Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(назва освітнього ступеня)

на тему: Розробка програмної системи управління енергоспоживанням міських об’єктів розумного міста

Виконав: студент IV курсу, групи СН-41
спеціальності 122 Комп’ютерні науки
(шифр і назва спеціальності)

_____ Денисюк І.М.
(підпис) (прізвище та ініціали)

Керівник _____ Палка О.В.
(підпис) (прізвище та ініціали)

Нормоконтроль _____ Шимчук Г.В.
(підпис) (прізвище та ініціали)

Завідувач кафедри _____ Боднарчук І.О.
(підпис) (прізвище та ініціали)

Рецензент _____ Мудрик І.Я.
(підпис) (прізвище та ініціали)

Тернопіль
2026

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)
«_8_» _червня_ 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп’ютерні науки
(шифр і назва спеціальності)

Студенту Денисюку Івану Михайловичу
(прізвище, ім’я, по батькові)

1. Тема роботи Розробка програмної системи управління енергоспоживанням міських об’єктів розумного міста

Керівник роботи Палка Олег Вікторович, доктор філософії, асистент кафедри КН
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «*14*» *травня* 2026 року № *4/9-239*

2. Термін подання студентом завершеної роботи 22 червня 2026 р.

3. Вихідні дані до роботи Технічне завдання, літературні джерела

4. Зміст роботи (перелік питань, які потрібно розробити) Вступ. 1 Аналіз предметної області та постановка завдання. 1.1 Концепція розумного міста та роль енергоменеджменту. 1.2 Дослідження характеристик енергоспоживання об’єктів міської інфраструктури. Аналіз та класифікація систем енергомоніторингу. 1.4 Деталізація вимог до програмної системи. 1.5 Обгрунтування постановки задачі дипмлоної роботи. 1.6 Висновок до 1 розділу. 2 Проектування програмної системи управління енергоспоживанням. 2.1 Архітектура програмної системи. 2.2 Проектування бази даних на основі MongoDB. 2.3 Моделювання Функціональних процесів системи. 3.Реалізація та дослідження програмної системи. 3.1 Обгрунтування технологій та реалізація моделя зберігання і обробки даних. 3.2 Реалізація модуля аналізу енергоспоживання. 3.3 Реалізація графічного інтерфейсу користувача. 3.4 Тестування та аналіз результатів роботи системи. 3.5 Висновок до 3 розділу. 4. Безпека життєдіяльності, основи охорони праці. Висновки. Список використаних джерел. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	к.т.н,, доц. кафедри МТ Гурик О.Я.	15.05.2026	29.05.2026

7. Дата видачі завдання 15 травня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	26.01.2026	*Виконано*
2.	Аналіз наукових джерел та сучасних технологій управління енергоспоживанням	27.01.2026-07.02.2026	*Виконано*
3.	Виконання дослідження енергоспоживання міських об'єктів. Розроблення програмної системи управління енергоспоживанням	10.02.2026-28.02.2026	*Виконано*
4.	Оформлення розділу «Аналіз предметної області та	02.03.2026-21.04.2026	*Виконано*
	постановка задачі»		
5.	Оформлення розділу «Проектування програмної	23.04.2026-03.05.2026	*Виконано*
	системи управління енергоспоживанням»		
6.	Оформлення розділу «Реалізація та дослідження	25.05.2026-31.05.2026	*Виконано*
	програмної системи»		
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»	15.05.2026-28.05.2026	*Виконано*
8.	Виконання завдання до підрозділу «Основи охорони праці»	18.05.2026-29.05.2026	*Виконано*
9.	Оформлення кваліфікаційної роботи	01.06.2026-05.06.2026	*Виконано*
10.	Нормоконтроль	12.06.2026-15.06.2026	*Виконано*
11.	Перевірка на плагіат	16.06.2026	*Виконано*
12.	Попередній захист кваліфікаційної роботи	18.06.2026	*Виконано*
13.	Захист кваліфікаційної роботи	22.06.2026	

Студент _____ Денисюк І.М.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Палка О.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Розробка програмної системи управління енергоспоживанням міських об’єктів розумного міста // Денисюк Іван // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедра комп’ютерних наук, група СН-41 // Тернопіль, 2026 // С. 59, рис. – 17, табл. – 4, додат. – 1, бібліогр. – 43.

Ключові слова: розумне місто, енергоспоживання, моніторинг, аналіз даних, MongoDB, Python, інформаційна система, управління ресурсами.

Кваліфікаційна робота присвячена розробці програмної системи управління енергоспоживанням міських об’єктів розумного міста.

У першому розділі кваліфікаційної роботи описано концепцію "розумного міста" та особливості управління енергоспоживанням міських об’єктів. Висвітлено сучасні підходи до моніторингу енергетичних ресурсів. Розглянуто існуючі програмні рішення та технології, що застосовуються для збору, обробки й аналізу даних про енергоспоживання. Проведено аналіз аналогів та визначено вимоги до програмної системи.

У другому розділі кваліфікаційної роботи виконано проєктування програмної системи. Досліджено функціональні та нефункціональні вимоги до програмного забезпечення. Подано архітектуру системи, структуру бази даних MongoDB, модель взаємодії програмних модулів та алгоритми обробки інформації. Розроблено схему інформаційних потоків та логіку функціонування системи.

У третьому розділі кваліфікаційної роботи описано реалізацію програмної системи засобами мови програмування Python. Проаналізовано особливості використання MongoDB для зберігання даних, бібліотеки PyMongo для роботи з базою даних та фреймворку PyQt5 для створення графічного інтерфейсу користувача. Реалізовано модулі збору, зберігання, обробки та аналізу даних про енергоспоживання міських об’єктів. Проведено тестування

системи, побудовано графіки споживання енергії та виконано аналіз результатів роботи програмного забезпечення.

Об’єкт дослідження: процес управління енергоспоживанням міських об’єктів в умовах функціонування концепції "розумного міста".

Предмет дослідження: методи, моделі та програмні засоби збору, зберігання, аналізу та візуалізації даних про енергоспоживання міських об’єктів.

ANNOTATION

Development of a Software System for Energy Consumption Management of Smart City Urban Facilities // Ivan Denysyuk // Ivan Puluj Ternopil National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, Group SN-41 // Ternopil, 2026 // P. 59, Fig. – 17, Tables – 4, Appendices – 1, References – 43.

Keywords: smart city, energy consumption, monitoring, data analysis, MongoDB, Python, information system, resource management.

The qualification thesis is devoted to the development of a software system for managing energy consumption of smart city facilities. The relevance of the research is determined by the need to improve the efficiency of energy resource utilization, reduce costs, and implement modern digital technologies for monitoring and analyzing energy consumption.

The first chapter describes the concept of a smart city and the specific features of energy consumption management in urban facilities. Modern approaches to energy resource monitoring are highlighted. Existing software solutions and technologies used for collecting, processing, and analyzing energy consumption data are reviewed. An analysis of existing analogues is carried out, and the requirements for the software system are defined.

The second chapter presents the design of the software system. Functional and non-functional requirements for the software are investigated. The system architecture, MongoDB database structure, interaction model of software modules, and information processing algorithms are described. An information flow scheme and the system operation logic are developed.

The third chapter describes the implementation of the software system using the Python programming language. The features of using MongoDB for data storage, the PyMongo library for database interaction, and the PyQt5 framework for

developing the graphical user interface are analyzed. Modules for collecting, storing, processing, and analyzing energy consumption data of urban facilities are implemented. System testing is conducted, energy consumption graphs are generated, and the results of the software operation are analyzed.

Object of research: the process of energy consumption management of urban facilities within the framework of the smart city concept.

Subject of research: methods, models, and software tools for collecting, storing, analyzing, and visualizing energy consumption data of urban facilities.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

CRUD (англ. *Create, Read, Update, Delete*) – створення, читання, оновлення та видалення даних.

IoT (англ. *Internet of Things*) – Інтернет речей.

API (англ Application Programming Interface) – інтерфейс програмування застосунків.

KPI (англ. *Key Performance Indicators*) – ключові показники ефективності

UI (англ. *User Interface*) – інтерфейс користувача.

UX (англ. *User Experience*) – користувацький досвід.

UML (англ. *Unified Modeling Language*) – уніфікована мова моделювання.

ЗМІСТ

ВСТУП 10

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ 12

1.1 Концепція розумного міста та роль енергоменеджменту 12

1.2 Дослідження характеристик енергоспоживання об'єктів міської інфраструктури 13

1.3 Аналіз та класифікація систем енергомоніторингу 17

1.4 Деталізація вимог до програмної системи 22

1.5 Обґрунтування постановки задачі дипломної роботи 23

1.6 Висновок до 1 розділу 24

РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ УПРАВЛІННЯ ЕНЕРГОСПОЖИВАННЯМ 25

2.1 Архітектура програмної системи 25

2.1.1 Вибір архітектурного підходу 28

2.2 Проектування бази даних на основі MongoDB 30

2.2.1 Обґрунтування вибору NoSQL для реалізації логічної моделі даних 30

2.2.2 Проектування структури бази даних та її оптимізація 31

2.3 Моделювання функціональних процесів системи 33

2.3.1 Сценарії використання (Use Case) та UML-діаграми системи 33

2.3.2 Опис бізнес-логіки системи 36

2.4 Проектування графічного інтерфейсу користувача 37

2.5 Висновок до 2 розділу 39

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ ПРОГРАМНОЇ СИСТЕМИ 41

3.1 Обґрунтування технологій та реалізація модуля зберігання і обробки даних 41

3.2 Реалізація модуля аналізу енергоспоживання 43
3.3 Реалізація графічного інтерфейсу користувача 44
3.4 Тестування та аналіз результатів роботи системи 46
3.5 Висновок до 3 розділу 47
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ 49
4.1 Ергономічні проблеми безпеки життєдіяльності 49
4.2 Заходи щодо автоматизації виробничих процесів, які сприяють покращенню умов праці.....50
4.3 Висновок до 4 розділу 52
ВИСНОВКИ.....53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 55
ДОДАТКИ

ВСТУП

Актуальність теми. Міста стають розумнішими швидкими темпами, проте разом з ними росте і обсяг споживаної в них енергії. Сьогоднішній мегаполіс, це не тільки безліч комп'ютерів і мобільних пристроїв, це ще і глобальна енергомережа, яка забезпечує роботу комп'ютерів і мобільних пристроїв у місті. Тому актуальним сьогодні є питання контролю містом своїх енергоресурсів.

Мета і задачі дослідження. Об'єктом дослідження є споживання енергії в об'єктах нерухомості, що входять до інфраструктури міст. Предметом ж цього дослідження є методи та інструменти програмного забезпечення для збору та автоматизованої обробки інформації про стан, і ефективність споживання енергії в будівлях. Мета роботи полягає в розробці програмної системи управління енергоспоживанням будівель у межах концепції розумного міста. Системі повинна бути під силу виконувати повний цикл операції з даними починаючи від автоматизованого збору інформації з різноманітних джерел, її зберігання, обробки та аналізу, і навіть візуалізації отриманих результатів для підтримки прийняття управлінських рішень.

Досягнення цієї мети потребує вивчення сучасних рішень при проектуванні систем моніторингу та управління енергії в будівлях у межах «Розумних» міст; аналізу принципів роботи концепції «Розумного» міста, для обґрунтування можливих рішень по управлінню енергетичними ресурсами в межах цієї концепції, окреслення вимог до системи збору та обробки даних, а також до її архітектури, проектування самої інформаційної системи з урахуванням її масштабності, надійності та можливостей її майбутнього розширення. Окрему увагу надають проектуванню сховища даних для енергетичних характеристик будівель, розробці програмних модулів, які призначені для збору, обробки та візуалізації даних, для аналітичної обробки даних з можливістю виділення як визначених закономірностей, так і відхилень

у споживанні енергії, а так само тестуванню та оцінці програмного забезпечення за різноманітними сценаріями;

Практичне значення одержаних результатів. Практичне значення одержаних результатів полягає в створенні програмної системи, що дозволяє автоматизувати процеси моніторингу та аналізу енергоспоживання об’єктів міської інфраструктури. Запропоноване рішення може бути використане для підвищення ефективності управління енергетичними ресурсами, зменшення втрат та оптимізації витрат, а також для підтримки прийняття обґрунтованих управлінських рішень у сфері енергоменеджменту.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Концепція розумного міста та роль енергоменеджменту

У всьому світі люди живуть і працюють у містах. Цифрова трансформація і раціональне керування муніципальними ресурсами стали гострою проблемою. За останні даними, понад половина населення планети (більше 55%), мешканці міст. Очікується, що частка людей, які живуть у містах, зросте до 68% у 2050 році. Енергетична інфраструктура сильно навантажена, інтенсивно зростає міське населення, однак значна частина енергетичних систем застарілі і не в змозі витримати підвищене навантаження [1].

За цих обставин поняття «розумного» міста модель якого зображена на рисунку 1.1, вже не сприймається лише як нова порція технологій, а розглядається як повна стратегія сталого розвитку, орієнтована на координацію міських процесів за допомогою розумних систем. Управління енергією щось на зразок "кровоносної системи" цієї екосистеми, тому що ефективне управління означає розумне виробництво через створення мікро мереж для використання сонячної та вітрової енергії, інноваційний розподіл із застосуванням технологій, таких як «розумна» мережа, що володіє здатністю до самовідновлення після збоїв, а також збалансоване споживання через програмні платформи, які роблять для кінцевих користувачів легшим процес подання та розуміння витрат у режимі реального часу [2].

Діючі стандарти — муніципалітети можуть створювати чіткі процедури для покращення енергетичних показників, використання та споживання ресурсів відповідно.

Рисунок 1.1 – Основні складові концепції «Розумного» міста [2]

Головні ж бар'єри для впровадження цих стандартів — дефіцит доступного програмного забезпечення, здатного об'єднати тисячі розрізнених лічильників у єдину аналітичну панель для ефективного моніторингу [3] [4].

1.2 Дослідження характеристик енергоспоживання об'єктів міської інфраструктури

Процеси енергоспоживання в сучасних муніципальних системах є динамічними, неоднорідними та залежать від багатьох факторів. Проектування ефективного програмного забезпечення у цій сфері вимагає попереднього детального дослідження структури енергетичних витрат, обґрунтування ключових показників ефективності (KPI) та виявлення технічних і організаційних перешкод, які обмежують раціональне водо- та енергокористування.

Енергоспоживання у муніципальному секторі настільки різняться, що можна навіть подумати, що єдина модель аналізу енергоспоживання для муніципальних будівель просто не існує. На рисунку 1.2 зображена схема

класифікації міських об'єктів. Почнемо із розподілу будівель за типом і режимом споживання енергії на наступні типи [5]:

Об’єкти адміністративно-соціальної сфери (управління, ЦНАПи): Зазвичай це «параболоїдний» графік споживання. Пік споживання, о 8-й ранку, велика кількість комп'ютерів, світло, кондиціонування, вентиляція, все це « з’їдає » енергію протягом усього робочого дня. Але в цьому ваша сила: можна заощадити, адже вентиляції і кондиціонування не потрібно працювати вночі.

Навчальні заклади (школи, дитячі садки тощо): Зазвичай це будівлі із схожим графіком споживання, як в адміністративних будівлях, але з істотним перепадом між літнім та зимовим сезоном (через канікули) і строгими вимогами з точки зору санітарних правил і нормщодо регулювання клімату при вході дітей

Заклади охорони здоров'я (лікарні і т.д.): Зазвичай це будівлі, в них «не знають» часу. Тут ми маємо навантаження, яке «завжди» споживається і це « база». Відповідно, велика частина обладнання, постійний споживач енергії, її не відключиш.

Житловий фонд : У цих будівлях два пікові навантаження, вранці (7,00 9,00) і ввечері (18,00-23,00). В особливості цієї групи будівель в тому, що обсяги споживання значною мірою залежать від рухливої складової, тобто від «людського фактора», від кількості членів сім'ї, від способу життя і т. п. Відповідно, в існує велика різниця між втратою енергії в різних житлових будинках.

Об’єкти комунального господарства (вуличне освітлення, комунальний транспорт і т.д.): Особливі з точки зору графіків, це будівлі або з інверсним графіком (наприклад, вуличне освітлення, максимум енергії вночі) або імпульсним (за графіком роботи міського електротранспорту).

Для побудови моделі у програмному забезпеченні використовуються наступні характеристики [7].

Рисунок 1.2 – Класифікація міських об’єктів за типом енергоспоживання [6]

Специфічне споживання енергії, це показник, що визначає обсяг електроенергії, спожитої на одиницю площі будівлі. Цей показник служить базовим орієнтиром для порівняння ефективності використання енергії.

Максимальне навантаження, це максимальна потужність споживання за певний час. Цей показник є ключовим для запобігання перевантаженню електромереж.

Коефіцієнт використання профілю навантаження, це відношення середнього значення навантаження до максимального навантаження. Чим більше цей коефіцієнт наближається до 1, тим більш рівномірною і ефективною є робота системи.

Основне навантаження, це мінімальне, базове споживання електроенергії будівлею, яке зберігається навіть у нічний час. Якщо цей показник має занадто високі значення, це може свідчити про витоки електроенергії або про несправність обладнання.

Аналіз проблемної області дозволив виявити деякі фундаментальні проблеми, які є істотним гальмівним фактором для впровадження та підвищення рівня енергоефективності у містах:

Фрагментарність даних: інформація про електроенергію, газ і воду, як правило, зберігається у різних підрозділах і у різних форматах (Excel-файли, паперові записи, окремі бази даних). Це створює серйозні проблеми для об'єднання інформації та її аналізу.

Відсутність своєчасності: у більшості випадків муніципальні будівлі збирають показники лічильників лише раз на місяць. Це дуже обмежує можливість своєчасного виявлення аварійних збільшень споживання, наприклад, у разі аварійного прориву труби або короткого замикання.

Помилки при ручному введенні: ручне введення даних часто є джерелом неточностей, що знижує достатність звітності і негативно впливає на її достовірність.

Проблема "останньої милі": віддаленість точок обліку і низька стабільність каналів передачі даних у віддалених місцях ускладнює утворення єдиного інформаційного простору системи.

Впровадження спеціалізованого програмного забезпечення вважається одним з ключових напрямків виходу з ситуації, описаної вище. У рамках концепції Розумного міста автоматизована система дає такі можливості [8]:

Прозорість: створення єдиного інформаційного простору для всіх рівнів управління від відповідальних працівників організацій до міської адміністрації.

Передбачувана аналітика: можливість прогнозувати витрати бюджетних коштів на енергоносії з урахуванням погодних умов, сезонності та змін тарифів.

Виявлення аномалій: автоматизоване визначення відхилень від нормальних режимів споживання з оперативним інформуванням відповідальних осіб.

Економічний ефект: реальні дослідження показують, що навіть без модернізації обладнання, лише за рахунок інформаційного контролю можна досягти економії енергії в розмірі 5-12% за рахунок підвищення дисципліни споживання.

На рисунку 1.3. наведено архітектурну модель потоків даних, яка ілюструє процес збору, обробки та візуалізації показників енергоспоживання в режимі реального часу.

Рисунок 1.3 – Загальна схема системи енергомоніторингу [9]

Використання сучасних стеків MQTT та WebSocket дозволяє одержувати та обробляти метрики з високою швидкістю, що дає змогу приймати рішення в режимі онлайн.

1.3 Аналіз та класифікація систем енергомоніторингу

Сучасні розробки програмного забезпечення у сфері енергоменеджменту є комплексами. Їх можна розділити за кількома критеріями, що дає можливість вибирати найбільш прийнятний варіант для створення своєї програми.

За способом створення архітектури:

On-premise: програмне забезпечення встановлюється на внутрішніх серверах компанії. Це дозволяє забезпечити максимальний захист даних, але, в той же час, обмежує масштабованість і ускладнює віддалений доступ до системи [10].

SaaS: обробка і зберігання даних відбувається в хмарі (наприклад, на платформі AWS чи Microsoft Azure). Це усуває потребу в підтримці власної серверної інфраструктури і дозволяє здійснювати доступ до аналітики через веб-інтерфейс з будь-якого пристрою.

За показником «інтелектуальність»:

Пасивні системи: зосереджені на здійсненні збору, зберігання і візуалізації аналітичних даних і не проводять їх подальшу обробку. Активні системи: використовують інтелектуальні технології, зокрема машинне навчання, для реалізації прогнозної аналітики. Наприклад, прогнозне рішення може бути таким: » через 2 години « внаслідок підвищення температури повітря « споживання енергії на охолодження приміщень збільшиться на 20% « і з певним випередженням затребує налаштувати роботу кондиціонерів відповідним чином « [11].

Протоколи обміну даними з пристроями червоніють набути вагоме значення. В рамках «Розумного» міста особливої важливості набувають економічні протоколи, такі як LoRaWAN та MQTT, оскільки вони забезпечують тривалий ресурс незалежної роботи давачів і здатні транспортувати дані на великі відстані навіть в умовах щільної забудови міста.

При розробці програмної системи управління енергоспоживанням вибір архітектурного підходу є одним з ключових факторів, що визначає масштабованість і довгострокову ефективність рішення. У.сучасних дослідженнях відзначається тенденція переходу від монолітних до мікросервісних архітектур [12].

1. Монолітна архітектура: всі функції системи (збір, обробка, аналітика, користувацький інтерфейс) реалізовані в межах однієї програми. Такий підхід полегшує процес розробки і впровадження, проте має недоліки, зокрема, поломка одного компонента суттєво позначається на роботі всієї системи.

2. Мікросервісна архітектура: система складається з набору незалежних сервісів, кожен з яких відповідає за окрему функцію (наприклад, обробка данних IoT-пристроїв або аналітичні розрахунки). Така структура забезпечує

гнучкість і можливість горизонтального масштабування: при зростанні кількості об'єктів можно збільшувати ресурси лише для окремих сервісів без заміни всю систему.

Важливим є і вибір системи керування базами даних. У сфері енергоспостережень традиційні реляційні СУБД (SQL) часто виявляються неефективними, коли маємо справу з великою кількістю часових даних. Так, якщо лічильники направляють показники що хвилину, за рік виходить мільйони записів [13].

Time-Series Databases: спеціалізовані бази даних, такі як InfluxDB або TimescaleDB, оптимізовані для зберігання та обробки даних з часовими мітками. Вони забезпечують значно швидше виконання аналітичних запитів, зокрема розрахунку середнього споживання за певний період.

NoSQL-рішення: застосовуються для зберігання неструктурованої або слабоструктурованої інформації, наприклад метаданих об'єктів, технічних характеристик будівель або описів обладнання.

Для проведення порівняльного аналізу функціональних можливостей існуючих систем енергомоніторингу корисно розглянути їх основні характеристики, наведені в таблиці 1.1.

Таким чином, проведений аналіз свідчить, що наявні програмні рішення володіють розвиненим функціоналом, проте часто мають суттєві обмеження. Зокрема, це може проявлятися у високій вартості впровадження або недостатній гнучкості для адаптації під специфічні вимоги конкретного користувача чи організації.

Оскільки система управління енергоспоживанням належить до критичної інфраструктури міста питання інформаційної безпеки набувають найбільшого значення. Згідно з дослідженнями, основними загрозами для подібних систем вважаються [14]:

Таблиця 1.1 – Порівняльна характеристика існуючих систем енергомоніторингу

Назва системи	**Основні функції**	**Переваги**	**Недоліки**
Siemens Desigo CC	Моніторинг енергоспоживання, автоматизація будівель, аналітика	Висока функціональність, підтримка великої кількості пристроїв	Висока вартість впровадження
Schneider Electric EcoStruxure	Управління енергоспоживанням, збір даних	Масштабованість, підтримка хмарних сервісів	Складність налаштування
Honeywell Building Management System	Моніторинг та керування інженерними системами	Надійність та стабільність роботи	Обмежена гнучкість
Open Energy Monitor	Аналіз енергоспоживання, візуалізація даних	Відкритий код, доступність	Обмежений функціонал
Розроблювана система	Збір даних, аналіз, формування звітів, візуалізація	Гнучкість, адаптація під потреби користувача	Потребує подальшого розвитку

Оскільки система управління енергоспоживанням належить до критичної інфраструктури міста питання інформаційної безпеки набувають найбільшого значення. Згідно з дослідженнями, основними загрозами для подібних систем вважаються [14]:

Введення підроблених даних: зловмисник може переписати дані з лічильників, що дозволяє приховувати фактичне перевищення споживання або, навпаки, створювати видиму брак енергоресурсів.

DDoS-атаки на шлюзи збору даних: перевантаження каналів передачі інформації з сенсорів призводить до того, що дані не є актуальними, а система фактично „втрачає видимість“ у режимі реального часу.

Несанкціонований доступ до керуючих модулів: доступ до адміністративних функцій може створити ризик віддаленого впливу на об’єкти енергопостачання, включно з їх вимкненням чи зміною режимів роботи.

Використання системи енергоменеджменту забезпечує вищу ефективність управління енергоресурсами і є вагомим фактором при прийняті управлінських рішень. Переваги такої системи зображені на рисунку 1.4.

Рисунок 1.4 – Основні переваги використання системи енергоменеджменту [15]

Також для зниження названих ризиків у системі повинні бути реалізовані механізми наскрізного шифрування даних та протоколи автентифікації відповідно до стандарту OAuth 2.0. Крім того, потрібне впровадження системи

детального журналювання дій користувачів, що створить умови для аудиту та виявлення потенційних інцидентів безпеки.

1.4 Деталізація вимог до програмної системи

Для реалізації повного циклу енергоменеджменту програма має набувати наступних функцій:

Детальний аналіз та специфікація функціональних вимог визначають ключові можливості розроблюваної системи.

Модуль ієрархічного управління об’єктами: включає класифікацію об’єктів за районами, типами (наприклад, школи, лікарні) або відомчою розподільністю. Для кожного об’єкта створюється цифрова картка, що включає основні технічні параметри: площу, тип утеплення, кількість персоналу тощо.

Інструменти порівняльної аналітики: програма автоматично порівнює споживання енергії аналогічними об’єктами. Якщо, наприклад, одна школа, у порівнянні з іншими аналогічними, використовує на 30% більше енергії, це вважається потенційним відхиленням або проблемою.

Гнучкий конструктор звітів: користувач отримує можливість самостійно створювати аналітичні звіти, обираючи параметри та види візуалізації, такі як лінійні графіки, теплові карти споживання, або кругові діаграми для розподілу витрат на електроенергію, газ, воду тощо.

Прогнозний модуль: на основі історичних даних та інтеграції з метеорологічними сервісами (API) система прогнозує споживання енергії на майбутні дні (тиждень або місяць), беручи до уваги погодні умови..

Особливу увагу приділено розгляду та опису нефункціональних вимог до програмних засобів.

Швидкодія: час обробки запиту на формування річного звіту не повинен перевищувати 3,5 сек.

Доступність: система має працювати без зупинок 24/7 із залишковим часом відмови не більше 10 хвилин на місяць (99.999%).

Адаптивність інтерфейсу – користувацький інтерфейс має коректно відображати аналітичні панелі (дашборди) як на мобільних пристроях інспекторів, так і на великих дисплейних екранах у диспетчерських центрах.

1.5 Обґрунтування постановки задачі дипломної роботи

Підсумовуючи виконане дослідження предметної області, дійдемо висновку, що попри наявність комерційних продуктів на ринку, актуальною залишається потреба у розробці вузькоспеціалізованого програмного забезпечення, пристосованого до українських реалій. Це, перш за все, стосується специфіки тарифоутворення, організаційної структури муніципального управління, а також фактичного технічного стану існуючих засобів обліку енергоресурсів.

Метою цієї роботи є проектування архітектури і розробка програмної системи, що має кілька взаємопов’язаних рівнів:

Нижній рівень – здійснює збір інформації безпосередньо з приладів обліку за допомогою шлюзів передачі даних (MQTT/HTTP).

Середній рівень – відповідає за інтелектуальну обробку отриманих даних і за їх зберігання у відповідних часових базах даних.

Верхній рівень: це веб-орієнтований інтерфейс для аналітиків, диспетчерів та керівників міських служб.

Запровадження такої архітектури дозволить перейти від моделі пасивного моніторингу до активного управління енергоспоживанням, що є важливим кроком у напрямі практичної реалізації концепції «Розумного» міста.

1.6 Висновки до першого розділу

Розглянуто предметну область, в межах якої одним із сучасних напрямків реалізації концепції "Розумного міста" є створення системи управління енергоспоживанням міської інфраструктури. Аналіз проведених досліджень особливостей енергоспоживання різних категорій об’єктів, сучасних підходів до моніторингу та застосовуваних технологій обробки даних виявив актуальну проблематику неповноти інформаційного поля, недостатньої оперативності контролю та складності обробки і об’єднання різнорідних джерел даних.

Доведено, що використання автоматизованих систем енергомоніторингу дає можливість підвищити ефективність використання енергоресурсів, своєчасно виявляти нетипові ситуації та підтримувати процес прийняття управлінських рішень. Підсумовуючи результати дослідження сформульовані вимоги до програмної системи управління енергоспоживанням об’єктів міської інфраструктури, які використовуються у подальшому при проектуванні та розробці відповідного програмного забезпечення.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ УПРАВЛІННЯ ЕНЕРГОСПОЖИВАННЯМ

2.1 Архітектура та структура програмної системи

Вибір архітектури програмного комплексу є ключовим етапом, що задає напрямок усій подальшій розробці і використанню системи в контексті мінливих потреб міського господарства. Архітектура має не тільки відповідати поточним функціональним вимогам, а й бути гнучкою і піддатливою до розширення і змін.

Загальна архітектура запропонованої системи базується на багаторівневій конструкції, що дозволяє чітко розмежувати зони відповідальності. Ця архітектура передбачає :

Рівень апаратних пристроїв: включає мережу смарт-лічильників і шлюзів. На цьому рівні пристрої безпосередньо вимірюють і збирають електричні, водні, газові та інші параметри споживання, попередньо обробляють дані.

Рівень мережевої взаємодії : відповідає за транспортування даних від пристроїв до центральної системи. На цьому рівні реалізуються протоколи та методи зв’язку, оптимізовані для нестабільних мереж або вузькосмугових ліній, наприклад MQTT або LoRaWAN.

Рівень застосунків і бізнес-логіки: є серцем системи і представляється у вигляді набору мікросервісів, що працюють у контейнерній оркестрації. Кожен мікросервіс відповідає за конкретний функціонал, валідацію даних, розрахунки енергоефективності відповідно до ISO 50001, прогнозування навантажень, керування правами доступу користувачів.

Рівень зберігання даних: спирається на концепцію комбінованого зберігання даних (Polyglot Persistence). Реляційні дані про користувачів, міські об’єкти і т.д. зберігаються у PostgreSQL, тоді як час від часу і як правило з великою дискретизацією збирається телеметрія. Вона фіксується в тайм-серії базі даних TimescaleDB.

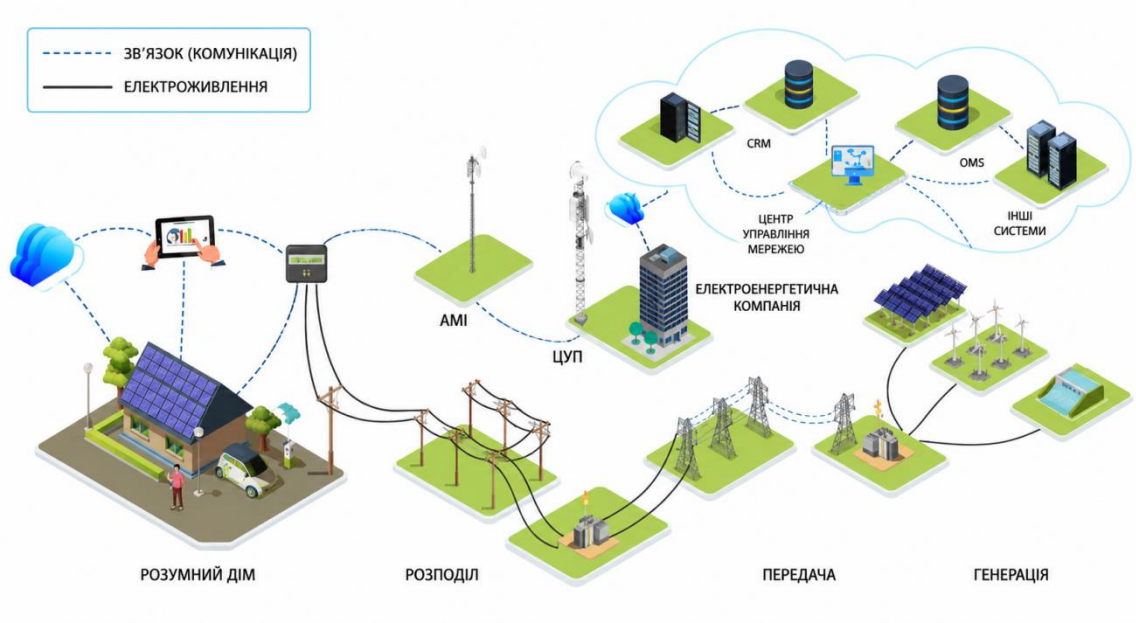

Рисунок 2.1 – Архітектура програмної системи [16]

Рівень представлення: реалізований у вигляді адаптивного веб-застосунку, що дозволяє користувачу взаємодіяти з системою за допомогою графіків і карт.

Внутрішня організація обміну даними між компонентами системи реалізується на базі комбінованої моделі, яка поєднує синхронні та асинхронні взаємодії, на рисунку 2.2 зображена схема такої системи.

Асинхронний обмін даними через шину повідомлень: Основа передачі телеметрії асинхронний обмін даними через обмінник повідомлень (наприклад Kafka). Після того, як дані від приладів обліку отримує відповідний шлюз, він формує і публікує події у чергу повідомлень. Таким чином обробка даних відбувається відокремлено від запису в базу. Якщо запис перевантажений або іноді не досягається, сервер продовжує отримувати дані, накопичуючи їх у черзі без втрат.

Синхронний обмін даними через API Gateway : Операції, що вимагають негайної відповіді користувачу (наприклад, авторизація або змін налаштувань системи), реалізуються за допомогою синхронного REST API. API Gateway є єдиною точкою входу у систему, через яку організується маршрутизація запитів

до відповідних сервісів. Також він виконує роль балансувальника навантаження і початкового контролю безпеки запитів.

Оновлення інформації в реальному часі за допомогою WebSockets : Для мінімізації часу від зміни інформації до її відображення на екрані оператора використовується двонаправлений канал зв’язку WebSockets. Це дає змогу серверній частині миттєво відправляти оновлені дані без необхідності перезапитувати або оновлювати сторінку, що дозволяє отримувати практично миттєво всі графіки після обробки аналітичними модулями [17].

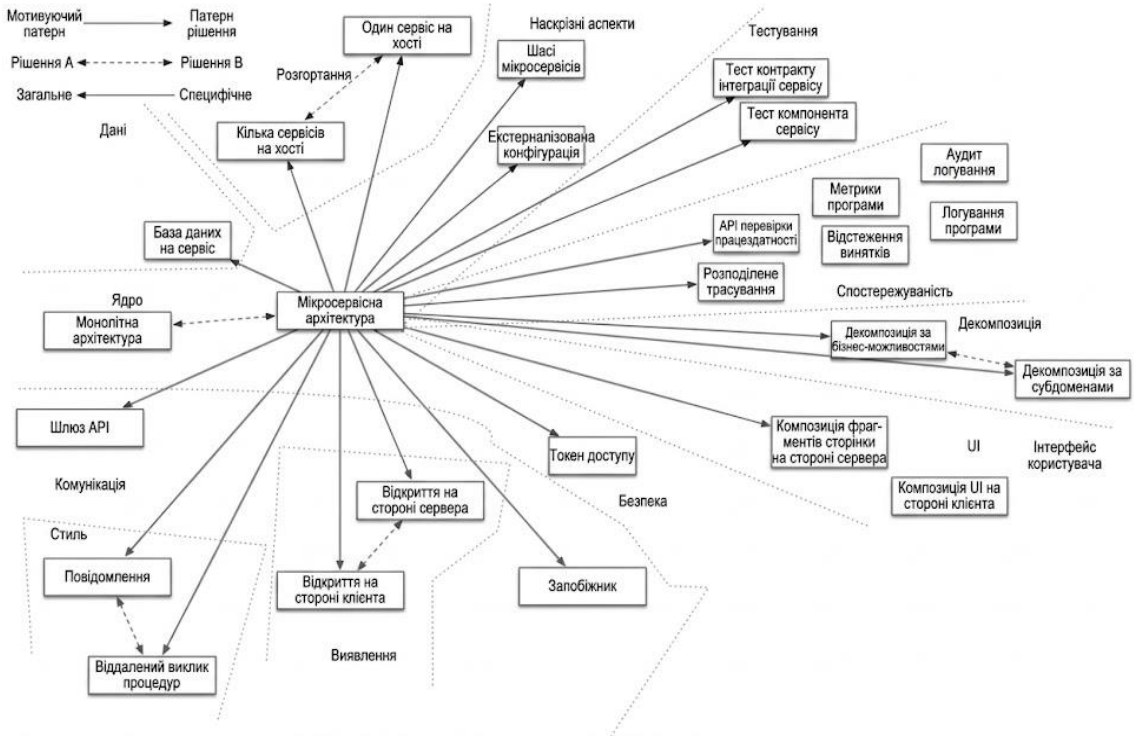

Рисунок 2.2 – Схема взаємодії компонентів програмної системи [17]

Така взаємодія дозволяє системі безперервно працювати навіть що одночасно підключено тисячі лічильників, що дуже важливо для великих муніципальних систем.

2.1.1 Вибір архітектурного підходу

На початковому етапі проектування було проведено аналіз існуючих архітектурних підходів. Основною дилемою було вибрати між монолітом і мікросервісами. Рисунок 2.3 ілюструє порівняння монолітної та мікросервісної архітектур. Ураховуючи, що система повинна цілодобово (24/7) збирати телеметрію і обробляти великі обсяги даних, було зроблене рішення на користь мікросервісної архітектури.

За результатами вивчень, мікросервісний підхід дає змогу розбити складну бізнес-логіку на сукупність автономних, слабо пов’язаних сервісів. Це приносить чимало переваг [18]:

Технологічна автономність (технологічна ізоляція): дає можливість обирати різні технологічні інструменти для різних частин системи, наприклад для високопродуктивного збору даних писати на Go, а для аналітичних обчислень використовувати Python.

Живучість (відмовостійкість): збій, наприклад, у модулі звітності, не зачіпає всю систему і не зупиняє потік даних від лічильників.

Контенеризація : використання Docker дає можливість створювати стандартизоване середовище для контейнерних застосунків, завдяки чому програмні продукти поводяться однаково як на локальних машинах розробників, так і на промислових серверах.

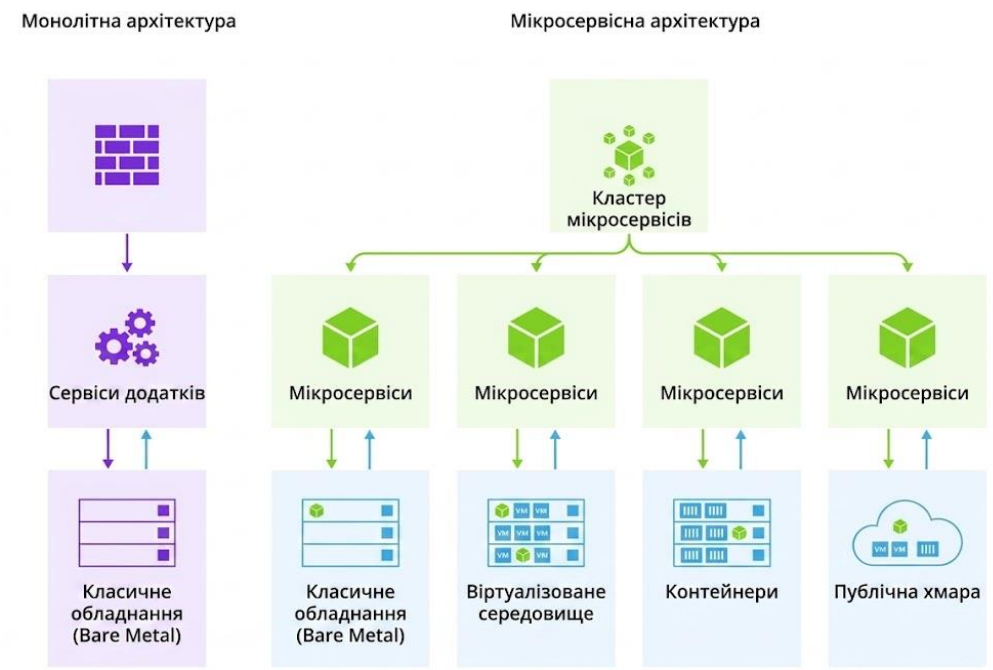

Рисунок 2.3 – Порівняння монолітної та мікро сервісної архітектури [19]

Для того, щоб система була живучою і адаптивною при розробці, було обрано набір технологічних інструментів, які в основному орієнтовані на сучасні технології потоків даних у реальному часі. Кожен інструмент вибирався з урахуванням вимог системи і правил її роботи, від первинного джерела даних, тобто від давачів, до її інтерфейсного відображення для користувача. Основні мови програмування і бібліотеки, що є необхідною умовою для реалізації технологічної платформи запропонованого рішення, подано у таблиці 2.1.

Таблиця 2.1 – Основні технології, використані при розробці системи

Технологія	**Призначення**
Python	Реалізація логіки програмної системи
MongoDB	Зберігання даних про енергоспоживання
PyMongo	Взаємодія Python із MongoDB
Tkinter / PyQt	Реалізація графічного інтерфейсу

Продовження таблиці 2.1

Технологія	Призначення
Matplotlib	Побудова графіків та візуалізація даних
Pandas	Аналіз та обробка даних

Застосування вказаних технології дозволяє досягти ефективності роботи системи, зручності для розробника і можливості масштабування програмного продукту.

2.2 Проектування бази даних на основі MongoDB

2.2.1 Обґрунтування вибору NoSQL для реалізації логічної моделі даних

Складність вибору певної системи керування базами даних полягає в необхідності ефективної обробки великих потоків телеметрії у реальному часі. Класичні реляційні бази даних вимагають суворо визначеної структури даних, що є перешкодою для їх використання з різними типовими IoT приладами або сенсорами. Використання MongoDB дає змогу застосувати більш гнучку модель зберігання даних, орієнтовану на документи, яка, як свідчать дослідження, є природнішою при роботі з напівструктурованими даними у форматі BSON [20].

Однією з головних переваг NoSQL у цьому випадку є підтримка горизонтального масштабування за рахунок шардінгу. Це дозволяє розподіляти навантаження по декількох серверах або вузлах без зниження продуктивності операцій запису. Враховуючи те, що кількість лічильників у системі може постійно зростати, можливість MongoDB динамічно змінювати структуру без паузи в роботі стане в нагоді для моніторингу міської інфраструктури.

Побудова логіки зберігання даних відбувається на основі компромісу між повною нормалізацією і продуктивністю запитів. Замість того, щоб сильно нормалізувати структуру за рахунок створення величезної кількості таблиць та зв’язків, що буде створювати колосальне навантаження при запитах з вибірками, було прийнято рішення нормалізувати тільки те, що стосується часто використовуваної метаінформації, а у великих наборах вимірювань виконувати нормалізацію за допомогою посилань.

Структура даних побудована згідно з принципом ієрархічної моделі об’єктів. Кожен документ, що стосується фасиліті, має список ідентифікаторів пов’язаних лічильників. Таким чином можна отримати повну картину споживання енергії конкретної будівлі за один запит до бази. Такий підхід скорочує кількість вводу-виводу (I/O) запитів, що є критичним при побудові інтерактивних аналітичних інформаційних панелей у реальному часі [21].

2.2.2 Організація структури колекцій, приклади документів, індексація

Фізична модель зберігання даних побудована на основі чотирьох основних колекцій, структура яких зображена на рисунку 2.4, кожна з яких призначена для виконання певного класу запитів [22]:

Колекція Organizations: відбиває ієрархію власників та структуру адміністративного управління.

Колекція Facilities: містить геопросторові і технічні характеристики об’єктів нерухомості. Для оптимізації виконання картографічних запитів застосовуються 2dsphere-індекси, які дозволяють швидко знаходити об’єкти у просторі.

Колекція Devices: зберігає технічні характеристики кожного приладу обліку, включаючи налаштування протоколів обміну даними.

Колекція Measurements: служить оптимізованим сховищем часових рядів, в якому реалізована індексація за часовими відмітками для прискорення аналітичних операцій кращих.

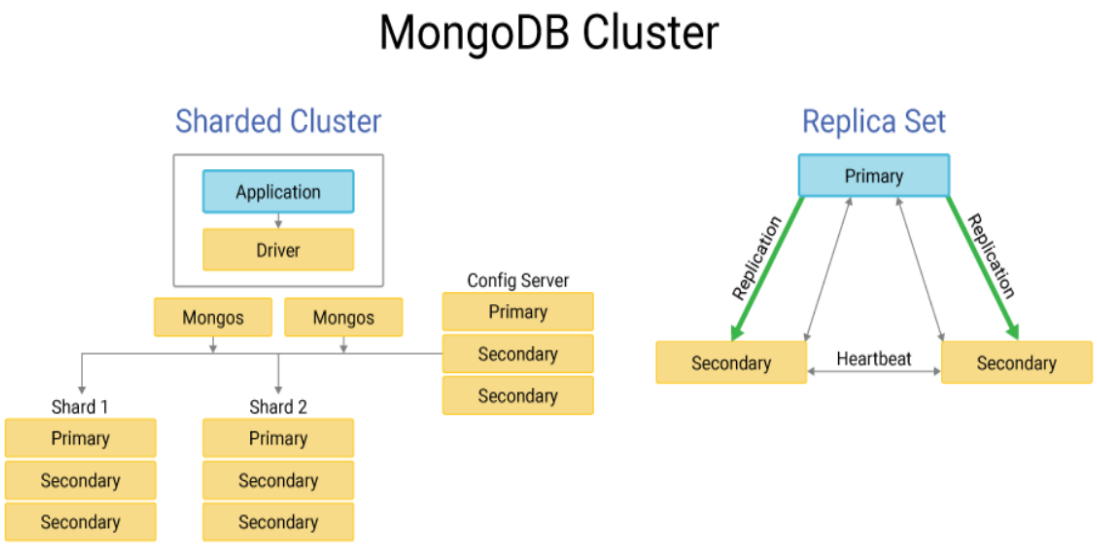

Рисунок 2.4 – Структура колекцій MongoDB [22]

Нижче у лістингу 2.1 наведено приклади документів (документ об’єкта Facilities**)**, які ілюструють архітектурні рішення щодо збереження інформації про енергоспоживання.

Лістинг 2.1 – Структура документа міського об'єкта в базі даних MongoDB

```
JSON
{
  "_id": ObjectId("65f1a2b3c4d5e6f7"),
  "name": "Муніципальна лікарня №3",
  "geo": { "type": "Point", "coordinates": [30.5234, 50.4501] },
  "metadata": { "area": 12500, "energy_class": "B" },
  "active_meters": ["ELE-9982", "WAT-4412"]
}
```

Документ вимірювань (Measurements): щоб зменшити кількість індексованих структур, використовується агрегування даних. Завдяки цьому агреговані значення за певний період залишаться в межах одного документа, що, в свою чергу, зробить його корисним при побудові систем, що мають справу з великими потоками даних, це наведено у лістингу 2.2.

Лістинг 2.2 – Структура документа показників енергоспоживання

```
JSON
{
  "meter_id": "ELE-9982",
  "period": "2026-04-24",
  "daily_stats": { "max": 45.2, "min": 10.5, "avg": 28.3 },
  "hourly_readings": [
    { "h": 0, "v": 12.1 }, { "h": 1, "v": 11.8 }, { "h": 2, "v":
10.5 }
    // ... інші 21 запис
  ]
}
```

Заключним етапом проектування стала реалізація індексації даних з метою підвищення швидкості запитів. Так, були створені комбіновані індекси за полями meter_id і timestamp, що дозволяє швидко отримувати історичні виміри, впродовж декількох мілісекунд. Крім цього, реалізовано TTL-індексацію, завдяки якій застарілі «сирі» записи автоматично видаляються, або направляються в архів, дозволяючи зберігати в активному наборі даних в оперативній пам’яті серверної частини [22].

2.3 Моделювання функціональних процесів системи

2.3.1 Сценарії використання (Use Case) та UML-діаграми системи

Функціонування системи енергомоніторингу Smart City базується на взаємодії програмного комплексу з користувачами та IoT-обладнанням, яке здійснює збір і передачу інформації. Основними учасниками системи є користувачі, адміністратори та сенсорні пристрої. На рисунку 2.5 показана діаграма, яка це ілюструє.

Користувач системи здійснює взаємодію через програмний інтерфейс і має доступ до поточних і архівних даних про споживання енергії та інших ресурсів. Окрім цього, система надає аналітичну інформацію, за допомогою якої можна оцінювати ефективність використання ресурсів і виявляти потенційні перевитрати.

Адміністративний користувач несе відповідальність за управління обліковими записами, налаштуванням параметрів платформи і моніторингом стану системи. Також адмін має доступ до інформації про роботу серверів і підключених сенсорів.

IoT-пристрої виконують роль джерел телеметрії, безперервно надаючи інформацію про використання електроенергії, води та інших ресурсів. Зібрані дані надсилаються на центральний сервер для подальшої обробки і аналізу.

Рисунок 2.5 – Use Case діаграма системи Smart City енергомоніторингу [24]

До основних сценаріїв використання системи належать автентифікація користувачів, отримання телеметричних даних у режимі реального часу, перегляд аналітичних показників, адміністрування платформи та обробка потоків даних [23].

Для опису архітектури і логіки функціонування системи використовувалося UML-моделювання, що надає наочне уявлення про взаємодію між компонентами і пргрес обробки інформації.

Діаграма варіантів яка зображена на рисунку 2.6 використання демонструє основні можливості системи і зв’язки між користувачами, програмними модулями та іншими учасниками процесу. За допомогою діаграми визначаються межі системи і типові сценарії її використання.

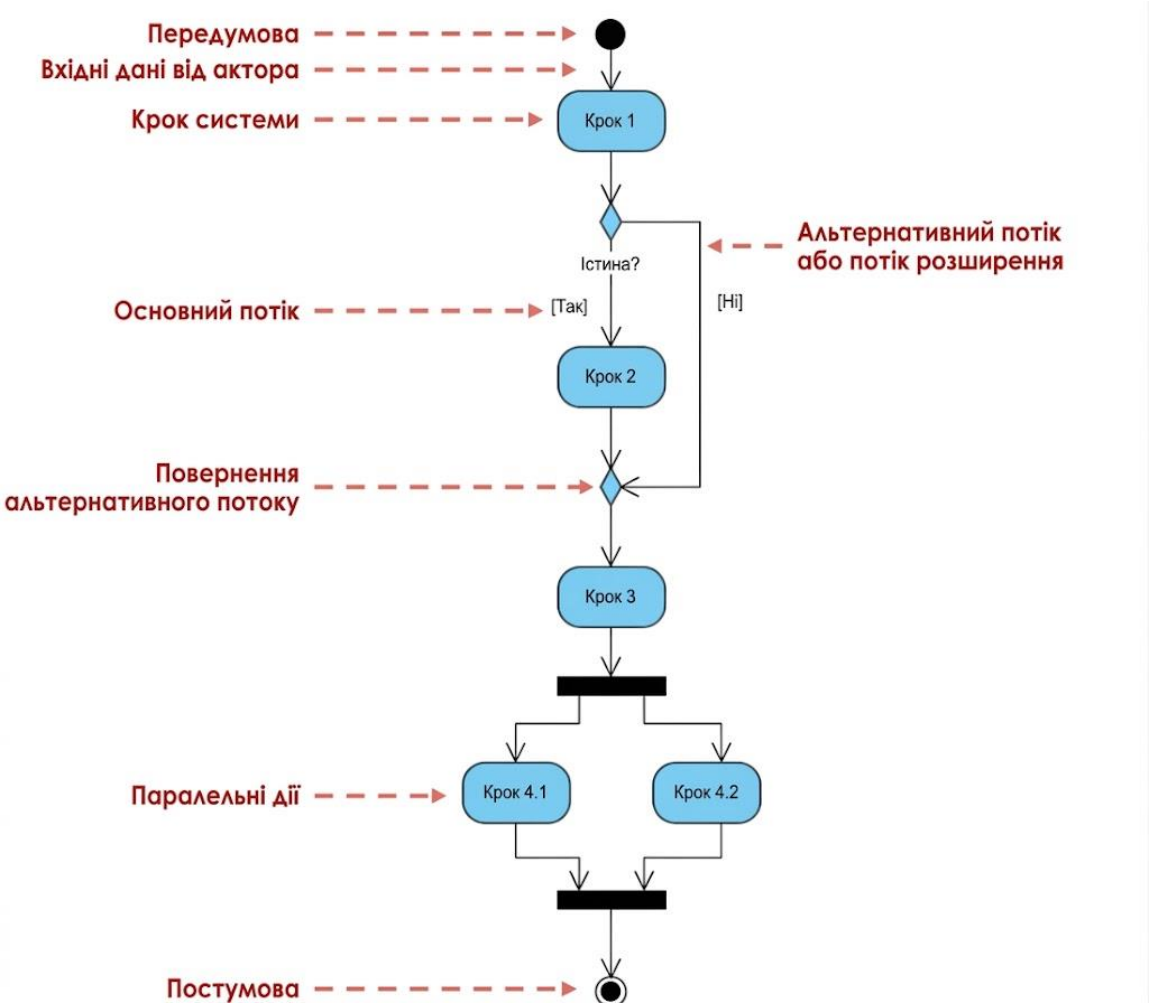

Рисунок 2.6 – UML-діаграма активностей обробки даних [26]

Діаграма діяльності використовується для показу послідовності виконання операцій всередині системи. Перша операція починається з отримання даних від сенсорних пристроїв, далі інформація надходить до серверної частини. Потім в системі відбувається безліч процедур фільтрації,

перевірки коректності і так далі. Нарешті данні записуються в базу. Заключний етап, це вебсайт, де результати показуються у вигляді графіків, панелей та звітів [25].

2.3.2 Опис бізнес-логіки системи

Бізнес-логіка системи енергомоніторингу Smart City полягає у безперервному циклі збору, обробки і аналізу інформації в реальному часі.

Спочатку надходить телеметрія від IoT-пристроїв, розташованих на об’єктах міської інфраструктури. Сенсори реєструють споживання енергетичних ресурсів і передають ці дані до центральної платформи через відповідні мережеві протоколи.

Далі відбувається попередня обробка отриманої інформації. У цьому процесі виконується очищення від шумів, перевірка достовірності показників, а також нормалізація даних для подальшого аналізу.

Після цього підготовлені дані записуються до сховища, що забезпечує можливість накопичення історії та здійснення аналітичних досліджень за різні періоди часу.

На стадії аналітики система шукає аномалії, генерує статистичні закономірності та рекомендації для ефективного використання енергетичних ресурсів.

В кінці виступає відображення для користувача у вигляді вебінтерфейсу або мобільного застосунку. Система пропонує доступ до графічних звітів, аналітичних панелей та повідомлень у зручній формі.

Отже, бізнес-логіка реалізована у вигляді повного циклу оперування інформацією, від отримання даних до їхньої аналітичної обробки і подальшого відображення для користувача [27].

2.4 Проєктування графічного інтерфейсу користувача

Створення графічного інтерфейсу користувача є одним з найважливіших етапів розробки системи Smart City, оскільки саме через інтерфейс користувач знайомиться з основними можливостями програмної платформи. Основним завданням є забезпечення зручного, інтуїтивно зрозумілого та швидкого доступу до інформації, пов’язаної із системою енергомоніторингу.

Інтерфейсна частина системи розробляється з урахуванням сучасних принципів користувацького досвіду (UX) та дизайну користувацького інтерфейсу (UI), що дозволяє полегшити взаємодію користувачів з платформою і підвищити ефективність роботи з даними [28].

У системі передбачено декілька категорій користувачів, кожна з яких має окремий рівень доступу та визначений набір функціональних можливостей, таблиці 2.2 це зображено. Такий підхід забезпечує безпеку інформації та дозволяє реалізувати чітке розмежування прав доступу.

Таблиця 2.2 – Категорії користувачів системи

Категорія користувача	**Основні функції**
Кінцевий користувач	Перегляд даних, аналітика, отримання звітів
Адміністратор	Управління системою, користувачами, налаштування
IoT-пристрої (сенсори)	Передача даних у систему

Архітектура графічного інтерфейсу користувача побудована на модульній основі і включає кілька функціонально відокремлених компонентів, приклад такої структури зображено на рисунку 2.7: вступний інформаційний екран,

аналітичний модуль, модуль оперативного спостереження та панель адміністрування.

Головна панель призначена для швидкого отримання найважливіших параметрів системи, зокрема загального стану платформи та узагальнених показників енергоспоживання. Аналітичний блок дає доступ до статистичних даних, історії та візуалізацій у вигляді графіків та діаграм. Модуль оперативного контролю використовується для перегляду актуальних показників, що надходять від сенсорних пристроїв у режимі реального часу. Адміністративна частина інтерфейсу відповідає за налаштування системних параметрів, управління користувачами та контроль доступу до функцій платформи [29].

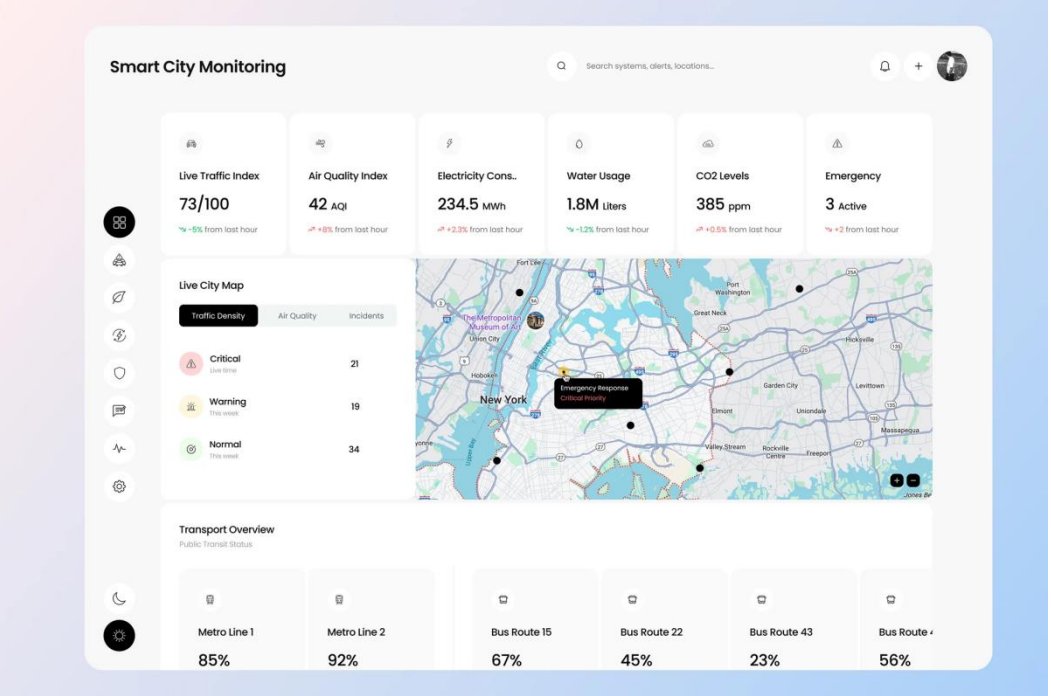

Рисунок 2.7 – Структура графічного інтерфейсу користувача системи енергомоніторингу Smart City [30]

При розробці користувацького інтерфейсу основна увага приділялася базовим принципам UX, головна ідея яких, зробити взаємодію користувача із системою максимально комфортною.

Одним з головних критеріїв стало зрозумілість інтерфейсу, завдяки якій користувач може швидко навчитися працювати з платформою без потреби в додаткових інструкціях чи навчанні. Мінімалістичний підхід до дизайну, інтерфейс тільки з необхідними елементами, без перевантаження інформацією.

Авжеж, враховані вимоги адаптивності та доступності. Інтерфейс працює коректно на різних пристроях, включаючи смартфони, планшети та десктопи. Щоб інформацію було легко сприймати, оптимізовано кольоровий контраст, розташування елементів та читабельність тексту, завдяки чому інтерфейс зручний для широкого кола користувачів [31].

Структура користувацького інтерфейсу складається з декількох окремих екранів, кожен із яких реалізує певний набір функцій платформи. До основних елементів належать стартова панель із зведеними показниками, модуль аналітичного перегляду даних та екран оперативного спостереження за поточними параметрами системи.

Стартовий екран забезпечує користувача узагальненою інформацією про стан системи та містить елементи навігації для переходу між функціональними розділами. Аналітичний модуль призначений для візуального представлення статистичних даних у вигляді графіків і діаграм, що дає змогу досліджувати зміни показників споживання ресурсів у часі. Екран оперативного контролю відображає актуальні телеметричні значення, які автоматично оновлюються безпосередньо під час роботи системи.

2.5 Висновки до другого розділу

У другому розділі дипломної роботи представлено проєкт програмної системи керування енергоспоживанням міських об’єктів розумного міста. Спроектовану систему розроблено з використанням мікросервісної архітектури, що дає змогу гарантувати масштабованість та відмовостійкість, а також дозволяє реалізувати ефективну обробку потоків великих обсягів телеметричних даних у реальному часі. Визначено архітектуру взаємодії

компонентів системи, обрано технології та засоби для реалізації програмної платформи.

Розроблено архітектуру бази даних із використанням MongoDB, побудовано модель даних, визначено структуру колекцій бази даних та методи індексування, що забезпечують високу продуктивність при зберіганні та аналізі інформації. Здійснено моделювання функціональних процесів системи з використанням Use Case та UML-діаграм, що допомогло визначити ключові сценарії використання системи та бізнес-логіку системи. Розроблено концепцію графічного інтерфейсу користувача з урахуванням сучасних трендів дизайну, зручності, доступності та адаптивності. Результати, наведені в розділі, є фундаментом для побудови практичної реалізації програмної системи та подальшого розвитку її функціоналу.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ДОСЛІДЖЕННЯ ПРОГРАМНОЇ СИСТЕМИ

3.1 Вибір програмних засобів і технологій

Для реалізації програмної системи моніторингу та аналізу енергоспоживання обрано мову програмування Python, оскільки вона забезпечує високу швидкість розробки, просту інтеграцію з базами даних та широкий набір бібліотек для аналітики і побудови графічних інтерфейсів. В якості системи керування базами даних використовується MongoDB документально-орієнтована NoSQL база даних, що дозволяє ефективно працювати з великими об’ємами даних та забезпечує функціональність структури документів.

Для реалізації графічного інтерфейсу використано PyQt5 який дозволяє створювати сучасні десктоп-застосування з інтерактивними елементами керування[32]. Для аналізу даних та побудов графіків використано бібліотеки:

- Pandas;
- NumPy;
- Matplotlib.

У лістингу 3.1 наведено приклад підключення бібліотек.

Лістинг 3.1 – Приклад підключення бібліотек

```
from pymongo import MongoClient
import pandas as pd
import matplotlib.pyplot as plt
from PyQt5.QtWidgets import *
```

Для зберігання інформації про енергоспоживання реалізовано окремий модуль роботи з MongoDB, це показано у лістингу 3.2 [33].

Лістинг 3.2 – Реалізація CRUD-операцій для роботи з MongoDB

```
from pymongo import MongoClient

/*Підключення до бази даних:*/
```

```
client = MongoClient("mongodb://localhost:27017/")
db = client["energy_db"]
collection = db["consumption"]

/*Реалізація CRUD-операцій*/
Додавання запису:
collection.insert_one({
    "object": "Building A",
    "consumption": 120,
    "date": "2026-05-01"
})

/*Отримання даних:*/
records = collection.find()

for item in records:
    print(item)

/*Оновлення запису:*/
collection.update_one(
    {"object": "Building A"},
    {"$set": {"consumption": 150}}
)

/*Видалення запису:*/
collection.delete_one(
    {"object": "Building A"}
)
```

На рисунку 3.1 зображена структура бази даних MongoDB

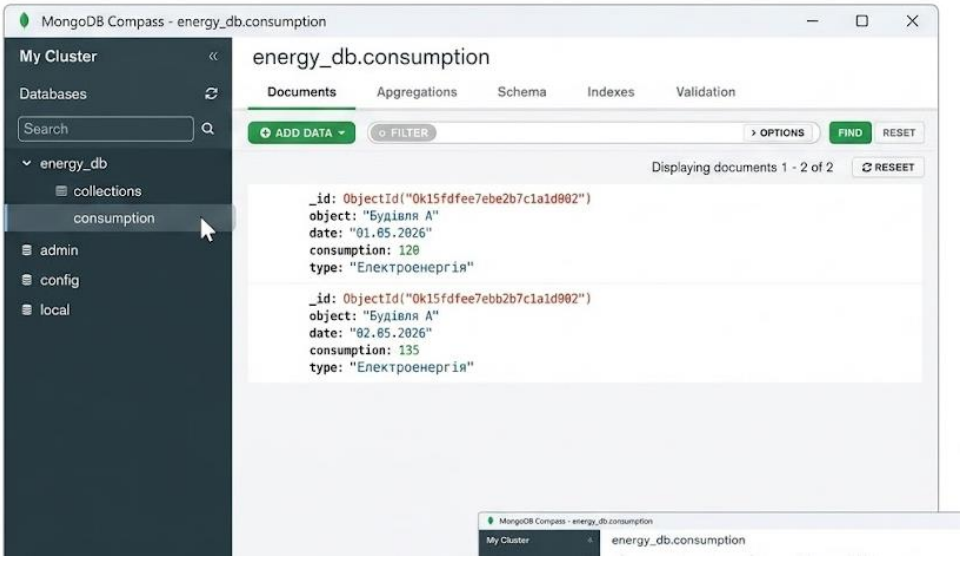

Рисунок 3.1 – Дані у базі MongoDB

3.2 Реалізація модуля аналізу енергоспоживання

Для аналізу даних реалізовано модуль статистичної обробки показників енергоспоживання. У лістингу 3.3 подано розрахунок статистичних показників.

Лістинг 3.3 – Розразунок показників

```
import pandas as pd

data = pd.DataFrame(list(collection.find()))

avg = data["consumption"].mean()
mx = data["consumption"].max()
mn = data["consumption"].min()

print(avg, mx, mn)

/*Виявлення аномальних значень*/

threshold = data["consumption"].mean() * 1.5

anomalies = data[data["consumption"] > threshold]

print(anomalies)
```

На рисунку 3.2 наведено результати аналізу даних.

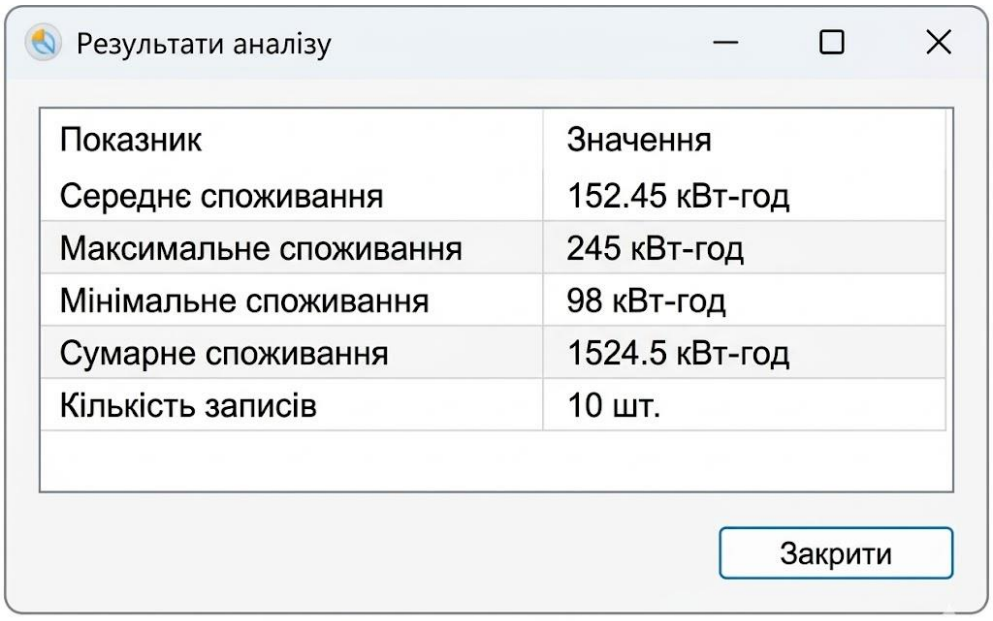

Рисунок 3.2 – Результати аналізу даних

3.3 Реалізація графічного інтерфейсу користувача

Графічний інтерфейс користувача реалізовано з використанням PyQt5. Інтерфейс містить:

1. головне вікно;
2. таблицю даних;
3. кнопки CRUD-операцій;
4. модуль аналізу;
5. графіки енергоспоживання.

Реалізація головного вікна наведена у лістингу 3.4:

Лістинг 3.4 – Реалізація головного вікна

```
import sys
from PyQt5.QtWidgets import QApplication, QMainWindow

app = QApplication(sys.argv)

window = QMainWindow()
window.setWindowTitle("Energy Monitoring System")
window.show()

sys.exit(app.exec_())
```

Інтерфейс програмної системи наведено на рисунку 3.3.

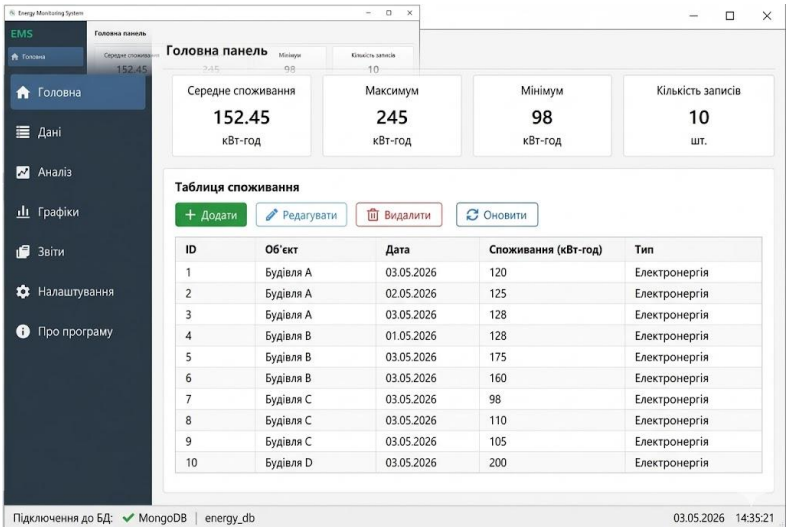

Рисунок 3.3 – Головне вікно системи

У лістингу 3.5 подано код для створення таблиці даних у системі.

Лістинг 3.5 – Код для створення таблиці даних

```
self.table.setColumnCount(2)
self.table.setHorizontalHeaderLabels([
    "Object",
    "Consumption"
])
```

Побудову графіка динаміки енергоспоживання наведено у лістингу 3.6.

Лістинг 3.6 – Код для побудови графіків

```
plt.plot(data["date"], data["consumption"])
plt.title("Energy consumption")
plt.show()
```

На рисунку 3.4 наведено графік динаміки енергоспоживання.

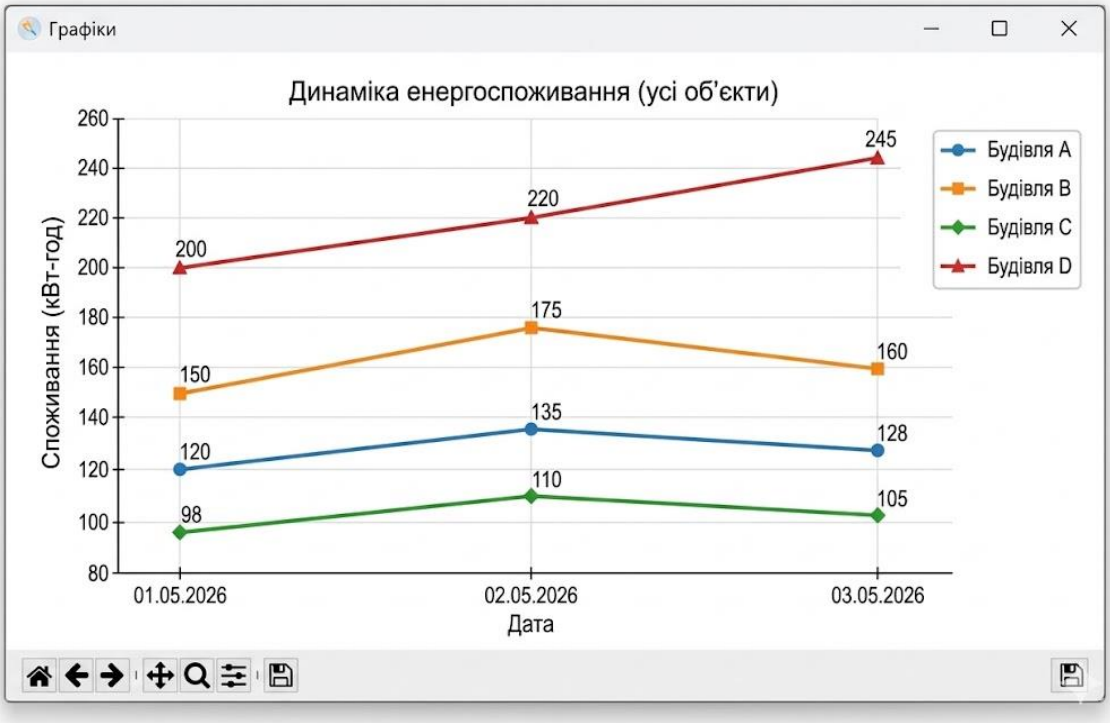

Рисунок 3.4 – Графік динаміки енергоспоживання

На рисунку 3.5 наведено порівняння середнього споживання по об'єктах.

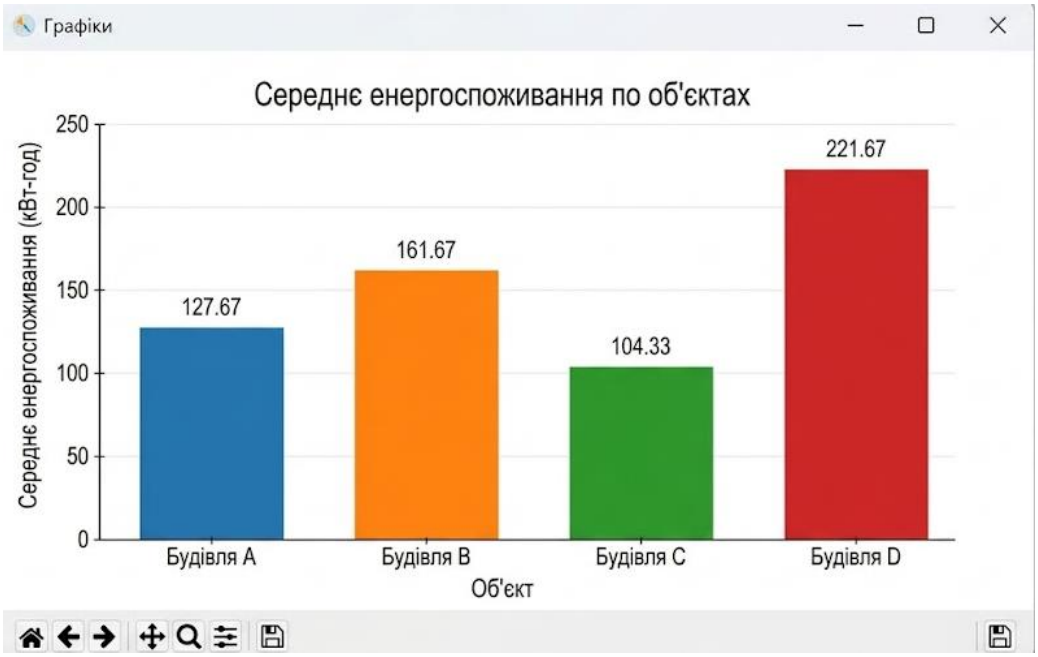

Рисунок 3.5 – Порівняння середнього споживання по об'єктах

3.4 Тестування та аналіз результатів роботи системи

Для перевірки працездатності системи проведено функціональне тестування основних модулів програмного забезпечення.

Результати тестування CRUD-операцій показано у таблиці 3.1

Таблиця 3.1 – Результати тестування CRUD-операцій

Операція	**Результат**
Create	Успішно
Read	Успішно
Update	Успішно
Delete	Успішно

Тестування продуктивності наведено у лістингу 3.7.

Лістинг 3.7 – Тестування продуктивності

```
import time

start = time.time()

for i in range(1000):
    collection.insert_one({"value": i})

print(time.time() - start)
```

Проведене тестування показало стабільну роботу системи при виконанні операцій збереження, аналізу та відображення даних. Використання MongoDB дозволило забезпечити високу швидкість обробки інформації, а застосування PyQt5 — створити зручний та сучасний інтерфейс користувача.

Отримані результати підтверджують ефективність реалізованої програмної системи для задач моніторингу та аналізу енергоспоживання.

3.5 Висновки до третього розділу

У третьому розділі реалізовано програмну частину системи моніторингу та аналізу енергоспоживання будівель, розроблено її основні архітектурні модулі та проведено експериментальне дослідження. Обґрунтовано вибір технологічного стеку, зокрема використання Python як основної мови програмування та MongoDB для ефективного зберігання й обробки великих обсягів динамічних даних. Клієнтську частину та засоби візуалізації реалізовано на базі PyQt5 із використанням бібліотеки Matplotlib.

Розроблено CRUD-модуль, що забезпечує повний цикл роботи з даними, а також аналітичний модуль на основі Pandas і NumPy для обчислення статистичних показників і виявлення аномалій енергоспоживання. Інтерфейс користувача поєднує таблиці даних, елементи керування та графічну візуалізацію динаміки споживання енергії.

Проведено функціональне та навантажувальне тестування, яке підтвердило коректність роботи CRUD-операцій, високу продуктивність

системи та її стабільність під інтенсивним навантаженням. Отримані результати засвідчили відповідність розробленого програмного забезпечення поставленим вимогам.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Ергономічні проблеми безпеки життєдіяльності

Багато проблем безпеки життєдіяльності людини розв'язуються через ергономіку, а саме через появу відповідних робочих умов, техніки, програм, які відповідають можливостям людини, знижуючи навантаження (фізичний і психоемоційний), підвищуючи продуктивність праці і знижуючи ризик помилки. Сучасні інформаційні системи, з одного боку, є, з іншого боку, створюють загрозу для таких конструкторських рішень у вигляді великих обсягів інформації, складної взаємодії з системою, і необхідності швидко приймати рішення, що може призводити до втоми, втрати концентрації і, відповідно, до помилкових рішень [37].

Особливо це видається в системах «Розумне» місто. Системи «Розумного» міста відслідковують і керують багатьма об'єктами та послугами міської інфраструктури, надаючи оператору інформацію про них у реальному часі [38]. Все це створює навантаження на оператора. В таких випадках важливо втручання інтерфейсу користувача, як дані представлені, візуалізуються, чи у вигляді графіків, діаграм, таблиць, чи є звук, повідомлення [39].

Ваша дипломна робота, пов'язана з розробкою програмного комплексу енергозбереження міських будівель у рамках "розумного міста" важлива у плані ергономіки. Система, яку ви пропонуєте, для користувача, для адміністратора або оператора, повинна бути інтуїтивною і легкою у використанні, що може бути досягнуто, зокрема, через інформаційні панелі (dashboard) для показників енергоспоживання, це дозволяє швидко розуміти стан речей, не витрачаючи багато часу на аналіз, коли всі показники в нормі, а якщо є відхилення, будуть індикатори, графіки і автоматичні повідомлення.

Отже, правильна розробка програмних систем для «Розумних» міст допомагає користувачам цих систем не тільки бути ефективними, а й безпечно

експлуатуватися в складній системі планування і пристроїв для управління/моніторингу міської інфраструктури [40].

4.2 Заходи щодо автоматизації виробничих процесів, які сприяють покращенню умов праці

Автоматизація виробничих і управлінських процесів є одним із заходів, спрямованих на підвищення безпеки життєдіяльності та покращення умов праці працівників, які працюють з інформаційними системами. Вона полягає у зниженні частки ручних операцій, передачі рутинних функцій технічним і програмним засобам, а також у підвищенні точності, швидкості та надійності виконання операцій. Це дозволяє знизити фізичне і психоемоційне навантаження на оператора і звести до мінімуму можливість людських помилок.

У сучасних реаліях особливого значення набуває впровадження автоматизованих інформаційних систем, які забезпечують збір, обробку і аналіз даних в оперативному режимі. Такі системи дають можливість оперативно виявляти відхилення в роботі обладнання, формувати попередження і підтримувати прийняття управлінських рішень без постійного втручання людини. Це істотно підвищує рівень безпеки і стабільності функціонування складних технічних систем [41].

Автоматизація лежить в основі побудови сучасних інформаційних систем управління енергетичними ресурсами міських об’єктів, оскільки вона дає можливість мінімізувати ручне втручання, підвищувати точність обробки даних і оперативність прийняття рішень в рамках концепції «Розумного» міста. Запропоноване програмне рішення передбачає автоматичний збір даних з міських енергетичних об’єктів за допомогою IoT-давачів і лічильників, їх передачу по мережі до центрального сервера, подальшу обробку в аналітичному модулі і формування рекомендацій з оптимізації енергоспоживання [42].

Окремим елементом системи є автоматизований модуль моніторингу, який у реальному часі відстежує стан енергосистеми міста, визначає перевищення норм споживання і формує повідомлення для оператора. Таким чином, зменшується потреба у постійному ручному контролі, а також підвищується швидкість реагування на можливі несправності.

Схематично роботу запропонованої системи можна подати у вигляді наступної діаграми яка зображена на рисунку 4.1 [43]:

Міські об’єкти → IoT-давачі та лічильники → шлюз збору даних → хмарний сервер → база даних → аналітичний модуль → система оптимізації енергоспоживання → інтерфейс оператора та система сповіщень.

Рисунок 4.1 – Схема автоматизації енергоменеджменту в системі розумного міста

Завдяки такій автоматизації підвищується ефективність управління міськими енергетичними ресурсами, зменшується навантаження на персонал та мінімізуються ризики помилок, що безпосередньо позитивно впливає на умови праці та рівень безпеки життєдіяльності.

4.3 Висновки до четвертого розділу

За результатами реалізованого проєкту розглянуто обґрунтовані пропозиції щодо ергономіки безпечної життєдіяльності при використанні інформаційних систем у середовищі «розумного» міста та вивчено роль автоматизації виробничих процесів у покращенні умов праці. Встановлено, що ергономічно спроєктовані програмні засоби знижують фізичне і психологічне навантаження на користувачів, підвищують зручність роботи з інформацією, а також знижують ймовірність помилок при прийнятті рішень.

Визначено, що автоматизація процесів збору, обробки та аналізу даних сприяє підвищенню ефективності роботи систем управління енергоспоживанням, знижує кількість ручних операцій і дає можливість для швидкого реагування на відхилення в роботі міської інфраструктури. Запропоновані рішення сприяють підвищенню безпеки праці операторів, поліпшенню умов їх роботи, а також забезпечують надійне функціонування програмного комплексу енергозбереження в рамках концепції «Розумного» міста.

ВИСНОВКИ

У першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр»:

- Подано загальну характеристику концепції «Розумного міста» та визначено роль енергоменеджменту в управлінні міською інфраструктурою.

- Розглянуто особливості енергоспоживання об’єктів міської інфраструктури та основні показники оцінювання їх енергоефективності.

- Висвітлено сучасні підходи до побудови систем енергомоніторингу, архітектурні рішення та технології зберігання даних.

- Проаналізовано існуючі системи енергомоніторингу, проблеми інтеграції, кібербезпеки та сформовано вимоги до розроблюваної програмної системи.

У другому розділі кваліфікаційної роботи:

- Досліджено архітектурні принципи побудови програмної системи управління енергоспоживанням міських об’єктів.

- Обґрунтовано вибір мікросервісного підходу, технологій реалізації та використання MongoDB для зберігання даних.

- Сформовано структуру бази даних, модель взаємодії компонентів системи, бізнес-логіку функціонування та концепцію графічного інтерфейсу користувача.

У третьому розділі кваліфікаційної роботи:

- Розроблено програмну систему моніторингу та аналізу енергоспоживання на основі мови програмування Python і бази даних MongoDB.

- Запропоновано модуль аналізу енергоспоживання для розрахунку статистичних показників та виявлення аномальних значень.

- Спроєктовано графічний інтерфейс користувача із засобами візуалізації даних та управління інформацією.

- Протестовано основні функціональні модулі програмного забезпечення та підтверджено коректність їх роботи.

У розділі «Безпека життєдіяльності, основи охорони праці»:

- Висвітлено ергономічні аспекти використання інформаційних систем у концепції «Розумного міста» та розглянуто вплив автоматизації процесів на покращення умов праці, зниження навантаження на персонал і підвищення рівня безпеки життєдіяльності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1.United Nations World Urbanization Prospects 2024: Technology, Energy and Sustainability in Megacities. [Електронний ресурс]. – Режим доступу: https://population.un.org/wup/ (дата звернення: 14.02.2026).

2.Simon Elias Bibri The Social Shaping of the Metaverse as a Next-Generation Smart City. [Електронний ресурс]. – Режим доступу: https://www.sciencedirect.com/ (дата звернення: 21.02.2026).

3.ДСТУ ISO 50001:2025. Системи енергетичного менеджменту. Вимоги та настанова щодо використання. [Електронний ресурс]. – Режим доступу: https://online.budstandart.com/ (дата звернення: 02.03.2026).

4.Мельник В. О. Інтеграція IoT-сенсорів у муніципальні системи контролю енергоресурсів. [Електронний ресурс]. – Режим доступу: https://ir.stu.cn.ua/ (дата звернення: 04.03.2026).

5.Anand P. A review of occupancy-based energy consumption modeling in smart buildings. [Електронний ресурс]. – Режим доступу: https://www.sciencedirect.com/ (дата звернення: 9.03.2026).

6. López-Moreno, H., Núñez-Peiró, M., Sánchez-Guevara, C., Neila, J. On the identification of Homogeneous Urban Zones for the residential buildings' energy evaluation // Building and Environment. – 2022. – Vol. 207. [Електронний ресурс]. – Режим доступу: https://doi.org/10.1016/j.buildenv.2021.108451 (дата звернення: 11.03.2026).

7.Маляренко В. А. Енергоефективність та енергоменеджмент у житлово-комунальній сфері. [Електронний ресурс]. – Режим доступу: https://eprints.kname.edu.ua/ (дата звернення: 13.03.2026).

8.Stuart, G., Ozawa-Meida, L. Supporting Decentralised Energy Management through Smart Monitoring Systems in Public Authorities // Energies. – 2020. – Vol. 13(20). [Електронний ресурс]. – Режим доступу: https://www.mdpi.com/1996-1073/13/20/5398 (дата звернення: 15.03.2026).

9.Energy Monitoring System. [Електронний ресурс]. – Режим доступу:

https://ecoenergies.co.in/energy-monitoring-system/ (дата звернення 18.03.2026)

10.Garcia M. Cybersecurity in Smart City Energy Grids. [Електронний ресурс]. – Режим доступу: https://www.sciencedirect.com/ (дата звернення: 21.03.2026).

11.Chen L. Deep Learning for Energy Forecasting in Smart Buildings. [Електронний ресурс]. – Режим доступу: https://www.sciencedirect.com/ (дата звернення: 28.03.2026).

12.Ferrer A. J. Cloud-native architectures for smart city energy platforms. [Електронний ресурс]. – Режим доступу: https://link.springer.com/ (дата звернення: 06.04.2026).

13.Brown T. NoSQL Databases in IoT Energy Platforms: Scalability and Performance. [Електронний ресурс]. – Режим доступу: https://www.oreilly.com/ (дата звернення: 12.04.2026).

14.Shafiq M. Security Challenges in IoT-based Smart City Energy Systems. [Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/ (дата звернення: 19.04.2026).

15.Benefits of Energy Management. [Електронний ресурс]. – Режим доступу: https://www.primeits.com.au/benefits-of-energy-management/

(дата звернення: 21.04.2026)

16.A Microservices-Based Solution with Hybrid Communication for Energy Management in Smart Grid Environments // Sensors. – 2026. [Електронний ресурс]. – Режим доступу: https://www.mdpi.com/1424-8220/26/5/1714 (дата звернення: 23.04.2026).

17.Scattone, F. F., Braghetto, K. R. A Microservices Architecture for Distributed Complex Event Processing in Smart Cities. [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/2008.07585 (дата звернення: 24.04.2026).

18.Newman, S. (2021). Building Microservices: Designing Fine-Grained Systems (2nd ed.). O’Reilly Media. [Електронний ресурс] – Режим доступу: https://samnewman.io/books/building_microservices/ (дата звернення: 26.04.2026).

19.Що таке мікросервісна архітектура. [Електронний ресурс]. – Режим доступу: https://blog.colobridge.net/uk/2024/01/what-is-microservices-architecture-ua/

(дата звернення 27.04.2026)

20.Chodorow K. MongoDB: The Definitive Guide. [Електронний ресурс]. – Режим доступу: https://www.mongodb.com/resources/books/ (дата звернення: 27.04.2026).

21.Алгоритми в програмуванні: що потрібно знати? [Електронний ресурс]. – Режим доступу: https://denzadnem.com.ua/blogy/korysni-porady/168016 (дата звернення: 28.04.2026).

22.Що таке MongoDB? Дізнайтеся про популярні системи управління базами даних NoSQL [Електронний ресурс]. – Режим доступу:

https://buimanhduc.com/mongodb-la-gi (дата звернення 29.04.2026).

23.Sommerville I. Software Engineering (10th Edition) [Електронний ресурс]. – Режим доступу:

https://www.pearson.com/en-us/subject-catalog/p/software-engineering/P200000003282 (дата звернення 30.04.2026)

24.UML Use Case Diagrams [Електронний ресурс]. – Режим доступу:https://www.uml-diagrams.org/use-case-diagrams.html (дата звернення: 30.04.2026)

25. UML 2.5 Specification (OMG) [Електронний ресурс]. – Режим доступу:https://www.omg.org/spec/UML/2.5/ (дата звернення: 30.04.2026)

26. What is Activity Diagram? [Електронний ресурс]. – Режим доступу: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/ (дата звернення 01.05.2026)

27.IBM Cloud Education – IoT (Internet of Things) [Електронний ресурс]. – Режим доступу: https://www.ibm.com/topics/internet-of-things (дата звернення: 01.05.2026)

28.Nielsen J. 10 Usability Heuristics for User Interface Design. [Електронний ресурс]. – Режим доступу: https://www.nngroup.com/articles/ten-usability-heuristics/ (дата звернення: 01.05.2026).

29.Google. Material Design 3 Guidelines. [Електронний ресурс]. – Режим доступу: https://m3.material.io/ (дата звернення: 03.05.2026).

30. Smart City Monitoring Dashboard UI. [Електронний ресурс]. – Режим доступу: https://elements.envato.com/smart-city-monitoring-dashboard-ui-2DGS2YJ (дата звернення: 03.05.2026).

31.ISO 9241-210:2019 Ergonomics of human-system interaction — Human-centred design for interactive systems. [Електронний ресурс]. – Режим доступу: https://www.iso.org/standard/77520.html (дата звернення: 03.05.2026).

32.Python Software Foundation. Python Documentation. – Режим доступу: Python Documentation (дата звернення: 05.05.2026).

33.MongoDB Inc. MongoDB Documentation. – Режим доступу: https://www.mongodb.com/docs/ (дата звернення: 08.05.2026).

34.The Pandas Development Team. Pandas Documentation. – Режим доступу: https://pandas.pydata.org/docs/ (дата звернення: 05.05.2026). .

35.Riverbank Computing Limited. PyQt5 Documentation. – Режим доступу: https://www.riverbankcomputing.com/static/Docs/PyQt5/ (дата звернення: 08.05.2026).

36.Matplotlib Development Team. Matplotlib Documentation. – Режим доступу: https://matplotlib.org/stable/index.html (дата звернення: 05.05.2026).

37.Методичні вказівки для написання розділу «Безпека життєдіяльності, основи охорони праці» в кваліфікаційних роботах здобувачів освітнього рівня ,,бакалавр” / Укладачі: Гурик О.Я., Окіпний І.Б. – Тернопіль: ТНТУ імені Івана Пулюя, 2021. – 20 с. – Режим доступу: https://dl.tntu.edu.ua/content.php?cid=392952 (дата звернення: 16.05.2026).

38.Palka O., Dmytrotsa L., Kozbur H., Nebesnyi R. Smart people: the role of big data analytics in digital transformation. – Proceedings of the BAITmp 2025: The 2nd International Workshop on Bioinformatics and Applied Information

Technologies for medical purpose (Ben Guerir, Morocco, November 12-13, 2025). CEUR Workshop Proceedings (CEURWS.org). 2025. Vol. 4159, pp. 163-174. – Режим доступу: https://ceur-ws.org/Vol-4159/paper14.pdf (дата звернення: 11.06.2026).

39.Ystgaard, K. F., Atzori, L., Palma, D. та ін. Review of the theory, principles, and design requirements of human-centric Internet of Things (IoT) // Journal of Ambient Intelligence and Humanized Computing. – 2023. – Vol. 14. [Електронний ресурс]. – Режим доступу: https://doi.org/10.1007/s12652-023-04539-3 (дата звернення: 17.05.2026).

40.Liu, Z., Sang, G. Research on IoT Design Strategies Based on HCD in Smart City Development // Proceedings of the MSIEID Conference. – 2024. [Електронний ресурс]. – Режим доступу: https://doi.org/10.4108/eai.8-12-2023.2344472 (дата звернення: 19.05.2026).

41.Bibri, S. E., Alahi, A., Sharifi, A. та ін. Environmentally sustainable smart cities and their converging AI, IoT, and big data technologies and solutions // Energy Informatics. – 2023. – Vol. 6. [Електронний ресурс]. – Режим доступу: https://doi.org/10.1186/s42162-023-00259-2 (дата звернення: 21.05.2026).

42.Палка О. Аналіз інтегрованої архітектури розумного міста з блокчейном та IoT. – Науковий вісник НЛТУ України. 2023. №33(6). С. 94-99. – Режим доступу: https://doi.org/10.36930/40330612 ; https://nv.nltu.edu.ua/index.php/journal/article/view/2554 (дата звернення: 11.06.2026).

43.Tekinerdogan, B., Köksal, Ö., Çelik, T. System architecture design of IoT-based smart cities // Applied Sciences. – 2023. – Vol. 13(7). [Електронний ресурс]. – Режим доступу: https://doi.org/10.3390/app13074173 (дата звернення: 24.05.2026).

ДОДАТКИ

Додаток А

Вихідний код програмної системи управління енергоспоживанням міських об'єктів «Розумного» міста

```
import sys
import random
from PyQt5.QtWidgets import (
    QApplication, QMainWindow, QWidget, QVBoxLayout,
    QHBoxLayout, QPushButton, QTableWidget, QTableWidgetItem,
    QLabel, QLineEdit, QMessageBox
)

from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg
as FigureCanvas
from matplotlib.figure import Figure


# ---------------------- GRAPH ----------------------
class MplCanvas(FigureCanvas):
    def __init__(self, parent=None):
        fig = Figure()
        self.ax = fig.add_subplot(111)
        super().__init__(fig)


# ---------------------- MAIN APP ----------------------
class EnergyApp(QMainWindow):
    def __init__(self):
        super().__init__()

        self.setWindowTitle("Energy Monitoring System")
        self.setGeometry(200, 200, 1000, 600)

        self.data = []  # тут імітація БД

        self.init_ui()

    def init_ui(self):
        main = QWidget()
        self.setCentralWidget(main)

        layout = QHBoxLayout()

        # ---------------- LEFT PANEL ----------------
        left_layout = QVBoxLayout()

        self.object_input = QLineEdit()
        self.object_input.setPlaceholderText("Object name")
```

```
        self.value_input = QLineEdit()
        self.value_input.setPlaceholderText("Consumption (kWh)")

        add_btn = QPushButton("Add record")
        delete_btn = QPushButton("Delete selected")
        analyze_btn = QPushButton("Analyze")

        add_btn.clicked.connect(self.add_record)
        delete_btn.clicked.connect(self.delete_record)
        analyze_btn.clicked.connect(self.analyze)

        self.info_label = QLabel("Statistics: -")

        left_layout.addWidget(self.object_input)
        left_layout.addWidget(self.value_input)
        left_layout.addWidget(add_btn)
        left_layout.addWidget(delete_btn)
        left_layout.addWidget(analyze_btn)
        left_layout.addWidget(self.info_label)

        # ---------------- TABLE ----------------
        self.table = QTableWidget()
        self.table.setColumnCount(2)
        self.table.setHorizontalHeaderLabels(["Object",
"Consumption"])

        left_layout.addWidget(self.table)

        # ---------------- GRAPH ----------------
        self.canvas = MplCanvas(self)
        left_layout.addWidget(self.canvas)

        layout.addLayout(left_layout)
        main.setLayout(layout)

        self.load_demo_data()

    # ---------------------- DATA ----------------------
    def load_demo_data(self):
        demo = [
            ("Building A", 120),
            ("Building B", 150),
            ("Building C", 90),
            ("Building D", 200),
        ]

        for obj, val in demo:
            self.data.append((obj, val))

        self.refresh_table()
        self.update_graph()
```

```
def add_record(self):
    obj = self.object_input.text()
    try:
        val = int(self.value_input.text())
    except:
        QMessageBox.warning(self, "Error", "Invalid number")
        return

    self.data.append((obj, val))
    self.refresh_table()
    self.update_graph()

def delete_record(self):
    row = self.table.currentRow()
    if row >= 0:
        self.data.pop(row)
        self.refresh_table()
        self.update_graph()

# ---------------------- TABLE ----------------------
def refresh_table(self):
    self.table.setRowCount(len(self.data))

    for i, (obj, val) in enumerate(self.data):
        self.table.setItem(i, 0, QTableWidgetItem(obj))
        self.table.setItem(i, 1, QTableWidgetItem(str(val)))

# ---------------------- ANALYSIS ----------------------
def analyze(self):
    if not self.data:
        return

    values = [v for _, v in self.data]

    avg = sum(values) / len(values)
    mx = max(values)
    mn = min(values)

    self.info_label.setText(
        f"AVG: {avg:.2f} | MAX: {mx} | MIN: {mn}"
    )

# ---------------------- GRAPH ----------------------
def update_graph(self):
    self.canvas.ax.clear()

    labels = [x[0] for x in self.data]
    values = [x[1] for x in self.data]

    self.canvas.ax.bar(labels, values)
    self.canvas.ax.set_title("Energy consumption")
    self.canvas.ax.set_ylabel("kWh")
```

```
        self.canvas.draw()


# ---------------------- RUN ----------------------
app = QApplication(sys.argv)
window = EnergyApp()
window.show()
sys.exit(app.exec_())
```