматичного опитування лічильників локальним центром збору та обробки даних дозволяє вирішувати такі завдання:

– точне вимірювання параметрів постачання/споживання;

– комплексний автоматизований комерційний та технічний облік енергоресурсів по підприємству, його інфраструктурним елементам (котельня та об'єкти житлового побуту, цеху, підрозділи, субабоненти);

– контроль енергоспоживання та параметрів якості електроенергії (ПЯЕ) за точками та об'єктами обліку в заданих часових інтервалах (5 хвилин, 30 хвилин, зони, зміни, доба, декади, місяці, квартали та роки) щодо заданих лімітів та технологічних обмежень потужності;

– обробка даних та формування звітів з обліку електроенергії та контролю ПКЕ;

– фіксація відхилень контрольованих параметрів енергоресурсів, їх оцінка в абсолютних та відносних одиницях для аналізу як енергоспоживання, так і виробничих процесів;

– сигналізація (кольором, звуком) про відхилення контрольованих величин від допустимого діапазону значень;

– діагностика повноти даних;

– опис електричних з'єднань об'єктів та їх характеристик;

– параметризація комунікацій та характеристик опитування;

– діагностика системи;

– підтримка єдиного системного часу. [10]

Організація багаторівневої АСКОЕ для територіально розподіленого середнього та великого підприємства чи енергосистеми.

Основна частина лічильників постійно пов'язана із центрами збору даних першого рівня прямими каналами зв'язку та опитується відповідно до заданого розкладу опитування, як у третьому способі організації АСКОЕ. Між деякими лічильниками та центром збору даних першого рівня

може бути постійного зв'язку, вони можуть опитуватись за допомогою переносного комп'ютера, як у другому способі організації АСКОЕ. Первинна інформація з лічильників записується в БД центрів збору даних першого рівня, на них відбувається обробка даних. У центрах збору даних другого рівня здійснюється додаткове агрегування та структурування інформації, запис її до БД центрів збору даних другого рівня.

Організація багаторівневої АСКОЕ для територіально розподіленого середнього та великого підприємств або енергосистеми дозволяє вирішувати такі завдання:

– точне вимірювання параметрів постачання/споживання;

– комплексний автоматизований комерційний та технічний облік енергоресурсів по підприємству, його інфраструктурним елементам (котельня та об'єкти житлового побуту, цеху, підрозділи, субабоненти);

– ведення договорів та формування платіжних документів для розрахунків за електроенергію;

– контроль енергоспоживання та показників якості електроенергії за точками та об'єктами обліку в заданих часових інтервалах (5 хвилин, 30 хвилин, зони, зміни, доба, декади, місяці, квартали та роки) щодо заданих лімітів та технологічних обмежень потужності;

– супровід нормативно-довідкової інформації;

– обробка даних та формування звітів з обліку електроенергії та контролю ПКЕ;

– фіксація відхилень контрольованих параметрів енергоресурсів, їх оцінка в абсолютних та відносних одиницях для аналізу як енергоспоживання, так і виробничих процесів;

– сигналізація (кольором, звуком) про відхилення контрольованих величин від допустимого діапазону значень;

– діагностика повноти даних;

– опис електричних з'єднань об'єктів та їх характеристик;

– параметризація комунікацій та характеристик опитування;

– діагностика системи;

– підтримка єдиного системного часу. [11]

1.3 Програмне забезпечення АСКОЕ

Програмне забезпечення призначене для автоматизації збору даних від приладів (пристроїв), програмно-технічних комплексів з цифрових інтерфейсів, їх обробку, зберігання та передачу в інші програмно-технічні комплекси, пристрої або автоматизовані робочі місця та може застосовуватись в автоматизованих робочих системах обліку споживання електричної та теплової енергії, води чи газу. [12]

Загальні вимоги до програмного забезпечення АСКОЕ:

1 Для АСКОЕ повинен використовуватися програмний комплекс верхнього рівня та забезпечувати можливість встановлення його на комп'ютер у вигляді локальної або мережевої версії.

2 Програмний комплекс верхнього рівня системи обліку повинен реалізовувати функції збору даних обліку з її нижніх рівнів (рівнів лічильників), накопичення, зберігання, обробки, відображення, документування та розповсюдження цих даних, синхронізації годинників засобів обліку, а також інші функції, що залежать від вимог до конкретної системи обліку із боку замовника.

3 Програмний комплекс верхнього рівня повинен забезпечувати процедури:

– гнучкого налаштування як на вигляд запитуваних і збережених даних, так і за періодом та обсягом запитів;

– повного збору даних за автоматичними та ручними запитами з використанням додаткових запитів та мажоритарного голосування для вибору правильних даних;

– забезпечення зручного інтерфейсу користувача;

– контролю цілісності та достовірності даних обліку;

– автоматичного архівування даних у стандартних базах даних тощо.

4 У регіональній АСКОЕ необхідно забезпечити спільну роботу програмного комплексу верхнього рівня системи обліку з обраною стандартною базою даних під відповідною СУБД.

5 Програмний комплекс повинен використовувати єдині класифікатори об'єктів бази даних, дозволяти фіксувати заміну лічильників у точках обліку, задавати режими їхнього опитування, забезпечувати коректність даних і параметрів, що зчитуються з лічильників і розміщуються в базі, а також безперервність та повноту даних у базі.

6 Повинна бути забезпечена можливість перегляду бази даних за вибраними точками обліку, інтервалами часу та типами даних, а також можливість встановлення для кожної точки обліку при автоматичному опитуванні допустимого значення часу запізнення даних/параметрів, після перевищення якого має генерувати аварійне повідомлення.

7 У разі неможливості дистанційного зчитування даних з нижнього рівня системи обліку має бути передбачена можливість альтернативного зчитування та занесення даних до бази (наприклад, з переносного засобу приладового обліку). При планових та аварійних замінах лічильників має бути передбачена можливість санкціонованої ручної корекції бази даних з урахуванням часу відсутності приладового обліку у відповідній точці обліку.

8 Програмний комплекс повинен фіксувати всі події, що спотворюють функціонування системи (збій зв'язку, збій операційної системи або прикладної програми, несправність лічильника тощо), видавати вчасно відповідну інформацію адміністратору системи та генерувати добові та місячні зведені звіти про помилки. Повинна бути забезпечена можливість автоматичної та ручної корекції бази даних після різних збоїв.

9 Програмний комплекс має забезпечувати реєстрацію прав користувачів за рівнями доступу, ідентифікацію та аудит всіх їхніх дій. Залежно від рівня доступу користувачу має бути обмежена за рівнем кількість інформації.

10 Загальне програмне забезпечення зі складу програмного забезпечення комплексу має забезпечувати розробку та надійне функціонування прикладного програмного забезпечення в реальному часі та включати мережеву операційну систему та засоби автоматизованої розробки та системної підтримки виконання прикладного програмного забезпечення (сервісне програмне забезпечення).

11 Мережева операційна система повинна забезпечувати виконання функцій з диспетчеризації процесу розподіленої обробки даних у реальному часі, керування введенням-виводом даних, забезпечення мережевої підтримки.

12 Сервісне програмне забезпечення має являти собою інтегровану систему програмних продуктів, засновану на відкритій компонентній технології та містить два взаємопов'язані середовища: середовище розробки, середовище виконання та забезпечувати функціонування прикладного програмного забезпечення.

1.4 Зарубіжний досвід впровадження автоматизованої системи контролю та обліку електроенергії

За останні роки увагу фахівців енергетичної сфери все більше зосереджено на розвитку «розумних» або «інтелектуальних» мереж (Smart Grid). США, європейські держави, Китай (залежно від підприємства-розробника) їх називають відповідно: Future Grid, Empowered Grid, Wise Grid, Modern Grid, IntelliGrid. У більшості розвинених країн вживаються заходи щодо активізації впровадження складових таких електромереж у процесі розвитку енергосистем.

Електричні мережі нового покоління повинні на технологічному рівні об'єднувати споживачів і виробників електроенергії в єдину автоматизовану

систему, що дає можливість у реальному часі відслідковувати й контролювати режими роботи всіх учасників процесу виробництва, передавання і споживання електроенергії, в автоматичному режимі оперативно реагувати на зміни параметрів у енергосистемі та здійснювати електропостачання з необхідною надійністю та економічною ефективністю. Завдяки впровадженню сучасних технологій електрична мережа зможе залежно від ситуації змінювати свої характеристики, збільшуючи за рахунок автоматичного регулювання пропускну здатність і якість постачання електроенергії, особливо в умовах розвитку поновлюваних джерел енергії в енергосистемах.

Однією із складових «розумних» мереж (Smart Grid) є оснащення існуючих розподільних мереж сучасними автоматизованими пристроями обліку енергоспоживання, об'єднаними в єдину інформаційну мережу (система Smart Metering), що створює умови для ефективного використання енергоресурсів [14].

У зарубіжній практиці є яскраві приклади підвищення ефективності передавання електроенергії за допомогою нових технологій. Зокрема, компанія Siemens вирішила проблему перевантаженості електромереж Сан-Франциско за допомогою впровадження автоматизації системи HVDC (уставок постійного струму), що дало змогу відмовитись від спорудження нових генеруючих потужностей у центрі великого міста.

Розвиток технології «розумних» мереж викликаний не тільки вимогами підвищення енергоефективності та енергозбереження, але й необхідністю регулювання режимів роботи енергосистем в умовах зростання частки альтернативної енергетики в Європі та інших регіонах [15]. Основною особливістю поновлюваних джерел електроенергії, насамперед, вітряних і сонячних електростанцій, є нерівномірне та важкопрогнозоване надходження енергії в мережу. «Розумні» мережі оптимізують режими роботи енергосистем з великою кількістю малопотужних електростанцій із змінними в часі обсягами генерації. За оцінкою експертів впровадження технологій Smart Grid

дозволить знизити викиди парникових газів (CO2) до 2020 р. більш ніж на 1 млрд. тонн.

Технології Smart Grid мають розширену функціональну можливість переходу від управління за фактом аварійного порушення електропостачання до попередження пошкодження елементів мережі, що підвищує інформаційну та енергетичну безпеку. Нова концепція розвитку електромереж з урахуванням впровадження сучасних інформаційних технологій забезпечує необхідний рівень надійності і якості енергопостачання в різних цінових сегментах. Крім того, за допомогою цих технологій розвивається мотивація активності кінцевих споживачів до самостійного регулювання обсягу і характеристики електропостачання (обсяги енергоспоживання, рівень надійності, якості тощо) на підставі балансу своїх запитів і можливостей енергосистеми з використанням інформації про параметри цін, обсяги централізованої і місцевої генерації тощо. Впровадження зазначених технологій сприяє розширенню ринків електроенергії та потужності з включенням у їхню діяльність кінцевих споживачів і наданням їм відкритого доступу на ринки і до систем генерації, що сприяє підвищенню результативності та ефективності роздрібного енергетичного ринку.

Сьогодні у світовій енергетичній сфері реалізується ряд проектів на впровадження однієї з концептуальних технологій Smart Grid – систем Smart Metering (інтелектуального обліку) [16]. Зокрема, Нова Зеландія вже повністю перейшла на 100% використання «розумних» приладів обліку електроенергії, Австралія, Китай у свою чергу планують довести до 100% оснащення споживачів «розумними» приладами обліку в найближчі три роки. У Великобританії на державному рівні прийнято Кодекс балансування та розрахунків у енергетичній сфері країни. Відповідно до цього компанія ELEXON (дочірня компанія оператора системи магістральних мереж National Grid Elexon) здійснює адміністрування механізмів балансування і розрахунків на ринку електричної енергії країни. У більшості країн Євросоюзу

реалізуються Програми розвитку і впровадження сучасних автоматизованих систем обліку та інших складових «інтелектуальних мереж».

Заслуговують особливої уваги плани впровадження систем «розумного» обліку в різних регіонах світу. Зокрема, досвід Євросоюзу, який щорічно спрямовує понад 1,2 млрд євро на дослідження у сфері розвитку інтелектуальних мереж і систем обліку.

1.5 Впровадження автоматизованої системи контролю та обліку електроенергії в Україні

Однією з найважливіших проблем, що стоять перед енергетикою України, є організація точного і достовірного комерційного обліку електроенергії та потужності на оптовому і роздрібному ринках електроенергії. Впровадження ринкових відносин потребує якісно нових технічних і програмних засобів обліку, систем контролю, передавання та оброблення інформації [17].

В Україні реформування Оптового ринку електричної енергії (ОРЕ) пов'язане з упровадженням моделі ринку двосторонніх договорів, балансуючого ринку електроенергії та на подальшому етапі – біржі електроенергії. Реформований (лібералізований) оптовий ринок електроенергії не зможе ефективно працювати без запровадження автоматизованої системи комерційного обліку і контролю виробництва, постачання і споживання електроенергії в реальному масштабі часу.

Необхідність невідкладного завершення роботи по впровадженню АСКОЕ на ОРЕ України зумовлено значним небалансом обліку виробленої та спожитої електроенергії, що спричиняє додаткові втрати електроенергії. Це зумовлено також тим, що значну кількість точок обліку електроенергії до впровадження сучасних автоматизованих систем обліку електроенергії було оснащено різними за типами і класами точності приладами обліку, більша половина яких застаріла морально і фізично. Крім цього майже 40% точок обліку електроенергії не мало дублюючих приладів.

Відповідно до Угоди між Урядом України та Європейською Комісією щодо покращення системи обліку на українському ОРЕ передбачено впровадження проекту «Створення автоматизованої системи обліку (АСОЕ) для Оптового ринку електроенергії». Реалізація зазначеного проекту створить умови для:

– забезпечення комерційного обліку активної та реактивної електроенергії відповідно до вимог ОРЕ щодо погодинного обліку електроенергії на межі балансової належності суб'єктів ринку електроенергії (генеруючі компанії, оператор магістральних і міждержавних електромереж і обленерго);

– підвищення точності та достовірності комерційного обліку електроенергії серед суб'єктів ОРЕ;

– підвищення оперативності управління режимами виробленої, переданої та відпущеної електроенергії, а також удосконалення комерційних розрахунків на ОРЕ;

– зниження витрат на передавання і постачання електроенергії за рахунок підвищення точності обліку та максимальному скороченню нетехнічних (комерційних) втрат. Система обліку та її окремі компоненти повинні відповідати вимогам міжнародної та української сертифікації для здійснення комерційних розрахунків.

Відповідно до постанов НКРЕ України від 15.07.2010 р. №№ 815 - 820 про внесення змін до Умов і Правил здійснення підприємницької діяльності з виробництва та постачання електричної енергії купівля-продаж електроенергії на ОРЕ має здійснюватися з використанням даних, отриманих із АСКОЕ головного оператора та суб'єктів оптового ринку електроенергії.

Відповідно до цього ДП «Енергоринок» України та суб'єкти ОРЕ зобов'язані привести свою діяльність у відповідність до вимог зазначених постанов НКРЕ. З метою забезпечення підготовки до розрахунків з купівлі-продажу електроенергії в ОРЕ з використанням даних, отриманих із АСКОЕ,

станом на кінець першого кварталу 2011 р. проведено значну роботу щодо впровадження автоматизованої системи комерційного обліку електроенергії.

Із врахуванням методики складання структури балансу електроенергії в електричних мережах 0,38 – 150 кВ, аналізу його складових і нормування технологічних витрат електроенергії потребує нагального перегляду та уточнення, насамперед щодо розрахунку нормативів витрат електроенергії в електромережах обласних енергопостачальних компаній, з метою стимулювання їх до активної роботи з реального зниження, перш за все, комерційних втрат електроенергії як за рахунок підвищення рівня організації енергозбутової роботи, так і впровадження сучасних систем обліку спожитої електроенергії [18].

1.6 Постановка задачі

Проведений аналіз стану проблема дозволяє сформулювати мету досліджень, визначити напрямок і завдання магістерської роботи.

Метою даної роботи є підвищення швидкості збору інформації про споживання електроенергії шляхом розробки системи обліку та аналізу споживання електроенергії в домогосподарствах на основі збору та аналізу інформації.

Система має бути спроектована таким чином, щоб у майбутньому інженери з розробки мали можливість легко підтримувати, повторно використовувати та розширювати функціональні дані, якщо в ході розробки продукту будуть раптово змінюватися вимоги або функціонал, і потрібно буде швидко змінювати програмний код. Організація АСКОЕ буде реалізовуватися із проведенням автоматичного опитування лічильників локальним центром збору та обробки даних, про варіанти організації описано у пункті 1.2. Програмне забезпечення має відповідати більшості пунктів з загальних вимог до програмного забезпечення АСКОЕ, що описані у пункті 1.3.

Висновки по розділу

У цьому розділі проведено науково-технічне обґрунтування розробки моделей та методів автоматизованого системного контролю та обліку енергоресурсів . На основі проведеного огляду підходів до побудови систем, для подальшого детального аналізу та побудови математичних моделей обрана реалізація із проведенням автоматичного опитування лічильників локальним центром збору та обробки даних,. На основі проведеного у пункті 1.6 опису об’єкту, створене технічне завдання, яке наведене в додатку А.

2 РОЗРОБКА МАТЕМАТИЧНИХ МОДЕЛЕЙ СПОЖИВАННЯ ЕЛЕКТРОЕНЕРГІЇ В ДОМОГОСПОДАРСТВАХ

2.1 Розрахунок споживання

Дані, отримані від розумних лічильників, для більшості лічильників є лише кількістю ват на одиницю часу. Для розрахунку кВт-год на одиницю часу потрібно використати наведену нижче формулу.

$$kWhPerSample = kwatt * kwhSampleConstant \quad (2.1)$$

kwhSampleConstant є константою, що залежить від кількості зразків для отриманих даних.

$$kwhSampleConstant = \frac{sampleLenght}{3600} \quad (2.2)$$

SampleLength — це час у секундах між кожним отриманим зразком. Наприклад, це може бути 10 секунд, і тоді *kWhSampleConstant* буде:

$$kwhSampleConstant = \frac{10}{3600} = 0.00278 \quad (2.3)$$

Коли лічильник надає значення кВт, а користувач не запитує дані, ми зберігаємо лише позначку часу та значення кВт.

Коли користувач запитує дані, сервіс починає обчислювати значення кВт-год, а також витрати на кВт-год. Приклади наведено на рисунку 2.1.

На рисунку 2.2 нижче ми бачимо приклад двох хвилин даних для Wh і витрати на Wh.

Timestamp	Reading of WattPerSample
2021-06-23T00:00:00	1000
2021-06-23T00:00:10	1400
2021-06-23T00:00:20	1300
2021-06-23T00:00:30	1200
2021-06-23T00:00:40	1200
2021-06-23T00:00:50	1300
2021-06-23T00:01:00	4000
2021-06-23T00:01:10	4000
2021-06-23T00:01:20	3800
2021-06-23T00:01:30	3200
2021-06-23T00:01:40	2400
2021-06-23T00:01:50	2400

Рис. 2.1 – отримані дані кВт-год за дві хвилини

Timestamp	WhPerSample	WhAccumulated	CostsPerSample	CostsAccumulated
2021-06-23T00:00:00	2.78	2.78	0.00217	0.00217
2021-06-23T00:00:10	3.89	6.67	0.00303	0.00520
2021-06-23T00:00:20	3.61	10.28	0.00282	0.00802
2021-06-23T00:00:30	3.33	13.61	0.00260	0.01062
2021-06-23T00:00:40	3.33	16.94	0.00260	0.01322
2021-06-23T00:00:50	3.61	20.56	0.00282	0.01603
2021-06-23T00:01:00	11.11	31.67	0.00867	0.02470
2021-06-23T00:01:10	11.11	42.78	0.00867	0.03337
2021-06-23T00:01:20	10.56	53.33	0.00823	0.04160
2021-06-23T00:01:30	8.89	62.22	0.00693	0.04853
2021-06-23T00:01:40	6.67	68.89	0.00520	0.05373
2021-06-23T00:01:50	6.67	75.56	0.00520	0.05893

Рис. 2.2 – отримані дані та ціни за дві хвилини

2.2 Відображення витрат у реальному часі

Представлення даних включає поточне значння кВт, а також ціну за кВт-год. Для отримання витрат у реальномі часі (*kWPriceNow*) поточне споживання (кВт) множиться на поточне значення (totalPerKwh) з додаванням ціни на годину (*totalPerSample*): *(Power_used_now (кВт) * totalPerKwh) + totalPerSample = kWPriceNow*

Дані про ціни надходять у форматі JSON:

{

"sample": "0",

"spotPerKwh": "0.34",

"powerPerKwh": "0.50",

"powerPerSample": "0.22",

"gridPerKwh" : "0.40",

"gridPerSample" : "0.33",

"extraPerKwh" : 0.00,

"totalPerKwh" : "0.9" (comments: "powerPerKwh": "0.50" + "gridPerKwh" : "0.40")

"totalPerSample" : "0.55" (comments: "powerPerSample": "0.22" + "gridPerSample" : "0.33")

}

2.3 Обмеження споживання

Рівні потужності використовуються мережевими компаніями для зменшення піків споживання електроенергії. Такі обмеження споживання електроенергії типові для скандинавських країн. У наступних прикладах будуть описані варіанти використання у Норвегії.

Коли користувач починає новий місяць, він перебуває на першому рівні потужності. Рівень потужності – це середнє значення найвищого максимального значення кВт/год за три дні в місяці. Наприклад, 8 кВт/год, 10 кВт/год і 9 кВт/год тощо. Кожен рівень потужності має відповідну ціну за місяць, яку клієнт повинен платити. Приклади рівнів потужності та цін за кожен рівень споживання наведено на рисунку 2.3.

Levels	Capacity (kWh per hour)	Price (NOK per month)
1	0-5	250
2	5-10	350
3	10-15	475
4	15-20	625
5	20-25	750

Рис. 2.3 – рівні споживання на ціни

Приклад обрахунку по трьом дням максимального споживання:

Day1 - max1: 14 кВт/год

Day2 - max2: 5 кВт/год

Day3 - max3: 10 кВт/год

Тоді середнє значення дорівнює (14 + 5 + 10) / 3 = 9,6667

Якщо, наприклад, max2 стає 6, клієнт перейде на наступний рівень потужності, коли середнє становитиме 10,333. Розрахунки трьох рівнів розділені на дві основні частини:

1 Розрахунок середнього значення кВт/год із трьох максимальних значень. Це називається *AbsoluteMax*.

2 Розрахунок кількості кВт (енергії (кВт·год) за хвилину), яку можна використати за хвилину протягом однієї години.

Користувач повинен бути проінформований, скільки енергії можна використати протягом наступної години, щоб користувач не перейшов на наступний рівень потужності, а потім не повинен був платити більше мережевій компанії. Тому вводиться дві основні ключові змінні: *MaxAbsolute* і *MaxRecommended*.

MaxAbsolute — це максимальна кількість кВт-год, яку користувач може використати протягом години, щоб не переходити до наступного рівня потужності.

MaxRecommended — максимальний рекомендований показник для години. Причина створення цього значення полягає в тому, що необхідно

давати користувачеві нижню межу для max-споживання, яка вказує те, що максимальна встановлена, наприклад, на початку місяця.

$$\left(\frac{max1+max2+max3}{3}\right) = CapacityLevel \quad (2.4)$$

Якщо *max1* є найнижчим значенням, тоді, щоб не перейти до наступного рівня ємності, новий *max1* (*MaxAbsolute*) може бути таким, як показано у формулі нижче:

$$(CapacityLevel * 3) - max2 - max3 = MaxAbsolute \quad (2.5)$$

MaxRecommended — це максимальне значення, яке система рекомендує використовувати сьогодні. MaxAbsolute повідомляє про максимальне значення, яке має залишатися нижче поточного рівня потужності, але якщо був пік (наприклад, високе значення для Max1) на початку місяця, то у буде менше енергії, якою можна варіювати до кінця місяця. Через це вводиться MaxRecommended, який є фактором ємності, яку ми повинні використовувати, і кількості днів, що залишилися в місяці. Вводиться константуа CapacityToSave у формулу для MaxRecommendation, а також використовуємо кількість днів, що залишилися в місяці, як фактор.

Наприклад якщо dayNumberInMonth дорівнює 2 (другий день у місяці), варто використовувати це, щоб сказати, скільки можна використати ємності для дня в місяці.

$$MaxAbsolute * (1 - CapacityToSave) + \frac{MaxAbsolute*CapacityToSave*DayInMonth}{AllDaysInMonth} \quad (2.6)$$

З розрахунків рівня потужності є MaxAbsolute і MaxRecommended. Це два значення, які допомагають встановити MaxHourAbsolute, MaxHourAbsoluteOffset і MaxHourRecommended.

MaxHourAbsolute обчислюється, знаючи, скільки енергії можна використати за кожну хвилину протягом години. Він використовує вхідні дані для обчислення.

MaxHourAbsolute полягає в тому, щоб використовувати MaxAbsolute і зменшити його на накопичену енергію, використану до цього моменту за цю годину.

$$MaxHourAbsolute = \frac{kWh*60}{hour-MinutesLeft} \quad (2.7)$$

Для поточної години обчислюється MaxHourAbsoluteOffset шляхом множення MaxHourAbsolute на постійний MaxAbsoluteOffsetPercent.

$$MaxHourAbsoluteOffset = MaxHourAbsolute * MaxAbsoluteOffsetPercent \quad (2.8)$$

MaxHourRecommended використовується, щоб дати користувачеві рекомендацію щодо того, скільки енергії слід використовувати за кожну хвилину протягом години. MaxHourRecommended обчислюється шляхом множення MaxHourAbsolute на MaxRecommended Percent.

$$MaxHourRecommended = MaxHourAbsolute * MaxRecommendedPercent \quad (2.9)$$

2.4 Формування вимог до розроблюваної системи

Ціллю розробки є ведення автоматизації процесу обліку споживання електроенергії та спрощення процесу отримання вихідних даних у вигляді виконаного звіту, обґрунтування доцільності використання, виявлення сильних і слабких сторін такого рішення.

Для виконання поданих вимог необхідно вирішити такі задачі:

– обрати спосіб передачі даних від користувача до системи;

– описати спосіб підрахунку та збереження даних споживання;

– обрати інструменти для реалізації автоматизованої системи обліку споживання електроенергії;

– створити гнучкий додаток для обліку споживання електроенергії із подальшою можливістю корегування, розширення та удосконалення проекту;

– створити набір тестів для перевірки функціоналу;

Висновки до розділу

В даному розділі сформовано вимоги до розроблюваної системи автоматизованого обліку даних енергоносіїв, проведено опис математичний моделей споживання електроенергії в домогосподарствах, які покращать контроль споживання та будуть надавати дані користувачам у реальному часі.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СПОЖИВАННЯ ЕЛЕКТРОЕНЕРГІЇ В ДОМОГОСПОДАРСТВАХ

3.1 Основні функціональні вимоги програмного забезпечення

Для розробки програмного забезпечення споживання електроенергії у домогосподарствах необхідно обрати структуру системи управління, стандарт комунікації між пристроями та сервісом, метод серіалізації даних; обґрунтувати вибір програмних засобів для створення програмного забезпечення; створити UML-діаграми основних компонентів та відобразити залежності між ними; організувати структуру таблиць у базі даних; розробити тести для перевірки коректної роботи програмного забезпечення.

3.2 Структура сучасної системи управління

Широка різноманітність апаратного та програмного забезпечення дозволяє створити системи управління, структура і функції яких оптимально підібрані. Аналізуючи доступні рішення, можна виділити кілька груп таких систем:

– невеликі системи управління, оснащені центральним блоком з низькою обчислювальною потужністю та локальними модулями введення/виведення

– системи середньої ефективності оснащені промисловими стандартами, що ґрунтуються на комунікаційних шинах, що підтримують віддалені модулі введення/виведення,

– системи великої продуктивності, складені як набір розподілені контролери та модулі введення/виведення.[19]

Крім того, важливі елементи сучасного контролю — це системи візуалізації процесів, які передають дані з контролерів безпосередньо через посередників технології. Коли необхідно тестувати системи в змодельованих

умовах або для оптимізації процесів обміну даними між контролерами та програмним забезпеченням, що використовується для моделювання або оптимізації [20].

Враховуючи це, очевидно, що створити альтернативи сучасним, але запатентованим системам управління використовуючи відкриті технології, необхідно забезпечити різні апаратні та програмні технології.

3.3 Відкриті стандарти комунікації

Сучасні системи керування зазвичай складаються з кількох пристроїв які обмінюються даними. У випадку зв'язку між пристроями побудованими за різними технологіями та що постачаються різними виробниками, необхідно використовувати відкриті стандарти обміну даними. Ця особливість є головною при спробі створити систему управління на основі на відкритих технологій. [21]

Найбільш використовувані відкриті стандарти, для яких є реалізації з відкритим кодом, включають:

- Modbus
- OPC
- OPC UA
- MQTT
- DDS/RTPS
- POWERLINK

Modbus є одним із найпопулярніших протоколів обміну даними в системах автоматизації. Це реалізовано в більшості комерційно доступних контролерах PLC . Існує також багато варіацій з відкритим кодом на цей протокол.

OPC — це відкрита комунікаційна платформа на основі стандартів Windows COM (Component Object Model) і DCOM (distributed COM). Розробкою стандарту OPC займається OPC Foundation. OPC вимагає тільки

один сервер для інтеграції великих промислових систем управління з використанням різних протоколів зв'язку.

У разі створення систем з використанням цього стандарту зв'язку, обмеженням є необхідність використання операційної системи Windows. Є також відносно невелика кількість відкритих бібліотек, і вони зазвичай реалізують лише специфікації доступу до даних OPC. Приклади таких бібліотек це LightOPC, OpenOPC і PyOPC.

OPC UA (Undefined Architecture) – це покращений стандарт обміну, розроблений OPC Foundation. OPC UA базується на універсальних та незалежних від платформи комунікаційних стандартах, таких як:

– TCP/IP
– HTTP
– SOAP

Протокол може працювати на різних операційних системах, на відміну від OPC, для якого потрібна Windows.

MQTT — це спрощений протокол передачі даних за принципом публікації-підписки. Перевага цього протокол — це простота та механізм, який використовується для обміну даними, що дозволяє знизити витрати на мережу. Головна частина комунікаційної системи вимагає спеціального додатка під назвою брокер повідомлень, який можна порівняти з сервером. Найпопулярнішими брокерами є:

– Mosquitto,
– RabbitMQ
– HiveMQ
– Brocker Erlang MQTT.

Кожне повідомлення MQTT має містити тему, яку брокер може використовувати для оновлення вже існуючої теми або для пересилки повідомлення клієнтам, підписаним на певну тему. Клієнти MQTT повідомляють брокеру назву теми, на яку вони хочуть підписатися. Нові дані, які потребують брокера повідомлення надсилаються клієнтами брокеру.

Брокерська програма надсилає повідомлення всім підписаним клієнтам.[22] Обмін даними показано на рис. 3.1.

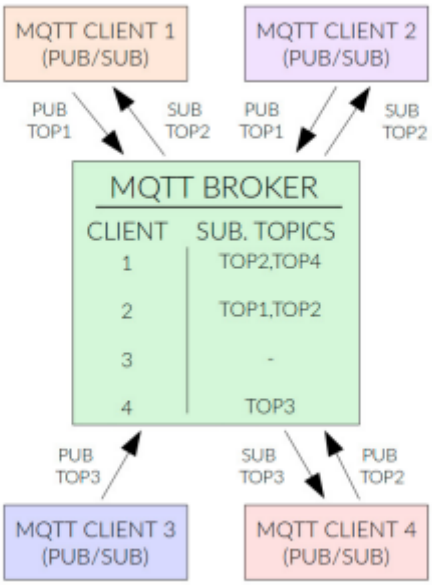

Рис 3.1 — Обмін даними MQTT брокера

Практично, MQTT — простий протокол обміну даними, що не вимагає налаштування. Недолік розробок на основі цього протоколу є спрощені механізми безпеки. [23]

У випадку більш складних і вимогливих систем необхідно використовувати більш складні протоколи.

DDS (Data Distribution Service) — це OMG (Object Management Group), оптимізований для високопродуктивного обміну даними в реальному часі. Він реалізує шаблон publish/subscribe для надсилання та отримання даних, подій і команд між вузлами. Це забезпечує відкритий механізм для видавців і передплатників, що означає що програма не повинна знати заздалегідь (тобто під час проектування або компіляції) усі вузли, оскільки вони автоматично виявляються DDS під час виконання.

Протокол RTPS (Real Time Publish Subscribe Protocol) забезпечує сумісність реалізацій DDS і був стандартизований OMG. RTPS протокол розроблено для роботи через ненадійні транспортні

протоколи, такі як UDP. Протокол забезпечує надійність і сприяє детермінізму. Це надає багатоадресні можливості транспортного механізму, де одне повідомлення від відправника може досягти кількох одержувачів.[24]

В основі нового протоколу RTPS лежить версія ROS (Robots Operating System) [25].

ROS може бути успішно використаний у передових системах управління та проектах IoT. Якщо необхідно оснастити систему управління приладами, що працюють у режимі реального часу та використовують промисловий Ethernet, доступним відкритим рішенням є протокол POWERLINK. Це єдине відкрите стандартне програмне забезпечення рішення для промислового Ethernet, яке гарантує реальний час операцій. На відміну від інших промислових стандартів Ethernet, таких як EtherCAT і Profinet, він не вимагає жодних спеціальні компоненти, відмінні від стандартних мережевих адаптерів працювати. OpenPOWERLINK — це реалізація протоколу POWERLINK з відкритим кодом, що поширюється під BSD ліцензія. [26]

RabbitMQ — це легкий і надзвичайно потужний інструмент для створення архітектур розподіленого програмного забезпечення, які варіюються від дуже простих до неймовірно складних. RabbitMQ забезпечує надзвичайну гнучкість у підході та вирішенні проблем. Як проект із відкритим вихідним кодом, написаний мовою Erlang, RabbitMQ відрізняється свободою та гнучкістю, одночасно використовуючи потужність Pivotal як продукту, що стоїть за ним. Розробники та інженери спільноти RabbitMQ можуть вносити вдосконалення та доповнення, а Pivotal може запропонувати комерційну підтримку та стабільну базу для постійного вдосконалення продукту.

RabbitMQ забезпечує гнучкість у контролі керування надійного обміну повідомленнями з пропускною здатністю та продуктивністю. Оскільки це не універсальний тип програми, у повідомленнях можна вказати, чи потрібно їх зберігати на диску перед доставкою, і, якщо це налаштовано в кластері, черги

можна налаштувати як високодоступні, охоплюючи кілька серверів, що буде гарантувати, що повідомлення не будуть втрачені у разі збою сервера.

Оскільки не всі топології та архітектури мережі однакові, RabbitMQ забезпечує обмін повідомленнями в середовищах із низькою затримкою та підтримує плагіни для середовищ із більшою затримкою. Це дозволяє кластеризувати RabbitMQ в одній локальній мережі та обмінюватися об’єднаними повідомленнями в кількох центрах обробки даних.

RabbitMQ надає гнучку систему плагінів. Наприклад, існують плагіни сторонніх розробників для зберігання повідомлень безпосередньо в базах даних, використовуючи RabbitMQ безпосередньо для запису в базу даних.

У RabbitMQ безпека забезпечується кількома рівнями. Клієнтські з’єднання можуть бути захищені за допомогою зв’язку лише за протоколом SSL і перевірки сертифіката клієнта. Доступом користувачів можна керувати на рівні віртуального хосту, забезпечуючи ізоляцію повідомлень і ресурсів на високому рівні. [27]

3.4 Серіалізація даних

При використанні простих протоколів, таких як MQTT, і передачі великої кількісті тем, є значні накладні витрати пов'язані з використанням протоколу TCP/IP. Дані методи серіалізації можна використовувати для підвищення ефективності обміну даними та обмеження кількості тем. Найпопулярніші методи серіалізації:

– XML;
– JSON;
– MessagePack;

Усі ці методи дозволяють обмінюватися даними між різними платформами, але вони мають різну ефективність. Серед перерахованих методів найпростіший у використанні і в той же час одним з найефективніших є JSON. Він дозволяє обмінюватися даними між сервісами з кількома мовами програмування, він невеликий і швидкий. [28]

3.5 Збереження даних

Для збереження даних з лічильників яку поступають з брокера, була обрана нереляційна база даних InfluxDB. Основною перевагою цієї бази даних є швидке вичитування даних по будь-якому часу запису. Це реалізується завдяки можливості Time Series Databases.

InfluxDB — це безсхемна база даних часових рядів із відкритим вихідним кодом і необов'язковими компонентами із закритим кодом InfluxData. Вона написана на мові програмування Go та оптимізованиа для обробки даних часових рядів. [29]

Можна впевнено виділити переваги InfluxDB:

1 Спеціально для даних часових рядів: InfluxDB розроблено для ефективної обробки даних часових рядів. Він створений для забезпечення вражаючої пропускної здатності запису та читання.

2 Рішення для будь-яких потреб: InfluxData забезпечує відкритий вихідний код, що включає InfluxDB, Kapacitor, Telegraf і Chronograf. Він також надає рішення SaaS, а саме InfluxCloud, який пропонує високу доступність, масштабованість і розширені можливості функції резервного копіювання та відновлення для користувачів із більшими потребами та обмежена інфраструктура.

3 Розширена підтримка мов програмування: InfluxDB пропонує підтримку різних мов програмування, включаючи, але не обмежуючись: .Net, Java, Perl, PHP, Python, R, Ruby, Scala тощо.

4 SQL-подібна мова запитів: InfluxDB поставляється з SQL-подібним запитом мова InfluxQL, що означає, що її легше писати.

5 Автоматичне закінчення терміну дії: дані часових рядів можуть стати менш релевантними з часом. InfluxDB з використання політик збереження, увімкне автоматичне закінчення терміну дії застарілих даних.

6 Вбудований інтерфейс веб-адміністратора: як і SQL, InfluxDB надає вбудований онлайн-інтерфейс для користувачів, яким не подобається

інтерфейси командного рядка та віддають перевагу більш інтуїтивно зрозумілому рішенню.

7. Розширена документація: InfluxData надає розширений посібник з документації для InfluxDB від встановлення до складних запитів і оптимізація схеми. [30]

3.6 Вибір фрейморка для реалізації серверної частини

Серед найпопулярніших фреймворків серверної розробки, станом на останні роки, можна виділити Laravel на мові PHP, Express.js на мові JavaScript (платформа Node.js), ASP.NET на мові C#, Ruby on Rails на мові Ruby, фреймворки Django та Flask на мові Python та Spring Boot на мові Java (та Kotlin яка базується на Java).[31] Безумовно кожен з них має свої переваги та недоліки які виражаються у особливостях мов програмування, на яких вони написані, та у різних можливостях, які вони пропонують розробникам для швидкої та зручної розробки.

Spring Boot – це програмний каркас, який, в свою чергу, побудований на платформі-фреймворку Spring. Однією з головних властивостей Spring Boot є те, що в ньому вже є все готове для дії «просто запусти». Тобто, перевагою Spring Boot є те, що він налаштований на те, щоб користувачу було максимально легко сконфігурувати своє застосування. В ньому вже є вбудовані сервери: широко відомий Tomcat а також інші – і Jetty та GlassFish. «З коробки» в Spring Boot легко додаються так звані залежності-стартери, тобто надбудови над бібліотеками та збірками бібліотек, які дуже часто досить просто додати в список залежностей – і вони вже працюють, маючи автоконфігурації, які, звісно ж, можна перевизначити на власний розсуд та манер. [32]

3.7 Опис середовища системи

Майбутнє за цифровими даними, і споживачі все більше хочуть контролювати власні дані, щоб вони могли адаптувати свої моделі

використання. В останні роки інтерес до ознайомлення з детальною інформацією про власне споживання електроенергії в домогосподарствах зріс. Використовуючи розумні пристрої можна дізнатися більше про власну поведінку та отримати контроль над власним споживанням електроенергії. PowerLink можна легко підключити до лічильника та отримувати детальну інформацію в режимі реального часу.

PowerLink використовує різні технології бездротового зв’язку, що забезпечує безперебійну роботу, чудове покриття, легке підключення та моніторинг стану споживання, також PowerLink отримує дані в реальному часі з усіх типів автоматичних лічильників електроенергії.

Від розумного пристрою PowerLink дані надсилаються по протоколу передачі MQTT на сервер підрахунку статистики споживання.

В основі MQTT лежить проста ідея перенесення ресурсозатратної частини системи на один елемент, у якого ці ресурси є. При цьому всі інші учасники системи майже повністю звільнені від роботи, що дозволяє їм економити ресурси. Реалізовано це наступним чином: пристрої, які хочуть передати повідомлення (під назвою видавці в термінах MQTT), відправляють його серверу (брокеру), пристрої які хочуть отримати повідомлення (передплатники), також підключаються до брокера і отримують повідомлення від нього. Всю обробку повідомлень бере на себе MQTT- брокер. Даний підхід показав себе дуже добре і ефективно в таких завданнях, як, наприклад, збір даних, телеметрія і управління простими пристроями. [33]

Схематичне зображення середовища системи зображено на рисунку 3.2

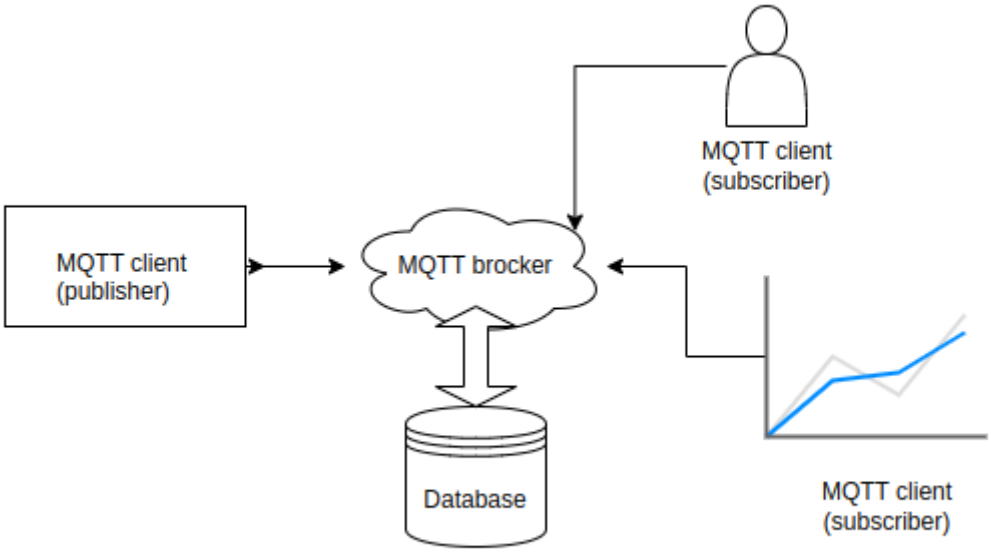

Рис 3.2 — Схематичне зображення середовища системи

Для розробки серверної частини обрано фрейморк програмування Spring Boot, нереляційна база даних InfluxDB, MQTT-брокер RabbitMQ, переваги обраних інструментів описані у попередніх підпунктах.

3.8 Серверна обробка даних

Сервер має підписатися на певну чергу MQTT-брокера, щоб забезпечити отримання даних. Після отримання даних сервер починає обробку, збереження та обрахунок споживання користувача.

Обробка відбувається в декілька етапів. Спочатку відбувається виокремлиння корисної інформації з потоку даних, що передає розумний пристрій у чергу. Поточні показники потужності енергоспоживання сервер зберігає в базу даних. На наступному кроці сервер обраховує активне споживання електроенергії за поточну хвилину та зберігає дані в іншу таблицю бази даних. Цей процес має повторюватися кожні 15 секунд для поточної хвилини для забезпечення точності результатів. Кожну годину отримуючи реальні показники лічильника, проводиться обрахунок споживання електроенергії за годину. Паралельно з цими процесами

відбувається поступове накопичення даних та обрахунок споживання за день. Маючи сконфігурований параметр та актуальну інформацію по тарифам споживання електроенергії проводиться обрахунок споживання за кожну хвилину, годину та день.

3.9 Розробка UML–діаграми класів сутностей системи

Для перетворення даних, що отримані з лічильників користувачів, за об’єктно-орієнтованими принципами, необхідно створити таки класи у програмі: DailyConsumption, HourlyConsumption, MinMaxConsumption, MinuteConsumption, RegularConsumption. Кожна сутність містить дані про конкретний тип споживання користувачів, відповідно до назви. Сутності мають спільні поля (sourceSystem, value, time, locationId) значення яких позначають основні дані для користувацької інформації. За допопомою анотації @Measurement бібліотеки influxdb.annotation для кожної сутності створюється таблиця у базі даних з відповідною назвою.

На рисунку Додатку Б.1 зображені сутності системи, які обгортають клієнтські дані у класи програмного середовища.

3.10 Розробка UML–діаграми основних функціональних класів системи

Архітектура серверної частини системи виглядає як трьохрівнева ієрархія прошарків: Presentation — Service — Persistence. У прошарку контролерів відбувається спілкування з клієнтами через RestApi, також там знаходяться класи, що прослуховують черги MQTT-брокера. У прошарку сервісів відбуваються транзакційні події запису або зчитування даних та математичні обрахунки. Прошарок репозиторіїв відповідає за взаємодію з базою даних, тобто перетворення даних з класів-сутностей у табличні дані.

У системі можна виділити основний клас DataService. У ньому описана реалізація більшості ключових методів, таких як: отримання та запис даних останнього або поточного споживання, отримання історії споживань, розрахунок вартості користування за годину/день/місяць тощо.

Для реалізації роботи клас DataService має залежності з класами репозиторіями, обрахунку кількості споживання та цін і прошарком контролерів. Детальніше з функціональністю класу можна ознайомитися у лістингу програми Додатку Г. UML–діаграма основних функціональних класів системи зображена на рисунку Додатку Б.2.

Основні репозиторії програми поділяються на дві частини, репозиторій, що віддає дані, та той що обробляє біль складні запити зображено на рисунках Додатку Б.3 та Б.4 відповідно.

3.11 Тестування програмного забезпечення

Для тестування програмного забезпечення використовувалися модульні та інтеграційні тести, які покривають функціональні сценарії. Тести перевіряють як і тривіальні випадки використання функціоналу, так і особливі, що важливі для коректного функціонування програмного забезпечееня. Приклади тестів для основному класу DataService зображені на рисунку Додатку Б.5.

3.12 Інтерфейс користувача

Серверна частина системи надає API, яким користується мобільний клієнтський додаток. На інтерфейсі мобільного додатку відображається поточні значення споживання енергії, кількість споживчої енергії протягом доби та вартість. Перелічені показники у користувацькому додатку зображені на рисунку 3.3.

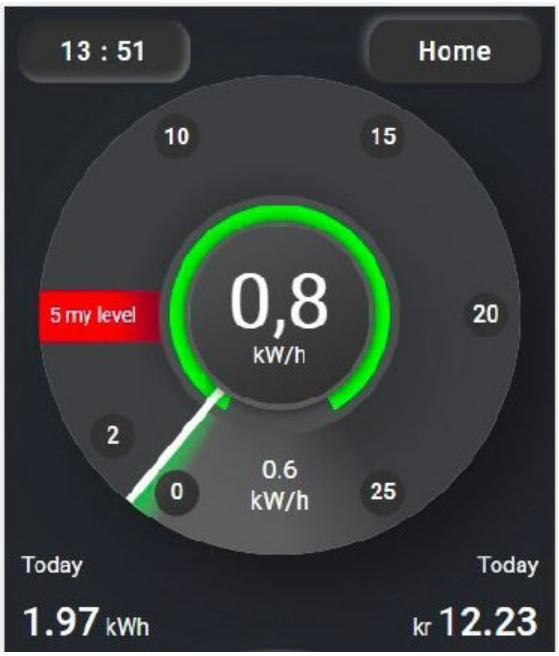

Рис 3.3 — Інтерфейс користувача

Висновки по розділу

Отже у даному розділі було проведено розробку програмного забезпечення системи обліку та аналізу споживання електроенергії. Було розроблено функціональні моделі системи, яку побудовано на основі математичних моделей з попереднього розділу, проведено її тестування та відлагодження.

Особливістю розробленого програмного продукту є можливість отримання інформації про споживання електроенергії у домогосподарствах з врахуванням встановлених обмежень на споживання електроенергії. Розробка також дає можливість передавати дані про споживання по відкритому протоколу MQTT для подальшої інтеграції з сторонніми системами що забезпечує можливість розширення функціоналу сторонніми розробниками.

У розділі описані функціональні вимоги до програмного забезпечення. Результати тестування довели працездатність системи.

4. ЕКОНОМІЧНИЙ РОЗДІЛ

4.1 Технологічний аудит розробленої автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах

Як було зазначено раніше, значне зростання кількості електроприладів, які використовуються в сучасних домогосподарствах, викликає збільшення кількості спожитої електроенергії (та інших ресурсів) і як наслідок – збільшення їх вартості. Тому актуальними стають чіткий облік спожитих ресурсів та їх правильний підрахунок, підвищення контролю за використанням ресурсів, раціоналізація споживання і попередження несанкціонованого відбору або витоків ресурсів тощо. Швидко виконувати ці операції вручну дуже складно.

У зв’язку з цим, все більшої популярності набуває модель розумного споживання, застосування якої мінімізує надлишок вироблених і неспожитих домогосподарствами ресурсів. Модель розумного споживання передбачає створення системи, яка відображає у реальному часу поточне споживання ресурсів (електроенергії, теплової енергії, води). Наявність цієї інформації дозволяє приймати рішення стосовно регулювання споживання енергоносіїв і тим самим зменшувати витрати на їх споживання.

Тому метою виконаної нами магістерської кваліфікаційної роботи була розробка автоматизованої системи обліку і аналізу споживання електроенергії в домогосподарствах, а також шляхи вдосконалення обліку та контролю за споживанням електричної енергії.

Для досягнення поставленої мети було проведено огляд науково-літературних джерел за тематикою досліджуваного об’єкту; сформульовано та описано вимоги до автоматизованої програмної системи; визначено технологію та підхід до її реалізації; створено тестові сценарії для тестування системи; описано архітектуру розробленого додатку з висновками, його переваги та недоліки тощо.

Для встановлення комерційного потенціалу розробленої автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах було запрошено 3-х відомих експертів –кандидатів технічних наук, професорів Кривогубченка С.Г. і Папінова В.М. та к.т.н., доценткиню Богач І.В.

Встановлення комерційного потенціалу розробленої автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах було здійснено за критеріями, наведеними в таблиці 4.1,

Таблиця 4.1 – Рекомендовані критерії оцінювання технічного рівня та комерційного потенціалу будь-якої розробки і їх бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
Ринкові перспективи					
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренті в немає

Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Практична здійсненність					
	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
0	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

1	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
2	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Запрошені експерти оцінили розроблену нами автоматизовану систему обліку та аналізу споживання електроенергії в домогосподарствах таким чином (див. таблицю 4.2):

Таблиця 4.2 – Результати технологічного аудиту розробленої автоматизованої системи обліку і аналізу споживання електроенергії в домогосподарствах, (за шкалою оцінювання 0-1-2-3-4)

Критерії	Прізвище, ініціали експертів		
	Кривогубченко С.Г.	Папінов В.М.	Богач І.В.
	Бали, що їх виставили експерти:		
1	4	3	3
2	4	3	4
3	3	3	3
4	4	4	4
5	3	3	3
6	4	4	4
7	3	3	4
8	4	3	3
9	4	4	3
10	3	3	4
11	4	4	3
12	3	3	4
Сума балів	$СБ_1 = 43$	$СБ_2 = 40$	$СБ_3 = 42$
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_1^3 СБ_i}{3} = \frac{43+40+42}{3} = \frac{125}{3} = 41,67$		

Встановлення комерційного потенціалу розробленої автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах будемо здійснювати на основі рекомендацій, наведених в таблиці 4.3 [34].

Таблиця 4.3 – Рівні комерційного потенціалу будь-якої наукової розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього

21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Оскільки середньоарифметична сума балів, що їх виставили експерти, складає 41,67 балів, то це свідчить, що розроблена нами автоматизована система обліку та аналізу споживання електроенергії в домогосподарствах має рівень комерційного потенціалу, який вважається «високим».

Це пояснюється тим, що розроблена нами автоматизована система обліку та аналізу споживання електроенергії в домогосподарствах базується на сучасних підходах, методах та алгоритмах, що дозволяє більш ефективно організовувати процес обліку та аналізу споживання електроенергії в домогосподарствах.

4.2 Розрахунок витрат на розроблення автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах

При розробленні автоматизованої системи обліку та аналізу споживання електроенергії були зроблені певні витрати. Зокрема:

А). Основна заробітна плата $З_о$ розробників, яка визначається за формулою:

$$З_о = \frac{М}{Т_р} \cdot t \text{ грн}, \quad (4.1)$$

де М – місячний посадовий оклад розробника, грн; приймемо, що

М = (6700…20000) грн/місяць;

$Т_р$ – число робочих днів в місяці; приймемо $Т_р$ = 21 день;

t – число днів роботи розробників.

Зроблені розрахунки зведемо до таблиці 4.4:

Таблиця 4.4 – Основна заробітна плата розробників

Найменування посади виконавця	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на оплату праці, грн
1. Науковий керівник магістерської роботи	19000	904,76	20 годин	≈ 3016
2. Магістрант-студент-виконавець	2000 (беремо 6700)	319,04	88	≈ 28076
3. Консультант з економічної частини	16750	797,62	1,5 години	≈ 199 (при 6-годинному робочому дні)
Загалом				$З_о$ = 31201 грн

Б). Додаткова заробітна плата $З_д$ розробників розраховується як (10…12)% від величини їх основної заробітної плати, тобто:

$$З_д = \alpha \cdot З_о = (0{,}1…0{,}12) \cdot З_о. \quad (4.2)$$

Приймемо, що α = 0,1. Тоді для нашого випадку отримаємо:

$$З_д = 0{,}1 \times 31291 = 3129{,}1 \approx 3129 \text{ грн.}$$

В). Нарахування на заробітну плату $НЗП_{зп}$ розробників (дослідників) розраховуються за формулою:

$$НЗП_{зп} = (З_о + З_д) \cdot \frac{\beta}{100}, \quad (4.3)$$

де β – ставка обов'язкового єдиного внеску на державне соціальне страхування, %. β = 22%. Тоді:

$$НЗН_{зп} = (31291 + 3129) \times 0{,}22 = 7572{,}40 \approx 7573 \text{ грн.}$$

Г). Амортизація основних засобів А, які використовувались під час виконання цієї роботи:

$$А = \frac{Ц \cdot Н_а}{100} \cdot \frac{Т}{12} \text{ грн}, \quad (4.4)$$

де Ц – загальна балансова вартість основних засобів, грн;

$Н_а$ – річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що $Н_а$ = (2,5...25)%;

Т – термін використання основних засобів, місяці.

Зроблені розрахунки зведено в таблицю 4.5.

Таблиця 4.5 – Розрахунок амортизаційних відрахувань

Найменування обладнання, приміщень тощо	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
1. Комп'ютерна техніка, обладнання тощо	55000	25	3,2 (при 70% використанні	2566,7 ≈ 2567
2. Приміщення університету, кафедри	22000	3	3,2 при 20% використанні	35,2 ≈ 36
Всього				А = 2603 грн

Д). Витрати на матеріали М розраховуються за формулою:

$$М = \sum_1^n Н_і \cdot Ц_і \cdot К_і - \sum_1^n В_i \cdot Ц_в \text{ грн.}, \quad (4.5)$$

де $Н_і$ – витрати матеріалу *і*-го найменування, кг; $Ц_і$ – вартість матеріалу *і*-го най-менування; $К_і$ – коефіцієнт транспортних витрат, $К_і$ = (1,1…1,15); $В_і$ – маса відходів матеріалу *і*-го найменування; $Ц_в$ – ціна відходів матеріалу *і*-го найменування; n – кількість видів матеріалів.

Е). Витрати на комплектуючі К розраховуються за формулою:

$$К = \sum_1^n Н_i \cdot Ц_i \cdot К_i \text{ грн}, \quad (4.6)$$

де $Н_і$ – кількість комплектуючих *і*-го виду, шт.; $Ц_і$ – ціна комплектуючих *і*-го виду; $К_і$ – коефіцієнт транспортних витрат, $К_і$ = (1,1…1,15); n – кількість видів комплектуючих.

Під час виконання роботи загальні витрати на матеріали та комплектуючі склали приблизно 1200 грн.

Ж). Витрати на силову електроенергію $В_е$ розраховуються за формулою:

$$B_e = \frac{B \cdot П \cdot Ф \cdot К_п}{К_д}, \quad (4.7)$$

де В – вартість 1 кВт-год. електроенергії, в 2022 р. В ≈ 3,0 грн/кВт;

П – установлена потужність обладнання, кВт; П = 1,15 кВт;

Ф – фактична кількість годин роботи обладнання, годин.

Приймемо, що Ф = 215 годин;

$К_п$ – коефіцієнт використання потужності; $К_п < 1 = 0,82$.

$К_д$ – коефіцієнт корисної дії, $К_д = 0,77$.

Тоді витрати на силову електроенергію будуть дорівнювати:

$$В_е = \frac{В \cdot П \cdot Ф \cdot К_п}{К_д} = \frac{3 \cdot 1,15 \cdot 215 \cdot 0,82}{0,77} = 789,9 \approx 790$$ грн.

И). Інші витрати $В_{інш}$ можна прийняти як (50…300)% від основної заробітної плати розробників, тобто:

$$В_{інш} = (0,5....3) \times З_о. \quad (4.8)$$

Для нашого випадку отримаємо:

$$В_{інш} = 1,25 \times 31291 = 39113,75 \approx 39114 \text{ грн.}$$

К). Сума всіх попередніх статей витрат складає витрати на виконання нашої роботи (безпосередньо розробником-магістрантом) – В.

$$В = 31291 + 3129 + 7573 + 2603 + 1200 + 790 + 39114 = 85700 \text{ грн.}$$

Л). Загальні витрати на розробку та можливе впровадження розробленої нами автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах $В_{заг}$ розраховуються за формулою:

$$В_{заг} = \frac{В}{\beta}, \quad (4.9)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання цієї роботи. Можна прийняти, що, $\beta \approx 0,75$ [34], оскільки робота потребує незначного доопрацювання.

Тоді: $$В_{заг} = \frac{85700}{0,75} = 114266,67$$ грн або приблизно 115 тисяч грн.

Тобто прогнозовані загальні витрати на розробку та можливе впровадження (комерціалізацію) автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах становлять приблизно 115 тисячі грн.

4.3 Розрахунок економічного ефекту від можливої комерціалізації нашої розробки

Економічний ефект від впровадження та можливої комерціалізації розробленої автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах пояснюється її значно кращими функціональними можливостями. Тому нашу розробку можна реалізовувати на ринку дещо дорожче, ніж аналогічні за функціями розробки. Так, якщо подібна за функціями автоматизована система для обслуговування 500 користувачів (домогосподарств) у 2022 році коштувала на ринку приблизно 200 тисяч грн (це вартість обслуговування – 40 тис грн, вартість пристроїв 150 тис грн тощо), то нашу розробку можна буде реалізовувати на ринку приблизно за 230 тисяч грн або на 30 тисяч грн дорожче.

Аналіз місткості ринку показує, що на сьогодні в Україні кількість подібних систем автоматизованого обліку може складати приблизно 50 систем. Це, насамперед, численні домогосподарства, заклади харчування тощо), але їх кількість у зв’язку із суттєвим підвищенням вартості ресурсів буде стрімко зростати . Тому можна очікувати зростання попиту на нашу розробку принаймні протягом 3-х років після її впровадження.

Тобто, якщо наша розробка буде впроваджена з 1 січня 2024 року (оскільки потребує що доопрацювання), то її результати будуть виявлятися протягом 2024-го, 2025-го та 2026-го років.

Прогноз зростання попиту на нашу розробку складає по роках:

а) 2024 р. – приблизно +20 шт. до базового року;

б) 2025 р. – +25 шт. до базового року;

в) 2026 р. – +30 шт. до базового року.

Можливе збільшення чистого прибутку $\Delta П_і$, що його може отримати потенційний інвестор від комерціалізації, тобто виведення нашої розробки на ринок, становитиме:

$$\Delta П_і = \sum_1^n (\Delta Ц_о \cdot N + Ц_о \cdot \Delta N)_і \cdot \lambda \cdot \rho \cdot (1 - \frac{\upsilon}{100}), \qquad (4.10)$$

де $\Delta Ц_о$ – покращення основного якісного показника від впровадження результатів нашої розробки у цьому році. Для нашого випадку це є збільшення ціни реалізації нашої розробки $\Delta Ц_о$ = 230 - 200 = + 30 тисяч грн;

N – основний кількісний показник, який визначає обсяг діяльності у році до впровадження результатів розробки; N = 50 шт.;

ΔN – покращення основного кількісного показника від впровадження результатів розробки. Таке покращення становитиме по роках, відповідно: у 2024 році – + 20 шт., у 2025 році +25 шт, та у 2026 році + 30 шт. (відносно базового 2022 року);

$Ц_о$ – основний якісний показник (тобто ціна), який визначає обсяг діяльності у році після впровадження результатів розробки, грн; $Ц_о$= 230 тисяч грн;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки; для нашого випадку n = 3;

λ – коефіцієнт, який враховує сплату податку на додану вартість; $\lambda = 0{,}8333$;

ρ – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати ρ= (0,2...0,5); візьмемо ρ= 0,5;

υ – ставка податку на прибуток. У 2022-23 роках υ = 18%. У 2024 році також очікуємо на 18%.

Тоді можливе зростання чистого прибутку $\Delta П_1$ для потенційного інвестора протягом першого року від можливого впровадження нашої розробки (2024 р.) складе:

$$\Delta П_1 = [30 \cdot 50 + 230 \cdot 20] \cdot 0{,}8333 \cdot 0{,}5 \cdot (1 - \frac{18}{100}) \approx 2084 \text{ тис. грн.}$$

Можливе зростання чистого прибутку $\Delta П_2$ для потенційного інвестора від можливого впровадження нашої розробки протягом другого (2025 р.) року складе:

$$\Delta П_2 = [30 \cdot 50 + 230 \cdot 25] \cdot 0,8333 \cdot 0,5 \cdot (1 - \frac{18}{100}) \approx 2477$$ тис. грн.

Можливе зростання чистого прибутку $\Delta П_3$ для потенційного інвестора від можливого впровадження нашої розробки протягом третього (2026 р.) року складе:

$$\Delta П_3 = [30 \cdot 50 + 230 \cdot 30] \cdot 0,8333 \cdot 0,5 \cdot (1 - \frac{18}{100}) \approx 2870$$ тис. грн.

Приведена вартість зростання всіх чистих прибутків від можливого впровадження нашої розробки становитиме:

$$ПП = \sum_1^T \frac{\Delta П_i}{(1+\tau)^t}, \quad (4.11)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої роботи, грн;

т – період часу, протягом якого виявляються результати впровадженої роботи, роки. Для нашого випадку т = 3 роки;

τ – ставка дисконтування. Приймемо τ = 0,10 (10%);

t – період часу від моменту початку розроблення автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах до моменту отримання можливих чистих прибутків потенційним інвестором.

Тоді приведена вартість зростання всіх можливих чистих прибутків ПП, що їх може отримати потенційний інвестор від комерціалізації нашої розробки, складе:

$$ПП = \frac{2084}{(1+0,1)^2} + \frac{2477}{(1+0,1)^3} + \frac{2870}{(1+0,1)^4} \approx 1722 + 1861 + 1960 = 5543$$

тисяч грн.

Теперішня вартість інвестицій PV, що повинні бути вкладені для реалізації нашої розробки: PV = (1,0…5) × $В_{заг}$.

Для нашого випадку PV = (1,0…5) × 115 = 5 × 115 = 575 тисяч грн.

Розраховуємо абсолютний ефект від можливих вкладених інвестицій $Е_{абс}$.

$$Е_{абс} = ПП – PV, \quad (4.12)$$

де ПП – приведена вартість збільшення всіх чистих прибутків для інвестора від можливого впровадження нашої розробки, грн;

PV – теперішня вартість інвестицій PV = 575 тисяч грн.

Абсолютний ефект від можливого впровадження нашої розробки (при прогнозованому ринку збуту) за три роки складе:

$$Е_{абс} = 5543 – 575 = 4968 \text{ тисяч грн.}$$

Оскільки $Е_{абс} > 0$, то комерціалізація нашої розробки може бути доцільною.

Далі розрахуємо внутрішню дохідність $Е_в$ вкладених інвестицій:

$$Е_в = \sqrt[Т_ж]{1 + \frac{Е_{абс}}{PV}} - 1, \quad (4.13)$$

де $Е_{абс}$ – абсолютний ефект вкладених інвестицій; $Е_{абс}$ = 4968 тис. грн;

PV –теперішня вартість початкових інвестицій PV = 575 тис. грн;

$Т_ж$ – життєвий цикл розробки, роки.

$Т_ж$ = 4 років (2023-й, 2024-й, 2025-й, 2026-й роки)

Для нашого випадку отримаємо:

$$Е_в = \sqrt[4]{1 + \frac{4968}{575}} - 1 = \sqrt[4]{1 + 8,64} - 1 = \sqrt[4]{9,64} - 1 = 1,76 - 1 = 0,76 = 76\%.$$

Далі визначимо ту мінімальну дохідність, нижче за яку потенційному інвестору не вигідно буде займатися комерціалізацією нашої розробки.

Мінімальна дохідність або мінімальна (бар'єрна) ставка дисконтування $\tau_{мін}$ визначається за формулою:

$$\tau_{мін} = d + f, \quad (4.14)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні d = (0,10...0,12);

f – показник, що характеризує ризикованість вкладень; f = (0,05...0,30).

Для нашого випадку отримаємо:

$$\tau_{мін} = 0,12 + 0,30 = 0,42 \text{ або } \tau_{мін} = 42\%.$$

Оскільки величина $E_в = 76\% > \tau_{мін} = 42\%$, то потенційний інвестор у принципі може бути зацікавлений у фінансуванні та комерціалізації розробленої автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах.

Далі розраховуємо термін окупності коштів, вкладених у можливу комерціалізацію розробленої автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах.

Термін окупності $T_{ок}$ розраховується за формулою:

$$T_{ок} = \frac{1}{E_в}. \quad (4.15)$$

Для нашого випадку термін окупності $T_{ок}$ коштів становитиме:

$$T_{ок} = \frac{1}{0,76} = 1,32 \text{ років} < 3 \text{ років},$$

що свідчить про потенційну доцільність комерціалізації розробленої нами автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах.

Далі проведено моделювання залежності величини внутрішньої дохідності вкладених потенційних інвестицій від рівня інфляції в країні. Як відомо, на наступні роки, на жаль, прогнозується високий рівень інфляції (в межах 30% і вище), що обумовлюється військовою агресією росії проти України.

Прийнявши рівень інфляції у 30% отримаємо:

$$ПП = \frac{2084}{(1+0,3)^2} + \frac{2477}{(1+0,3)^3} + \frac{2870}{(1+0,3)^4} \approx 1233 + 1127 + 1005 = 3365$$

тисяч грн.

Тоді абсолютний ефект від можливого впровадження нашої розробки за три роки складе:

$$Е_{абс} = 3365 – 575 = 2790 \text{ тисяч грн.}$$

Внутрішня дохідність $Е_в$ вкладених інвестицій становитиме:

$$Е_в = \sqrt[Т_ж]{1 + \frac{Е_{абс}}{PV}} - 1,$$

де $Е_{абс}$ – абсолютний ефект вкладених інвестицій; $Е_{абс} = 2790$ тисяч грн;

PV –теперішня вартість початкових інвестицій PV = 575 тисяч грн.

Для нашого випадку отримаємо:

$$Е_в = \sqrt[4]{1 + \frac{2790}{575}} - 1 = \sqrt[4]{1 + 4,8522} - 1 = \sqrt[4]{5,8522} - 1 = 1,555 - 1 = 0,555 = 55,5\%.$$

Прийнявши рівень інфляції у 50% отримаємо:

$$ПП = \frac{2084}{(1+0,5)^2} + \frac{2477}{(1+0,5)^3} + \frac{2870}{(1+0,5)^4} \approx 926 + 734 + 567 = 2227 \text{ тисяч грн.}$$

Тоді абсолютний ефект від можливого впровадження нашої розробки за три роки складе:

$$Е_{абс} = 2227 – 575 = 1652 \text{ тисяч грн.}$$

Внутрішня дохідність $Е_в$ вкладених інвестицій становитиме:

$$Е_в = \sqrt[Т_ж]{1 + \frac{Е_{абс}}{PV}} - 1,$$

де $Е_{абс}$ – абсолютний ефект вкладених інвестицій; $Е_{абс} = 1652$ тисяч грн;

PV –теперішня вартість початкових інвестицій PV = 575 тисяч грн.

Для нашого випадку отримаємо:

$$Е_в = \sqrt[4]{1 + \frac{1652}{575}} - 1 = \sqrt[4]{1 + 2,8730} - 1 = \sqrt[4]{3,8730} - 1 = 1,4 - 1 = 0,4 = 40\%.$$

Зроблені розрахунки у вигляді графіків наведено на рис. 4.1.

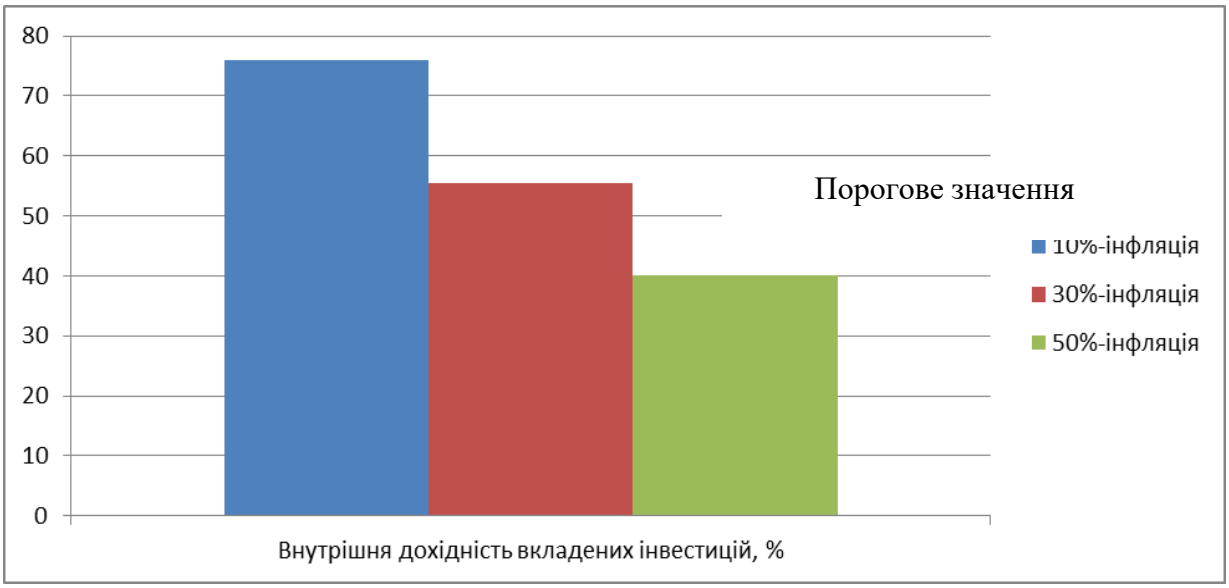

Рисунок 4.1 – Моделювання залежності величини внутрішньої дохідності
потенційних інвестицій від рівня інфляції в країні

Аналіз графіка на рис 4.1 показує, що при рівні інфляції в 10% величина внутрішньої дохідності інвестицій становить $E_в$ = 76%, що значно більше порогового значення $\tau_{мін} = 42\%$ і тому комерціалізація нашої розробки може бути доцільною. При рівні інфляції в 30% величина внутрішньої дохідності інвестицій, вкладених в комерціалізацію нашої розробки, становить $E_в$ = 55,5%, що також більше порогового значення $\tau_{мін} = 42\%$, і тому комерціалізація нашої розробки (при нинішніх умовах) може бути для інвестора доцільною. І тільки при інфляції у 50% величина внутрішньої дохідності інвестицій, вкладених в комерціалізацію нашої розробки, становить $E_в = 40{,}0\%$, що менше порогового значення $\tau_{мін}$ = 42%. і комерціалізація нашої розробки є проблематичною. Для прийняття остаточного рішення потрібно провести додаткові обгрунтування і розрахунки (наприклад, знизити рівень прийнятого ризику, підняти ціну реалізації нашої розробки тощо).

Результати виконаної економічної частини магістерської кваліфікаційної роботи зведено у таблицю:

Показники	Задані у ТЗ	Досягнуті у магістерській кваліфікаційній роботі	Висновок
1. Витрати на розробку	Не більше 120 тис. грн	115 тис. грн.	Досягнуто
2. Абсолютний ефект від впровадження розробки (в майбутній вартості грошей), тисяч грн	Не менше 5000 тисяч грн (за три роки)	4968 тисяч грн	Практично виконано
3. Внутрішня дохідність інвестицій, %	не менше 42%	76%	Досягнуто
4. Термін окупності інвестицій, роки	до 3-ти років	1,32 років	Виконано

Таким чином, основні техніко-економічні показники розробленої нами автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах, визначені у технічному завданні, виконані.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Рішення обміну даних в системах автоматизації [Електронний ресурс] Молодь в науці: дослідження, проблеми, перспективи (МН-2023) - Режим доступу: URL: https://conferences.vntu.edu.ua/index.php/mn/mn2023/paper/view/16657

2. Вісник Харківського національного технічного університету сільського господарства імені Петра Василенка. Технічні науки. Випуск 165 "Проблеми енергозабезпечення та енергозбереження в АПК України". – Харків: ХНТУСГ, 2015. – 135 с.

3. Хижняк І. А. Пристрій збору даних цифрової лабораторії : дис. – КПІ ім. Ігоря Сікорського, 2021.

4. Деркач В. О. Замкнена мережа 110 кВ та аналіз засобів обліку електричної енергії : дис. – КПІ ім. Ігоря Сікорського, 2021.

5. Карпалюк І. Т. Комп’ютерні інформаційні технології в енергетиці: конспект лекцій (для студентів 5 курсу денної, 6 курсу заочної форми навчання освітньо-кваліфікаційного рівня «магістр» за спеціальністю 141–Електроенергетика, електротехніка та електромеханіка). – 2018.

6. Клендій Г., Сухорукова І. РЕАЛІЗАЦІЯ АСКОЕ НА ОСНОВІ МУЛЬТИПЛЕКСОРІВ-РОЗШИРЮВАЧІВ //Рекомендовано Вченою радою ВП НУБіП України «Бережанський агротехнічний інститут»(Протокол № 3/1 від 29.11. 2019 року). – С. 107.

7. Шуллє Ю. А., Рогозянський І. С. Використання АСКОЕ для підвищення ефективності енерговикористання на промислових підприємствах //Інформаційні технології та комп'ютерна інженерія. – 2016. – Т. 35. – №. 1. – С. 60-63.

8. Буркало В. І. Системи обліку та передачі даних спожитої електроенергії : дис. – Тернопільський національний технічний університет імені Івана Пулюя, 2019.

9. Бабенко О. В., Ясько Я. А. Автоматизована система контролю і обліку енергоресурсів : дис. – ВНТУ, 2020.

10. Махлін П. В., Махлин П. В. Програма, методичні вказівки з вивчення дисципліни “«Автоматизована система обліку та керування в енергозбереженні» та контрольні завдання для студентів спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» заочної форми навчання. – 2016.

11. Махлін П. В., Махлин П. В. Програма, методичні вказівки з вивчення дисципліни “«Автоматизована система обліку та керування в енергозбереженні» та контрольні завдання для студентів спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» заочної форми навчання. – 2016.

12. Гаман В. В., Курсін С. М. ОЦІНЮВАННЯ ВІДПОВІДНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАСОБІВ ОБЛІКУ ЕЛЕКТРОЕНЕРГІЇ ТА ЙОГО ВІДОБРАЖЕННЯ У НОРМАТИВНІЙ ДОКУМЕНТАЦІЇ //Відновлювана енергетика та енергоефективність у XXI столітті: матеріали XXII міжнародної науково-практичної конференції (Київ, 20-21 травня 2021р.).–К.: Інтерсервіс, 2021.–1104 с. У збірнику викладено матеріали доповідей учасників конференції, присвяченої розвитку відновлюваної енергетики з. – 2021. – С. 190.

13. Забелло Е. П., Булах В. Г., Качалко А. С. Автоматизированные системы контроля и учета энергоресурсов. – 2016.

14. Кириленко О.В., Smart Grid та організація інформаційного обміну в електроенергетичних системах / Кириленко О.В., Блінов І.В., Танкевич С.Є.; Технічна електродинаміка 2012, №3 – С. 47- 48.

15. Стогній Б.С. Інтелектуальні електричні мережі: світовий досвід і перспективи України. / Стогній Б.С., Кириленко О.В., Праховник А.В. Денисюк С.П.; Праці Інституту електродинаміки НАН України, 2011,Часина 1. – С. 5-20.

16. EPRI Smart Grid Demonstration Initiative. Two ycar update. – Electric Power Research Institute (ERP). – USA, California, 2010.

17. Енергетична стратегія України на період до 2030 року.[Електронний ресурс]. - Режим доступу: URL: www.cfin. ua/press/management/2001-6/13.pshtml

18. Правила улаштування електроустановок. - 5-те вид., переробл. й доповн. Міненерговугілля України, 2014 р

19. Теслюк Т. и др. Архітектура багаторівневої системи управління енергоефективністю регіону //Вісник Національного університету Львівська політехніка. Комп'ютерні науки та інформаційні технології. – 2017. – №. 864. – С. 201-209.

20. M. Bonvini, F. Donida, and A. Leva. Modelica as a design tool for hardware-in-the-loop simulation. In Proceedings 7th Modelica Conference, 09 2009. doi: 10.3384/ecp09430087. або налаштування контролера M. Laskawski. The tuning method of continuous controllers using scilab/xcos environment. In Przeglad Elektrotechniczny, 2017.

21. Kazala R., Straczynski P. The most important open technologies for design of cost efficient automation systems //IFAC-PapersOnLine. – 2019. – Т. 52. – №. 25. – С. 391-396.

22. R. Kazala, P. Straczynski, A. Taneva, and S. Penkov. The use of iot technologies for the monitoring of electrotechnological systems. In 2018 Conference on Electrotechnology: Processes, Models, Control and Computer Science (EPMCCS), pages 1–7, Nov 2018.

23. Yassein M. B. et al. Internet of Things: Survey and open issues of MQTT protocol //2017 international conference on engineering & MIS (ICEMIS). – Ieee, 2017. – С. 1-6.

24. Schulzrinne H., Rao A., Lanphier R. RFC2326: Real time streaming protocol (RTSP).

25. Yuya Maruyama, Shinpei Kato, and Takuya Azumi. Exploring the performance of ros2. In Proceedings of the 13th International Conference on Embedded Software, EMSOFT '16, pages 5:1–5:10, New York, NY, USA, 2016.

26. M. Cereia and S. Scanzio. A user space ethercat master architecture for hard real-time control systems. In Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies Factory Automation (ETFA 2012), pages 1–8, Sep. 2012.

27. Roy G. M. RabbitMQ in depth. – Simon and Schuster, 2017.

28. Sumaray A., Makki S. K. A comparison of data serialization formats for optimal efficiency on a mobile platform //Proceedings of the 6th international conference on ubiquitous information management and communication. – 2012.

29. Naqvi S. N. Z., Yfantidou S., Zimányi E. Time series databases and influxdb //Studienarbeit, Université Libre de Bruxelles. – 2017. – Т. 12.

30. InfluxData — Documentation — Continuous Queries. 2017. [Електронний ресурс]. - Режим доступу: URL: https: //docs.influxdata.com/influxdb/v1.2/query_language/continuous_queries/.

31. Kadamtech – Top backend frameworks in 2020 [Електронний ресурс]. - Режим доступу: URL: https://www.kadamtech.com/top-backend-frameworks-in-2020/

32. Spring.io – Spring Boot [Електронний ресурс]. - Режим доступу: URL: https://spring.io/projects/spring-boot

33. Akbar S. R. et al. Message queue telemetry transport protocols implementation for wireless sensor networks communication—A performance review //2017 International Conference on Sustainable Information Engineering and Technology (SIET). – IEEE, 2017.

34. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. / Укладачі В.О. Козловський, О.Й. Лесько, В.В.Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

ДОДАТКИ

Додаток А (обов'язковий).

Технічне завдання

ЗАТВЕРДЖЕНО
Зав. кафедри АІІВ ВНТУ,
д.т.н., проф.
_____О.В. Бісікало
"__" _____ 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на виконання магістерської кваліфікаційної роботи
РОЗРОБКА АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБЛІКУ ТА АНАЛІЗУ СПОЖИВАННЯ ЕЛЕКТРОЕНЕРГІЇ В ДОМОГОСПОДАРСТВАХ
08-02.МКР.000.00.009 ТЗ

Керівник магістерської кваліфікаційної роботи:
доцент
Я.А. Кулик
"___" _____ 2022р.

Виконавець: ст. гр. 1АКІТ-21м
Є.О. Рудич
"___" _____ 2022 р.

Вінниця 2022

1. Назва та галузь застосування:

Розробка автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах. Розроблене програмне забезпечення та алгоритм, що дозволить проводити облік споживання електроенергії в домогосподарствах.

2. Підстава для проведення розробки:

Тема дипломного проекту (роботи) затверджена наказом по ВНТУ № від _____ р.

3. Мета та призначення розробки:

Метою даної роботи є підвищення швидкості збору інформації про споживання електроенергії шляхом розробки системи обліку та аналізу споживання електроенергії в домогосподарствах на основі збору та аналізу інформації.

4. Джерела розробки:

1. Рішення обміну даних в системах автоматизації [Електронний ресурс] Молодь в науці: дослідження, проблеми, перспективи (МН-2023) - Режим доступу: URL: https://conferences.vntu.edu.ua/index.php/mn/mn2023/paper/view/16657

2. Бабенко О. В., Ясько Я. А. Автоматизована система контролю і обліку енергоресурсів : дис. – ВНТУ, 2020.

3. Карпалюк І. Т. Комп'ютерні інформаційні технології в енергетиці: конспект лекцій (для студентів 5 курсу денної, 6 курсу заочної форми навчання освітньо-кваліфікаційного рівня «магістр» за спеціальністю 141– Електроенергетика, електротехніка та електромеханіка). – 2018.

4. Теслюк Т. и др. Архітектура багаторівневої системи управління енергоефективністю регіону //Вісник Національного університету Львівська політехніка. Комп'ютерні науки та інформаційні технології. – 2017. – №. 864. – С. 201-209.

5. Технічні вимоги:

Система обліку та аналізу споживання електроенергії має працювати в автоматичному режимі.

Результати роботи програми:

- розробка серверного додатку для проведення операцій збереження та передання даних та математичну обробку даних;

- надання обліку на аналізу споживання електроенергії користувачу;

6. Економічні показники:

- прогнозовані витрати на розробку – 115000 грн.;

- абсолютна ефективність розробки – 4968000 грн.;

- термін окупності витрат для виробника – не більше 1,32 року.

7. Стадії та етапи розробки:

1	Аналіз стану проблеми і постановка задач	__.__.2022
2	Науково-технічне обгрунтування розробки моделей та методів автоматизованих систем контролю та обліку енергоресурсів	__.__.2022
3	Розробка математичних моделей споживання електроенергії в домогосподарствах	__.__.2022
4	Розробка програмного забезпечення споживання електроенергії в домогосподарствах	__.__.2022
5	Підготовка економічної частини	__.__.2022
6	Оформлення пояснювальної записки, графічного матеріалу і презентації	__.__.2022

8. Порядок контролю і приймання:

1.Рубіжний контроль провести до __.__.2022

2.Попередній захист магістерської кваліфікаційної роботи. Провести до __.__.2022

3.Захист магістерської кваліфікаційної роботи. Провести в період з __.__.2022 до __.__.2022

Додаток Б (обов'язковий).

Графічна частина

Зав. кафедри АІІТ _____ професор О.В. Бісікало
(підпис) (прізвище та ініціали)

Науковий керівник _____ доцент Я.А. Кулик
(підпис) (прізвище та ініціали)

Тех. контроль _____ доцент Я.А. Кулик
(підпис) (прізвище та ініціали)

Норм. контроль _____ доцент Я.А. Кулик
(підпис) (прізвище та ініціали)

Рецензент _____ доцент О.О. Ковалюк
(підпис) (прізвище та ініціали)

Ст. групи 1АКІТ-21м _____ Є.О. Рудич
(підпис) (прізвище та ініціали)

Продовження додатку Б

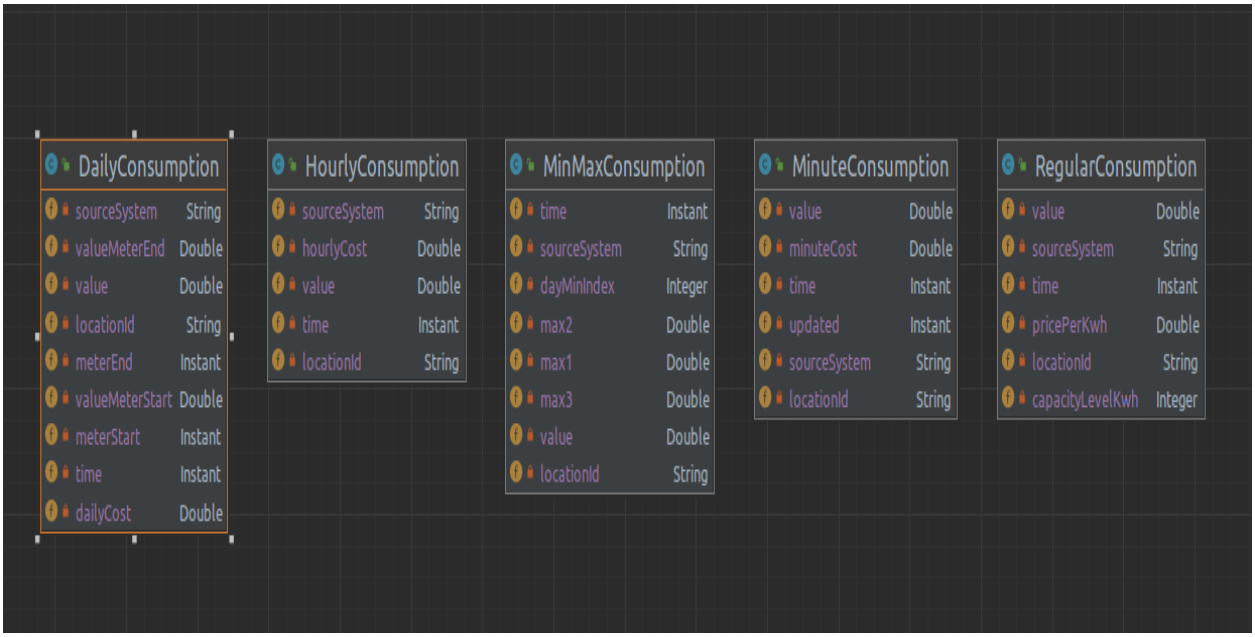

Рис Б.1 — UML–діаграма класів сутностей системи

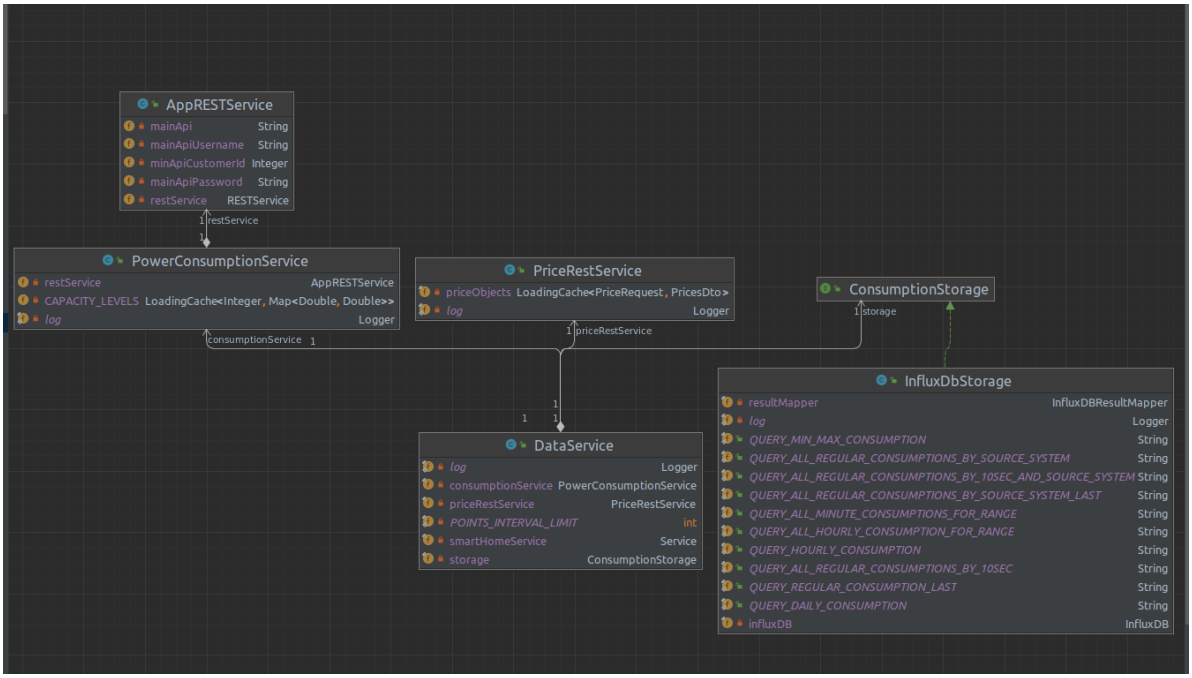

Рис Б.2 — UML–діаграма основних функціональних класів системи

Продовження додатку Б

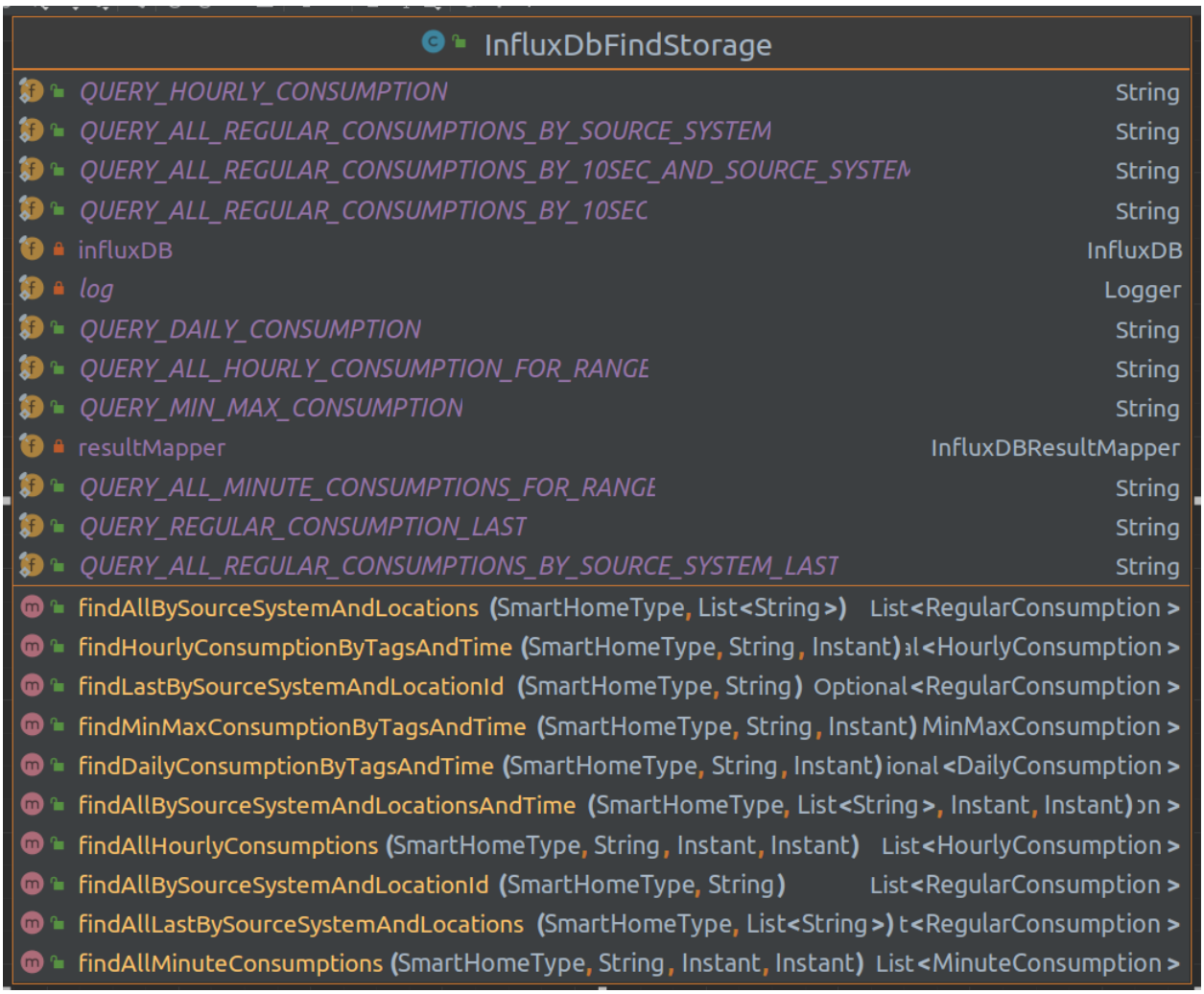

Рис Б.3 — UML–діаграма репозиторію InfluxDbFindStore

Продовження додатку Б

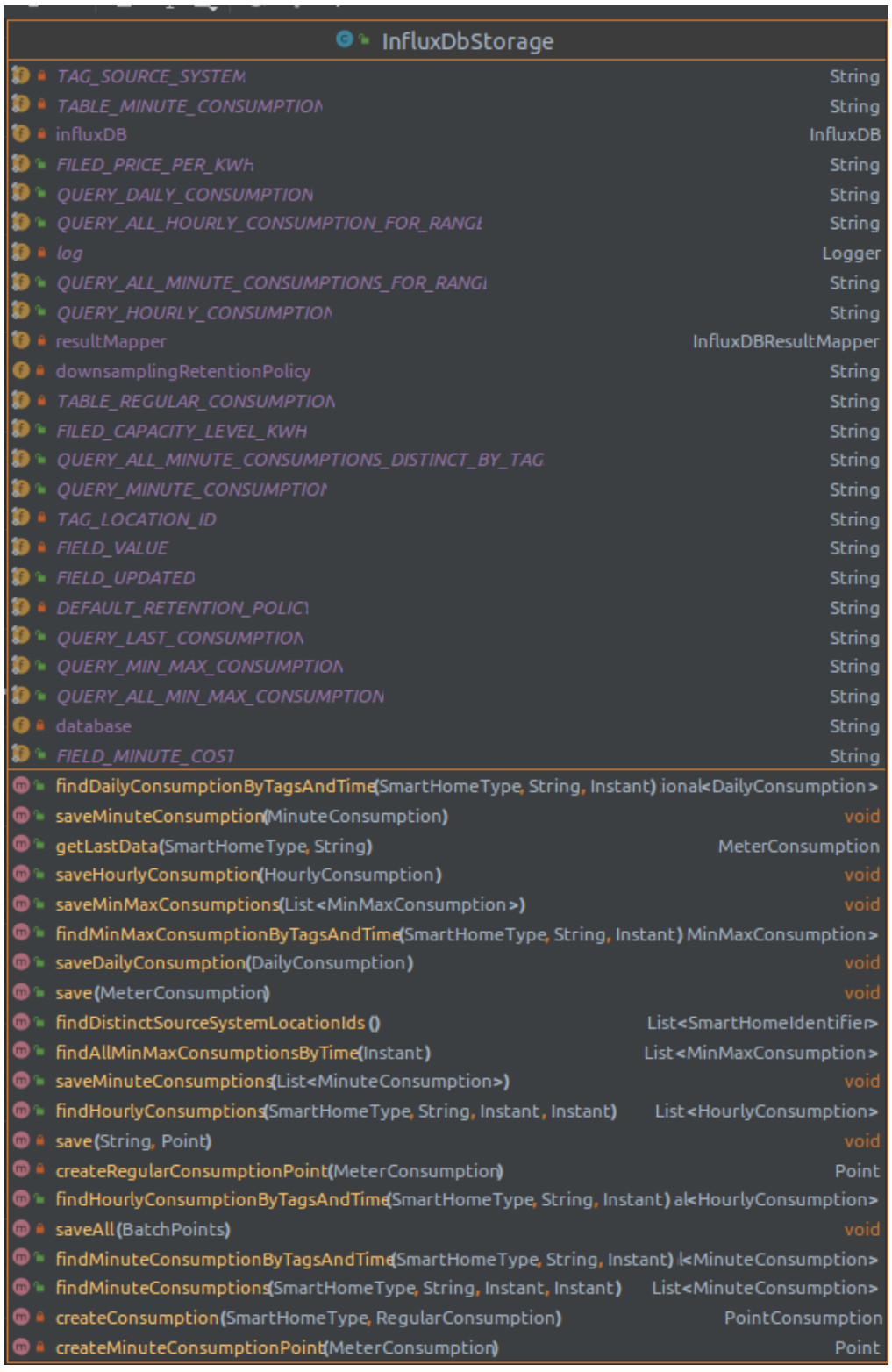

Рис Б.4 — UML–діаграма репозиторію InfluxDbStore

Продовження додатку Б

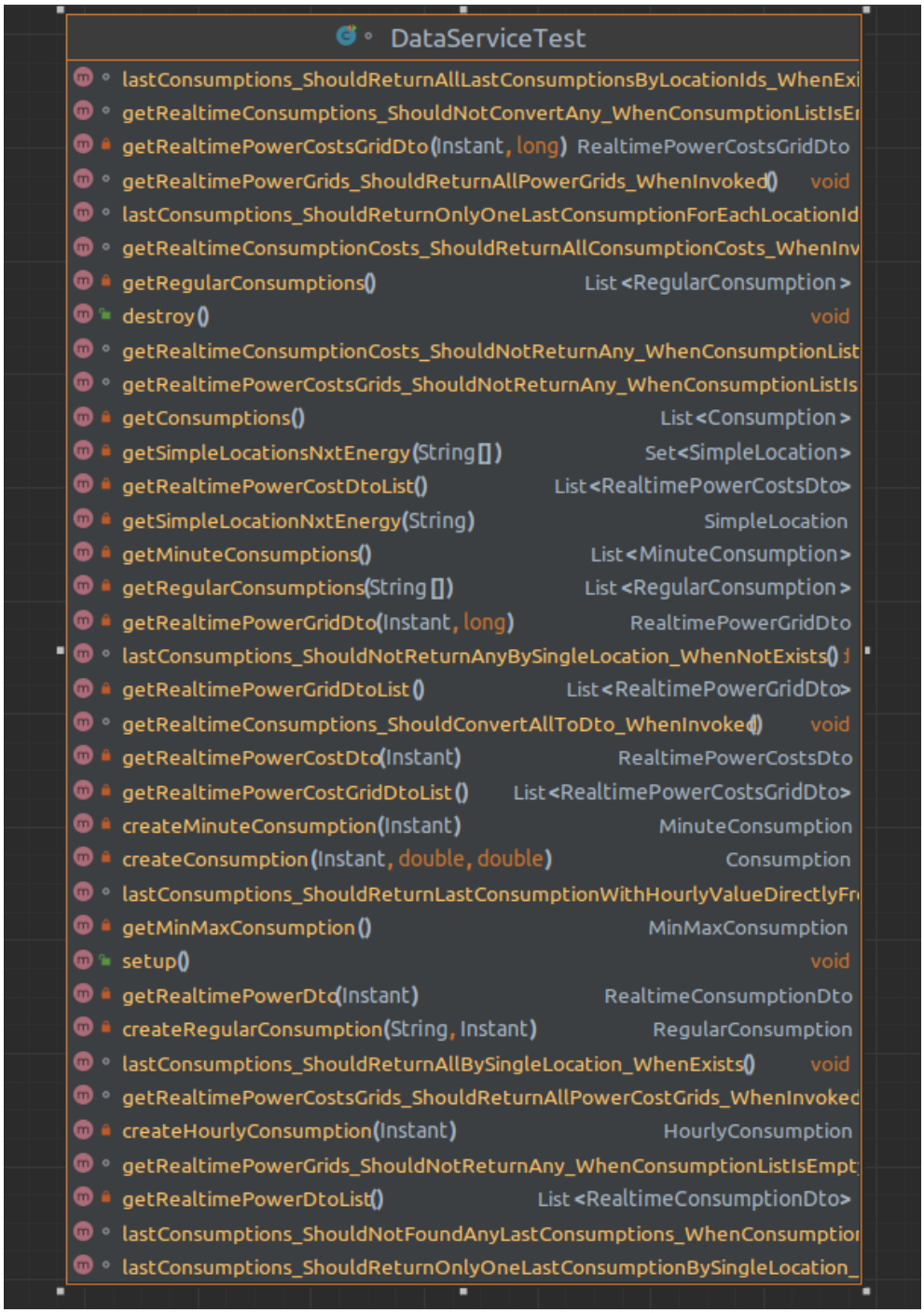

Рис Б.5 — UML-діаграма тестів класу DateServiceTest

Додаток В (обов'язковий).

Акт про впровадження

Товариство з обмеженою відповідальністю «Джи Ай Бі Групп»
КОД ЄДРПОУ 43823104
Індивідуальний податковий номер платника податку на додану вартість 438231002288
Розрахунковий рахунок:
IBAN UA 083348510000000026007126754
АТ «ПУМБ» м. Вінниця
МФО 334851

Юридична адреса:
21034, Україна , м. Вінниця ,вул. Тичини П. буд.21 кв.95
тел.+380991594398
gibgroup170920@gmail.com

ЗАТВЕРДЖУЮ
Джи Ай Бі Групп»
Гроян В.А.
08» грудня 2022 р.

АКТ
про впровадження результатів магістерської роботи
Рудич Єлизавети Олександрівни

«Розробка автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах»

Комісія у складі головного інженера Карватко М.А. та керівника проектів Карватко А.В. розглянула матеріали магістерської роботи Рудич Є. О. і встановила що при розробці систем автоматизації споживання електричної енергії, для яких потребується проводити детальний розрахунок споживання електричної енергії за певними параметрами, перспективним буде використання розробленого алгоритму та математичної моделі розрахунку споживання електроенергії у домогосподарствах.

За рахунок побудованої математичної моделі можна легко передбачати можливий рівень споживання електроенергії в майбутньому та тим самим обмежувати споживання економлячи кошти. Також є дуже корисним розроблене програмне забезпечення яке наочно показує споживання електроенергії в реальному часі з встановленими обмеженнями.

Отже розроблене Рудич Є.О. математичне, алгоритмічне та програмне забезпечення може бути прийнято для використання при побудові систем, що пов'язані з обліком електричної енергії. Таким чином, актуальність та практична цінність роботи підтверджена. Також, на думку комісії, підтверджена і наукова новизна розробки, оскільки в відомих програмних системах аналогічних засобів для розв'язання таких задач.

Члени комісії:
Головний інженер Карватко М.А.
Керівник проектів Карватко .А.В.

Додаток Г (обов‘язковий).

Лістинг програми

```
@Slf4j
@org.springframework.stereotype.Service
public class DataService {    private static final int POINTS_INTERVAL_LIMIT = 100;
   private final ConsumptionStorage storage;
   private final PriceRestService priceRestService;
   private final PowerConsumptionService consumptionService;
   private final Service smartHomeService;


   public DataService(ConsumptionStorage storage, PriceRestService priceRestService,
                      PowerConsumptionService consumptionService,
                      Service smartHomeService) {
      this.storage = storage;
      this.priceRestService = priceRestService;
      this.consumptionService = consumptionService;
      this.smartHomeService = smartHomeService;
   }


   public Map<SimpleLocation, List<Consumption>> lastConsumptions(Set<SimpleLocation> locations) {


      Map<Integer, List<SimpleLocation>> groupedLocations =
            locations.stream().collect(Collectors.groupingBy(SimpleLocation::getSmartHomeType));
      Map<SimpleLocation, List<Consumption>> locationsMap = new HashMap<>();
      for (Map.Entry<Integer, List<SimpleLocation>> groupedLocation : groupedLocations.entrySet()) {
         List<String> locationIds = locations.stream()
               .filter(l -> l.getSmartHomeType().equals(groupedLocation.getKey()))
               .map(SimpleLocation::getLocationId)
               .collect(Collectors.toList());


         List<RegularConsumption> consumptions =
storage.findAllBySourceSystemAndLocations(SmartHomeType.fromId(groupedLocation.getKey()),
locationIds);


         locationsMap.putAll(
               consumptions.stream()
                     .map(this::createConsumption)
                     .collect(Collectors.groupingBy(consumption -> new SimpleLocation(groupedLocation.getKey(),
```

```
consumption.getLocationId())));

        List<String> noCurrentDataLocations = locations.stream()
            .filter(l -> l.getSmartHomeType().equals(groupedLocation.getKey()) &&
!locationsMap.containsKey(l))
            .map(SimpleLocation::getLocationId)
            .collect(Collectors.toList());

        List<RegularConsumption> lastConsumptions =
storage.findAllLastBySourceSystemAndLocations(SmartHomeType.fromId(groupedLocation.getKey()),
noCurrentDataLocations);
        locationsMap.putAll(
            lastConsumptions.stream()
                .map(this::createConsumption)
                .collect(Collectors.groupingBy(consumption -> new SimpleLocation(groupedLocation.getKey(),
consumption.getLocationId()))));
      }
      return locationsMap;
    }

    public List<Consumption> lastConsumptions(SimpleLocation location) {
      List<RegularConsumption> consumptions =
storage.findAllBySourceSystemAndLocationId(SmartHomeType.fromId(location.getSmartHomeType()),
location.getLocationId());
      if (consumptions.isEmpty()) {
        return
storage.findLastBySourceSystemAndLocationId(SmartHomeType.fromId(location.getSmartHomeType()),
location.getLocationId())
            .map(this::createConsumption)
            .map(List::of).orElse(Collections.emptyList());
      }
      return consumptions.stream().map(this::createConsumption).collect(Collectors.toList());
    }

    public List<RealtimeConsumptionDto> getRealtimeConsumptions(List<Consumption> consumptions) {
      Instant now = Instant.now();
      return consumptions.stream()
          .map(consumption -> RealtimeConsumptionDto.fromConsumption(consumption, now))
          .collect(Collectors.toList());
    }
```

```
  public List<RealtimePowerCostsDto> getRealtimeConsumptionCosts(List<Consumption> consumptions) {
    Instant now = Instant.now();
    return consumptions.stream()
        .map(consumption -> RealtimePowerCostsDto.fromConsumption(consumption, now))
        .collect(Collectors.toList());
  }

  public List<RealtimePowerGridDto> getRealtimePowerGrids(List<Consumption> consumptions) {
    Instant now = Instant.now();
    return consumptions.stream()
        .map(consumption -> RealtimePowerGridDto.from(consumption, createPowerGrid(consumption),
now))
        .collect(Collectors.toList());
  }

  public List<RealtimePowerCostsGridDto> getRealtimePowerCostsGrids(List<Consumption>
consumptions) {
    Instant now = Instant.now();
    return consumptions.stream()
        .map(consumption -> RealtimePowerCostsGridDto.from(consumption,
createPowerCostsGrid(consumption), now))
        .collect(Collectors.toList());
  }



  public List<RealtimeConsumptionDto> realtimeConsumptions(SimpleLocation location) {
    return getRealtimeConsumptions(lastConsumptions(location));
  }

  public List<RealtimePowerCostsDto> realtimePowerCosts(SimpleLocation location) {
    return getRealtimeConsumptionCosts(lastConsumptions(location));
  }

  public List<RealtimePowerGridDto> realtimePowerGrid(SimpleLocation location) {
    return getRealtimePowerGrids(lastConsumptions(location));
  }
```

```
  public List<RealtimePowerCostsGridDto> realtimePowerCostsGrid(SimpleLocation location) {
    return getRealtimePowerCostsGrids(lastConsumptions(location));
  }

  public Map<SimpleLocation, List<RegularConsumption>> historicConsumptions(HistoryRequest
historyRequest,
                                                Set<SimpleLocation> locations) {
    Integer points = historyRequest.getPoints();
    Integer interval = historyRequest.getInterval();
    if (points == null || interval == null) {
      throw new IllegalArgumentException("`points` or `interval` are not provided");
    }
    if (points > POINTS_INTERVAL_LIMIT) {
      throw new IllegalArgumentException("Cannot request more than 100 points");
    }
    if (interval > POINTS_INTERVAL_LIMIT) {
      throw new IllegalArgumentException("Interval cannot be more than 100 seconds");
    }
    return findFor(locations, points, interval);
  }

  public List<HistoricConsumptionDto> historicConsumptions(HistoryRequest historyRequest,
                                      SimpleLocation location) {
    Map<SimpleLocation, List<RegularConsumption>> consumptions =
        historicConsumptions(historyRequest, Set.of(location));
    Instant now = Instant.now();
    return consumptions.getOrDefault(location, Collections.emptyList())
        .stream()
        .map(consumption -> HistoricConsumptionDto.fromConsumption(consumption, now))
        .collect(Collectors.toList());
  }

  public HourlyConsumptionsDto hourlyConsumptionsKwh(SimpleLocation location) {
    Instant time = Instant.now();
    int hour = time.atZone(ZONE_ID).toLocalDateTime().getHour();
    if (hour < 1) return new HourlyConsumptionsDto(location.getLocationId(), time, Collections.emptyList());
    List<HourlyConsumption> hourlyConsumptions = storage.findAllHourlyConsumptions(
        SmartHomeType.fromId(location.getSmartHomeType()), location.getLocationId(),
        getTimeAtStartOfDay(time), getTimeAtEndOfHour(time.minus(1, ChronoUnit.HOURS)));
```

```
        List<HourConsumptionDto> observed = hourlyConsumptions.stream().map(c -> {
            int chour = c.getTime().atZone(ZONE_ID).toLocalDateTime().getHour();
            return new HourConsumptionDto(String.valueOf(chour), c.getValue());
        }).collect(Collectors.toList());

        return new HourlyConsumptionsDto(location.getLocationId(), time, observed);
    }

    public PricesDto dailyPrices(SimpleLocation location) {
        return priceRestService.getPrices(SmartHomeType.fromId(location.getSmartHomeType()),
location.getLocationId(), Instant.now());
    }

    public HourlyPriceConsumptionsDto hourlyConsumptionsCosts(HourlyConsumptionsDto
kwhConsumptions, PricesDto prices) {
        List<HourConsumptionDto> observedCosts = new ArrayList<>();
        kwhConsumptions.getObserved().forEach(e -> {
            Double price = prices.getPrices().get(Integer.parseInt(e.getId()));
            price = Optional.ofNullable(price).map(p -> p * e.getValue() / 1000).orElse(null);
            observedCosts.add(new HourConsumptionDto(e.getId(), price));
        });
        return new HourlyPriceConsumptionsDto(kwhConsumptions.getLocationId(),
kwhConsumptions.getTime(), observedCosts,
                prices.getVatIncluded(), prices.getPayVat(), prices.getVatPercent());
    }

    public WarningsDto getWarnings(SimpleLocation location) {
        WarningsDto warningDto = new WarningsDto();
        List<Warnings> warnings = new ArrayList<>();
        List<Consumption> lastConsumptions = lastConsumptions(location);
        if (lastConsumptions.isEmpty()) {
            warnings.add(Warnings.NO_DATA);
        } else if (lastConsumptions.get(0).getTime().isBefore(Instant.now().minus(30, ChronoUnit.MINUTES))) {
            warnings.add(Warnings.DATA_DELAY);
        }

        var prevDayDailyConsumption = storage.findDailyConsumptionByTagsAndTime(
                SmartHomeType.fromId(location.getSmartHomeType()),
                location.getLocationId(),
```

```
            DateUtils.getTimeAtStartOfDay(Instant.now()).minus(1, ChronoUnit.DAYS));
        if (prevDayDailyConsumption.isEmpty()) {
          warnings.add(Warnings.COLD_START);
        }

        warningDto.setWarnings(warnings);
        return warningDto;
    }

    private Consumption createConsumption(RegularConsumption consumption) {
        double realtimeHourlyConsumption = getRealtimeHourlyConsumption(consumption);
        double dailyConsumptionTillLastHour =
calcDailyConsumptionTillLastHour(SmartHomeType.valueOf(consumption.getSourceSystem()),
consumption.getLocationId(), consumption.getTime());
        double realtimeDailyConsumption = dailyConsumptionTillLastHour + realtimeHourlyConsumption;

        double realtimeHourlyCosts = getRealtimeHourlyCosts(consumption);
        double dailyCostsTillLastHour =
calcDailyCostsTillLastHour(SmartHomeType.valueOf(consumption.getSourceSystem()),
consumption.getLocationId(), consumption.getTime());
        double realtimeDailyCost = dailyCostsTillLastHour + realtimeHourlyCosts;

        VatDto vatInfo = priceRestService.getVatInfo(SmartHomeType.valueOf(consumption.getSourceSystem()),
consumption.getLocationId(), Instant.now());

        return Consumption.fromConsumption(consumption, realtimeDailyConsumption,
realtimeHourlyConsumption, realtimeDailyCost,
            vatInfo.isVatIncluded(), vatInfo.isPayVat(), vatInfo.getVatPercent());
    }

    private Map<SimpleLocation, List<RegularConsumption>> findFor(Set<SimpleLocation> locations,
                                                  Integer points,
                                                  Integer interval) {
        Instant start = Instant.now().minus((long) points * interval, ChronoUnit.SECONDS);
        Instant stop = Instant.now();

        Map<Integer, List<SimpleLocation>> groupedLocations =
            locations.stream().collect(Collectors.groupingBy(SimpleLocation::getSmartHomeType));
```

```
    Map<SimpleLocation, List<RegularConsumption>> locationsMap = new HashMap<>();
    for (Map.Entry<Integer, List<SimpleLocation>> groupedLocation : groupedLocations.entrySet()) {
      List<String> locationIds = locations.stream()
          .filter(l -> l.getSmartHomeType().equals(groupedLocation.getKey()))
          .map(SimpleLocation::getLocationId)
          .collect(Collectors.toList());


      List<RegularConsumption> regularConsumptionsForPeriod =
storage.findAllBySourceSystemAndLocationsAndTime(SmartHomeType.fromId(groupedLocation.getKey()),
locationIds, start, stop);
      Map<SimpleLocation, List<RegularConsumption>> consumptionsForPeriod =
regularConsumptionsForPeriod.stream()
          .collect(Collectors.groupingBy(consumption -> new SimpleLocation(groupedLocation.getKey(),
consumption.getLocationId())));


      consumptionsForPeriod.forEach((location, consumptions) -> {
        consumptions.sort(Comparator.comparing(RegularConsumption::getTime));
        double[] times = consumptions.stream().map(RegularConsumption::getTime)
            .map(Instant::toEpochMilli).mapToDouble(c -> c / 1000.0).toArray();
        double[] values = consumptions.stream().map(RegularConsumption::getValue)
            .mapToDouble(c -> c).toArray();


        LinearInterpolator linearInterpolator = new LinearInterpolator();
        PolynomialSplineFunction interpolated =
            linearInterpolator.interpolate(times, values);


        List<RegularConsumption> thinnedConsumption = new ArrayList<>();
        double startDouble = start.toEpochMilli() / 1000.0;
        double stopDouble = stop.toEpochMilli() / 1000.0;


        while (startDouble < stopDouble) {
          Instant time = Instant.ofEpochMilli((long) startDouble * 1000);
          Double value = getConsumptionValue(interpolated, startDouble);
          String locationId = location.getLocationId();
          thinnedConsumption.add(new
RegularConsumption(SmartHomeType.fromId(groupedLocation.getKey()).name(), locationId, time, value));
          startDouble += interval;
        }
```

```
            locationsMap.put(location, thinnedConsumption);
        });
    }
    return locationsMap;
  }

  private double getConsumptionValue(PolynomialSplineFunction interpolationFunction, double v) {
    try {
      return interpolationFunction.value(v);
    } catch (OutOfRangeException e) {
      // add the extrapolation function: we use linear extrapolation based on the slope of the two points on
the left or right

      double[] interpolationKnots = interpolationFunction.getKnots();
      int n = interpolationKnots.length;
      double extrapolated;
      double extrapolatedValue;

      if (v < interpolationKnots[0]) {
        extrapolated = interpolationKnots[0];   // the leftest point
      } else {
        extrapolated = interpolationKnots[n - 1];  // the rightest point
      }

      extrapolatedValue = interpolationFunction.value(extrapolated);

      return extrapolatedValue;
    }
  }

  private PowerGrid createPowerCostsGrid(Consumption consumption) {
    PowerGrid powerGrid = createPowerGrid(consumption);
    Optional<SmartHomeLocation> location =
smartHomeService.getSmartHomeLocation(consumption.getSmartHomeType(),
consumption.getLocationId());
    Double capacityLevelCostsCurrent = null;
    Double capacityLevelCostsPredicted = null;
    if (location.isEmpty()) {
      log.error("No location found with id " + consumption.getLocationId() + ". Capacity levels are
```

```
unknown.");
      } else {
         capacityLevelCostsCurrent = consumptionService.getCosts(location.get().getGridCompanyOrgNo(),
powerGrid.getCapacityLevelCurrent());
         capacityLevelCostsPredicted = consumptionService.getCosts(location.get().getGridCompanyOrgNo(),
powerGrid.getCapacityLevelPredicted());
      }

      return new PowerGrid(powerGrid.getBurnRate(), powerGrid.getSampleStart(),
powerGrid.getSampleLeft(), powerGrid.getHourlyConsumptionPredicted(),
            capacityLevelCostsCurrent, capacityLevelCostsPredicted,
powerGrid.getCapacityLevelMaxToStayInCurrent(),
            powerGrid.getMaxAbsolute(), powerGrid.getMaxRecommended(),
powerGrid.getMaxAbsoluteInHour(), powerGrid.getMaxAbsoluteOffsetInHour(),
            powerGrid.getMaxAbsoluteRecommendedInHour(),
powerGrid.getMaxAbsoluteRecommendedCalculatedInHour(), powerGrid.getMaxHour1(),
            powerGrid.getMaxHour2(), powerGrid.getMaxHour3(), powerGrid.getMeanMax());
   }

   private PowerGrid createPowerGrid(Consumption consumption) {
      double burnRate = consumption.getBurnRate();
      Instant currentSampleTime = consumption.getTime();
      Instant sampleStart = consumption.getTime().truncatedTo(ChronoUnit.HOURS);
      Instant sampleEnd = consumption.getTime().plus(1,
ChronoUnit.HOURS).truncatedTo(ChronoUnit.HOURS);
      long sampleLeft = Duration.between(currentSampleTime, sampleEnd).toMillis();
      double hourlyConsumption = consumption.getHourlyConsumption();
      double hourlyConsumptionPredicted =
consumptionService.getHourlyConsumptionPredicted(currentSampleTime, hourlyConsumption);



      Double capacityLevelCurrent = null;
      Double capacityLevelPredicted = null;
      Double capacityLevelMaxToStayInCurrent = null;
      Double maxAbsolute = null;
      Double maxRecommended = null;
      Double maxAbsoluteInHour = null;
      Double maxAbsoluteOffsetInHour = null;
      Double maxAbsoluteRecommendedCalculatedInHour = null;
      Double maxAbsoluteRecommendedInHour = null;
```

```
    Double maxHour1 = null;
    Double maxHour2 = null;
    Double maxHour3 = null;
    Double meanMax = null;

    Instant thisDay = Instant.now().atZone(ZONE_ID).toInstant().truncatedTo(ChronoUnit.DAYS);
    Optional<MinMaxConsumption> thisDayMinMaxOptional =
storage.findMinMaxConsumptionByTagsAndTime(consumption.getSmartHomeType(),
consumption.getLocationId(), thisDay);

    if (thisDayMinMaxOptional.isPresent()) {
      Optional<SmartHomeLocation> location =
smartHomeService.getSmartHomeLocation(consumption.getSmartHomeType(),
consumption.getLocationId());
      MinMaxConsumption thisDayMinMax = thisDayMinMaxOptional.get();
      if (location.isEmpty() || location.get().getGridCompanyOrgNo() == null) {
        log.error("No location found with id " + consumption.getLocationId() + " or grid organization no is
null. Capacity levels are unknown.");
        capacityLevelCurrent = 0D;
        capacityLevelPredicted = 0D;
      } else {
        capacityLevelCurrent = consumptionService.getMinKwh(location.get().getGridCompanyOrgNo() ,
thisDayMinMax.getValue());
        capacityLevelPredicted = consumptionService.getNextLevel(location.get().getGridCompanyOrgNo() ,
capacityLevelCurrent);
      }
      maxHour1 = thisDayMinMax.getMax1();
      maxHour2 = thisDayMinMax.getMax2();
      maxHour3 = thisDayMinMax.getMax3();
      meanMax = thisDayMinMax.getValue();
      Double[] max = new Double[] {maxHour1, maxHour2, maxHour3};
      Integer dayMinIndex = thisDayMinMax.getDayMinIndex();
      maxAbsolute = consumptionService.getMaxAbsolute(max, dayMinIndex, capacityLevelCurrent);
      maxRecommended = consumptionService.getMaxRecommended(maxAbsolute);
      capacityLevelMaxToStayInCurrent =
consumptionService.getConsumptionBeforeGoingOverMax(maxAbsolute, hourlyConsumption);
      maxAbsoluteInHour = consumptionService.getMaxAbsoluteInHour(consumption,
capacityLevelMaxToStayInCurrent);
      maxAbsoluteOffsetInHour = consumptionService.getMaxAbsoluteOffsetInHour(maxAbsoluteInHour);
      maxAbsoluteRecommendedCalculatedInHour =
```

```
consumptionService.getMaxAbsoluteRecommendedInHour(maxAbsoluteInHour);
        maxAbsoluteRecommendedInHour = maxAbsoluteOffsetInHour - (0.2 * capacityLevelCurrent);
    }

    return new PowerGrid(burnRate, sampleStart, sampleLeft, hourlyConsumptionPredicted,
        capacityLevelCurrent, capacityLevelPredicted, capacityLevelMaxToStayInCurrent,
        maxAbsolute, maxRecommended, maxAbsoluteInHour, maxAbsoluteOffsetInHour,
        maxAbsoluteRecommendedInHour, maxAbsoluteRecommendedCalculatedInHour, maxHour1,
maxHour2, maxHour3, meanMax);
  }

  private double getRealtimeHourlyConsumption(RegularConsumption consumption) {
    if (Instant.now().isAfter(getTimeAtStartOfHour(consumption.getTime()).plus(1, ChronoUnit.HOURS))) {
      return
storage.findHourlyConsumptionByTagsAndTime(SmartHomeType.valueOf(consumption.getSourceSystem()),
consumption.getLocationId(), getTimeAtStartOfHour(consumption.getTime()))
          .map(HourlyConsumption::getValue)
          .orElse(0.0);
    }
    return calcRealtimeHourlyConsumption(SmartHomeType.valueOf(consumption.getSourceSystem()),
consumption.getLocationId(), consumption.getTime());
  }

  private double calcRealtimeHourlyConsumption(SmartHomeType sourceSystem, String locationId, Instant
time) {
    return storage.findAllMinuteConsumptions(sourceSystem, locationId, getTimeAtStartOfHour(time),
getTimeAtEndOfHour(time)).stream()
        .map(MinuteConsumption::getValue)
        .reduce(0.0, Double::sum);
  }

  private double getRealtimeHourlyCosts(RegularConsumption consumption) {
    if (Instant.now().isAfter(getTimeAtStartOfHour(consumption.getTime()).plus(1, ChronoUnit.HOURS))) {
      return
storage.findHourlyConsumptionByTagsAndTime(SmartHomeType.valueOf(consumption.getSourceSystem()),
consumption.getLocationId(), getTimeAtStartOfHour(consumption.getTime()))
          .map(HourlyConsumption::getHourlyCost)
          .orElse(0.0);
    }
    return calcRealtimeHourlyCosts(SmartHomeType.valueOf(consumption.getSourceSystem()),
```

```
consumption.getLocationId(), consumption.getTime());
  }

  private double calcRealtimeHourlyCosts(SmartHomeType sourceSystem, String locationId, Instant time) {
    return storage.findAllMinuteConsumptions(sourceSystem, locationId, getTimeAtStartOfHour(time),
getTimeAtEndOfHour(time)).stream()
        .map(MinuteConsumption::getMinuteCost)
        .reduce(0.0, Double::sum);
  }

  private double calcDailyConsumptionTillLastHour(SmartHomeType sourceSystem, String locationId,
Instant time) {
    int hour = time.atZone(ZONE_ID).toLocalDateTime().getHour();
    if (hour < 1) return 0.0;
    List<HourlyConsumption> hourlyConsumptions = storage.findAllHourlyConsumptions(sourceSystem,
locationId, getTimeAtStartOfDay(time), getTimeAtEndOfHour(time.minus(1, ChronoUnit.HOURS)));

    return hourlyConsumptions.stream()
        .map(HourlyConsumption::getValue)
        .reduce(0.0, Double::sum);
  }

  private double calcDailyCostsTillLastHour(SmartHomeType sourceSystem, String locationId, Instant time)
{
    int hour = time.atZone(ZONE_ID).toLocalDateTime().getHour();
    if (hour < 1) return 0.0;
    List<HourlyConsumption> hourlyConsumptions = storage.findAllHourlyConsumptions(sourceSystem,
locationId, getTimeAtStartOfDay(time), getTimeAtEndOfHour(time.minus(1, ChronoUnit.HOURS)));

    return hourlyConsumptions.stream()
        .map(HourlyConsumption::getHourlyCost)
        .filter(Objects::nonNull)
        .reduce(0.0, Double::sum);
  }

}
```

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: *Комплексна магістерська кваліфікаційна робота «Розробка автоматизованої системи обліку та аналізу споживання електроенергії в домогосподарствах.»*

Частина 2. Технічне проектування

Тип роботи: кваліфікаційна робота

(кваліфікаційна робота, курсовий проект (робота), реферат, аналітичний огляд, інше – зазначити)

Підрозділ: кафедра АІІТ, ФІІТА, 1АКІТ-21м

(кафедра, факультет, навчальна група)

Науковий керівник: Кулик Я.А., доц. каф. АІІТ

(прізвище, ініціали, посада)

Показники звіту подібності

Plagiat.pl (StrikePlagiarism)		*Unicheck*	
КП1		Оригінальність	__%
КП2			
Тривога/Білі знаки	/	Схожість	__%

Аналіз звіту подібності (відмітити потрібне)

X Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

Заявляю, що ознайомлений (-на) з повним звітом подібності, який був згенерований Системою щодо роботи (додається)

Автор _____ Рудич Є.О.
(підпис) (прізвище, ініціали)

Опис прийнятого рішення:

Допустити до захисту

Особа, відповідальна за перевірку _____ Маслій Р.В.
(підпис) (прізвище, ініціали)