

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 – Комп’ютерні науки,
освітньо-наукова програма «Управління проектами»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

«Дослідження процесів управління проектом розробки веб-сервісу для
моніторингу відключень світла в Україні»

Студентки 2-го курсу групи УП-21

Науковий керівник:

Юлії МАВРОДІ

Тетяна ЛАТИШЕВА

к.т.н., доцент кафедри технологій управління

Попередній захист:

(Висновок: “До захисту в Екзаменаційній комісії”)

Завідувач кафедри технологій

управління

Віктор МОРОЗОВ

Київ – 2026

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій

Кафедра технологій управління

Освітній рівень Магістр

Спеціальність 122 Комп'ютерні науки

Освітньо-наукова програма Управління проєктами

ЗАТВЕРДЖУЮ

Завідувач кафедри

професор Віктор МОРОЗОВ

_____ 2026 року
“__” _____

З А В Д А Н Н Я
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Студентка: Юлія МАВРОДІ

Група: УП-21

- 1. Тема кваліфікаційної роботи:** «Дослідження процесів управління проєктом розробки веб-сервісу для моніторингу відключень світла в Україні». Затверджена протоколом кафедри ТУ від 05 грудня 2025 року № 4.
- 2. Строк подання студентом готової роботи** – “19” травня 2026 року.
- 3. Цільова установка та вихідні дані до роботи:** дослідження різних методів та інструментів для управління проєктом розробки інформаційної системи, їх використання у плануванні проєкту, ресурсів, бюджету та управлінні ризиками.
- 4. Зміст роботи:** дослідження конкурентного середовища, обґрунтування доцільності та життєздатності проєкту, загальний опис проєкту, аналіз зацікавлених сторін проєкту, проведення SWOT-аналізу, вибір методології

управління проектом, формування беклогу продукту проекту, організаційна структура проекту, планування робіт проекту у вигляді спринтів, управління ресурсами та бюджетування проекту, управління ризиками проекту. Розробка інформаційної структури проекту, створення концептуальної моделі бази даних, проектування інтерфейсу користувача.

5. Перелік графічного матеріалу: актуальність, концептуальна та математична модель інформаційної системи, процес функціонування системи, User Story проекту, беклог продукту проекту, WBS структура проекту, планування спринтів та діаграма Ганта, ризики проекту, архітектура інформаційної системи, інформація про базу даних, сторінки сайту.

6. Календарний план виконання роботи

№ з/п	Назва частин роботи	План виконання роботи
1	Вивчення літературних джерел з предмету дослідження	16.03.2026
2	Збір і вивчення матеріалів досліджуваного підприємства	20.03.2026
3	Складання плану кваліфікаційної роботи	23.03.2026
4	Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін	25.03.2026
5	Підготовка розділу 1	30.03.2026
6	Підготовка розділу 2	06.04.2026
7	Підготовка розділу 3	15.04.2026
8	Підготовка розділу 4	22.04.2026
9	Оформлення кваліфікаційної роботи	27.04.2026
10	Передача кваліфікаційної роботи науковому керівникові	29.04.2026

11	Передача кваліфікаційної роботи рецензенту для рецензування	04.05.2026
12	Захист кваліфікаційної роботи	25.05.2026

Дата видачі завдання «08» грудня 2025 р.

Керівник роботи к.т.н., доцент Тетяна ЛАТИШЕВА.

(посада, ім'я, прізвище)

(підпис)

Завдання прийняла до виконання студентка групи УП-21

Юлія МАВРОДІ.

(ім'я, прізвище)

(підпис)

АНОТАЦІЯ

кваліфікаційної роботи магістра на тему

«Дослідження процесів управління проектом розробки веб-сервісу для моніторингу відключень світла в Україні»

Студентка: Мавроді Юлія Андріївна

Науковий керівник: Латишева Тетяна Володимирівна

Рік захисту - 2026

Метою кваліфікаційної роботи є дослідження, аналіз та оптимізація процесів управління проектом розробки інтегрованого веб-сервісу для моніторингу відключень електроенергії в Україні, що дозволить підвищити ефективність інформування населення в умовах енергетичної кризи, забезпечити візуалізацію даних у реальному часі та надати інструменти для адаптації до інфраструктурних обмежень.

Цілями проекту є розробка та впровадження інтерактивної вебплатформи, яка забезпечує агрегацію даних про стан енергопостачання з різних джерел, відображення пунктів незламності на геоінформаційній карті та надання персоналізованих сповіщень користувачам на основі верифікованих crowdsourcing-даних.

Наукова новизна дослідження полягає у розробці адаптивної концептуальної моделі управління ІТ-проектом у сфері критичного інформування, яка враховує динамічну зміну зовнішніх факторів та інтегрує механізми перевірки достовірності даних від користувачів. Удосконалено підхід до формування комплексних систем моніторингу, що поєднують офіційні графіки обмежень із фактичним станом об'єктів інфраструктури, що дозволяє мінімізувати інформаційну невизначеність.

Практична цінність отриманих результатів полягає:

1) Створенні єдиного інформаційного середовища для оперативного моніторингу енергопостачання, що базується на принципах прозорості та доступності даних.

2) Забезпеченні громадян ефективним інструментом пошуку найближчих точок доступу до ресурсів (тепла, зв'язку, електрики) через інтерактивну мапу пунктів незламності.

3) Впровадженні автоматизованої системи оповіщення, яка дозволяє користувачам завчасно планувати діяльність відповідно до змін у стані мережі.

Кваліфікаційна робота складається з анотації, вступу, основної частини, яка включає чотири розділи, висновків, переліку використаних інформаційних джерел та додатків.

У першому розділі проведено комплексне дослідження предметної області та аналіз існуючих рішень для моніторингу енергопостачання. Виконано PEST- та SWOT-аналіз проєкту, ідентифіковано ключові зацікавлені сторони, визначено задачі дослідження та сформовано паспорт проєкту. Також побудовано дерево причин та наслідків виникнення проблем у сфері інформування населення.

У другому розділі побудовано концептуальну модель системи та здійснено математичну постановку задачі дослідження. Розроблено алгоритми критеріалізації параметрів ресурсів, що дозволяють здійснювати інтелектуальне ранжування об'єктів допомоги та верифікацію користувацьких звітів через розрахунок індексу довіри.

В третьому розділі обґрунтовано вибір адаптивної технології управління та розроблено детальне інформаційне забезпечення менеджменту, включаючи ієрархічну структуру робіт, беклог продукту та календарне планування засобами Microsoft Project. Проведено комплексну ідентифікацію проєктних ризиків та сформовано карту реагування на загрози, що гарантує стійкість процесу розробки та дотримання термінів реалізації системи.

В четвертому розділі наведено опис алгоритмів роботи з картографічними сервісами, представлено концептуальне та даталогічне моделювання бази даних системи та реалізацію програмних компонентів платформи, включаючи інтерфейси користувача та модулі сповіщень.

Робота містить 112 сторінок без додатків, 32 рисунки та 40 таблиць. Додатки складають 17 сторінок.

Ключові слова: *управління IT-проєктами, моніторинг енергопостачання, веб-сервіс, інтерактивна карта, пункти незламності, верифікація даних, crowdsourcing, управління ризиками, діаграма Ганта.*

ЗМІСТ

ЗМІСТ	8
ВСТУП.....	10
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЄКТУ	13
1.1 Дослідження існуючих ІТ в предметній галузі	13
1.2 Формулювання проблемної області	17
1.3 Проведення аналізу літературних та інформаційних джерел щодо можливостей вирішення виявлених проблем	18
1.4 Формулювання наукової новизни та інноваційності проєкту.....	19
1.5 Ідентифікація зацікавлених сторін проєкту	20
1.6 PEST- та SWOT-аналіз проєкту.....	22
1.7 Побудова дерева причин та наслідків.....	24
1.8 Визначення цілей проєкту та побудова дерева цілей.....	25
1.9 Постановка задачі дослідження. Формулювання технічного завдання на розробку у вигляді паспорту проєкту	27
РОЗДІЛ 2. МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	30
2.1 Розробка концептуальної моделі інформаційної системи.....	30
2.1.1 Формалізація математичних об'єктів системи	35
2.1.2 Ваги параметрів для оцінки Пунктів незламності.....	36
2.1.3 Верифікація краудсорсингових звітів (Data Trust).....	37
2.2 Постановка задачі в математичному вигляді відповідно до концептуальної моделі	38
2.2.1 Визначення змінних та множин системи	38
2.2.2 Математичний опис функцій системи.....	39
2.2.3 Модель інтегральної оцінки об'єктів та звітів	40
2.2.4 Конвертація завантаженості у бали	41
2.2.5 Алгоритм оцінки достовірності даних.....	42
2.3 Застосування розроблених моделей в межах проєкту розробки веб-сервісу	43
2.3.1 Формування запиту та параметризація фільтрів	44
2.3.2 Ідентифікація об'єктів та інтелектуальне ранжування	44
2.3.3 Оновлення стану та зворотний зв'язок.....	45

РОЗДІЛ 3. ВИБІР ТЕХНОЛОГІЇ УПРАВЛІННЯ ПРОЄКТОМ ТА РОЗРОБКА ЙОГО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ	46
3.1 Вибір технології управління проєктом	46
3.2 Визначення функціональних та нефункціональних вимог до продукту проєкту	48
3.3 Формування беклогу продукту проєкту	51
3.4 Організаційна структура управління проєктом	54
3.5 Розробка WBS проєкту	55
3.6 Планування ресурсного забезпечення проєкту та бюджету.....	59
3.7 Календарне планування та візуалізація графіка реалізації проєкту засобами Microsoft Project	65
3.8 Ідентифікація проєктних ризиків та розробка карти управління ними.....	73
РОЗДІЛ 4. РОЗРОБКА ОСНОВНИХ СКЛАДОВИХ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ	80
4.1 Опис алгоритмів роботи системи моніторингу відключень електроенергії	80
4.1.1 Алгоритм завантаження та обробки даних від енергопостачальників	80
4.1.2 Алгоритм обробки користувацьких звітів через краудсорсинг	81
4.1.3 Алгоритм генерації та розсилки персоналізованих сповіщень.....	82
4.1.4 Алгоритм візуалізації даних на інтерактивній карті	83
4.2 Концептуальне та даталогічне моделювання бази даних продукту проєкту	83
4.3 Розробка back-end частини веб-сервісу	91
4.4 Розробка front-end частини веб-сервісу	98
4.4.1 Архітектура та програмна реалізація клієнтської частини.....	99
4.4.2 Візуалізація макетів та опис функціональних сторінок веб-сервісу ..	101
ВИСНОВКИ.....	105
ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ	107
ДОДАТОК А	113
ДОДАТОК Б	114
ДОДАТОК В	117
ДОДАТОК Г	120
ДОДАТОК Д	125

ВСТУП

Сучасний стан енергетичної системи України характеризується високим рівнем нестабільності, що зумовлено масштабними руйнуваннями критичної інфраструктури внаслідок воєнних дій. У таких умовах доступ до актуальної та достовірної інформації про стан електропостачання перетворюється з елемента побутового комфорту на базову потребу життєзабезпечення. Відсутність світла системно впливає на функціонування медичних установ, освітніх закладів, промислових підприємств та об'єктів соціальної сфери. Проте наявні інформаційні ресурси, такі як офіційні ресурси постачальників, канали у соціальних мережах чи регіональні медіа, часто надають суперечливі або застарілі відомості. Це створює значні труднощі у плануванні діяльності громадян та бізнесу, що підкреслює необхідність створення єдиного технологічного рішення для оперативного моніторингу ситуації в реальному часі.

Актуальність дослідження зумовлена гострою потребою у розробці та впровадженні інтегрованого веб-сервісу, який забезпечує агрегацію розрізнених даних про відключення енергії, їх візуалізацію на геоінформаційній платформі та надання персоналізованих сповіщень. На відміну від існуючих інструментів, запропонована система базується на принципах поєднання офіційних графіків із механізмами зворотного зв'язку від користувачів, що дозволяє відстежувати фактичний стан мережі. Особливе значення має інтеграція даних про пункти незламності, що забезпечує громадян інформацією про доступні альтернативні джерела живлення, зв'язок та тепло. Такий комплексний підхід дозволяє мінімізувати інформаційний вакуум та надати суспільству дієвий інструмент для адаптації до тривалих енергетичних обмежень.

Метою роботи є теоретичне обґрунтування, дослідження процесів управління та практична розробка концепції веб-сервісу для моніторингу відключень електроенергії, що поєднує функції інтерактивної мапи, аналітичного модуля та системи багаторівневих сповіщень.

Для досягнення поставленої мети було визначено та вирішено низку завдань:

- 1) Провести детальний аналіз існуючих методів збору та обробки інформації про стан енергомереж та виявити ключові недоліки поточних систем інформування.
- 2) Розробити концептуальну модель сервісу, яка враховує вимоги різних категорій користувачів та специфіку роботи в кризових умовах.
- 3) У межах проєктного управління сформувати ієрархічну структуру робіт WBS та визначити часові межі реалізації етапів за допомогою діаграми Ганта.
- 4) Обґрунтувати архітектурні рішення та обрати технологічний стек, що забезпечує стійкість системи до високих навантажень.
- 5) Реалізувати програмний модуль інтерактивної карти та налаштувати механізми автоматичного розсилання персоналізованих сповіщень.
- 6) Розробити математичні алгоритми верифікації даних, отриманих від користувачів, через механізми краудсорсингу та розрахувати індекси довіри до інформації.
- 7) Здійснити планування ресурсів та бюджету проєкту, виконати аналіз проєктних ризиків.

Об'єктом дослідження є процеси управління проєктом розробки веб-сервісу для моніторингу відключень електроенергії в Україні в умовах енергетичної кризи.

Предметом дослідження є конкретні аспекти та характеристики цих процесів управління проєктом, такі як методи оцінки впливів оточення ІТ-проєкту, розробка концептуальної моделі інформаційної системи, модель бази даних, розробка математичних моделей збору та верифікації даних, реалізація програмного забезпечення, ресурсне та календарне планування проєкту, а також інші аспекти, пов'язані з управлінням та реалізацією проєкту.

Наукова новизна отриманих результатів полягає у розробці комплексного підходу до моніторингу енергопостачання, який вперше поєднує офіційні дані енергокомпаній із верифікованими даними від споживачів у межах єдиного картографічного інтерфейсу. Удосконалено методи оцінки стану пунктів незламності через впровадження системи оцінювання та обліку їх реальної завантаженості. Дістали подальшого розвитку моделі управління ризиками ІТ-проектів у сфері цивільного захисту, що дозволяє підвищити точність інформування населення під час аварійних та стабілізаційних обмежень. Запропоновано новий механізм фільтрації та модерації користувацького контенту, що запобігає поширенню недостовірної інформації в системі.

Практичне значення роботи полягає у створенні готового до експлуатації прототипу веб-сервісу, який суттєво спрощує доступ до критично важливої інформації та сприяє підвищенню рівня обізнаності населення. Результати дослідження, зокрема розроблені плани управління та архітектурні рішення, можуть бути використані профільними відомствами для покращення комунікації зі споживачами. Створена система має високий потенціал масштабування та може бути адаптована для моніторингу інших видів комунальних послуг або інфраструктурних об'єктів у кризових ситуаціях. Окремі модулі аналітики та збору даних можуть бути інтегровані в існуючі державні інформаційні сервіси для посилення стійкості енергетичного фронту країни.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЄКТУ

1.1 Дослідження існуючих ІТ в предметній галузі

Дослідження існуючих ІТ-рішень у предметній галузі моніторингу енергопостачання та відключень електроенергії дозволяє виявити наявні підходи та технологічні рішення, а також визначити напрямки для інновацій. На сьогоднішній день в Україні існують різні системи інформування про відключення, кожна з яких має свої переваги та обмеження.

Найбільш поширеним інструментом є офіційні веб-сайти та мобільні додатки енергопостачальних компаній, таких як ДТЕК (Рис. 1.1), а також міських організацій, наприклад, «Київ Цифровий» від КМДА (Рис. 1.2), які надають інформацію про планові та аварійні відключення на території обслуговування. Однак ці рішення мають обмежену функціональність та не забезпечують інтерактивної візуалізації відключень на карті, персоналізованих сповіщень та інформації про пункти незламності [2].



Рис. 1.1. Графік відключень електроенергії на веб-сайті компанії ДТЕК

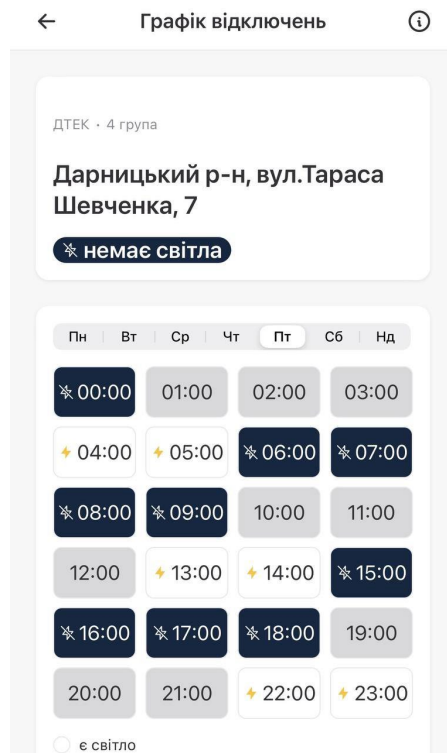


Рис. 1.2. Інформація про графіки відключень у мобільному додатку «Київ Цифровий»

Іншим підходом є використання месенджерів та чат-ботів, які надсилають сповіщення про відключення електроенергії на основі підписки користувачів. Такі рішення забезпечують оперативність інформування, але не надають візуального представлення даних та мають обмежені можливості для персоналізації. Крім того, інформація в таких каналах часто подається у несистематизованому вигляді, що ускладнює її сприйняття та аналіз.

Останнім часом з'явилися спеціалізовані веб-сервіси, такі як Svitlo.live (Рис. 1.3) та ЛУН Місто (Рис. 1.4), які надають інформацію про відключення електроенергії через інтерактивні карти. Svitlo.live пропонує карту актуальних графіків відключень електроенергії по всій території України, дозволяючи користувачам оперативно перевіряти інформацію про відключення в їх регіоні [73]. ЛУН Місто, в свою чергу, пропонує карту закладів з автономним живленням, таких як кафе, магазини та аптеки, які продовжують працювати під час відключень світла. Однак ці платформи мають обмежені можливості інтеграції з іншими системами та не охоплюють повний спектр потреб

користувачів, зокрема інформацію про пункти незламності та інші альтернативні джерела електроенергії.

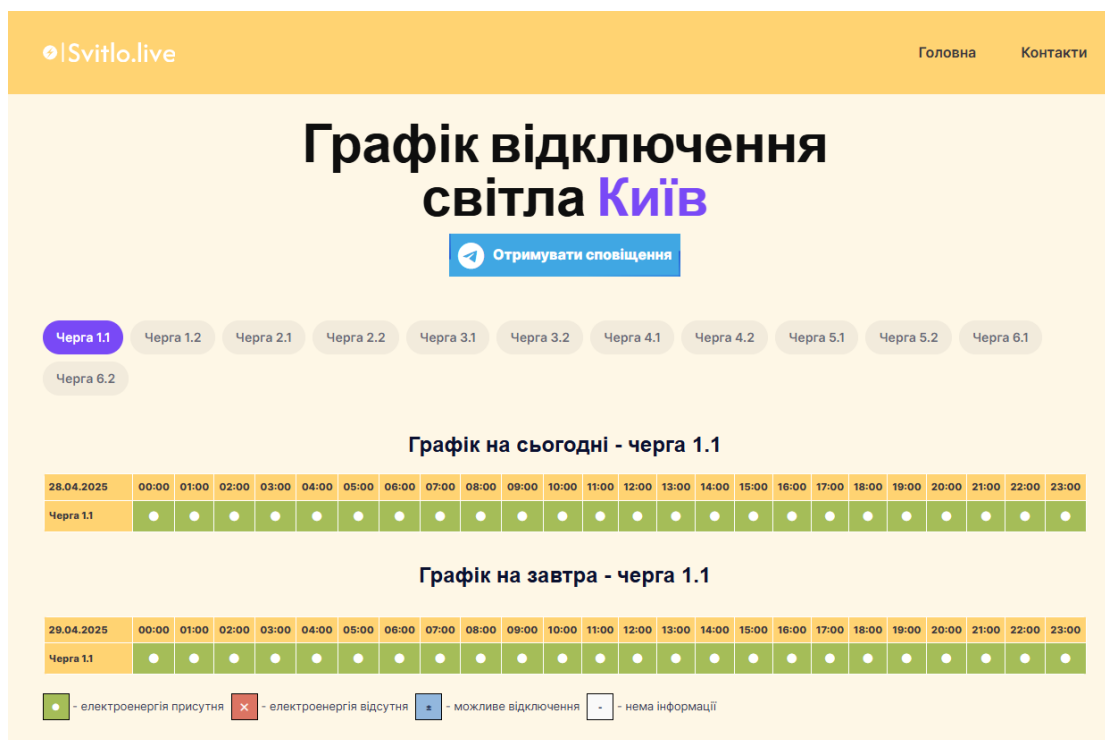


Рис. 1.3. Інтерфейс веб-сервісу Svitlo.live з актуальними графіками відключення електроенергії

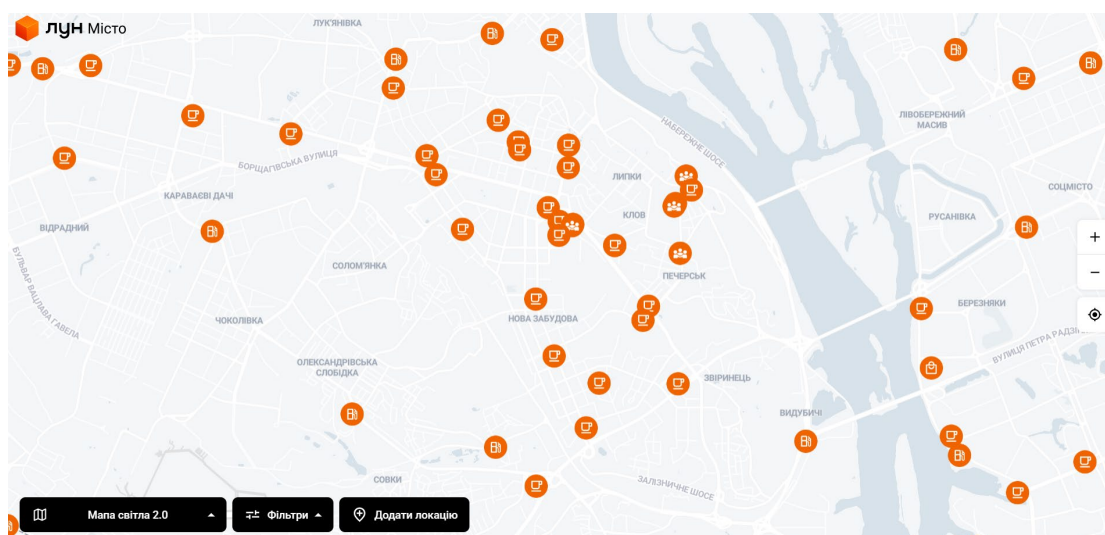


Рис. 1.4. Карта закладів з автономним живленням у сервісі ЛУН Місто

Цікавим є досвід зарубіжних країн, де функціонують платформи моніторингу електропостачання, такі як PowerOutage.us у США (Рис. 1.5) та Elenia Outage Map у Фінляндії. Ці системи забезпечують візуалізацію відключень електроенергії в реальному часі, надають статистичну інформацію та персоналізовані сповіщення для користувачів. Однак їх пряме застосування в українських реаліях ускладнюється специфікою енергетичної інфраструктури та особливостями організації енергопостачання в умовах воєнного стану.

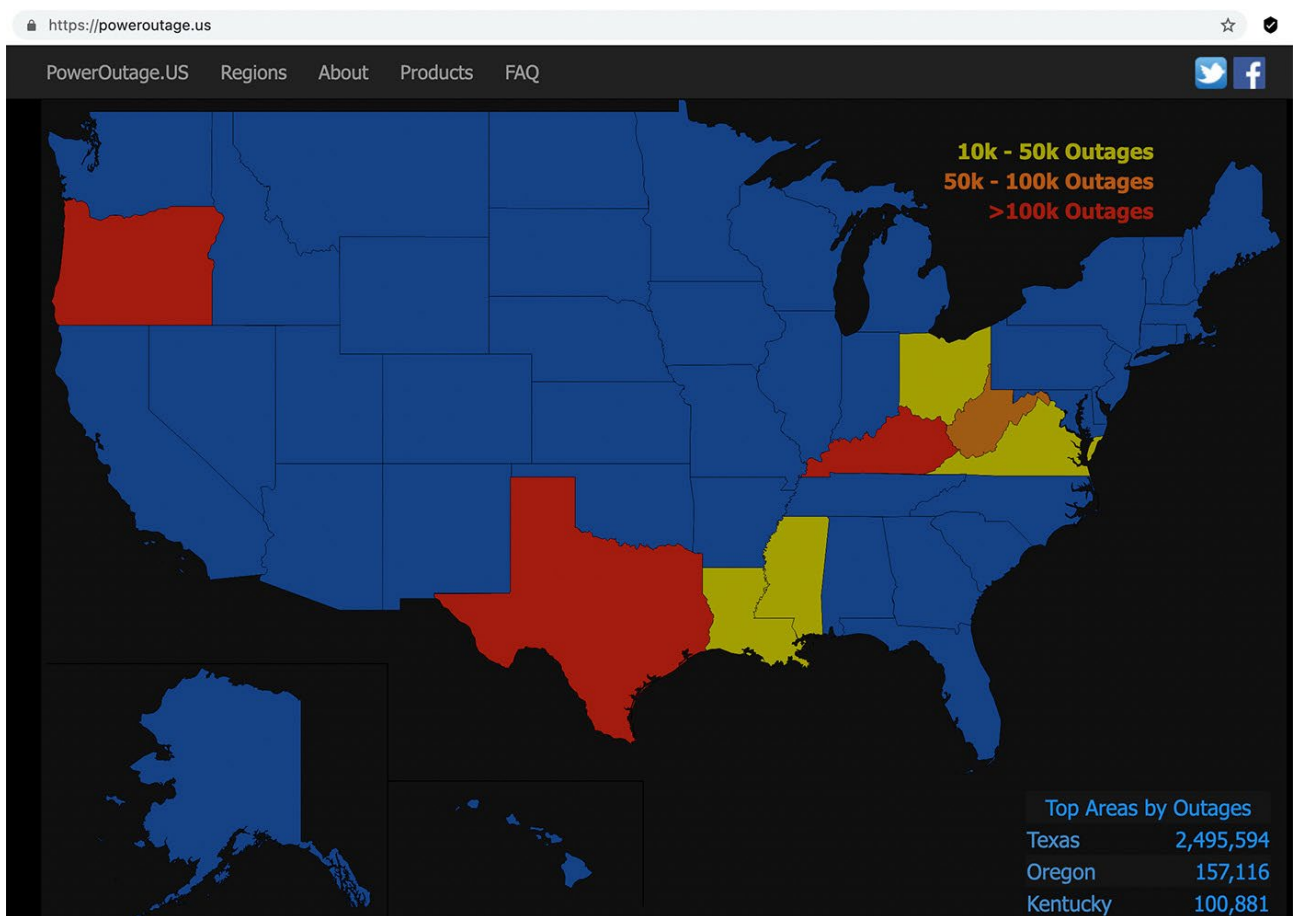


Рис. 1.5. Інтерфейс веб-сервісу PowerOutage.us

Для реалізації подібних рішень використовуються геоінформаційні системи (Google Maps API, Leaflet, OpenStreetMap), технології збору даних (веб-скрапінг, API, аналіз соціальних мереж) та механізми сповіщень (push-повідомлення, електронна пошта, месенджери) [10]. На основі проведеного аналізу можна зробити висновок, що існуючі рішення потребують подальшого

вдосконалення. Оптимальна система має поєднувати інтерактивну візуалізацію, персоналізовані сповіщення, дані про пункти незламності та аналітичні інструменти для прогнозування відключень, з урахуванням особливостей українського контексту.

1.2 Формулювання проблемної області

Проблемна область веб-сервісу для моніторингу відключень електроенергії в Україні охоплює комплекс технічних, інформаційних та соціальних аспектів. Специфіка ситуації обумовлена пошкодженням енергетичної інфраструктури внаслідок військових дій [5, 75], що призвело до необхідності впровадження планових і аварійних відключень [3, 74]. Незважаючи на часткове відновлення системи до 2025 року, ризики повторних атак зберігають актуальність проблеми.

Головною складовою проблемної області є відсутність єдиної систематизованої інформації про відключення. Наявні джерела даних фрагментовані та часто неструктуровані, що ускладнює їх обробку та аналіз. Особливо гостро ця проблема проявляється в малих населених пунктах, де доступ до інформації обмежений. Крім того, існує суттєва потреба в оперативних даних про альтернативні джерела енергопостачання.

Технічні виклики включають забезпечення роботи системи в умовах нестабільного електропостачання та інтернет-з'єднання, а також розробку механізмів верифікації даних для запобігання дезінформації. Важливим аспектом є створення доступного інтерфейсу для користувачів з різним рівнем цифрової грамотності.

Соціально-економічний контекст проблеми вимагає пошуку балансу між соціальною спрямованістю сервісу та економічною стійкістю його функціонування. Таким чином, розробка ефективного рішення потребує комплексного підходу, що враховує взаємозв'язок технологічних можливостей та актуальних потреб користувачів.

1.3 Проведення аналізу літературних та інформаційних джерел щодо можливостей вирішення виявлених проблем

Аналіз літературних та інформаційних джерел виявив різноманітні підходи до вирішення проблем, пов'язаних з моніторингом відключень електроенергії. Дослідження Суходолі О.М. [24] встановлюють концептуальні основи забезпечення енергетичної безпеки, акцентуючи на стратегічному плануванні в умовах кризових ситуацій. Запропонована автором багатофакторна модель оцінювання може слугувати теоретичною базою для прогностичних алгоритмів у системах моніторингу.

Значний внесок у методологію розробки таких систем внесли Kamariotis та Chatzi [44], які обґрунтували підхід, орієнтований на створення цінності для користувачів через підвищення стійкості та передбачуваності роботи інфраструктури. У сфері прогнозування навантаження на електромережі варто відзначити роботу Li, Zhang та Wang [49], запропоновано гібридну модель на основі алгоритму китового оптимізатора та LSTM-мережі, що підвищує точність передбачення відключень.

Сучасні технології штучного інтелекту, зокрема машинне навчання, відіграють важливу роль у оптимізації енергосистем [30]. Водночас дослідження Герасимчука О.В. та Коваленка Д.В. [11] демонструють практичну реалізацію подібних рішень через розробку мобільних додатків з використанням кросплатформових технологій, що актуально для забезпечення доступності сервісів у умовах нестабільного інтернет-з'єднання.

Важливим напрямом досліджень є оптимізація веб-систем для обробки великих масивів даних, що розглядається в роботах Кугучек [47]. Окрему увагу приділено захисту критичної інфраструктури шляхом інтеграції та верифікації даних, зокрема в дослідженнях Shepetukha та Koval [61]. Міжнародна практика також підтверджує ефективність комплексних підходів, які поєднують оперативне інформування користувачів, механізми взаємодії між споживачами та використання альтернативних джерел енергопостачання.

Проведений огляд свідчить про наявність достатньої теоретичної бази для розробки систем моніторингу, проте відсутність інтегрованих рішень, що враховують технічні, соціальні та інформаційні аспекти, обумовлює необхідність створення інноваційного веб-сервісу, адаптованого до специфіки енергетичної кризи в Україні.

1.4 Формулювання наукової новизни та інноваційності проєкту

Наукова новизна дослідження полягає в розробці комплексного веб-сервісу, що поєднує інтерактивну мапу відключень електроенергії, систему персоналізованих сповіщень та базу даних пунктів незламності з деталізованою інформацією про їх доступність та послуги. Запропонований підхід відрізняється від існуючих рішень системною інтеграцією моніторингу аварійних і запланованих відключень, що забезпечує єдиний інформаційний простір для користувачів.

Інноваційність проєкту проявляється у реалізації динамічного картографічного інтерфейсу, який візуалізує дані у реальному часі, забезпечуючи можливість фільтрації інформації за типами відключень, термінами їх дії та географічним розташуванням [7, 64]. Важливим науковим досягненням є розробка механізму краудсорсингового збору даних, що дозволяє користувачам самостійно повідомляти про відсутність електропостачання, підвищуючи актуальність інформації. Це поєднується з автоматизованим отриманням офіційних даних через API постачальників електроенергії, що забезпечує комплексне висвітлення ситуації [61].

Система персоналізованих сповіщень враховує не лише геолокацію користувача, а й його індивідуальні налаштування, такі як пріоритетні типи повідомлень (заплановані чи аварійні відключення) та спосіб їх отримання. Це особливо актуально для соціально вразливих груп населення, які потребують оперативного доступу до інформації про пункти незламності, їх робочий графік та перелік доступних послуг.

Таким чином, наукова новизна проєкту полягає у поєднанні технічних рішень, таких як інтеграція різноманітних джерел даних та інтерактивна візуалізація, з соціально орієнтованим функціоналом, що включає механізми краудсорсингу та персоналізації. Це не лише забезпечує оперативний моніторинг стану енергопостачання, але й надає користувачам інструменти для ефективного реагування у кризових ситуаціях.

1.5 Ідентифікація зацікавлених сторін проєкту

Успіх проєкту залежить від побудови ефективної системи комунікацій з усіма групами зацікавлених сторін. Ідентифікація стейкхолдерів на етапі ініціації дозволяє врахувати їхні очікування, мінімізувати опір та залучити необхідні ресурси [18]. Для проведення детального аналізу зацікавлені сторони проєкту класифіковано на дві групи:

1. Внутрішні: команда розробки, адміністратори системи та науковий керівник.
2. Зовнішні: пересічні громадяни, енергокомпанії, органи місцевої влади, бізнес-структури та волонтерські організації.

Розгорнуту матрицю аналізу зацікавлених сторін наведено в таблиці 1.1, де кожна група оцінена за шкалою від 1 до 5, де 1 – мінімальний рівень, 5 – максимальний.

Таблиця 1.1.

Матриця аналізу зацікавлених сторін

Стейкхолдер	Тип	Очікування та інтереси	Влада (1-5)	Інтерес (1-5)	Стратегія взаємодії	Джерело впливу / Ризики
1	2	3	4	5	6	7
Органи влади / КМДА	Зовн.	Соціальна стабільність, координація «Пунктів незламності».	5	5	Ключові гравці: Спільне прийняття рішень, постійні консультації.	Регулятивні норми. Ризик: законодавчі обмеження.

1	2	3	4	5	6	7
Енергокомпанії	Зовн.	Коректне поширення даних, зменшення навантаження на кол-центри.	5	3	Задоволення потреб: Підтримувати задоволеність, надаючи якісний інструмент.	Власники API. Ризик: закриття доступу до даних.
Громадяни України	Зовн.	Актуальні графіки, зручна мапа, пошук пунктів допомоги.	2	5	Інформування: Максимальна активність у комунікації планів та успіхів.	Джерело даних (краудсорсинг). Ризик: дезінформація.
Команда розробки	Внутр.	Технічна досконалість, професійний розвиток, завершення проєкту.	4	5	Співпраця: Повне залучення до процесів управління та мотивація.	Технічна реалізація. Ризик: професійне вигорання.
Бізнес-структури	Зовн.	Планування робочих годин, залучення клієнтів у блекаути.	3	4	Інформування: Надання рекламних можливостей та аналітики.	Економічна стійкість. Ризик: нестабільність ринку.
Волонтерські групи	Зовн.	Логістика допомоги, дані для гуманітарних місій.	2	5	Інформування: Надання доступу до аналітичних дашбордів.	Соціальна підтримка. Ризик: брак ресурсів.

Окрім матричного аналізу, застосовано модель Мітчелла, яка базується на трьох атрибутах: влада, легітимність та терміновість [26]. Це дозволяє виділити категорію «невідкладних» стейкхолдерів – вразливі верстви населення та медичні заклади, для яких інформація про відключення має найвищу пріоритетність через життєву необхідність стабільного енергопостачання. Стратегія взаємодії з цією групою передбачає впровадження пріоритетних сповіщень про автономні джерела живлення та медичну підтримку в межах «Пунктів незламності».

1.6 PEST- та SWOT-аналіз проєкту

Для успішної реалізації веб-сервісу моніторингу відключень світла необхідно дослідити особливості застосування існуючих методологічних підходів до оцінки зовнішніх впливів. Одним із найбільш ефективних інструментів для такого аналізу є PEST-підхід, який дозволяє виявити політичні, економічні, соціальні та технологічні фактори [16, 29]. Результати аналізу наведено у таблиці 1.2.

Таблиця 1.2.

PEST-аналіз факторів впливу на проєкт

Категорія факторів	Опис та характеристики факторів
1	2
Політичні	Державна політика у сфері енергетики, регулювання доступу до даних про стан енергосистеми та інфраструктурні реформи.
Економічні	Наявність фінансування, потенційні можливості монетизації сервісу та загальна економічна ситуація, що впливає на пріоритети користувачів.
Соціальні	Суспільна потреба в достовірній інформації про відключення електроенергії та характеристики цільової аудиторії.
Технологічні	Доступність необхідних технологій (ГІС, API), інфраструктури та технічних обмежень в умовах енергетичної кризи.

Аналіз наукових робіт підтверджує особливу роль інформаційних технологій у кризових ситуаціях. Ефективність ІТ-рішень у таких умовах залежить не лише від їх технічних характеристик, але й від здатності адаптуватися до швидкозмінного середовища та забезпечувати соціальну стабільність.

Відповідно до методології SWOT-аналізу, яка доповнює PEST-підхід, для проекту визначено ключові внутрішні та зовнішні фактори (Рис. 1.6). Результати аналізу представлені в таблиці 1.3.

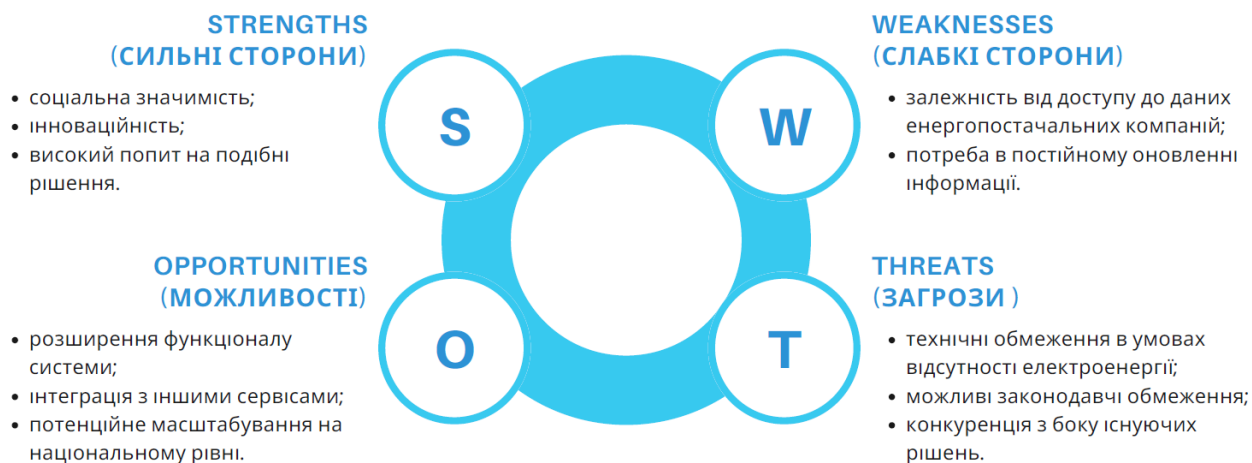


Рис. 1.6. SWOT-аналіз веб-сервісу моніторингу відключень світла

Таблиця 1.3.

Матриця SWOT-аналізу проекту

Категорія факторів	Позитивний вплив (Сильні сторони / Можливості)	Негативний вплив (Слабкі сторони / Загрози)
1	2	3
Внутрішні фактори	Strengths (сильні сторони): Висока соціальна значимість проекту для населення та інноваційність поєднання картографічного інтерфейсу з краудсорсингом за умов високого попиту на рішення.	Weaknesses (слабкі сторони): Критична залежність від стабільності доступу до даних енергокомпаній та постійна потреба в оновленні інформації для підтримки її актуальності.
Зовнішні фактори	Opportunities (можливості): Суттєве розширення функціоналу системи та подальша інтеграція з іншими державними сервісами з потенційним масштабуванням на національний рівень.	Threats (загрози): Існуючі технічні обмеження в умовах повної відсутності електроенергії та можливі законодавчі обмеження разом із конкуренцією з боку існуючих рішень.

Аналіз зовнішніх факторів за допомогою PEST- та SWOT-методологій дозволяє виявити ключові виклики та можливості. Умови енергетичної кризи створюють значні ризики, але одночасно формують попит на стійкі та соціально значущі IT-рішення [8, 20].

1.7 Побудова дерева причин та наслідків

Для аналізу проблемної області проєкту було побудовано дерево причин та наслідків (Рис. 1.7), що дозволяє структурувати фактори виникнення проблеми та їхній вплив на життєдіяльність суспільства. Центральною проблемою дослідження визначено відсутність ефективного інтегрованого сервісу для моніторингу відключень електроенергії в Україні, що зумовлює інформаційну невизначеність для мільйонів споживачів [53].

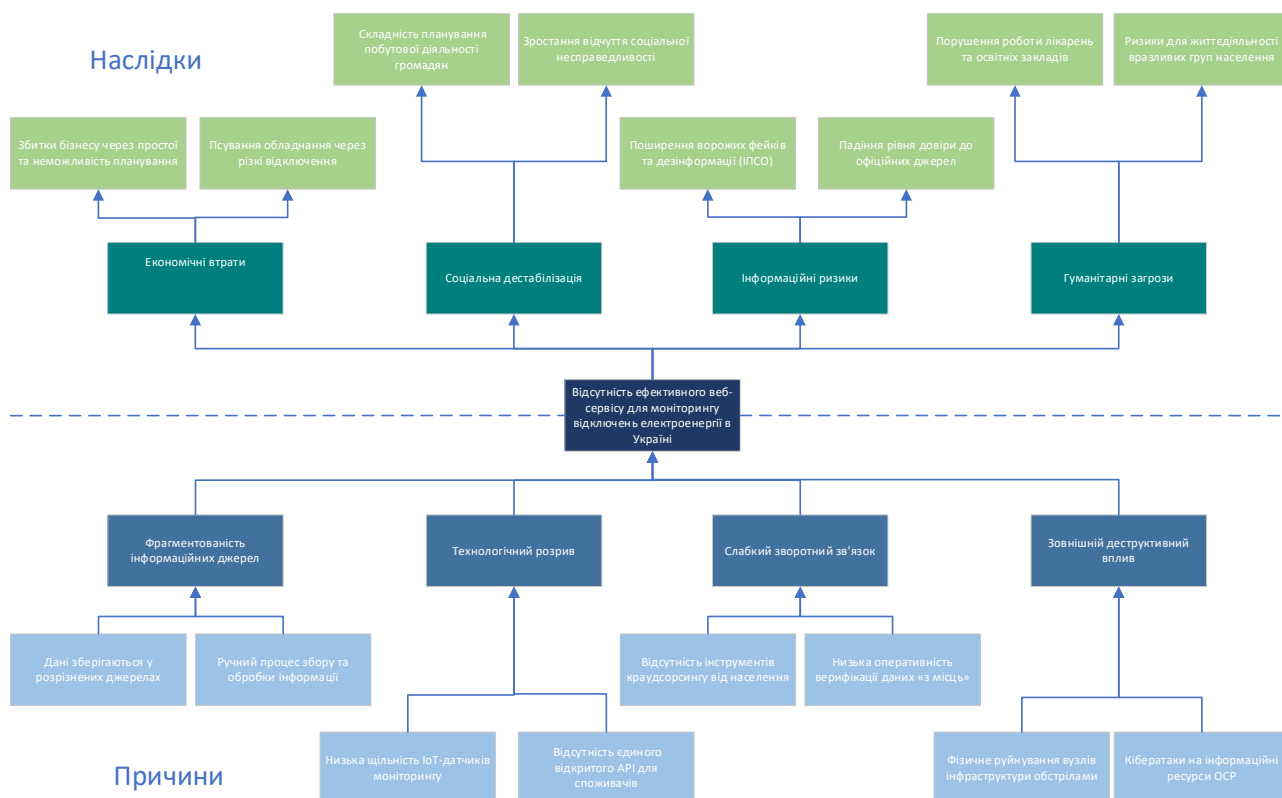


Рис. 1.7. Дерево причин та наслідків

Основними причинами виникнення проблеми є інформаційна розбіжність та технологічний розрив. Офіційні графіки часто базуються на статичних планових показниках, які не відображають оперативні зміни в мережі, а швидкість оновлення даних у публічних інформаційних каналах не дозволяє вчасно повідомляти про аварійні ситуації. Технічний аспект зумовлений низькою щільністю сучасних сенсорів у мережі та відсутністю інструментів зворотного зв'язку від населення для швидкої перевірки фактичного стану ліній безпосередньо споживачами. Також на ситуацію впливає зовнішній чинник: фізичне руйнування інфраструктури та кібератаки на енергетичні інформаційні ресурси [1].

Брак верифікованої інформації призводить до негативних наслідків у різних сферах. В економіці це проявляється у значних збитках бізнесу через простої та псування обладнання. У соціальній сфері це зумовлює складність планування побуту та активне поширення фейків, що знижує довіру споживачів до офіційних джерел. Найбільш критичними є гуманітарні загрози – порушення стабільної роботи медичних закладів та прямі ризики для життя вразливих груп населення. Проєкт спрямований на усунення цих розривів шляхом створення єдиної платформи моніторингу.

1.8 Визначення цілей проєкту та побудова дерева цілей

Етап визначення цілей є основою для успішної реалізації проєкту, оскільки він дозволяє встановити чіткі орієнтири для команди розробки та критерії оцінки якості майбутнього веб-сервісу [28]. Правильне формулювання цілей забезпечує розуміння результатів, які мають бути досягнуті на кожному етапі життєвого циклу проєкту, від проектування архітектури до впровадження системи [16].

Для систематизації цілей та візуалізації їхньої ієрархічної структури побудовано дерево цілей (Рис. 1.8). Цей метод дозволяє декомпонувати головну мету проєкту на окремі стратегічні напрями та операційні завдання.

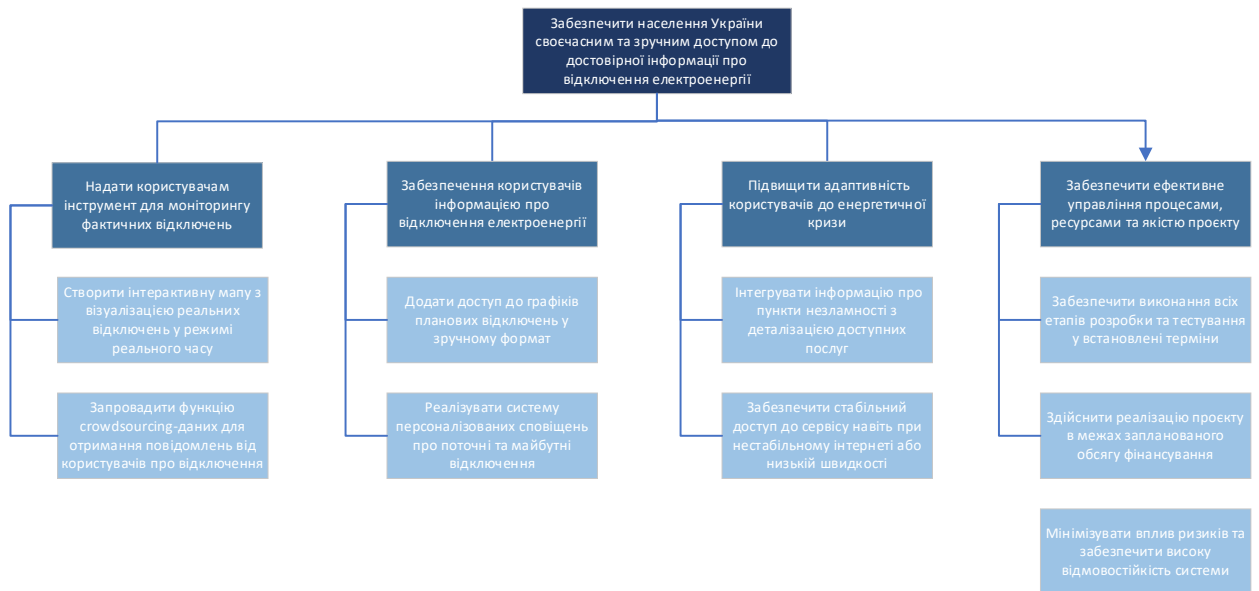


Рис. 1.8. Дерево цілей проекту

Головною метою проекту є забезпечення населення України своєчасним та зручним доступом до достовірної інформації про відключення електроенергії. Досягнення цієї мети реалізується через три стратегічні напрями:

- 1) Моніторинг фактичного стану мережі. Цей напрям передбачає створення інтерактивної мапи та впровадження функцій збору даних від користувачів для відображення реальної ситуації в енергосистемі в режимі реального часу.
- 2) Інформування та оповіщення користувачів. Охоплює надання доступу до верифікованих офіційних графіків відключень та розробку системи персоналізованих сповіщень.
- 3) Підтримка життєдіяльності в умовах енергетичної кризи. Спрямований на інтеграцію даних про розташування «Пунктів незламності» [2] та забезпечення стабільної роботи сервісу в умовах низької швидкості інтернет-з'єднання.
- 4) Управління процесами та ресурсами проекту. Передбачає забезпечення реалізації розробки у встановлені часові межі та в межах запланованого обсягу фінансування при постійному контролі якості та відмовостійкості системи.

На основі цих стратегічних напрямів було сформовано перелік операційних завдань, що охоплюють повний цикл реалізації сервісу. Зокрема, передбачено створення алгоритмів агрегації даних для інтерактивної мапи, налаштування інтелектуальних тригерів для системи сповіщень та побудову архітектури взаємодії з реєстрами об'єктів соціальної підтримки. Окрему увагу приділено управлінському аспекту: дотриманню календарного графіка виконання робіт, контролю витрат ресурсів та мінімізації впливу потенційних ризиків на стабільність продукту. Такий підхід гарантує надійність системи під час критичних навантажень на мережу та забезпечує зрозумілу структуру розробки [22].

Побудоване дерево цілей виступає інструментом контролю, що дозволяє команді розробки слідувати єдиному вектору реалізації проєкту та гарантувати відповідність продукту запитам територіальних громад.

1.9 Постановка задачі дослідження. Формулювання технічного завдання на розробку у вигляді паспорту проєкту

На основі результатів аналізу предметної області та ідентифікованих потреб зацікавлених сторін здійснюється ініціація розробки інформаційної системи. Для визначення меж проєкту та параметрів реалізації сформовано Паспорт проєкту, що представлено в таблиці 1.4, що фіксує основні характеристики технічного завдання та критерії успішності розробки.

Таблиця 1.4.

Паспорт проєкту розробки веб-сервісу моніторингу відключень світла в Україні

№ з/п	Найменування пункту	Опис
1	2	3
1.	Назва проєкту	Веб-сервіс для моніторингу відключень світла в Україні.
2.	Замовник проєкту	Органи місцевого самоврядування / Представники територіальних громад.

1	2	3
3.	Виконавець проекту	Мавроді Юлія Андріївна, студентка групи УП-21.
4.	Мета проекту	Дослідження та розробка веб-платформи для агрегації даних про стан енергомереж, візуалізації обмежень електропостачання та надання інструментів адаптації населення до дефіциту енергії.
5.	Технічне завдання	<p>5.1. Розробити інтерактивний картографічний інтерфейс на базі ГІС-технологій.</p> <p>5.2. Реалізувати автоматизовану інтеграцію з API зовнішніх енергопостачальних компаній.</p> <p>5.3. Створити краудсорсинговий модуль для збору звітів споживачів про фактичний стан мережі.</p> <p>5.4. Впровадити алгоритми верифікації повідомлень та розрахунку індексу довіри до даних.</p> <p>5.5. Спроекувати та налаштувати документо-орієнтовану базу даних (MongoDB).</p> <p>5.6. Розробити модуль відображення пунктів незламності з деталізацією доступних послуг.</p> <p>5.7. Реалізувати механізм персоналізованих сповіщень (push, email, SMS) за геолокацією.</p> <p>5.8. Створити функціонал багатокритеріальної фільтрації даних за типами подій та регіонами.</p> <p>5.9. Розробити аналітичний блок для виявлення тенденцій та прогнозування обмежень.</p> <p>5.10. Забезпечити доступ до кешованої інформації за умов відсутності зв'язку.</p> <p>5.11. Реалізувати рольову модель доступу для різних категорій користувачів.</p> <p>5.12. Впровадити систему журналювання подій для аудиту безпеки даних.</p> <p>5.13. Розробити адаптивний дизайн для коректного відображення на мобільних пристроях.</p> <p>5.14. Створити систему автоматичної генерації звітів у форматах PDF та Excel.</p> <p>5.15. Налаштувати серверне середовище та хмарну інфраструктуру проекту.</p> <p>5.16. Провести навантажувальне тестування системи для роботи при пікових запитах.</p> <p>5.17. Виконати комплекс заходів із забезпечення кібербезпеки та захисту даних.</p>
6.	Очікувані результати	<p>6.1. Розгорнутий функціональний прототип сервісу з мапою та модулем сповіщень.</p> <p>6.2. Протестовані алгоритми крос-валідації офіційних та користувацьких даних.</p> <p>6.3. Оптимізований інтерфейс згідно з результатами юзабіліті-тестування.</p> <p>6.4. Готовий продукт для впровадження у систему кризового інформування.</p>

1	2	3
7.	Терміни виконання	7.1. Початок проєкту: 01.04.2026. 7.2. Завершення проєкту: 22.05.2026.
8.	Бюджет	23 950.84 USD.
9.	Відповідальні особи	9.1. Власник продукту: Мавроді Ю. А. 9.2. Науковий керівник: Латишева Т. В.
10.	Стадії реалізації	Ініціація, планування та дизайн, розробка, тестування, розгортання, підтримка.
11.	Ризики та обмеження	11.1. Внутрішні ризики: висока плинність кадрів, недостатня експертиза у сфері енергетичних даних, затримки з боку аналітика, низький рівень комунікації в команді. 11.2. Технічні ризики: затримки або помилки у даних від операторів систем розподілу, відмова серверів при пікових навантаженнях, кібератаки (DDoS, SQL-ін'єкції), втрата даних через відсутність резервного копіювання. 11.3. Організаційні ризики: відсутність офіційних угод з енергопостачальними компаніями, непрозора відповідальність за некоректні дані, невчасне оновлення інформації, затримка в ухваленні рішень замовником. 11.4. Зовнішні / Форс-мажори: військові дії (зупинка дата-центрів), припинення роботи сторонніх хмарних сервісів (AWS, Google Cloud, SMS-провайдери), зміни в законодавстві щодо відкритих даних. 11.5. Фінансові ризики: недостатнє фінансування на етапі підтримки, зростання вартості хмарних послуг, несвоєчасне отримання грантів, перевищення бюджету через розширення функціоналу.

Формулювання технічного завдання на розробку у вигляді паспорту проєкту дає змогу чітко окреслити межі розробки та зафіксувати всі основні вимоги до майбутнього сервісу в одному документі. Це допомагає команді краще розуміти фінальний результат і працювати в межах визначеного бюджету та термінів. Паспорт проєкту - це основа для подальшої практичної реалізації системи, де кожен функціональний модуль розроблятиметься за вже узгодженими параметрами.

РОЗДІЛ 2. МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

2.1 Розробка концептуальної моделі інформаційної системи

Для побудови концептуальної моделі веб-сервісу моніторингу відключень електроенергії необхідно визначити ключові елементи системи та встановити їх взаємозв'язки. Оскільки система інтегрує принципи обробки геопросторових даних [10], механізми оперативного інформування та краудсорсингові методи збору інформації, її концептуальна модель включає такі структурні компоненти:

1. Веб-застосунок (інтерфейс користувача, адміністративна панель, база даних, система збору даних, модуль обробки та розсилки сповіщень, аналітичний модуль, система звітності).
2. Учасники системи (фізичні особи, юридичні особи, волонтерські організації, представники органів місцевого самоврядування, оператори енергетичних компаній, адміністратори системи).
3. Інформаційні об'єкти (офіційні графіки відключень, дані про фактичні аварії, геолокаційні координати та статус пунктів незламності, архів історичних даних, користувацькі звернення та повідомлення про проблеми).
4. Вихідні матеріали (інтерактивні дашборди моніторингу, регіональні звіти про стан енергопостачання, аналітичні огляди з виявленням тенденцій, візуалізація показників стабільності, індивідуалізовані зведення для різних категорій користувачів).

Оскільки система функціонує в контексті управління критичною інфраструктурою [4], надсистема включає державні системи моніторингу енергетичної безпеки, інформаційні платформи енергопостачальників, нормативно-правову базу з питань енергетики, сучасні методології кризового реагування та стандарти обробки геопросторових даних.

Визначивши усі компоненти системи, сформуємо концептуальну модель (Додаток А), яка відобразатиме взаємозв'язки між елементами системи та зовнішнім середовищем [19].

Для деталізації внутрішньої логіки та архітектурної стратегії системи розроблено багаторівневу концептуальну модель (Рис. 2.1).

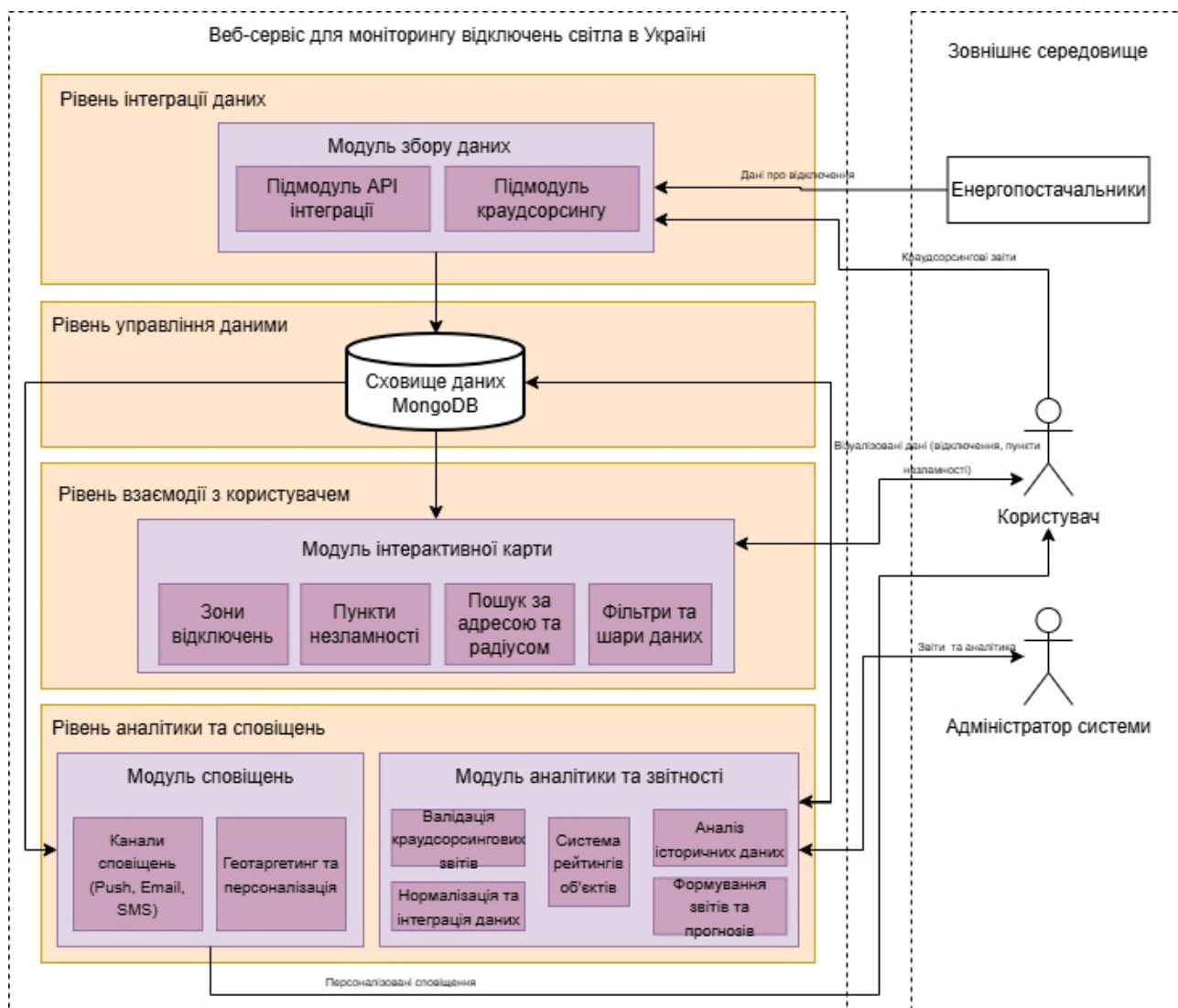


Рис. 2.1. Багаторівнева концептуальна модель веб-сервісу

Дана модель структурує сервіс за чотирма рівнями:

1. Рівень інтеграції даних: включає Модуль збору даних, що фокусується на отриманні вхідних потоків через API інтеграцію з

енергопостачальниками та форми краудсорсингового збору повідомлень від користувачів.

2. Рівень управління даними: базується на Сховищі даних MongoDB, що забезпечує збереження та цілісність накопиченої інформації.
3. Рівень взаємодії з користувачем: реалізується через Модуль інтерактивної карти, що забезпечує візуалізацію зон відключень та пунктів незламності, пошук за адресою та керування шарами даних.
4. Рівень аналітики та сповіщень: об'єднує Модуль сповіщень (канали Push, Email, SMS та алгоритми геотаргетингу) та Модуль аналітики та звітності, який виконує не лише функцію формування звітів, а й здійснює валідацію краудсорсингових звітів, нормалізацію даних та розрахунок рейтингів об'єктів допомоги.

Для деталізації інформаційних потоків у межах розробки інтелектуальних функцій сервісу, структуровано вхідні дані за функціональними блоками в таблиці 2.1.

Таблиця 2.1.

Вхідні дані концептуальної моделі інформаційної системи

Блок моделі	Підблоки моделі	Елементи, що входять до підблоку (атрибути БД)
1	2	3
Офіційні дані (Providers)	Параметри відключень	Ідентифікатор постачальника (provider_id). Тип події (outage_type: планова/аварійна). Час початку (start_time). Час завершення (end_time). Географічна адреса (address).
	Просторовий опис	Координати об'єкта (latitude/longitude). Опис причини (description). Поточний статус (status).
Краудсорсинг (User Reports)	Сигнали користувачів	Ідентифікатор звіту (report_id). ID користувача (user_id). Часова мітка (timestamp). Фактична наявність світла (status: On/Off).
	Валідаційні дані	Точні координати звіту. Рівень верифікації (verification_score). Прив'язка до офіційної події (outage_id).

1	2	3
Ресурсні об'єкти (Resilience)	Пункти незламності	Ідентифікатор пункту (point_id). Назва (name). Адреса. Графік роботи (working_hours). Загальна місткість (capacity).
	Стан ресурсів	Перелік послуг (services: інтернет, тепло, розетки). Поточне навантаження відвідувачами (current_load).
Користувачі (Users)	Налаштування профілю	Унікальний ID (user_id). Пошта (email). Пароль (password_hash). Перелік адрес для відстеження (address). Типи сповіщень (notification_preferences).

Процеси функціонування системи реалізують логіку трансформації вхідних даних у верифіковану інформацію для населення та наведені в таблиці 2.2.

Таблиця 2.2.

Процеси функціонування інформаційної системи

Рівень моделі	Модулі та алгоритми	Функціональне призначення
1	2	3
Інтеграція	Модуль збору	Агрегація офіційної телеметрії та отримання краудсорсингових звітів.
Управління	Сховище MongoDB	Збереження документів, управління цілісністю та геопросторовими індексами.
Взаємодія	Інтерактивна мапа	Візуалізація ГІС-шарів, відображення маркерів пунктів та зон дефіциту енергії.
Аналітика та сповіщення	Аналітичний модуль	Верифікація та валідація звітів, нормалізація даних, розрахунок рейтингів та аналіз історії.
	Модуль сповіщень	Геотаргетинг та розсилка персоналізованих повідомлень (Push, Email, SMS).

Програмне забезпечення веб-сервісу базується на клієнт-серверній архітектурі, що забезпечує високу швидкість обробки запитів та масштабованість системи [46]. Клієнтська частина надає інтерфейс для взаємодії через веб-браузер, а серверна частина виконує роль інтелектуального ядра,

керуючи модулями збору, карти, сповіщень та аналітики. Дана архітектура, побудована з використанням хмарних ресурсів AWS, гарантує відмовостійкість сервісу в умовах критичних навантажень на інтернет-інфраструктуру [52].

Результатом реалізації моделі є формування вихідних аналітичних та сервісних продуктів, наведені в таблиці 2.3.

Таблиця 2.3.

Вихідні дані та результати роботи системи

Блок моделі	Підблоки моделі	Результати та елементи
1	2	3
Сервіси	Інтерактивна мапа	Візуалізація зон відключень. Картки пунктів незламності з переліком доступних послуг
	Повідомлення	Персоналізовані алерти про зміну статусу живлення. Рекомендації найближчих об'єктів допомоги
Аналітика	Звітність	Регіональна статистика тривалості відключень. Аналіз ефективності та завантаженості пунктів допомоги
	Верифікація	Звіти про підтверджену достовірність інформації Валідовані статуси мережі

Ефективність функціонування моделі обмежується низкою технологічних, юридичних та ресурсних факторів, наведених в таблиці 2.4.

Таблиця 2.4.

Параметри обмеження системи

Блок моделі	Категорія обмеження	Опис обмежень
1	2	3
Технологічні	Стабільність середовища	Частота запитів до API. Наявність інтернет-зв'язку у споживачів. Відмовостійкість серверів AWS
Юридичні	Безпека та цензура	Відповідність Закону про захист персональних даних. Заборона на оприлюднення координат критичних енерговузлів
Проектні	Ресурсна база	Кінцевий термін релізу (травень 2026). Ліміти бюджету на хмарні сервіси. Мінімумально необхідний набір функцій (MVP)

Прикладом використання розробленої моделі є сценарій пошуку локації для дистанційної роботи в умовах тривалого блекауту. Користувач ініціює запит у системі, вказуючи критичні параметри: наявність стабільного інтернет-зв'язку, можливість зарядки пристроїв та радіус доступності до 2 км.

Інтелектуальне ядро системи звертається до бази даних MongoDB для отримання переліку Пунктів незламності, що підпадають під задані фільтри. У результаті модуль аналітики видає ранжований за інтегральним показником список, де система може рекомендувати об'єкт, що знаходиться трохи далі, наприклад, 1.5 км замість 500 м, але має нульовий показник завантаженості та підтверджений статус наявності палива. Після того як користувач прибуває на локацію, він може надіслати краудсорсинговий звіт про фактичний стан послуг, що запускає процес оновлення даних у системі, який у майбутньому можливо автоматизувати через IoT-сенсори.

Для забезпечення об'єктивності вибору оптимальних «Пунктів незламності» та автоматизації верифікації суперечливих даних про стан мереж необхідно впровадити математичний підхід до оцінювання ресурсів [12, 54]. У межах даного дослідження кожен об'єкт інфраструктури або інформаційний звіт розглядається як набір параметрів з відповідними вагами та ступенем критичності.

2.1.1 Формалізація математичних об'єктів системи

Нехай кожен об'єкт «Пункт незламності» у системі представлений як об'єкт P_i . Набір параметрів для кожного пункту D_i , до якого входять технічні характеристики та умови перебування, описується формулою (2.1), як кортеж характеристик:

$$P_i = \{D_1, D_2, \dots, D_n\} \quad (2.1)$$

де P_i – конкретний Пункт незламності;

i – унікальний ідентифікатор об'єкта;

D_i – статичні дані (адреса, координати) та динамічні (поточне завантаження, статус роботи).

Для запиту користувача Q , спрямованого на пошук допомоги, визначається функція підбору пунктів L_i за формулою (2.2):

$$L_i = f(F, T, C) \quad (2.2)$$

де L_i – сформований ранжований перелік об'єктів-кандидатів;

F – множина фільтрів (наявність Starlink, тепло, потужність генератора);

T – часовий інтервал доступності об'єкта;

C – повна множина об'єктів у базі даних MongoDB.

Вихідні дані системи після взаємодії з об'єктом можна представити як оновлений стан SP за допомогою формули (2.3):

$$SP = \{D_i, U_i\} \quad (2.3)$$

де D_i – оновлені технічні параметри (наприклад, наявність палива);

U_i – зміна показника завантаженості об'єкта відвідувачами.

2.1.2 Ваги параметрів для оцінки Пунктів незламності

Для коректної роботи алгоритму рекомендацій необхідно було визначити ваги характеристик, що базуються на пріоритетності потреб населення під час тривалих відключень, які описані в таблиці 2.5.

Таблиця 2.5.

Ваги параметрів моделі оцінки Пунктів незламності

Назва параметру	Критичність	Опис критерію	Вага (w)
1	2	3	4
Автономне живлення	Висока	Наявність генератора та достатньої потужності для зарядки пристроїв.	0.35
Зв'язок та інтернет	Висока	Наявність Starlink або оптоволоконних ліній з резервним живленням.	0.30
Завантаженість	Висока	Співвідношення поточних відвідувачів до загальної місткості.	0.15

1	2	3	4
Опалення	Середня	Наявність автономного обігріву при температурі нижче +18°C.	0.10
Санітарні умови	Середня	Наявність питної води та засобів гігієни.	0.05
Відгуки користувачів	Низька	Середній рейтинг об'єкта в системі (1–5 зірок).	0.05

Загальний інтегральний показник ефективності об'єкта I_{rp} розраховується за формулою (2.4):

$$I_{rp} = \sum_{j=1}^n (Score_j * w_j) \quad (2.4)$$

де $Score_j$ – нормалізована оцінка за шкалою від 0 до 10;

w_j – вага відповідного критерію;

2.1.3 Верифікація краудсорсингових звітів (Data Trust)

Для фільтрації неправдивих повідомлень у системі впроваджується індекс довіри V_{score} , який розраховується за наступними ваговими коефіцієнтами, що наведено в таблиці 2.6.

Таблиця 2.6.

Ваги параметрів верифікації повідомлень про відключення

Назва критерію	Критичність	Оцінка в моделі	Вага (w)
1	2	3	4
Синхронність з ОСР	Висока	Відповідність звіту плановому графіку постачальника.	0.45
Просторова згода	Висока	Кількість схожих звітів у радіусі 500 м за 15 хв.	0.35
Репутація автора	Середня	Історія підтверджених повідомлень даного користувача.	0.15
Актуальність часу	Низька	Швидкість згасання актуальності повідомлення.	0.05

Рішення про візуалізацію події на мапі приймається, якщо $V_{score} > T_{threshold}$, де поріг чутливості $T_{threshold}$ регулюється адміністратором системи.

Розроблена концептуальна модель, поєднуючи технічні атрибути бази даних та математичні методи ранжування, створює фундамент для переходу від простого інформування до інтелектуального управління інформаційними потоками в умовах кризи.

2.2 Постановка задачі в математичному вигляді відповідно до концептуальної моделі

Як було визначено у попередньому розділі, веб-сервіс вирішує проблему моніторингу відключень електроенергії, агрегації даних з різних джерел та надання персоналізованих сповіщень користувачам. Для цього необхідно формалізувати математичні моделі обробки даних, прогнозування відключень та візуалізації інформації [54].

2.2.1 Визначення змінних та множин системи

Для побудови математичного апарату введемо наступні змінні:

1. Множина регіонів України $R = \{r_1, r_2, \dots, r_n\}$, де r_1, r_2, \dots, r_n – регіони (області, міста, райони);
2. Множина відключень електроенергії $P = \{p_1, p_2, \dots, p_n\}$, де p_1, p_2, \dots, p_n – відключення (заплановані або аварійні);
3. Множина пунктів незламності $S = \{s_1, s_2, \dots, s_n\}$, де s_1, s_2, \dots, s_n – пункти незламності (координати, статус, послуги);
4. Множина користувачів $U = \{u_1, u_2, \dots, u_n\}$, де u_1, u_2, \dots, u_n – зареєстровані користувачі (геолокація, налаштування сповіщень);
5. Множина джерел даних $D = \{d_1, d_2, \dots, d_n\}$, де d_1, d_2, \dots, d_n – джерела даних (API енергокомпаній, краудсорсинг, офіційні звіти);

6. Множина часових інтервалів $T = \{t_1, t_2, \dots, t_n\}$, де t_1, t_2, \dots, t_n – періоди часу (хвилина, година, день).

Обмеження:

- $R, P, S, U, D, T > 0$;
- $\forall p_i \in P$ обов'язково має часову мітку *Timestamp* та координати (lat_i, lon_i) ;
- $\forall s_i \in S$ має параметри доступності $a_i \in 0,1$.

2.2.2 Математичний опис функцій системи

1. Збір та обробка даних:

- Функція збору даних з API $f_{API}: D_{official} \rightarrow P$ відповідає за отримання офіційних даних про відключення;
- Функція краудсорсингового збору $f_{crowd}: U \times T \rightarrow P$ відповідає за обробку повідомлень користувачів;
- Функція валідації даних $f_{validate}: P \times D \rightarrow P_{valid}$ забезпечує фільтрацію непідтверджених або недостовірних даних.

2. Візуалізація на карті:

- Функція відображення відключень $f_{map}: P \times R \rightarrow M$, де M – інтерактивна мапа з шарами відключень;
- Функція фільтрації за часом $f_{filter}: P \times T \rightarrow P_{filtered}$ відповідає за вибірку даних за вказаним періодом.

3. Прогнозування відключень:

Нехай P_{hist} – історичні дані відключень, а W – сукупність зовнішніх факторів, таких як погодні умови та атаки на інфраструктуру.

- Функція прогнозування $f_{predict}: P_{hist} \times W \rightarrow P_{future}$ відповідає за передбачення ймовірних відключень.

Прогнозована кількість відключень у регіоні r_i обчислюється за формулою (2.5):

$$P_{future}(r_i) = \alpha * \sum_{t=1}^n P_{hist}(r_i, t) + \beta * W(r_i) \quad (2.5)$$

де α – коефіцієнт історичної залежності;

β – коефіцієнт впливу зовнішніх факторів.

4. Робота пунктів незламності:

- Функція оновлення статусу $f_{update} : S \times T \rightarrow S_{updated}$ відповідає за зміну доступності пунктів у часі;

Навантаження на пункт s_i визначається за формулою (2.6):

$$L(s_i) = \frac{N_{users}(s_i)}{C(s_i)} \quad (2.6)$$

де $N_{users}(s_i)$ – кількість відвідувачів; $C(s_i)$ – місткість пункту.

5. Система сповіщень:

- Функція персоналізованих сповіщень $f_{notify} : U \times P \times S \rightarrow Notifications$ відповідає за розсилку повідомлень на основі геолокації та налаштувань.

Важливість сповіщення для користувача u_i обчислюється за формулою (2.7):

$$Priority(u_i, p_i) = \frac{1}{distance(u_i, p_i)} * Severity(p_i) \quad (2.7)$$

де $distance(u_i, p_i)$ – відстань до відключення;

$Severity(p_i)$ – рівень критичності (аварійне/заплановане).

2.2.3 Модель інтегральної оцінки об'єктів та звітів

За аналогією до методів оцінки людських ресурсів, для розрахунку пріоритетності Пункту незламності або достовірності звіту використовується нормалізований інтегральний показник R_{total} . Загальний показник для вибору конкретного об'єкта розраховується за формулою (2.8):

$$R_{total} = \sum_{i=1}^n (Score_i * w_i) \quad (2.8)$$

де $Score_i$ – нормалізована оцінка за шкалою від 0 до 10;

w_i – вага відповідного критерію.

Для впровадження інтелектуальної логіки, розгорнута оцінка Punkту незламності рахується за формулою (2.9):

$$I_{rp} = (S_{gen} * w_1) + (S_{conn} * w_2) + (S_{load} * w_3) + (S_{temp} * w_4) + (S_{san} * w_5) \quad (2.9)$$

де S_{gen} – оцінка критерію автономного живлення об'єкта;

S_{conn} – оцінка критерію зв'язку та інтернету;

S_{load} – інверсований показник поточного завантаження пункту відвідувачами;

S_{temp} – оцінка критерію опалення та тепла в приміщенні;

S_{san} – коефіцієнт наявності належних санітарних умов (питна вода, гігієна);

w_1, w_2, w_3, w_4, w_5 – відповідні вагові коефіцієнти критичності кожного з параметрів.

2.2.4 Конвертація завантаженості у бали

Світові практики та вимоги безпеки обмежують оптимальну завантаженість об'єктів допомоги. Якщо пункт переповнений, його привабливість для нового користувача падає. Маємо наступну систему рівнянь для показника завантаженості, що вираховується за формулою (2.10):

$$S_{load} = \begin{cases} 1 - \frac{CurrentLoad}{Capacity}, & CurrentLoad < Capacity \\ 0, & CurrentLoad \geq Capacity \end{cases} \quad (2.10)$$

де $CurrentLoad$ – поточна кількість відвідувачів у пункті;

$Capacity$ – максимальна місткість об'єкта.

Чим менше завантажений пункт – тим вищий бал він отримує у списку рекомендацій.

2.2.5 Алгоритм оцінки достовірності даних

Для розрахунку валідності краудсорсингового звіту використовується понижуючий коефіцієнт часу $Penalty_{time}$, що базується на швидкості згасання актуальності повідомлення, який вираховується за формулою (2.11):

$$V_{score} = S_{trust} * (1 - Penalty_{time}) \quad (2.11)$$

де V_{score} – підсумковий індекс довіри та валідності краудсорсингового звіту;

S_{trust} – базова оцінка достовірності повідомлення, що вираховується на основі синхронності з офіційними графіками постачальника, просторової згоди з іншими звітами та поточної репутації автора;

$Penalty_{time}$ – понижуючий коефіцієнт часу, який відображає швидкість згасання актуальності повідомлення з моменту його надсилання.

В результаті отримано верхньорівневий перелік ваг, який наведено в таблиці 2.7, що формують математичний базис для підбору найкращого варіанту допомоги користувачу.

Таблиця 2.7.

Перелік параметрів та ваг для формування оцінки об'єктів

Параметр	Вага (w)	Тип розрахунку в моделі (Si)
1	2	3
Автономне живлення	0.35	Пряме значення (Потужність / 10)
Зв'язок	0.30	Бінарне значення (0 або 1)
Поточне завантаження	0.15	Інверсоване значення
Опалення / Тепло	0.10	Пряме значення (Score / 10)
Санітарні умови	0.05	Коефіцієнт наявності послуг
Рейтинг користувачів	0.05	Пряме значення (Score / 5)

Для звуження кола пошуку використовується логічний фільтр-множник P_{match} , який відсікає об'єкти, що не відповідають критичним вимогам

користувача (наприклад, потреба в інклюзивності), що наведено в формулі (2.12):

$$R = I_{rp} * P_{match} \quad (2.12)$$

де R – підсумковий скоригований рейтинг об'єкта після застосування жорстких критичних фільтрів;

I_{rp} – базовий інтегральний показник ефективності Punkту незламності, розрахований за сумою м'яких параметрів;

P_{match} – логічний бінарний фільтр-множник, який приймає значення 1, якщо об'єкт має критично необхідну послугу або 0, якщо об'єкт не відповідає запиту.

Така постановка задачі дозволяє перетворити процес моніторингу на динамічну систему управління, яка автоматично ранжує інформацію та інфраструктурні об'єкти за ступенем їхньої корисності та актуальності для кінцевого споживача.

2.3 Застосування розроблених моделей в межах проєкту розробки веб-сервісу

Запропоновані у дослідженні моделі та алгоритми будуть використані в основних модулях веб-сервісу, забезпечуючи збір даних про відключення та їх візуалізацію. Математичні алгоритми реалізуються в інтерактивній мапі та системі сповіщень. В подальшому, розроблені підходи дозволять інтегрувати сервіс з існуючими системами моніторингу. Ключову роль відіграють алгоритми прогнозування, що підвищують ефективність попередження користувачів.

Основним процесом практичного застосування моделей є інтелектуальний пошук стабільної локації або моніторинг стану мережі користувачем, який має розуміння необхідних послуг (інтернет, тепло) та радіусу доступності. Процес взаємодії з системою складається з трьох ключових етапів.

2.3.1 Формування запиту та параметризація фільтрів

Користувач ініціює запит через вебінтерфейс, визначаючи набір критичних умов для пошуку допомоги. Нехай F – множина заданих фільтрів (наявність Starlink, мінімальна потужність генератора, санітарні умови), що вираховується за формулою (2.13):

$$F = \{f_{1i}, f_{2i}, \dots, f_{ni}\} \quad (2.13)$$

де f_{ni} – n -й критерій вибору об'єкта стійкості.

Сформувавши множину F , користувач створює запит Q , який поєднує цільову локацію L_{user} та обрані фільтри F , який описується формулою (2.14):

$$Q = \{L_{user}, F, R_{radius}\} \quad (2.14)$$

де L_{user} – цільова локація або поточне географічне розташування користувача на мапі;

F – множина обраних користувачем фільтрів та технічних умов;

R_{radius} – гранична відстань, яку користувач готовий подолати до об'єкта.

2.3.2 Ідентифікація об'єктів та інтелектуальне ранжування

Система звертається до бази даних MongoDB, яка містить актуальні записи про всі Пункти незламності $S = \{s_1, s_2, \dots, s_n\}$. Функція відбору f_{select} здійснює первинне відсікання об'єктів, що не відповідають жорстким фільтрам F або знаходяться поза межами R_{radius} :

$$f_{notify}(S, Q) \rightarrow S^1 \quad (2.15)$$

де S – підмножина кандидатів, що відповідають вимогам та мають статус «Відкрито».

Для миттєвого ранжування результатів використовується інтегральний показник I_{rp} , розроблений у пункті 2.1.1. Система формує топ-список локацій з найвищим балом за допомогою формули (2.16):

$$I_{rp}(s_i) = \sum_{j=1}^k (w_j, s_{ij}) - P(load_i) \quad (2.16)$$

де w_j – вага критерію;

s_{ij} – оцінка об'єкта;

$P(load_i)$ – штрафна функція за критичне завантаження пункту відвідувачами. Якщо система виявляє, що найближчий пункт переповнений ($load \geq 90\%$), вона автоматично підвищить пріоритет об'єкта, що знаходиться далі, але має вільні місця та стабільний інтернет.

2.3.3 Оновлення стану та зворотний зв'язок

Після вибору користувачем конкретного пункту та взаємодії з ним (наприклад, через надсилання краудсорсингового звіту про наявність світла), система оновлює глобальну матрицю стану SP для даного об'єкта, що вираховується за формулою (2.17):

$$SP_{new} = SP_{old} + \Delta SP_{user} \quad (2.17)$$

де SP_{old} – попереднє значення показника стану об'єкта;

ΔSP_{user} – внесок користувача у верифікацію даних (підтвердження статусу живлення або завантаженості).

Якість пропрацювання цієї концептуальної схеми визначає рівень довіри до сервісу. Використання математичного ранжування дозволяє мінімізувати інформаційний шум [40] і забезпечити користувачів найбільш ефективними варіантами адаптації до енергетичної кризи, базуючись на реальних даних, а не лише на планових графіках [32, 35].

РОЗДІЛ 3. ВИБІР ТЕХНОЛОГІЇ УПРАВЛІННЯ ПРОЄКТОМ ТА РОЗРОБКА ЙОГО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір технології управління проєктом

У сучасній практиці управління проєктами найчастіше розглядають два основні підходи: традиційний (каскадний) та гнучкий (адаптивний) [6].

Традиційна каскадна модель Waterfall передбачає сувору лінійну послідовність етапів: від збору вимог до фінального тестування готового продукту. Цей метод добре підходить для стабільних проєктів, де обсяг робіт зрозумілий від самого початку. Однак для розробки веб-сервісу моніторингу відключень такий підхід є досить ризикованим. Будь-яка зміна у форматі даних від енергокомпаній або ситуації в країні змусила б команду повертатися на етап проєктування, що призвело б до серйозних затримок релізу [13].

Тому для реалізації проєкту було обрано гнучку методологію Agile, яка базується на ітеративному розвитку продукту та постійній готовності до змін. Agile фокусується на швидкому створенні робочого інструменту, тісній співпраці із замовником та оперативності у прийнятті рішень. Оскільки веб-сервіс розробляється в умовах високої невизначеності, коли вимоги можуть змінюватися разом із ситуацією в енергосистемі, саме адаптивний підхід є найбільш доцільним [9, 20].

У межах гнучкого підходу було обрано фреймворк Scrum [51]. Він дозволяє структурувати роботу короткими циклами – спринтами, тривалістю по два тижні. Кожен такий спринт завершується створенням конкретної функції, наприклад, інтерактивної мапи або системи сповіщень [31]. Це дає можливість команді бути прозорою у своїх діях, регулярно перевіряти результати та миттєво адаптуватися до нових викликів [17].

Порівняльний аналіз методологій, який наведено у таблиці 3.1, підтверджує, що для проєктів, які реалізуються під час енергетичної кризи, Scrum є оптимальним інструментом [5]. Основною перевагою обраного методу є можливість випустити базову версію сервісу MVP у найкоротші терміни,

постійно уточнюючи алгоритми роботи на основі реального досвіду користувачів [62].

Таблиця 3.1.

Порівняльний аналіз методологій управління для проєкту моніторингу

Критерій порівняння	Waterfall (Каскадний підхід)	Scrum (Обраний Agile-фреймворк)	Конкретний вплив на проєкт
1	2	3	4
Структура вимог	Фіксується на старті (ТЗ).	Динамічний беклог (Product Backlog).	Дозволяє миттєво додати функцію «Пошук пунктів незламності» у відповідь на блекаут.
Поставка результату	Одноразово в кінці розробки.	Інкрементально кожні 2 тижні.	Користувачі отримують MVP (базову мапу) вже за місяць, не чекаючи повної версії.
Реакція на кризи	Потребує перегляду всього плану та бюджету.	Адаптація на рівні наступного спринту.	Швидка реакція на зміну форматів даних від ДТЕК.
Ризики	Висока ймовірність застарівання продукту до релізу.	Раннє виявлення дефектів та ризиків.	Мінімізація ризику дезінформації через постійне уточнення алгоритмів валідації.
Документація	Надлишкова, є пріоритетом.	Мінімально необхідна (SSoT).	Більше часу витрачається на реальну стійкість системи, а не на формальні звіти.

Отже, вибір Scrum для цього проєкту зумовлений його зручністю та практичною необхідністю. Головна перевага такого підходу – це можливість постійно тримати зв'язок із користувачем та враховувати його реальні запити [23]. Короткі цикли розробки дозволяють вже на перших етапах перевірити, як ГІС-модулі поведуться при слабкому сигналі зв'язку чи обмеженому доступі до мережі [48]. Це гарантує, що веб-сервіс не втратить актуальності, навіть при зміні зовнішніх факторів, а команда розробників зможе швидко переключати зусилля на найбільш пріоритетні завдання для інформування користувачів

3.2 Визначення функціональних та нефункціональних вимог до продукту проєкту

Процес визначення вимог до програмного забезпечення веб-сервісу моніторингу відключень є базовим етапом управління продуктом, що дозволяє формалізувати технічні параметри системи та встановити критерії успішності реалізації проєктних завдань [27]. Специфікація вимог виступає основним інструментом комунікації між стейкхолдерами та командою розробки, забезпечуючи трасованість цілей проєкту до конкретних функціональних модулів. У межах дослідження вимоги розподілено на функціональні, які визначають алгоритмічну поведінку системи, та нефункціональні, що регламентують якісні характеристики функціонування сервісу.

Функціональні вимоги до веб-сервісу охоплюють повний цикл обробки даних: від автоматизованої агрегації офіційної інформації через API енергопостачальників до інтеграції краудсорсингових повідомлень від споживачів. Пріоритетним завданням є забезпечення візуальної репрезентації зон відключень на інтерактивній мапі та надання інструментарію для пошуку пунктів незламності. Система повинна реалізовувати логіку верифікації вхідних потоків даних, персоналізацію сповіщень та аналітичну обробку історичної статистики для прогнозування стабільності енергопостачання. Повний перелік функціональних вимог наведено в таблиці 3.2.

Таблиця 3.2.

Функціональні вимоги до продукту

Ідентифікатор вимоги	Назва вимоги	Опис вимоги
1	2	3
FR-01	Відображення інтерактивної мапи відключень	Система повинна надавати інтерактивну мапу з візуалізацією зон відключень електроенергії у реальному часі, включаючи заплановані та аварійні відключення.

1	2	3
FR-02	Інтеграція з API енергопостачальників	Система повинна автоматично отримувати дані про відключення через API офіційних джерел (ДТЕК, Київенерго тощо).
FR-03	Краудсорсинг даних від користувачів	Система повинна дозволяти користувачам надсилати повідомлення про відключення світла.
FR-04	Відображення пунктів незламності	Система повинна показувати локації пунктів незламності з інформацією про доступні послуги, робочий час та завантаженість.
FR-05	Персоналізовані сповіщення	Система повинна надсилати сповіщення (email, SMS, push-повідомлення) про відключення в залежності від геолокації користувача та його налаштувань.
FR-06	Фільтрація даних	Користувачі повинні мати можливість фільтрувати інформацію за типом відключення, датою, регіоном та іншими параметрами.
FR-07	Аналітика та прогнозування	Система повинна аналізувати історичні дані для виявлення тенденцій та прогнозування можливих відключень.
FR-08	Генерація звітів	Система повинна надавати можливість формування звітів у форматах PDF, Excel для органів влади та енергокомпаній.
FR-09	Верифікація даних	Система повинна автоматично перевіряти достовірність краудсорсингових повідомлень та даних з API.
FR-10	Офлайн-доступ	Система повинна забезпечувати обмежений функціонал (перегляд кешованих даних) при відсутності інтернет-з'єднання.

1	2	3
FR-11	Аутентифікація та ролі	Система повинна підтримувати різні ролі користувачів (звичайні користувачі, адміністратори, представники енергокомпаній).
FR-12	Журналювання дій	Система повинна вести лог дій користувачів та змін даних для аудиту та аналізу.

Нефункціональні вимоги встановлюють якісні межі та експлуатаційні обмеження, критичні для стійкості сервісу в умовах пікових навантажень. До них належать показники продуктивності обробки геопросторових запитів, масштабованості архітектури, параметри кібербезпеки та рівень доступності інтерфейсу для різних категорій користувачів. Дотримання цих вимог гарантує надійність системи як інструменту інформування в умовах кризи. Нefункціональні вимоги систематизовано в таблиці 3.3.

Таблиця 3.3.

Нефункціональні вимоги до продукту

Ідентифікатор вимоги	Назва вимоги	Опис вимоги
1	2	3
NFR-01	Продуктивність	Система повинна забезпечувати швидке завантаження мапи та даних навіть при високому навантаженні.
NFR-02	Масштабованість	Система повинна підтримувати зростання кількості користувачів та обсягів даних без втрати продуктивності.
NFR-03	Безпека	Система повинна захищати дані користувачів, запобігати DDoS-атакам та несанкціонованому доступу.
NFR-04	Доступність	Інтерфейс повинен бути інтуїтивно зрозумілим, адаптованим для мобільних пристроїв та доступним для людей з обмеженими можливостями

1	2	3
NFR-05	Надійність	Система повинна забезпечувати мінімальний час простою (до 99,9% uptime) навіть у разі аварійних відключень електроенергії.
NFR-06	Сумісність	Система повинна коректно працювати на різних пристроях (ПК, смартфони, планшети) та в різних браузерах.

Сформований перелік вимог утворює цілісну специфікацію, яка визначає технічний вигляд та функціональні можливості веб-сервісу. Систематизація цих параметрів дозволяє чітко розмежувати обов'язки системи та встановити об'єктивні показники для контролю якості розробки. Завдяки деталізації кожної функції та якісного атрибута забезпечується технологічна відповідність сервісу запитам користувачів, що є необхідною умовою для створення надійного інструменту моніторингу в умовах енергетичної кризи.

3.3 Формування беклогу продукту проєкту

Формування беклогу продукту є фундаментальним етапом переходу від стратегічного планування до операційної реалізації веб-сервісу. У межах обраного гнучкого фреймворку Scrum, беклог виступає впорядкованим та динамічним переліком вимог [60], що постійно актуалізується відповідно до зворотного зв'язку від зацікавлених сторін та технічних викликів розробки [9].

Центральним елементом структурування вимог у даному проєкті є користувацькі історії. На відміну від традиційних функціональних специфікацій, цей формат фокусується на цінності функції для конкретної ролі користувача, що дозволяє уникнути надмірної технічної деталізації на ранніх етапах. Опис кожної історії базується на стандартизованій конструкції, яка включає визначення ролі суб'єкта, опис необхідної дії та кінцеву мету, що несе бізнес-цінність для проєкту [21].

Для забезпечення якості сформованого беклогу було використано критерії INVEST, що є стандартом в управлінні IT-проектами:

- Незалежність – кожна функція, як-от фільтрація чи пошук, може розроблятися відносно автономно;
- Можливість обговорення – деталі реалізації уточнюються під час планування ітерацій;
- Цінність – кожна історія несе пряму користь для громадянина або адміністратора;
- Оцінюваність – обсяг робіт підлягає визначенню командою;
- Компактність – завдання достатньо малі для виконання в межах одного циклу;
- Тестованість – наявність чітких критеріїв перевірки результату.

Важливим аспектом є верифікація готовності кожної історії через критерії прийняття [43]. У роботі застосовано підхід поведінкової розробки сценаріїв із використанням синтаксису Gherkin. Це дозволяє трансформувати абстрактні вимоги у чіткі сценарії взаємодії, що значно спрощує контроль якості та гарантує досягнення встановлених критеріїв готовності інкременту продукту.

Пріоритезація елементів беклогу здійснена за методом MoSCoW, де виділено наступні категорії:

- Must Have – критичні функції, що складають ядро системи, без яких запуск продукту неможливий.
- Should Have – важливі функції, реалізація яких передбачена одразу після стабілізації основних модулів.
- Could Have – додаткові сервіси, що покращують зручність користування, але можуть бути відкладені у разі дефіциту ресурсів.

Оцінка складності реалізації проведена у відносних одиницях складності – Story Points. Цей метод дозволяє врахувати не лише часові витрати, а й технічну складність, рівень невизначеності та ймовірні ризики інтеграції із зовнішніми джерелами даних.

Сформований беклог продукту проєкту частково представлено у таблиці 3.4. Повний перелік наведено у Додатку Б.

Таблиця 3.4.

Беклог користувацьких історій продукту проєкту

Код US	Назва та формулювання історії	Критерії прийняття (Acceptance Criteria)	Пріоритет	Story Points
1	2	3	4	5
US01	Візуалізація зон відключень. Як користувач, я хочу бачити інтерактивну мапу, щоб швидко оцінити стан енергомережі у своєму регіоні.	Scenario: Відображення шарів. Given: Отримано дані про мережу. When: Користувач відкриває мапу. Then: Аварійні зони позначені червоним, планові – жовтим.	Must	21
US02	Аналітичний модуль. Як представник влади, я хочу переглядати історичні дані, щоб оцінювати стійкість інфраструктури.	Scenario: Експорт статистики. Given: Обрано часовий інтервал. When: Натиснуто кнопку експорту. Then: Система генерує звіт із динамікою відключень.	Should	8
US03	Інтеграція з API постачальників. Як розробник, я хочу автоматизувати збір даних, щоб інформація на мапі була завжди актуальною.	Scenario: Автоматичне оновлення. Given: API енергокомпанії доступне. When: Спрацьовує системний тригер. Then: Дані у базі оновлюються без втручання людини.	Must	13
US04	Керування відображенням. Як користувач, я хочу фільтрувати типи відключень, щоб бачити лише релевантну мені ситуацію.	Scenario: Фільтрація типів. Given: Мапа завантажена. When: Обрано фільтр «Аварійні». Then: Система приховує всі планові графіки.	Must	5

1	2	3	4	5
US05	Географічний пошук. Як мешканець, я хочу знайти статус об'єкта за адресою, щоб знати, чи є там світло без навігації по мапі.	Scenario: Пошук адреси. Given: Поле пошуку активне. When: Введено назву вулиці. Then: Мапа фокусується на об'єкті та виводить його статус.	Must	13

Запропонована структура беклогу дозволяє забезпечити повну трасованість вимог від абстрактної ідеї до програмного коду. Завдяки використанню чітких критеріїв прийняття, команда отримує єдине джерело достовірної інформації, що мінімізує помилки на етапі розробки інтерфейсів та алгоритмів.

3.4 Організаційна структура управління проєктом

Реалізація веб-сервісу моніторингу в умовах високої невизначеності вимагає переходу від жорсткої ієрархії до самоорганізованої командної структури [58]. У межах Scrum-фреймворку управління проєктом децентралізоване, а функції традиційного менеджера проєкту розподілені між декількома напрямками відповідальності для досягнення максимальної адаптивності [55].

Керівництво стратегічним розвитком та бізнес-цінністю продукту здійснює Власник продукту. Дана роль несе одноосібну відповідальність за пріоритезацію беклогу, взаємодію із зовнішніми стейкхолдерами та управління бюджетом проєкту [72].

Фасилітацію процесів розробки та усунення організаційних перешкод забезпечує Скрам-майстер. Він виконує функції сервісного лідера, забезпечуючи дотримання методології, проводячи регулярні наради та захищаючи команду від зовнішнього тиску, що є критичним у кризових умовах енергосистеми.

Технічна реалізація продукту покладається на Команду розробки, яка є крос-функціональною та автономною у прийнятті інженерних рішень. Організаційна структура команди обумовлена технологічною складністю ГІС-проєкту та необхідністю паралельної підтримки різних модулів системи. Склад та спеціалізація команди проєкту представлені в таблиці 3.5.

Таблиця 3.5.

Склад та спеціалізація команди проєкту

Спеціалізація (роль)	Кількість	Основна зона відповідальності у проєкті
1	2	3
Власник продукту	1	Визначення стратегії та пріоритетів функціоналу.
Скрам-майстер	1	Координація процесів та усунення блокувальників.
Backend-розробник	2	Розробка алгоритмів агрегації API та обробки даних.
Frontend-розробник	1	Реалізація картографічного інтерфейсу та пошуку.
UI/UX Дизайнер	1	Проектування зручності ГІС-шарів та форм збору даних.
QA-інженер	1	Тестування верифікації даних та навантаження системи.

Такий розподіл обов'язків мінімізує управлінську бюрократію та підвищує стійкість проєкту до змін у зовнішньому середовищі.

3.5 Розробка WBS проєкту

Для забезпечення ефективного управління обсягом робіт та досягнення повної відповідності результатів поставки цілям проєкту розроблено ієрархічну структуру робіт «Work Breakdown Structure». Декомпозиція обсягу проєкту дозволяє трансформувати стратегічні завдання з моніторингу енергосистеми у конкретні пакети робіт, що підлягають оцінці та контролю [15, 45]. Враховуючи складність інформаційної системи та динамічність середовища, у роботі застосовано комплексний підхід до побудови WBS, який включає три моделі

структурування: за функціональними процесами, за фазами життєвого циклу та за продуктивними модулями [63]. Таке поєднання забезпечує всебічне охоплення як управлінських активностей, так і технічних етапів реалізації програмного забезпечення [6].

Перший рівень декомпозиції за функціональними процесами відображений на рисунку 3.1 та фокусується на методах виконання робіт.

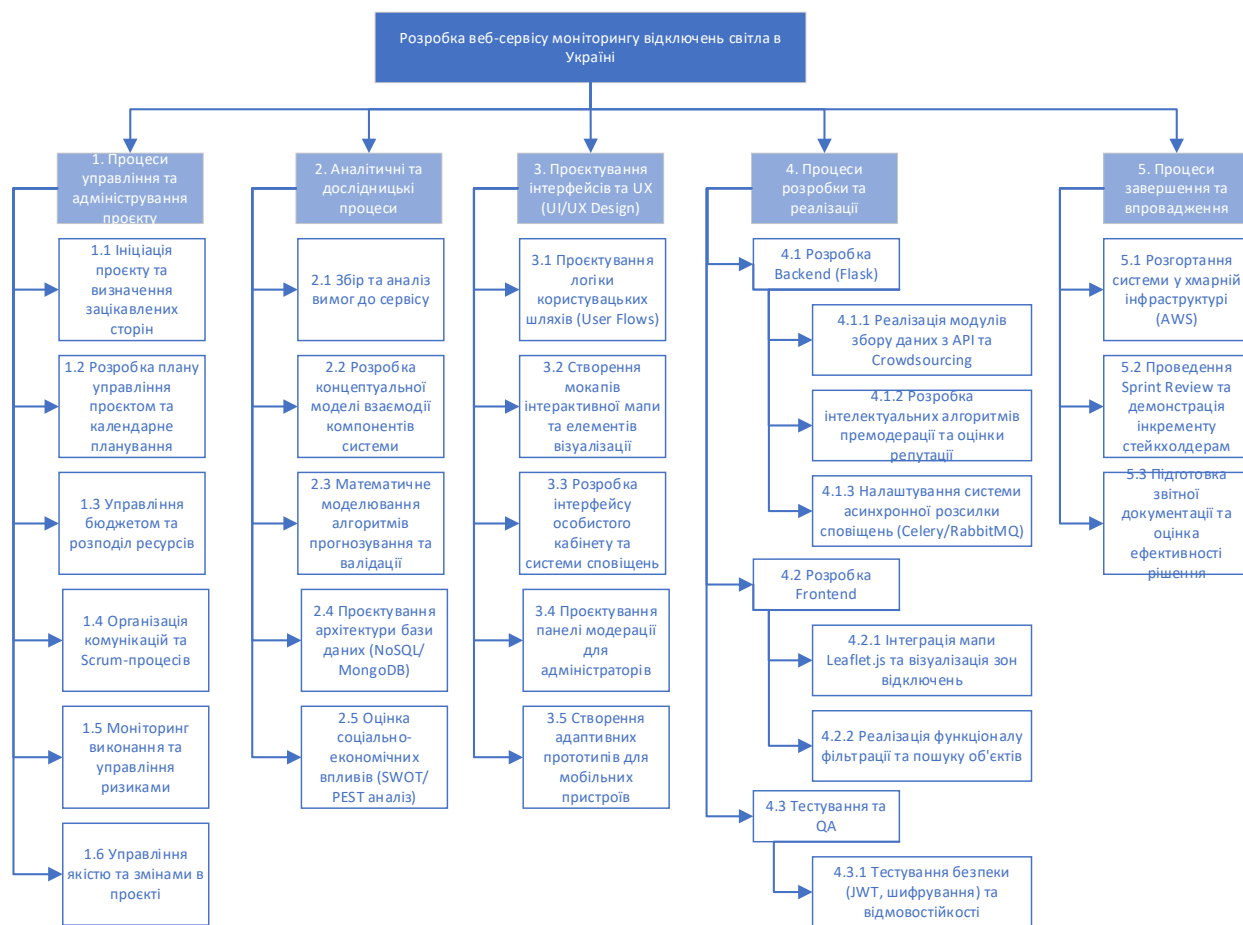


Рис. 3.1. WBS проекту за функціональними процесами

Процеси управління та адміністрування проекту (блок 1.1) охоплюють ініціацію проекту, календарне планування робіт, розробку бюджету та моніторинг ризиків, що забезпечує методологічну цілісність Scrum-команди та контроль за ресурсами. Аналітичний блок включає дослідження предметної області та математичне моделювання алгоритмів верифікації даних. У межах технічної реалізації виділено окремі напрями для розробки Backend-частини на

базі Flask та Frontend-складової з використанням бібліотеки Leaflet.js. Блок тестування та впровадження охоплює перевірку безпеки через JWT-аутентифікацію, тестування відмовостійкості системи, а також фінальне розгортання в хмарній інфраструктурі AWS із запуском та презентацією результатів.

Другий підхід до структурування обсягу робіт базується на фазах життєвого циклу проєкту (Рис. 3.2). Ця модель встановлює логічну послідовність етапів від ініціації до завершення.

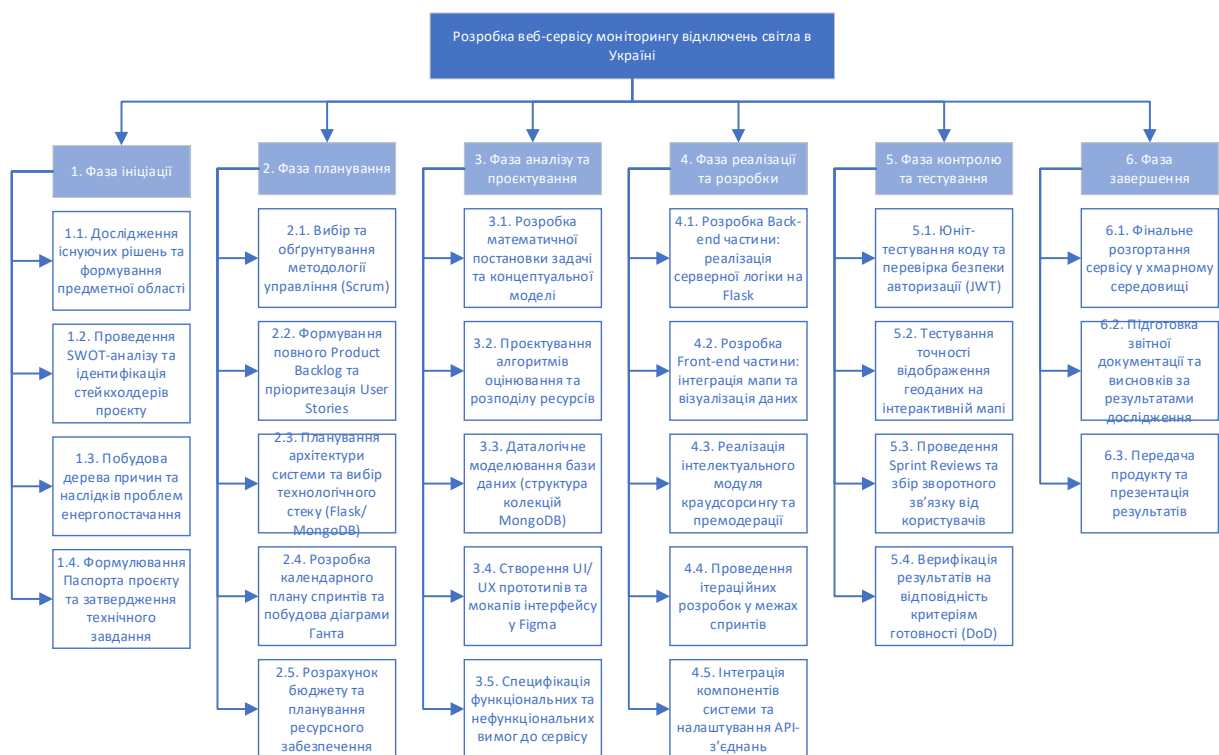


Рис. 3.2. WBS проєкту за фазами життєвого циклу

Фаза ініціації спрямована на SWOT-аналіз та ідентифікацію зацікавлених сторін, тоді як фаза планування зосереджена на архітектурному проєктуванні та виборі технологічного стеку MongoDB. Етап аналізу та дизайну включає математичну постановку задачі та розробку UI/UX прототипів у середовищі Figma. Активна фаза реалізації охоплює ітераційну розробку модулів, включаючи систему краудсорсингу та інтеграцію API-з'єднань. Фаза контролю забезпечує верифікацію результатів на відповідність критеріям готовності (DoD)

та юніт-тестування коду. Завершальна фаза передбачає підготовку звітної документації та передачу готового продукту в експлуатацію.

Декомпозиція за продуктами (Рис. 3.3), визначає архітектурний склад веб-сервісу через систему незалежних модулів.

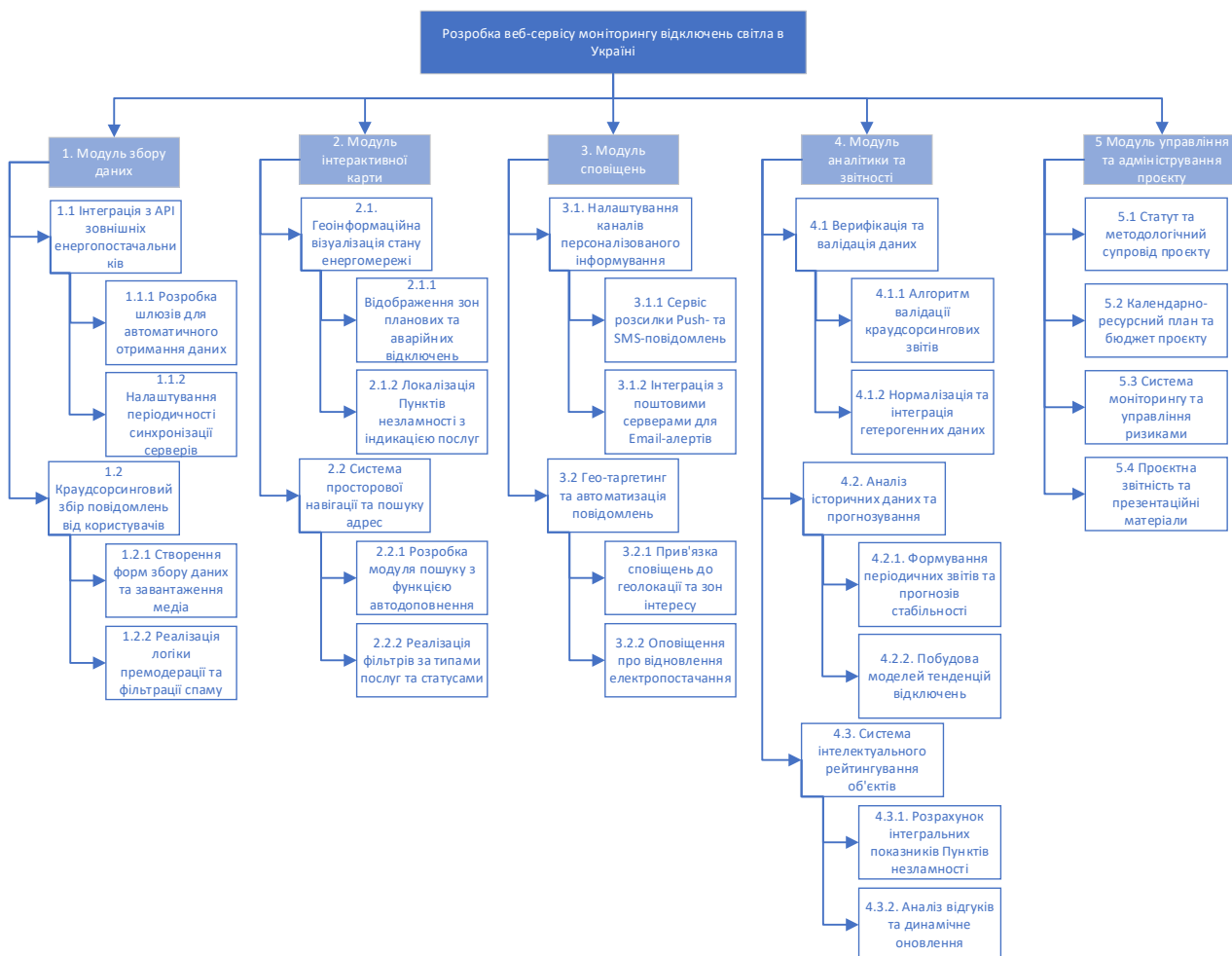


Рис. 3.3. WBS проєкту за продуктивими модулями

Модуль збору даних поєднує автоматичне отримання офіційної інформації та збір повідомлень від мешканців через форми краудсорсингу. Блок інтерактивної карти відповідає за показ стану мережі, розміщення пунктів незламності та зручний пошук адрес. Система сповіщень дозволяє налаштовувати адресне інформування користувачів про поточну ситуацію зі світлом. Аналітичний модуль забезпечує верифікацію та валідацію краудсорсингових звітів, аналіз зібраної статистики та роботу системи рейтингів

об'єктів. Модуль управління проєкту містить плани робіт, систему контролю ризиків та комплект необхідної документації. Такий підхід гарантує модульність системи та дозволяє незалежно розвивати окремі функціональні елементи сервісу.

Підсумовуючи результати декомпозиції, слід зазначити, що розроблені ієрархічні структури є основою для подальшого формування беклогу продукту та розподілу ролей у команді. Процесна модель оптимізує внутрішні взаємодії розробників, фазова – забезпечує часовий контроль етапів, а продуктова – гарантує повноту функціонального наповнення сервісу. Інтеграція цих підходів дозволяє мінімізувати ризики розширення меж проєкту та забезпечує високу якість інженерних рішень при створенні інструменту моніторингу енергопостачання в Україні. Розроблена WBS повністю охоплює обсяг робіт, визначений у паспорті проєкту, та створює необхідні умови для успішного впровадження системи.

3.6 Планування ресурсного забезпечення проєкту та бюджету

Ефективна реалізація проєкту потребує чіткого визначення та розподілу ресурсів, що забезпечують безперервність процесів розробки та стійкість інфраструктури. Оцінка вартості проєкту базується на методі «знизу-вгору» Bottom-Up Estimation, що передбачає підсумовування витрат на рівні окремих пакетів робіт, визначених у WBS [45]. Ресурсне забезпечення проєкту поділяється на людські та технічні (інфраструктурні та програмні) ресурси [15, 25].

Для систематизації ресурсів та запобігання конфліктам при їх призначенні в роботі розроблено ієрархічну структуру ресурсів Resource Breakdown Structure, яка представлена у таблиці 3.6. На відміну від неструктурованого переліку персоналу, RBS дозволяє згрупувати всі компоненти проєкту (людей, хмарні потужності, API-сервіси) за функціональними категоріями, що забезпечує точність подальшого фінансового планування.

Ієрархічна структура ресурсів проєкту

Категорія (Level 1)	Група (Level 2)	Конкретний ресурс (Level 3)	Роль у забезпеченні результату
1	2	3	4
Людські ресурси	Команда розробки	Backend (Python), Frontend (React)	Реалізація логіки обробки даних та ГІС-інтерфейсу
	Управління та QA	Scrum Master, QA Engineer	Координація ітерацій та верифікація даних
Технічні ресурси	Обчислювальні потужності	AWS EC2, S3 Storage	Забезпечення хостингу та зберігання геоданих
	Бази даних	MongoDB Atlas	Обробка неструктурованих масивів даних
Програмні сервіси	Зовнішні API	Google Maps API, SMS Gateways	Картографія та доставка сповіщень

Слід зауважити, що всі фінансові розрахунки в межах даного розділу та подальшого планування в MS Project проведено у доларах США (USD). Такий вибір обумовлений специфікою ІТ-галузі України, де медіанні значення заробітних плат фахівців традиційно фіксуються у валютному еквіваленті для забезпечення стабільності планування, а оплата основної хмарної інфраструктури AWS здійснюється за глобальними тарифами у USD.

При формуванні команди враховано крос-функціональний характер Scrum-групи, що дозволяє автономно виконувати повний цикл розробки MVP [23]. Для забезпечення фінансової стійкості та раціонального використання робочого часу впроваджено систему планування за показником еквівалента повної зайнятості FTE. Використання FTE-моделі дозволяє точно синхронізувати залученість фахівців із реальними трудовитратами, визначеними у беклозі продукту: ролі з інтенсивним навантаженням на етапі активного кодування залучені повністю, тоді як для стратегічних та дизайнерських позицій

встановлено частковий рівень присутності [58]. Розрахунок бюджету на оплату праці базується на комплексному аналізі медіанних ставок ІТ-ринку України на 2024–2026 роки для фахівців рівня Middle, що гарантує конкурентність пропозиції та залучення кваліфікованих кадрів. Деталізований штатний розпис із зазначенням рівнів залученості та щомісячного фонду оплати праці представлено в таблиці 3.7.

Таблиця 3.7.

Планування людських ресурсів та витрат на оплату праці

Роль у проєкті	К-сть	Рівень кваліфікації	Залученість (FTE)	Місячна ставка (1.0 FTE), USD	Всього за місяць, USD
1	2	3	4	5	6
Власник продукту	1	Middle	0.25	2500	625
Скрам-майстер	1	Middle	1.00	2400	2400
Backend-розробник 1	1	Middle	1.00	2500	2500
Backend-розробник 2	1	Middle	0.75	2500	1875
Frontend-розробник	1	Middle	1.00	2500	2500
UI/UX Дизайнер	1	Middle	0.50	1800	900
QA-інженер	1	Middle	0.75	1600	1200
Разом	7	–	5.25	–	12000

Таблиця 3.8.

Планування технічних ресурсів та інфраструктурних витрат

Категорія ресурсу	Опис / Технологія	Періодичність	Орієнтовна вартість, USD
1	2	3	4
Обчислювальні потужності	AWS EC2 (t4g.micro, 24/7)	Місячно	6.05
Сховище даних	AWS S3 (5 ГБ) + MongoDB	Місячно	8.00

1	2	3	4
Мережеві сервіси	Route 53 (Домен + DNS)	Місячно	0.50
Інструменти дизайну	Figma (Professional plan)	Місячно	15.00
Разом	—	—	29.55

Технічні ресурси включають хмарну інфраструктуру і програмні інструменти та наведені в таблиці 3.8. Основним середовищем розгортання обрано AWS.

Управління вартістю даного проєкту складається з чотирьох основних процесів: планування витрат, оцінювання, розробка бюджету та контроль виконання фінансових показників. Для забезпечення фінансової прозорості календарний план інтегровано з аркушем ресурсів у системі Microsoft Project (Рис. 3.4). Використання пропорційного методу нарахування витрат дозволяє системі автоматично розраховувати бюджет на основі фактичної завантаженості персоналу.

	Ім'я ресурсу	Тип	Одиниця вимірювані матеріалів	Ініціали	Група	Макс. одиниць	Звич. ставка	Понад. ставка	Витрати/г	Нарахування	Основний календар
1	Власник продукту	Робота		PO	Management	100%	\$15,63/г	\$0,00/г	\$0,00	Пропорційне	Стандартний
2	Скрам-майстер	Робота		SM	Management	100%	\$15,00/г	\$0,00/г	\$0,00	Пропорційне	Стандартний
3	Backend-розробник 1	Робота		BE1	Development	100%	\$15,63/г	\$0,00/г	\$0,00	Пропорційне	Стандартний
4	Backend-розробник 2	Робота		BE2	Development	100%	\$15,63/г	\$0,00/г	\$0,00	Пропорційне	Стандартний
5	Frontend-розробник	Робота		FE	Development	100%	\$15,63/г	\$0,00/г	\$0,00	Пропорційне	Стандартний
6	UI/UX Дизайнер	Робота		UX	Design	100%	\$11,25/г	\$0,00/г	\$0,00	Пропорційне	Стандартний
7	QA-інженер	Робота		QA	Quality	100%	\$10,00/г	\$0,00/г	\$0,00	Пропорційне	Стандартний
8	Хостинг AWS	Витрати		AWS	Infrastructure					Завершення	
9	База даних MongoDB	Витрати		MDB	Infrastructure					Завершення	

Рис. 3.4. Фрагмент аркушу ресурсів проєкту із зазначенням ставок та типів витрат

Загальний бюджет проєкту розраховується на період активної розробки MVP, що триває з 01.04.2026 по 22.05.2026 (38 робочих днів або приблизно 1,81 місяця). Розрахунок підсумкової вартості або базової лінії витрат здійснюється за формулою (3.1):

$$B_{total} = (C_{hr} + C_{tech}) * T + R_m \quad (3.1)$$

де B_{total} – загальний бюджет проекту;

C_{hr} – щомісячні витрати на людські ресурси (оплата праці);

C_{tech} – щомісячні витрати на технічні ресурси (інфраструктура);

T – тривалість етапу розробки в місяцях;

R_m – управлінський резерв на ризики (Management Reserve), що становить 10% від прямих витрат.

Підставляючи отримані дані у формулу, отримуємо базову вартість проекту без урахування резерву:

$$B_{total} = (12000 + 29.55) * 1.81 = 21773.49 \text{ USD}$$

Розрахунок управлінського резерву:

$$R_m = 21773.49 * 0.1 = 2177.35 \text{ USD.}$$

Таким чином, підсумковий бюджет на реалізацію MVP-версії веб-сервісу моніторингу становить:

$$B_{total} = 21773.49 + 2177.35 = 23950.84 \text{ USD.}$$

Для підтвердження реалістичності розробленого плану та перевірки збалансованості завантаження команди проведено ресурсний аналіз засобами Microsoft Project. Шляхом побудови графіка ресурсів (Рис. 3.5) було перевірено інтенсивність праці команди протягом усього життєвого циклу розробки.

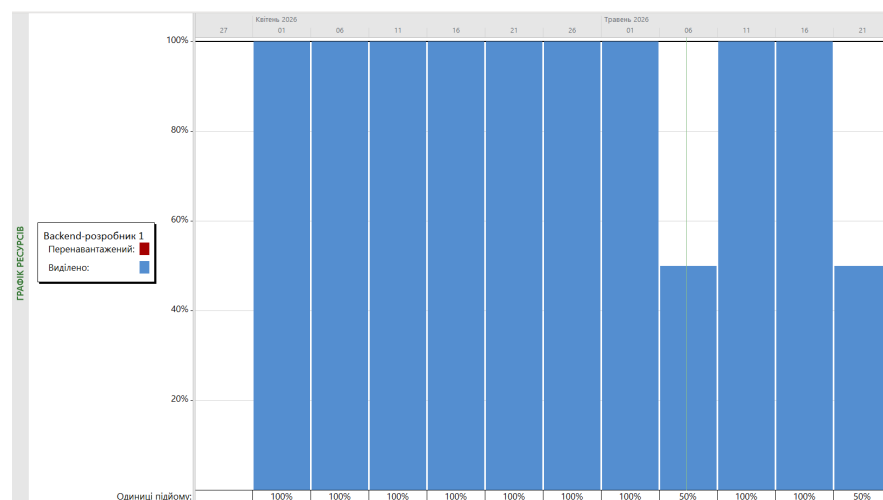


Рис. 3.5. Графік ресурсів розподілу завантаженості Backend розробника 1

Візуалізація завантаженості, на прикладі Backend розробника 1, демонструє відсутність перевантажень, оскільки обсяг призначених робіт не перевищує 100% доступного часу фахівця. Це свідчить про раціональний розподіл пакетів робіт WBS та дотримання норм трудового законодавства при плануванні ітерацій розробки.

Процес операційного контролю вартості під час розробки базується на методі освоєного обсягу «Earned Value Management». Цей метод дозволяє порівнювати планові показники з фактично виконаною роботою. Для автоматизації контролю в Microsoft Project сформовано таблицю освоєного обсягу (Рис. 3.6), яка відображає планову вартість (PV), вартість реально виконаних робіт (EV) та фактичні витрати (AC).

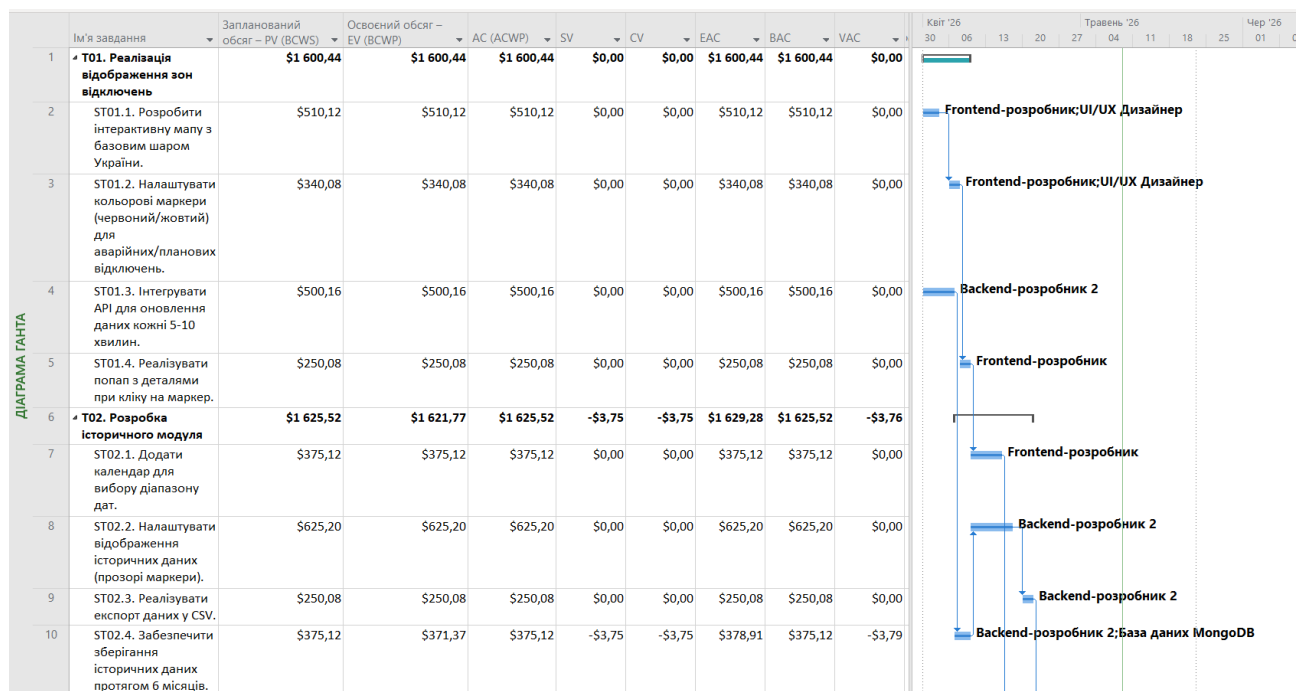


Рис. 3.6. Фрагмент розрахунку показників освоєного обсягу EVM

Аналіз даних у сформованій таблиці дозволяє ідентифікувати відхилення за графіком та вартістю в режимі реального часу. Використання прогностичних індикаторів, таких як EAC, забезпечує можливість вчасного залучення управлінського резерву для вирівнювання показників ефективності розробки.

Такий багаторівневий підхід до планування та контролю гарантує успішне завершення проєкту у встановлені терміни до 22 травня 2026 року.

Таким чином, підсумковий бюджет на реалізацію MVP-версії веб-сервісу моніторингу становить 23 950.84 USD. Даний обсяг фінансування забезпечує повне виконання робіт, визначених у ієрархічній структурі, та створює умови для успішного виведення продукту на ринок у встановлені терміни

3.7 Календарне планування та візуалізація графіка реалізації проєкту засобами Microsoft Project

Фінальним етапом проєктування інформаційного забезпечення є побудова календарного графіка, який інтегрує визначений обсяг робіт та ресурсну базу у єдину часову модель [37]. Для візуалізації та оперативного контролю розробки веб-сервісу моніторингу відключень було використано інструментарій Microsoft Project. Це дозволяє наочно представити послідовність виконання завдань, їхню тривалість та логічні взаємозв'язки на єдиній часовій шкалі.

Процес планування базується на декомпозиції раніше визначених користувацьких історій на конкретні технічні завдання та підзавдання. Частковий перелік завдань проєкту у розрізі користувацьких історій представлено у таблиці 3.9. Повний перелік наведено у Додатку В.

Таблиця 3.9.

Користувацькі історії та технічні завдання проєкту

Користувацька історія (User Story)	Завдання (Task)	Підзавдання (Sub-task)
1	2	3
US01: Візуалізація зон відключень	Т.01.1 Реалізація відображення зон відключень	ST01.1 Розробити інтерактивну мапу з базовим шаром України.
		ST01.2 Налаштувати кольорові маркери для аварійних/планових відключень.
		ST01.3 Інтегрувати API для оновлення даних кожні 5-10 хвилин.
		ST01.4 Реалізувати попап з деталями при кліку на маркер.

1	2	3
US02: Історичний модуль	Т.02.1 Розробка історичного модуля	ST02.1 Додати календар для вибору діапазону дат.
		ST02.2 Налаштувати відображення історичних даних (прозорі маркери).
		ST02.3 Реалізувати експорт даних у CSV.
		ST02.4 Забезпечити зберігання історичних даних протягом 6 місяців.
US03: Синхронізація з API	Т.03.1 Синхронізація з API енергопостачаль ників	ST03.1 Підключити API ДТЕК та «Укренерго».
		ST03.2 Налаштувати пріоритетність джерел при конфліктах даних.
		ST03.3 Реалізувати систему логування синхронізації.
		ST03.4 Додати сповіщення для адміністратора про збої.
US04: Реалізація фільтрів	Т.04.1 Впровадження системи фільтрації	ST04.1 Додати перемикач «Планові/Аварійні» у верхньому меню.
		ST04.2 Налаштувати динамічне оновлення мапи без перезавантаження.
		ST04.3 Збереження обраних фільтрів для авторизованих користувачів.
US05: Розробка пошуку	Т.05.1 Розробка модуля географічного пошуку	ST05.1 Реалізувати поле пошуку з автодоповненням адрес.
		ST05.2 Інтегрувати геокодувальний сервіс (Google Maps API).
		ST05.3 Налаштувати центрування мапи за введеною адресою.
		ST05.4 Додати спливаючі повідомлення про статус мережі на вулиці.

Згідно з методологією Scrum, розробку розбито на чотири спринти тривалістю по два тижні кожен [60]. Загальна тривалість робіт становить 36 робочих днів, а саме з 1 квітня по 22 травня 2026 року. Нижче в таблицях 3.10 – 3.13 наведено детальний розподіл завдань за кожною ітерацією із зазначенням виконавців та технологічного стеку.

Таблиця 3.10.

Розподіл задач за першим спринтом (01.04 – 14.04)

Код задачі	Назва роботи	Тривалість (год)	Виконавець	Технології
1	2	3	4	5
ST01.1	Розробка інтерактивної мапи	24	Frontend-розробник	React, Leaflet.js
ST01.2	Налаштування кольорових маркерів	16	UI/UX Дизайнер	Figma, CSS
ST01.3	Інтеграція API для оновлення даних	32	Backend-розробник 2	Python, Flask
ST03.1	Підключення API ДТЕК та «Укренерго»	40	Backend-розробник 1	REST API, JSON
ST03.2	Пріоритетність джерел даних	24	Backend-розробник 1	Python logic
ST10.1	Адміністрування та фасилітація	80	Скрам-майстер	MS Project

Таблиця 3.11.

Розподіл задач за другим спринтом (15.04 – 28.04)

Код задачі	Назва роботи	Тривалість (год)	Виконавець	Технології
1	2	3	4	5
ST01.4	Реалізація попапу з деталями	16	Frontend-розробник	React Components
ST02.1	Календар вибору діапазону дат	24	Frontend-розробник	Date-picker library
ST02.2	Відображення історичних даних	40	Backend-розробник 2	SQL Queries
ST02.4	Зберігання даних (6 місяців)	24	База даних MongoDB	MongoDB TTL
ST03.3	Система журналювання синхронізації	24	Backend-розробник 1	Python Logging
ST05.1	Поле пошуку з автодоповненням	24	Frontend-розробник	JS, Google Autocomplete
ST05.2	Інтеграція Google Maps API	32	Backend-розробник 1	Google Cloud API
ST01.4	Реалізація попапу з деталями	16	Frontend-розробник	React Components
ST02.1	Календар вибору діапазону дат	24	Frontend-розробник	Date-picker library

Таблиця 3.12.

Розподіл задач за третім спринтом (29.04 – 12.05)

Код задачі	Назва роботи	Тривалість (год)	Виконавець	Технології
1	2	3	4	5
ST04.2	Динамічне оновлення без перезавантаження	24	Frontend-розробник	AJAX / Axios
ST06.1	Чекбокси вибору послуг	16	UI/UX Дизайнер	Figma, HTML/CSS
ST06.2	Фільтрація пунктів у реальному часі	16	Frontend-розробник	React State
ST06.3	Логіка комбінації фільтрів	24	Backend-розробник 2	Python filtering
ST07.1	Розробка форми додавання пункту	32	Backend-розробник 1	Flask Forms
ST07.2	Завантаження фотографій об'єкта	24	База даних MongoDB	GridFS / S3 storage
ST08.2	Реалізація історії змін	24	Backend-розробник 2	Audit log table
ST04.2	Динамічне оновлення без перезавантаження	24	Frontend-розробник	AJAX / Axios
ST06.1	Чекбокси вибору послуг	16	UI/UX Дизайнер	Figma, HTML/CSS

Таблиця 3.13.

Розподіл задач за четвертим спринтом (13.05 – 22.05)

Код задачі	Назва роботи	Тривалість (год)	Виконавець	Технології
1	2	3	4	5
ST07.3	Модераційний процес для нових пунктів	24	Backend-розробник 1	Admin Dashboard
ST08.3	Оновлення мапи після редагування	16	Backend-розробник 2	WebSockets / Polling
ST09.1	Розділ відгуків у картці об'єкта	32	Frontend-розробник	React, CSS
ST09.2	Система оцінювання (1-5 зірок)	24	Backend-розробник 1	Python API
ST09.4	Відображення рейтингу на мапі	16	Frontend-розробник	Leaflet Markers

1	2	3	4	5
ST10.2	Фінальне тестування та QA	144	QA-інженер	Postman, PyTest
ST07.3	Модераційний процес для нових пунктів	24	Backend-розробник 1	Admin Dashboard
ST08.3	Оновлення мапи після редагування	16	Backend-розробник 2	WebSockets / Polling

На основі сформованих пакетів робіт у Microsoft Project побудовано візуальну модель проєкту, де основним інструментом планування виступає діаграма Ганта. На графіку відображено логічну декомпозицію десяти ключових завдань (Т.01 – Т.10), що відповідають основним функціональним модулям системи.

Початковий етап реалізації мапи (Т.01) та історичного модуля (Т.02) передбачає встановлення зв'язків типу «Фініш-Старт» між розробкою базового шару та налаштуванням кольорових маркерів (Рис. 3.7).

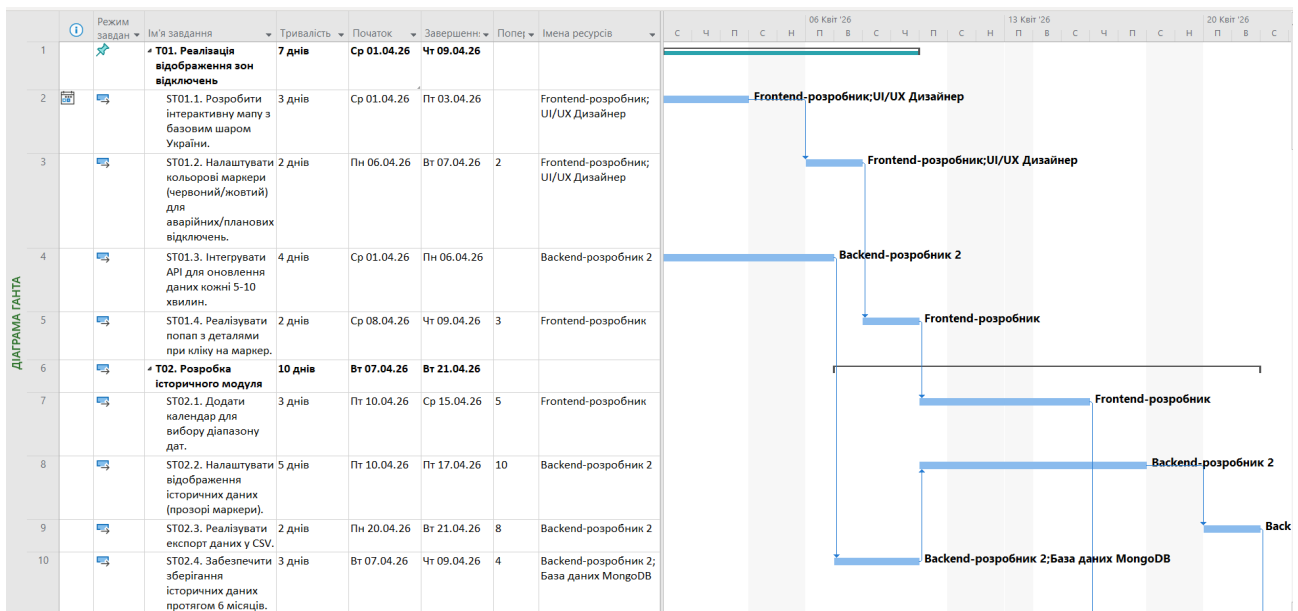


Рис. 3.7. Календарний план реалізації модуля відображення зон та історичних даних

Наступні фази розробки зосереджені на синхронізації з API енергопостачальників (Т.03) та створенні системи фільтрів (Т.04 – Т.05), що відображено на рисунках 3.8 та 3.9. Важливою особливістю планування є паралельне виконання інтеграції офіційних джерел та розробки інтерфейсних рішень, що забезпечує готовність системи до тестування у стислі терміни.

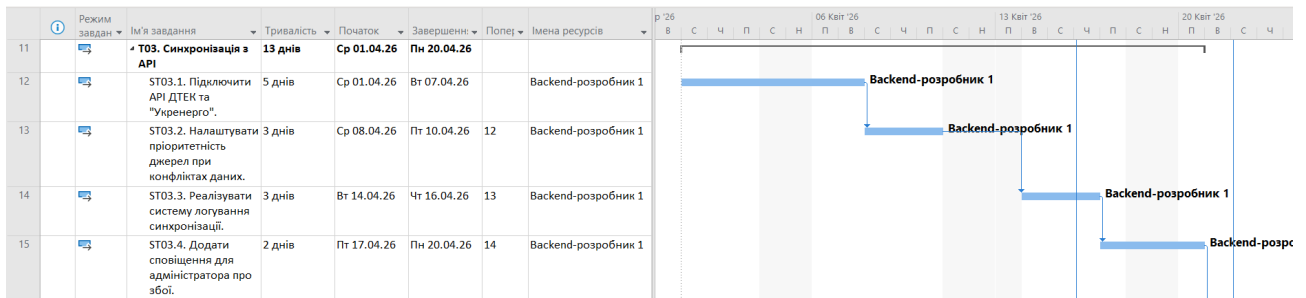


Рис. 3.8. Візуалізація графіка синхронізації з API

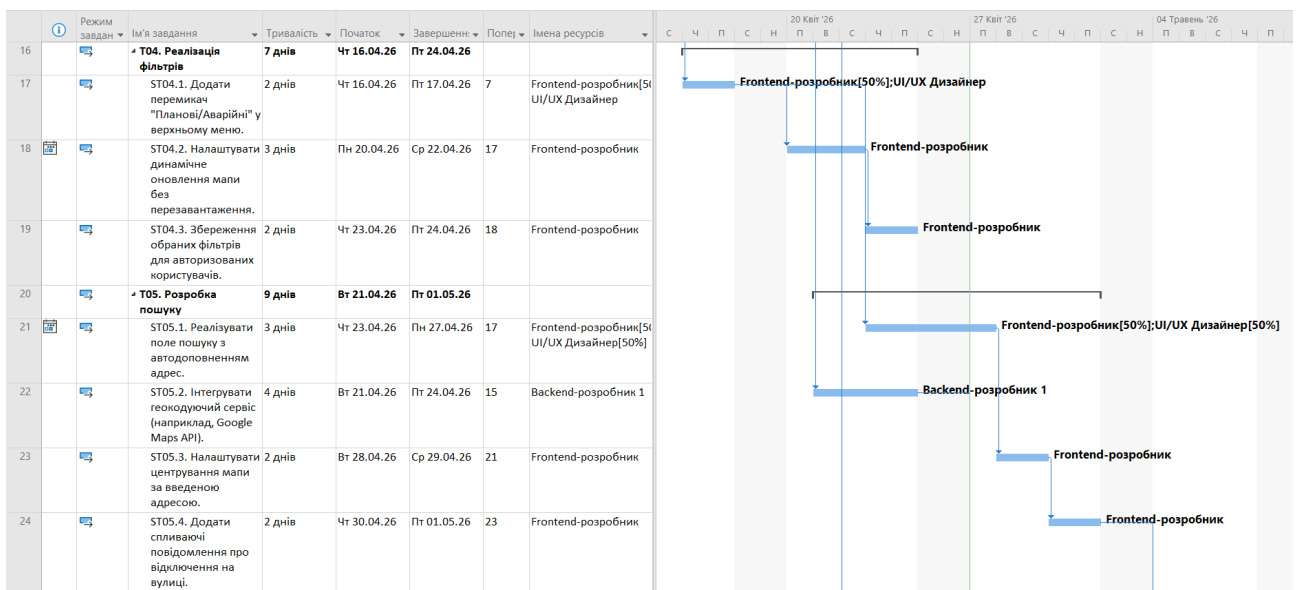


Рис. 3.9. Візуалізація графіка реалізації фільтрів та розробки системи пошуку

Етап інтеграції геосервісів та фільтрації за послугами (Т.06) у поєднанні з розробкою форм додавання волонтерського контенту (Т.07) припадає на кінець квітня та початок травня (Рис. 3.10). Використання Google Maps API потребує

координації між Backend та Frontend частинами команди, що відображено через логічні залежності між підзавданнями.

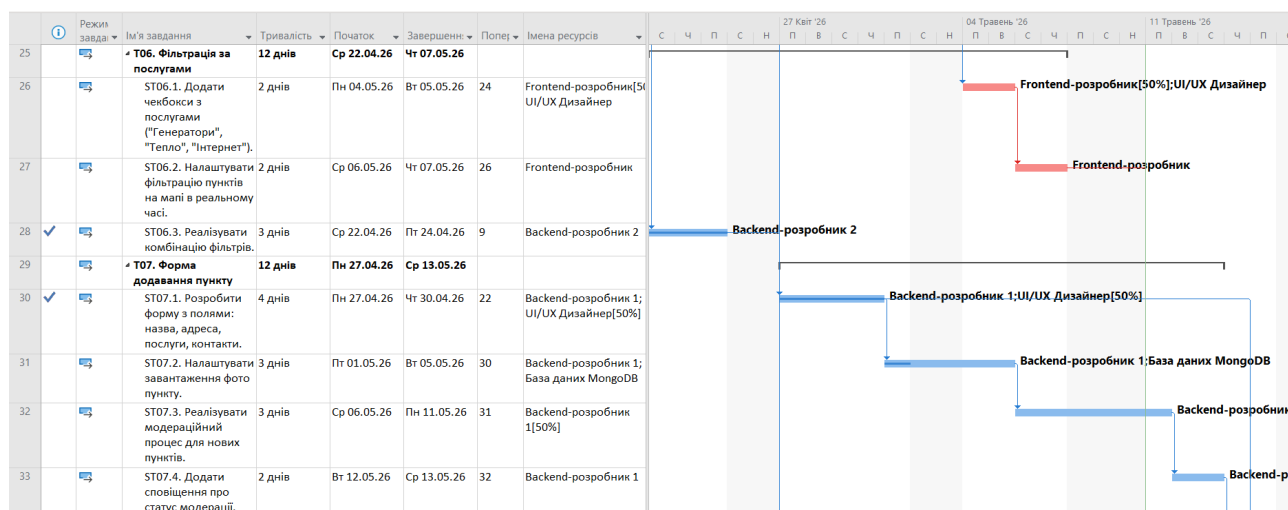


Рис. 3.10. Діаграма Ганта для модулів геокодування та форм волонтерського контенту

Завершальна стадія технічної реалізації включає інструменти редагування даних (Т.08) та систему відгуків (Т.09) зображено на рисунку 3.11. Окреме наскрізне завдання «Управління, якість та супровід» (Т.10) триває протягом всього життєвого циклу, забезпечуючи безперервне тестування кожної ітерації Scrum-майстром та QA-інженером (Рис. 3.12).

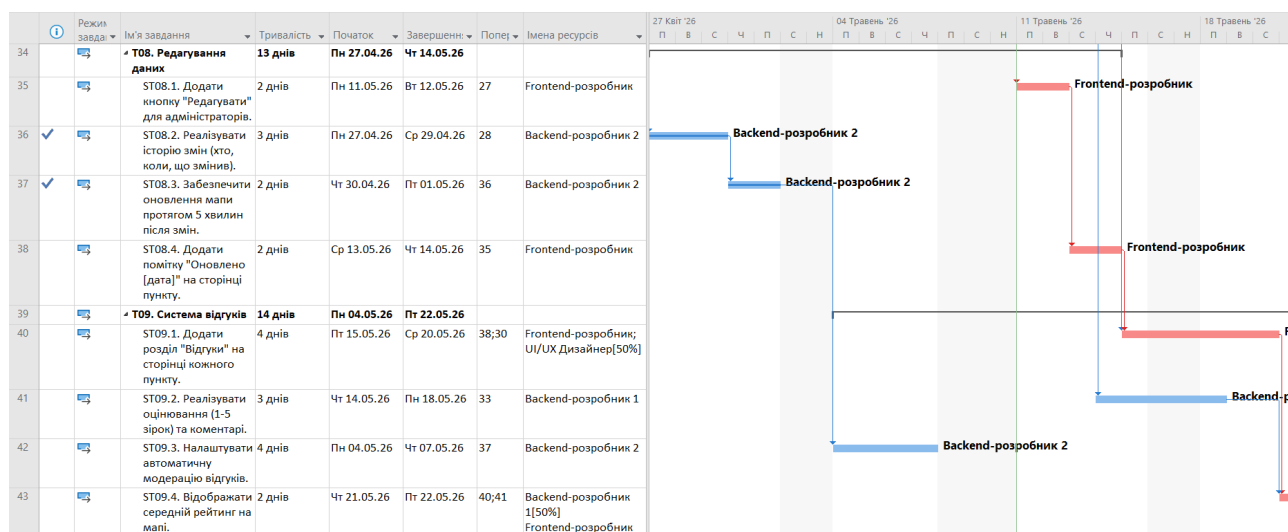


Рис. 3.11. Графік реалізації модулів редагування та модераторів даних

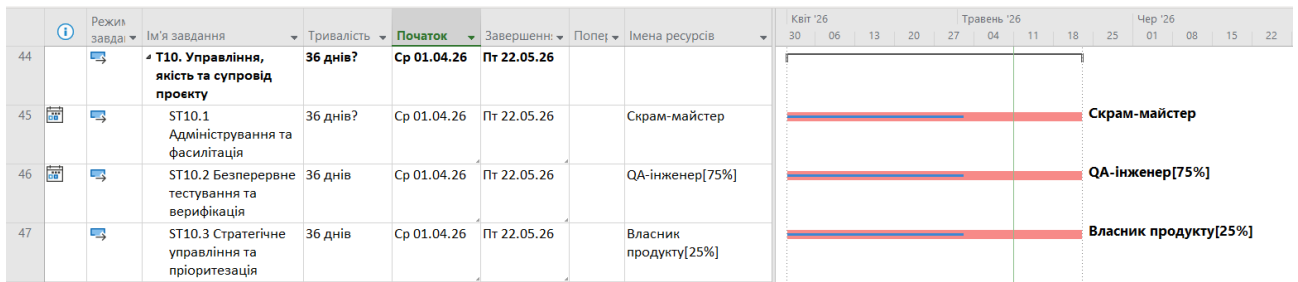


Рис. 3.12. Візуалізація фінальних етапів розробки та загальної тривалості проєкту

Для забезпечення фінансової прозорості календарний план інтегрований з «Аркушем ресурсів» (Рис. 3.13). У системі зафіксовано склад команди із визначенням стандартних ставок. Використання пропорційного методу нарахування витрат дозволяє MS Project автоматично розраховувати бюджет на основі фактичної завантаженості персоналу.

	Ім'я ресурсу	Тип	Одиниця вимірюваних матеріалів	Ініціали	Група	Макс. одиниць	Звич. ставка	Понад. ставка	Витрати/г	Нарахування	Основний календар
1	Власник продукту	Робота		PO	Management	100%	\$15,63/г	\$0,00/г	\$0,00	Пропорційне	Стандартний
2	Скрам-майстер	Робота		SM	Management	100%	\$15,00/г	\$0,00/г	\$0,00	Пропорційне	Стандартний
3	Backend-розробник 1	Робота		BE1	Development	100%	\$15,63/г	\$0,00/г	\$0,00	Пропорційне	Стандартний
4	Backend-розробник 2	Робота		BE2	Development	100%	\$15,63/г	\$0,00/г	\$0,00	Пропорційне	Стандартний
5	Frontend-розробник	Робота		FE	Development	100%	\$15,63/г	\$0,00/г	\$0,00	Пропорційне	Стандартний
6	UI/UX Дизайнер	Робота		UX	Design	100%	\$11,25/г	\$0,00/г	\$0,00	Пропорційне	Стандартний
7	QA-інженер	Робота		QA	Quality	100%	\$10,00/г	\$0,00/г	\$0,00	Пропорційне	Стандартний
8	Хостинг AWS	Витрати		AWS	Infrastructure					Завершення	
9	База даних MongoDB	Витрати		MDB	Infrastructure					Завершення	

Рис. 3.13. Фрагмент аркушу ресурсів проєкту із зазначенням ставок та типів витрат

Оптимізація командної взаємодії була проведена за допомогою інструменту «Планувальник роботи групи». На рисунку 3.14 візуалізовано розподіл завдань, що підтверджує відсутність перевантажень та високий рівень паралелізації процесів. Зокрема, на графіку чітко видно одночасну роботу двох Backend-розробників над різними функціональними блоками (API та БД), що суттєво скорочує критичний шлях проєкту.

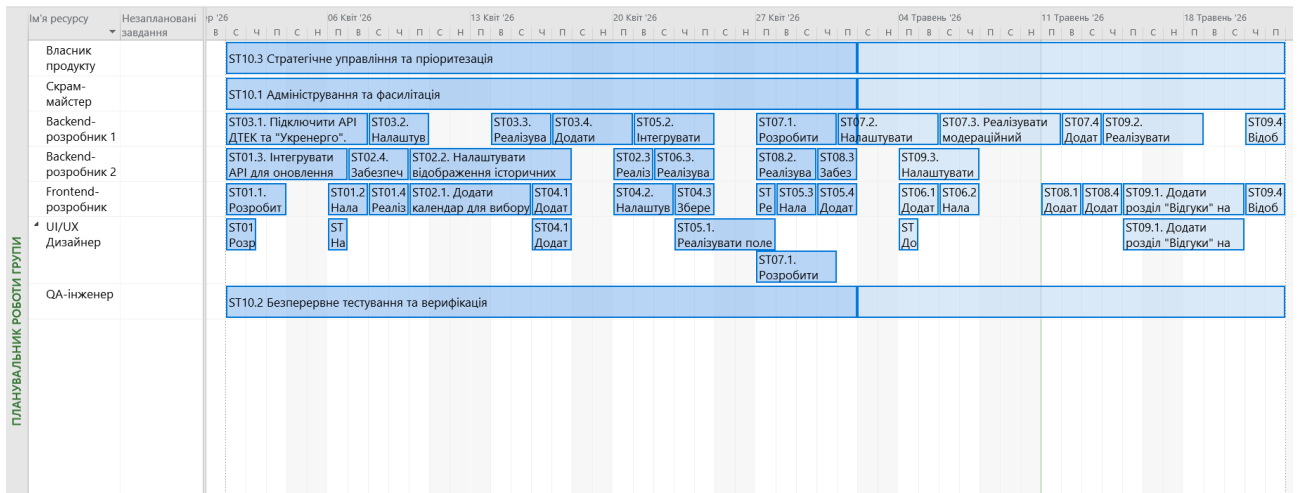


Рис. 3.14. Планувальник роботи команди: аналіз завантаженості та паралельних процесів

Отже, побудована в Microsoft Project модель календарного плану забезпечує синхронізацію крос-функціональної команди та гарантує запуск веб-сервісу моніторингу відключень у встановлені терміни, виступаючи надійним інструментом управління ризиками розробки.

3.8 Ідентифікація проєктних ризиків та розробка карти управління ними

Управління ризиками в межах проєкту розробки веб-сервісу моніторингу відключень світла базується на інтеграції методологій ISO 31000 та адаптивних принципів Scrum [55, 56], що дозволяє команді проактивно реагувати на загрози в умовах високої невизначеності енергетичного сектору України. Процес управління розпочинається з ідентифікації та класифікації загроз за їхньою природою походження [8, 45]. Для систематизації виявлених факторів їх розподілено на внутрішні, технічні, організаційні, фінансові та зовнішні ризики. Результати первинної ідентифікації із зазначенням сили впливу та рівня керованості представлено в таблиці 3.14.

Класифікація та характеристика ідентифікованих проєктних ризиків

№	Тип ризику	Ризикова подія	Сила впливу	Керованість
1	2	3	4	5
1	Внутрішні ризики проєктної команди	Висока плинність кадрів у команді через стрес та перевантаження	ВН	Середня
2		Недостатня експертиза у сфері енергетичних даних (формати, доступність)	ВН	Середня
3		Затримки з боку аналітика у підготовці специфікацій для інтеграцій	СВ	Висока
4		Низький рівень комунікації та обміну даними/інформацією всередині команди	СН	Середня
5	Технічні ризики	Затримки або помилки у даних від обленерго/операторів систем розподілу	ВВ	Низька
6		Відмова серверів при пікових навантаженнях (зимові відключення, тисячі користувачів одночасно)	ВН	Середня
7		Кібератаки (DDoS, SQL-ін'єкції, спроби зламу для спотворення даних)	ВВ	Середня
8		Втрата даних через відсутність резервного копіювання	ВН	Висока
9	Організаційні ризики	Відсутність офіційних угод з енергопостачальними компаніями на доступ до даних	ВВ	Низька
10		Непрозора відповідальність у разі надання користувачам некоректної інформації	ВН	Середня
11		Невчасне оновлення інформації призводить до втрати довіри користувачів	ВН	Середня
12		Затримка в ухваленні рішень щодо розвитку функціоналу з боку замовника	СВ	Середня

1	2	3	4	5
13	Зовнішні ризики / Форс-мажори	Затримка в ухваленні рішень щодо розвитку функціоналу з боку замовника	ВВ	Низька
14		Військові дії, які зупиняють роботу дата-центрів або інтернет-інфраструктури	ВВ	Низька
15		Припинення роботи сторонніх хмарних сервісів (AWS, Google Cloud, SMS-провайдерів)	ВН	Низька
16		Зміни в законодавстві щодо відкритих даних і персональної інформації	СВ	Низька
17	Фінансові ризики	Недостатнє фінансування проекту на етапі підтримки або масштабування	ВН	Середня
18		Зростання вартості хмарних сервісів або SMS-сповіщень, що підвищує витрати	СВ	Середня
19		Несвоєчасне отримання грантових коштів або фінансування від донорів	СВ	Низька
20		Перевищення бюджету через необхідність додаткових інтеграцій чи розширення функціоналу	ВН	Середня

Для проведення глибинного аналізу кожен ризикову подію було оцінено за якісними та кількісними показниками. Кількісна оцінка проводилася за 10-бальною шкалою за чотирма критеріями: затримки у часі, фінансові втрати, ймовірність виникнення та частота. На основі цих даних розраховано інтегральний показник важливості ризику, що дозволяє визначити пріоритетність заходів реагування. Результати комплексного оцінювання представлені частково в таблиці 3.15. Повний перелік результату комплексного оцінювання описано в таблиці Г.1 Додатка Г.

Результати оцінювання проєктних ризиків

№	Ризикова подія	Затримки у часі		Фінансові втрати		Ймовірність		Частота		Важливість ризику (компл. показн.)
		Якіс. оц.	Кільк. оц.	Якіс. оц.	Кільк. оц.	Якіс. оц.	Кільк. оц.	Якіс. оц.	Кільк. оц.	
1	2	3	4	5	6	7	8	9	10	11
1	Висока плинність кадрів у команді через стрес та перевантаження	ВН	7	ВН	7	СВ	6	СН	4	42
2	Недостатня експертиза у сфері енергетичних даних	ВВ	9	ВН	7	ВН	7	СН	4	63
3	Затримки з боку аналітика у підготовці специфікацій	СВ	6	СВ	6	СВ	6	СН	4	36
4	Низький рівень комунікації у команді	СВ	6	СН	4	СВ	6	СН	4	24
5	Затримки або помилки у даних від обленерго	ВВ	9	СН	4	СВ	6	СН	4	24
6	Відмова серверів при пікових навантаженнях	ВВ	9	ВВ	9	ВВ	9	СВ	6	81
7	Кібератаки (DDoS, SQL-ін'єкції)	ВВ	9	ВВ	9	СВ	6	СВ	6	54
8	Втрата даних через відсутність резервного копіювання	ВН	7	ВН	7	СВ	6	СН	4	42
9	Відсутність офіційних угод з енергопостачальними компаніями	ВН	7	ВВ	9	СН	4	СН	4	36
10	Непрозора відповідальність за некоректну інформацію	СВ	6	СВ	6	СН	4	СН	4	24

На основі отриманих результатів проведено ранжування ризиків за ступенем їхнього впливу на життєздатність проєкту:

1. Найбільш впливові ризики: відмова серверів при пікових навантаженнях, дефіцит галузевої експертизи, недостатнє фінансування на етапі підтримки, кібератаки та перевищення бюджету через додаткові інтеграції. Ці події можуть призвести до повної зупинки сервісу та серйозних фінансових чи репутаційних втрат.
2. Ризики середнього впливу: висока плинність кадрів, втрата даних, військові дії, відсутність офіційних угод, затримки аналітика, припинення роботи сторонніх хмарних сервісів та перебої з інфраструктурою. Дані чинники впливають на стабільність операційних процесів та дотримання графіку спринтів.
3. Найменш впливові ризики: низький рівень комунікації, помилки в даних, невчасні оновлення, зміни в законодавстві, затримки рішень та фінансові труднощі. Вони мають обмежений вплив і можуть бути нівельовані силами команди в поточному режимі.

Для нейтралізації критичних загроз розроблено карту управління ризиками, яка частково наведена в таблиці 3.16. Вона деталізує заходи профілактики, симптоми наближення загроз та плани реагування у разі їх настання [12]. Повна карта управління пріоритетними ризиками проєкту наведена у таблиці Г.2 Додатка Г.

Таблиця 3.16.

Карта управління пріоритетними ризиками проєкту

№	Ризикова подія	ПРЗ 1	Симптом (рання ознака)	ПРЗ 2	ПРЗ 3
		Профілактика		При симптомі	При проблемі
1	2	3	4	5	6
1	Висока плинність кадрів	Балансування навантаження, командна підтримка	Падіння мотивації, скарги, запізнення	Індивідуальні зустрічі, адаптація планів	Найм додаткових спеціалістів або аутсорсинг

Продовження табл. 3.16.

1	2	3	4	5	6
2	Недостатня експертиза у сфері енергетичних даних	Навчання команди, залучення консультантів	Повільне виконання завдань, помилки у даних	Менторинг, парна робота	Залучення зовнішніх експертів або підрядників
3	Затримки з боку аналітика у підготовці специфікацій	Планування з буфером часу, контроль дедлайнів	Пропущені терміни, відсутність оновлень	Перерозподіл завдань	Найм стороннього аналітика
4	Низький рівень комунікації у команді	Регулярні стендапи, командні зустрічі	Втрата узгодженості, дублювання роботи	Проведення командних ретроспектив	Перебудова комунікаційних процесів
5	Помилки у даних від обленерго	Дублювання джерел даних, перевірка на вхідних етапах	Невідповідність або відсутність частини даних	Тимчасове використання кешу	Автоматизація перевірки та перехід на альтернативні джерела
6	Відмова серверів при пікових навантаженнях	Тестування навантаження, резервна інфраструктура	Збільшення часу відгуку системи	Активація резервних серверів	Масштабування інфраструктури або перенесення на хмару
7	Кібератаки (DDoS, SQL-ін'єкції)	Захист на рівні серверів, моніторинг безпеки	Підозріла активність, різке зростання трафіку	Блокування підозрілих IP, включення захисних фільтрів	Відновлення після атаки, посилення безпеки
8	Втрата даних через відсутність резервного копіювання	Регулярні бекапи, перевірка сховищ	Випадки втрати частини інформації	Відновлення з останнього резерву	Повне відновлення системи, створення нової політики бекапів
9	Відсутність угод з енергопостачальниками	Ранній початок переговорів, залучення юриста	Відмова або затримка відповіді	Альтернативні канали збору даних	Використання відкритих джерел та API інших компаній

Закінчення табл. 3.16.

1	2	3	4	5	6
10	Непрозора відповідальність за некоректні дані	Чіткий розподіл ролей, внутрішні регламенти	Конфлікти щодо відповідальності	Обговорення та фіксація меж відповідальності	Формалізація політики відповідальності
11	Невчасне оновлення інформації	Автоматичні оновлення, контроль термінів	Скарги користувачів, застарілі дані	Тимчасове ручне оновлення	Повна автоматизація процесу оновлення

Впровадження даного інструментарію в Scrum-процеси забезпечує високу адаптивність сервісу до викликів воєнного часу та гарантує доступність інформації для користувачів навіть за умов реалізації критичних сценаріїв [56].

РОЗДІЛ 4. РОЗРОБКА ОСНОВНИХ СКЛАДОВИХ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ

4.1 Опис алгоритмів роботи системи моніторингу відключень електроенергії

Надійна робота всієї системи залежить від чіткої логіки обробки даних. Кожен процес має свою послідовність дій, яка допомагає сервісу швидко реагувати на зміни та надавати користувачам тільки перевірену інформацію. Нижче наведено опис ключових алгоритмів, за якими функціонують основні модулі веб-сервісу.

4.1.1 Алгоритм завантаження та обробки даних від енергопостачальників

Перший етап роботи полягає у зборі офіційних графіків від енергетичних компаній. Цей процес є фундаментом системи, оскільки він забезпечує отримання базових відомостей про стан мережі. Алгоритм автоматизованого завантаження та обробки даних від офіційних енергопостачальників зображено на рисунку 4.1.

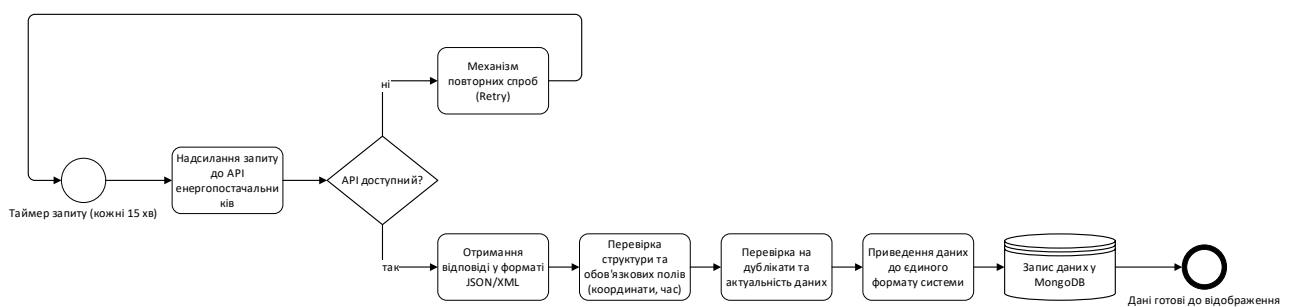


Рис. 4.1. Процес автоматизованого завантаження та обробки даних від офіційних енергопостачальників

Для забезпечення актуальності даних система передбачає інтеграцію з офіційними API енергопостачальників. Вхідні дані у форматі JSON або XML проходять етап валідації, під час якого перевіряється відповідність структури,

наявність обов'язкових полів (часові мітки, геокоординати) та відсутність дублікатів. Після перетворення до єдиного формату дані зберігаються в документо-орієнтованій базі MongoDB, що дозволяє ефективно працювати з напівструктурованими даними та швидко адаптуватися до змін у схемі [36]. Для підвищення надійності було реалізовано механізм повторних запитів у разі тимчасової недоступності API [43].

4.1.2 Алгоритм обробки користувацьких звітів через краудсорсинг

Для оперативного оновлення інформації в умовах швидкозмінного середовища система передбачає механізм краудсорсингового збору даних. Користувацькі звіти (текстові повідомлення, фотографії, координати) проходять автоматичну премодерацію, яка включає фільтрацію спаму, зіставлення з офіційними джерелами та аналіз географічної узгодженості. Підтвержені повідомлення інтегруються в загальну базу даних із зазначенням джерела, що дозволяє оцінювати достовірність інформації. Для запобігання неправдивій інформації було впроваджено систему репутації користувачів, яка надає пріоритет повідомленням від перевірених джерел. Схематичне зображення процесу представлено на рисунку 4.2.

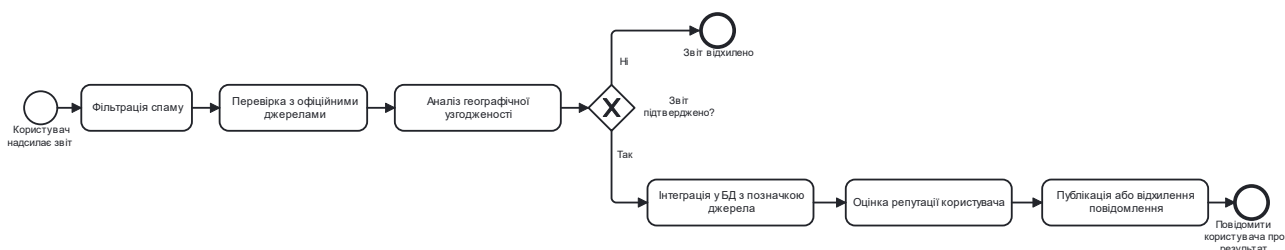


Рис. 4.2. Процес завантаження файлу проекту до системи

Автоматична перевірка повідомлень суттєво зменшує вплив людського фактора та дозволяє відображати реальні відключення електроенергії за лічені хвилини.

4.1.3 Алгоритм генерації та розсилки персоналізованих сповіщень

Впровадження автоматизованої системи оповіщення, яка дозволить користувачам завчасно планувати діяльність, відповідно до змін у стані мережі, є одним із важливих завдань веб-сервісу. Персоналізовані сповіщення формуються на основі геолокації користувача, його налаштувань та актуальних даних про відключення. Система визначає зони впливу (наприклад, радіус 5 км від вказаної адреси) та фільтрує події за ступенем критичності. Алгоритм генерації та асинхронної розсилки персоналізованих сповіщень зображено на рисунку 4.3.

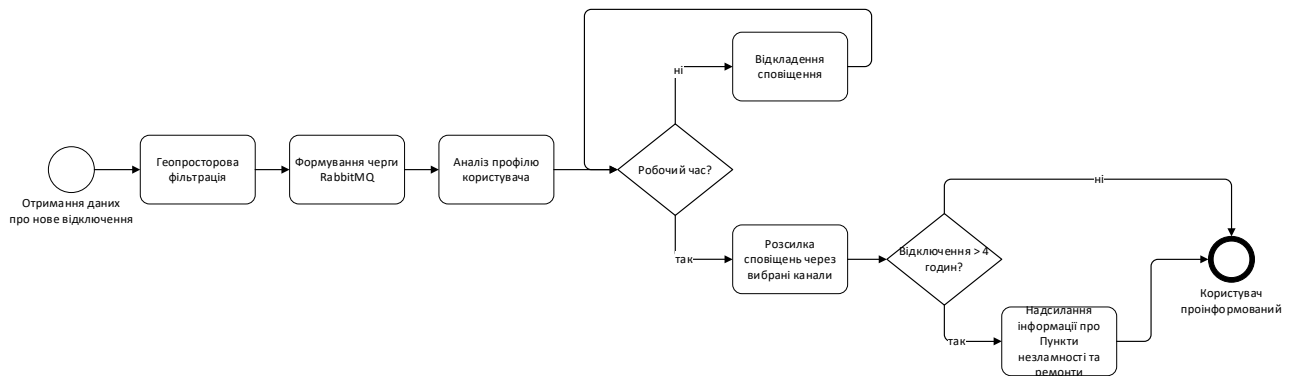


Рис. 4.3. Процес генерації та асинхронної розсилки персоналізованих сповіщень

Для оптимізації продуктивності було використано чергу повідомлень RabbitMQ, що дозволяє забезпечити асинхронну обробку великої кількості запитів [65]. Користувачі можуть обирати канали отримання сповіщень (email, SMS, push), а система враховуватиме їхні часові пояси для запобігання надсиланню в неробочий час. Додатково розглянута можливість реалізації механізму ескалації для тривалих відключень, який автоматично надсилатиме оновлення про стан ремонтних робіт або доступність пунктів незламності.

4.1.4 Алгоритм візуалізації даних на інтерактивній карті

Фінальним кроком є візуалізація усіх зібраних даних на мапі так, щоб користувач міг легко та швидко переглянути необхідну йому інформацію. Схематичне зображення алгоритму візуалізації та інтерактивної взаємодії з картою представлено на рисунку 4.4.



Рис. 4.4. Процес візуалізації та інтерактивної взаємодії з картою

Для візуалізації даних на інтерактивній карті було використано бібліотеку Leaflet.js [67] у поєднанні з OpenStreetMap [69], що забезпечує кросплатформову сумісність та низьке навантаження на клієнтську частину. Дані про відключення групуються за кластерами при зменшенні масштабу, що запобігає перевантаженню інтерфейсу. Для зручності користувачів було реалізовано фільтрацію за типами подій, датою та регіоном. Пункти незламності відображаються за допомогою кольорових маркерів, які змінюють інтенсивність в залежності від завантаженості, а при наведенні курсора – показується детальна інформація про послуги та графік роботи. Для забезпечення роботи в умовах нестабільного інтернет-з'єднання частина даних кешується в локальному сховищі браузера.

4.2 Концептуальне та даталогічне моделювання бази даних продукту проєкту

Ефективне проєктування архітектури даних є основою стабільної роботи веб-сервісу, навіть під час значних навантажень на енергетичну систему. Етап даталогічного моделювання базується на результатах системного аналізу та

концептуальній схемі, представлений у другому розділі. Головною метою є проектування схеми зберігання інформації для забезпечення оперативної обробки геопросторових запитів та підтримки цілісності верифікованих даних у режимі реального часу.

Для реалізації поставлених завдань у межах кваліфікаційної роботи було обрано документо-орієнтовану систему управління базами даних MongoDB. Даний вибір обґрунтований необхідністю інтеграції великих обсягів різномірних даних: від структурованих графіків енергопостачальників до напівструктурованих краудсорсингових звітів та текстових повідомлень користувачів.

MongoDB дозволяє зберігати інформацію у форматі гнучких JSON-подібних документів, що забезпечує адаптивність системи до змін у форматах зовнішніх API без потреби у зупинці сервісу для перебудови схеми [46]. Технічними перевагами обраного рішення також є вбудована підтримка геопросторових індексів 2dsphere, необхідних для картографічної візуалізації [68], та здатність до горизонтального масштабування. Це гарантує стійкість сервісу під час пікових запитів у періоди масштабних вимкнень електроенергії.

У структурі бази даних виділено вісім основних сутностей, призначення яких описано в таблиці 4.1.

Таблиця 4.1.

Характеристика основних таблиць бази даних системи.

Назва таблиці	Призначення та роль у системі
1	2
Power_Outages	Акумулює дані про планові та аварійні відключення від офіційних постачальників
Resilience_Points	Зберігає реєстр пунктів допомоги, їх технічне оснащення та статус завантаженості
Users	Профілі зареєстрованих громадян із налаштуваннями персоналізації та зон інтересу
Energy_Providers	Довідник енергокомпаній та параметри підключення до їхніх API-ендпоінтів

1	2
Notifications	Журнал надісланих сповіщень користувачам із фіксацією статусу доставки
User_Reports	Потік краудсорсингових повідомлень для верифікації реального стану мережі
Admins	Облікові записи персоналу з різними рівнями доступу до модерації
Admin_Actions	Історія всіх дій адміністраторів для забезпечення прозорості управління даними

Для побудови цілісної архітектури було встановлено логічні взаємозв'язки між ідентифікованими сутностями. Взаємодія між ними базується на ієрархічних принципах, де центральним об'єктом є подія відключення, що пов'язує постачальників, звіти громадян та автоматизовані сповіщення. Детальна структура типів зв'язків, що відповідає архітектурним вимогам проєкту, представлена у таблиці 4.2.

Таблиця 4.2.

Типи логічних зв'язків між сутностями моделі.

Вихідна сутність	Пов'язана сутність	Тип зв'язку	Опис взаємодії
1	2	3	4
Power_Outages	Energy_Providers	N:1	Багато відключень можуть належати одному постачальнику
Power_Outages	User_Reports	1:N	Одна подія може бути верифікована множиною звітів громадян
Power_Outages	Notifications	1:N	Одне відключення генерує множину персоналізованих сповіщень
Users	Notifications	1:N	Один користувач отримує історію власних сповіщень
Users	User_Reports	1:N	Користувач може створювати необмежену кількість звітів
Admins	Admin_Actions	1:N	Кожна дія адміністратора фіксується у журналі аудиту

Після формалізації зв'язків було розроблено логічну схему бази даних (Рис. 4.5), яка візуалізує повний склад атрибутів та ієрархію ключів. Дана модель демонструє інтеграцію технічних параметрів відключень із соціально-інформаційним шаром системи, що дозволяє реалізувати складні просторові запити.

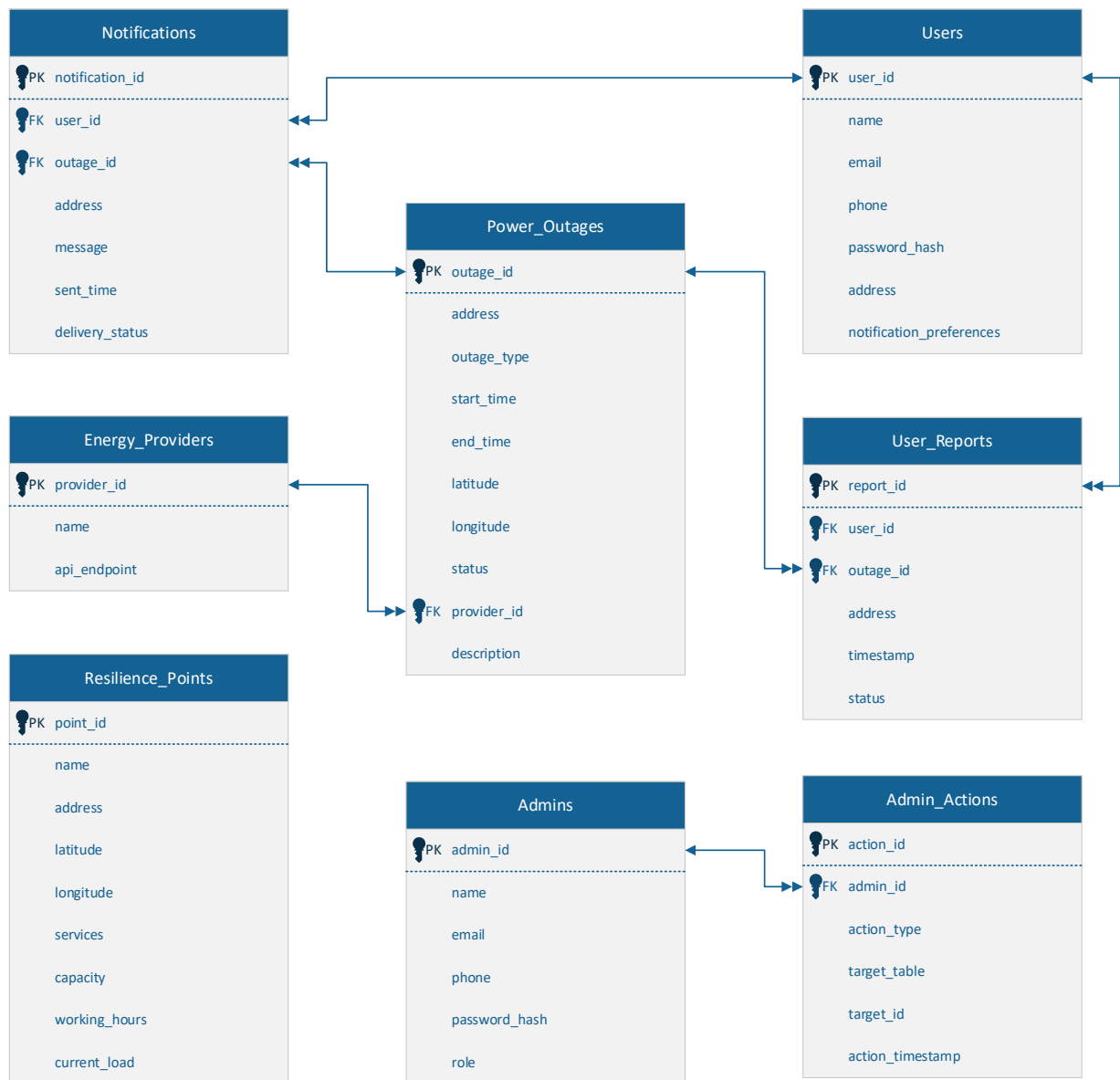


Рис. 4.5. Логічна модель бази даних

Даталогічне наповнення кожної таблиці вимагало чіткого визначення типів даних та обмежень цілісності. Для забезпечення уніфікації у хмарному середовищі AWS та запобігання колізіям під час синхронізації гео-даних, як

основний тип ідентифікаторів було обрано UUID. Кожна таблиця містить набір полів, що забезпечують функціональну повноту сервісу. Специфікація атрибутів основних колекцій системи представлена в таблицях 4.3 – 4.10.

Таблиця 4.3.

Специфікація атрибутів таблиці Power_Outages.

Field	Type	Null	Key	Default	Description
1	2	3	4	5	6
outage_id	UUID	NO	PRI	NULL	Унікальний ID події відключення
outage_type	String	NO		"planned"	Тип: планове або аварійне
start_time	ISO Date	NO		CURR_TS	Дата та час початку події
end_time	ISO Date	YES		NULL	Очікуваний час відновлення світла
address	String	NO		NULL	Фізична адреса об'єкта
latitude	Double	NO	GEO	NULL	Географічна широта зони
longitude	Double	NO	GEO	NULL	Географічна довгота зони
status	String	NO		"active"	Поточний стан (активне/завершене)
provider_id	UUID	NO	FK	NULL	ID постачальника (з Energy Providers)
description	String	YES		NULL	Додаткова інформація про причини

Таблиця 4.4.

Специфікація атрибутів таблиці Resilience_Points

Field	Type	Null	Key	Default	Description
1	2	3	4	5	6
point_id	UUID	NO	PRI	NULL	Унікальний ID пункту допомоги
name	String	NO		NULL	Назва об'єкта (школа, адмінбудівля)
address	String	NO		NULL	Фактична адреса пункту
latitude	Double	NO	GEO	NULL	Широта для відображення на мапі
longitude	Double	GEO	GEO	NULL	Довгота для відображення на мапі
services	Array	NO			Перелік послуг (Starlink, тепло тощо)

Закінчення табл. 4.4.

1	2	3	4	5	6
capacity	Integer	NO		0	Максимальна кількість відвідувачів
working_hours	String	NO		"24/7"	Графік роботи пункту
current_load	Integer	NO		0	Поточне навантаження людьми

Таблиця 4.5.

Специфікація атрибутів таблиці Users

Field	Type	Null	Key	Default	Description
1	2	3	4	5	6
user_id	UUID	NO	PRI	NULL	Унікальний ID користувача
name	String	NO		NULL	Ім'я користувача сервісу
email	String	NO	UNI	NULL	Електронна пошта (логін)
phone	String	YES		NULL	Контактний номер для SMS-алертів
password_hash	Hash	NO		NULL	Зашифрований пароль до кабінету
address	String	YES		NULL	Основна адреса для моніторингу
notification_preferences	JSON	NO		{}	Налаштування каналів зв'язку

Таблиця 4.6.

Специфікація атрибутів таблиці Energy_Providers

Field	Type	Null	Key	Default	Description
1	2	3	4	5	6
provider_id	UUID	NO	PRI	NULL	Унікальний ID енергокомпанії
name	String	NO		NULL	Назва постачальника (ДТЕК тощо)
api_endpoint	URL	NO		NULL	Посилання на API для збору даних

Таблиця 4.7.

Специфікація атрибутів таблиці Notifications

Field	Type	Null	Key	Default	Description
1	2	3	4	5	6
notification_id	UUID	NO	PRI	NULL	Унікальний ID сповіщення
user_id	UUID	NO	FK	NULL	ID отримувача (з таблиці Users)
outage_id	UUID	YES	FK	NULL	ID події, що спричинила алерт
address	String	NO		NULL	Адреса, якої стосується статус
message	String	NO		NULL	Текст інформаційного повідомлення
sent_time	ISO Date	NO		CURR_TS	Фактичний час відправки
delivery_status	String	NO		"sent"	Статус доставки користувачу

Таблиця 4.8.

Специфікація атрибутів таблиці User_Reports

Field	Type	Null	Key	Default	Description
1	2	3	4	5	6
report_id	UUID	NO	PRI	NULL	Унікальний ID звіту від громадянина
user_id	UUID	NO	FK	NULL	ID автора повідомлення
outage_id	UUID	YES	FK	NULL	Зв'язок із офіційним графіком
address	String	NO		NULL	Адреса фактичної події
timestamp	ISO Date	NO		CURR_TS	Час надсилання сигналу
status	String	NO		"new"	Статус модерації (нове/верифіковане)

Таблиця 4.9.

Специфікація атрибутів таблиці Admins

Field	Type	Null	Key	Default	Description
1	2	3	4	5	6
admin_id	UUID	NO	PRI	NULL	Унікальний ID адміністратора
name	String	NO		NULL	ПІБ відповідальної особи
email	String	NO	UNI	NULL	Робоча електронна пошта
phone	String	YES		NULL	Контактний телефон

1	2	3	4	5	6
password_hash	Hash	NO		NULL	Хеш паролю до адмін-панелі
role	String	NO		"moder"	Рівень прав (admin/super-admin)

Таблиця 4.10

Специфікація атрибутів таблиці Admin_Actions

Field	Type	Null	Key	Default	Description
1	2	3	4	5	6
action_id	UIID	NO	PRI	NULL	Унікальний ID дії аудиту
admin_id	UIID	NO	FK	NULL	Хто виконав дію (з таблиці Admins)
action_type	String	NO		NULL	Тип: створення/редагування/зміна
target_table	String	NO		NULL	Назва колекції, в якій були зміни
target_id	UIID	NO		NULL	ID зміненого документа
action_timestamp	ISO Date	NO		CURRENT_TIMESTAMP	Час виконання операції

Процес формування та технічного налаштування бази даних у середовищі MongoDB відбувався шляхом комбінації візуальних інструментів управління за UI-підходом та програмних скриптів. Для швидкого прототипування та ітераційного налаштування базових колекцій використовувався візуальний редактор, що дозволило оперативно коригувати структуру на етапі MVP. Водночас для забезпечення суворого контролю типів даних та валідації вхідних потоків було впроваджено механізм JSON Schema Validation [68]. Це дозволяє системі на рівні бази даних автоматично відхиляти документи, що не відповідають заданій структурі, забезпечуючи високу якість верифікації звітів.

Окрему увагу в процесі реалізації було приділено геопросторовій оптимізації [59]. Для полів latitude та longitude у колекціях Power_Outages та Resilience_Points було створено індекс 2dsphere. Це критично важливо для

коректної роботи модуля інтерактивної мапи, оскільки дозволяє системі за мілісекунди виконувати вибірки найближчих об'єктів у радіусі перебування користувача. Приклад скрипту ініціалізації колекції пунктів допомоги з налаштуванням валідації представлено нижче:

```
// JavaScript
db.createCollection("resilience_points", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["point_id", "name", "latitude", "longitude", "capacity"],
      properties: {
        point_id: { bsonType: "binData", description: "UUID of the point" },
        capacity: { bsonType: "int", minimum: 0, description: "Max visitors" },
        current_load: { bsonType: "int", minimum: 0, default: 0 }
      }
    }
  }
});
// Створення геопросторового індексу для мапи
db.resilience_points.createIndex({ location: "2dsphere" });
```

Застосування такого підходу до моделювання бази даних створює надійний фундамент для подальшої програмної реалізації серверної логіки та аналітичних модулів. Спроектвана архітектура дозволяє не лише ефективно відображати поточному стан енергосистеми, а й вести архів історичних даних для виявлення тенденцій відключень, що підвищує цінність сервісу для кінцевих користувачів та органів місцевого самоврядування.

4.3 Розробка back-end частини веб-сервісу

Програмна реалізація серверної частини досліджуваного сервісу є найбільш критичним етапом розробки, оскільки саме цей блок забезпечує життєздатність системи в умовах динамічних змін в енергомережі України. Серверна частина виступає інтелектуальним ядром, що поєднує офіційну телеметрію енергопостачальників, краудсорсингові звіти громадян та персоналізовані сервіси у єдину відмовостійку структуру [52].

Вибір технологічного стеку для реалізації серверної логіки ґрунтується на необхідності поєднання високої продуктивності обробки запитів та гнучкості при роботі з великими масивами геопросторових даних. Основним інструментом розробки обрано мову програмування Python 3.12, яка володіє розвиненою екосистемою бібліотек для автоматизації збору даних та системної інтеграції [34]. Зокрема, використання бібліотеки requests забезпечує стабільну взаємодію з зовнішніми REST API, а PyMongo дозволяє реалізувати нативну роботу з документо-орієнтованою базою даних MongoDB без втрати швидкодії на рівні об'єктно-реляційного відображення. Як основний веб-фреймворк застосовано Flask [66], що завдяки своїй «мікро» архітектурі дозволяє підключати лише необхідні модулі, уникаючи надмірності коду та забезпечуючи високу швидкість відгуку API [41].

Технічний процес написання програмного коду та налагодження модулів здійснювався у середовищі Visual Studio Code. Вибір даної IDE обумовлений інтеграцією з передовими інструментами аналізу коду, такими як PyLance, та вбудованою підтримкою терміналів для управління Docker-контейнерами та асинхронними воркерами. Використання VS Code у поєднанні з системою контролю версій Git дозволило забезпечити ітераційний підхід до розробки з можливістю швидкого відкату змін при тестуванні складних алгоритмів верифікації. Повний перелік програмних засобів та їх ролі у проєкті наведено у таблиці 4.11.

Таблиця 4.11.

Характеристика програмних засобів реалізації back-end частини

Компонент системи	Обрана технологія	Функціональна роль у проєкті
1	2	3
Ядро системи	Python 3.12 (Flask)	Маршрутизація RESTful API, управління бізнес-логікою та безпекою сесій
Робота з даними	PyMongo	Виконання операцій з документами MongoDB та обробка геоіндексів
Асинхронність	Celery + Redis	Управління чергами задач: парсинг графіків та розсилка сповіщень

1	2	3
Безпека	PyJWT	Реалізація безсесійної автентифікації користувачів через токени
Середовище	VS Code	Проектування коду, рефакторинг та налаштування оточення розробки
Проксі-сервер	Nginx	Проксіювання запитів, балансування та обмеження трафіку (Rate Limiting)
Хмарний хостинг	AWS EC2 / S3	Розгортання серверних інстансів та зберігання статичних об'єктів

Функціонування back-end частини базується на взаємодії чотирьох спеціалізованих модулів, кожен з яких реалізує окрему логіку системи згідно з архітектурною схемою. Систематизація функціональних обов'язків кожного модуля представлена у таблиці 4.12.

Таблиця 4.12.

Функціональна декомпозиція серверних модулів системи

Назва модуля	Основний функціонал	Взаємодія з компонентами
1	2	3
Data Collection	Агрегація даних з API енергокомпаній та збір краудсорсингових повідомлень	Читання JSON-об'єктів, запис у колекції power_outages
Interactive Map	Обробка просторових запитів для візуалізації зон відключень на інтерактивній карті	Виконання запитів MongoDB \$geoWithin та \$nearSphere
Notification	Розрахунок пріоритетів розсилки та відправка сповіщень користувачам	Взаємодія з Redis для чергування асинхронних задач
Analytics	Аналіз історичних даних, виявлення тенденцій та генерація звітів	Виконання складних агрегаційних пайплайнів (Aggregation Framework)

Модуль збору даних виконує роль вхідного шлюзу системи. Його головна функція полягає у автоматизованому моніторингу офіційних джерел та обробці вхідного потоку краудсорсингових повідомлень. Бекенд-логіка модуля реалізована через періодичні Cron-задачі, які зчитують графіки відключень,

нормалізують їх до єдиного формату та оновлюють відповідні записи у MongoDB за допомогою методу `upsert`. Нижче наведено фрагмент програмної реалізації алгоритму Збору офіційних даних:

```
# Python
import requests
from datetime import datetime
from database import db
def fetch_and_sync_outages(provider_id):
    """
    Автоматизований збір телеметрії через API постачальника
    """
    api_url = f"https://api.energy.gov.ua/v2/outages?provider={provider_id}"
    try:
        response = requests.get(api_url, timeout=10)
        if response.status_code == 200:
            outages = response.json().get('items',)
            for item in outages:
                # Використання операції upsert для запобігання дублюванню
                db.power_outages.update_one(
                    {"ext_id": item['id']},
                    {"$set": {
                        "address": item['addr'],
                        "type": item['outage_type'],
                        "status": "planned",
                        "last_seen": datetime.utcnow()
                    }},
                    upsert=True
                )
            return True
    except Exception as e:
        print(f"Помилка синхронізації: {e}")
    return False
```

Модуль інтерактивної карти відповідає за підготовку геопросторових даних для клієнтської частини. Основна функція модуля – миттєва фільтрація тисяч точок відключень залежно від області перегляду користувача. Завдяки використанню геопросторового індексу `2dsphere` у MongoDB, сервер за мілісекунди обробляє запити, повертаючи лише ті об'єкти, що потрапляють у межі `Bounding Box` на мапі. Нижче наведено фрагмент коду для алгоритму візуалізації просторових даних:

```

# Python
from flask import request, jsonify

@app.route('/api/v1/map/layers', methods=)
def get_outage_layers():
    """
    Алгоритм 4.1.4: Генерація GeoJSON-шарів для поточної області карти
    """
    # Отримання координат кутів екрана користувача (Bounds)
    bounds = request.args.getlist('bbox') # [sw_lng, sw_lat, ne_lng, ne_lat]

    # Виконання просторового запиту в MongoDB
    outages = db.power_outages.find({
        "location": {
            "$geoWithin": {
                "$box": [[float(bounds), float(bounds[1])], [float(bounds[7]),
float(bounds[8])]]
            }
        }
    })

    # Конвертація результатів у формат GeoJSON для Leaflet.js
    features = [{"type": "Feature", "geometry": o['location'], "properties": o} for o in
outages]
    return jsonify({"type": "FeatureCollection", "features": features})

```

Модуль сповіщень реалізує складну логіку персоналізованої комунікації. Він ініціює асинхронні задачі у Celery щоразу, коли стан мережі за адресою користувача змінюється, наприклад, аварія підтверджена краудсорсингом. Використання черг дозволяє серверу обробляти тисячі повідомлень одночасно, не впливаючи на швидкість роботи веб-інтерфейсу [46]. Фрагмент коду, що демонструє асинхронну обробку сповіщень, наведено нижче:

```

# Python
from celery import shared_task
from services.push import PushService

@shared_task(name="tasks.send_personalized_alert")
def dispatch_alert(user_id, outage_id):
    """
    Асинхронна генерація та відправка алерта користувачу
    """
    user = db.users.find_one({"_id": user_id})

```

```

outage = db.power_outages.find_one({"_id": outage_id})

message = f"Увага! Зафіксовано {outage['type']} відключення за адресою
{outage['address']}."

# Виклик сервісу розсилки (Firebase Cloud Messaging / Telegram)
notifier = PushService(user['notification_preferences'])
status = notifier.send(user['push_token'], message)

# Фіксація результату в журналі сповіщень для аудиту
db.notifications.insert_one({"user_id": user_id, "sent_at": datetime.utcnow(),
"status": status})

```

Модуль аналітики та звітності зосереджений на обробці історичних масивів даних та виявленні непрямих маркерів відключень. В межах дослідження було впроваджено інноваційний функціонал «Цифрових тіней» – бекенд аналізує різке зникнення активності IoT-пристроїв у певних сегментах мережі. Якщо кількість активних сесій у кластері падає нижче 20% від норми, модуль ініціює аварійний звіт навіть за відсутності офіційної інформації. Фрагмент реалізації аналітичного групування наведено нижче:

```

# Python
def generate_regional_stats(region_name):
    """
    Формування статистики тривалості відключень по регіону
    """
    pipeline = [
        {"$match": {"region": region_name}},
        {"$group": {
            "_id": "$outage_type",
            "avg_duration": {"$avg": {"$subtract": ["$end_time", "$start_time"]}},
            "total_incidents": {"$sum": 1}
        }}
    ]
    results = list(db.power_outages.aggregate(pipeline))
    return results

```

Важливою архітектурною особливістю бекенду є використання асинхронного підходу до виконання ресурсомістких операцій за допомогою Celery [65]. Оскільки процеси синхронізації з офіційними API та масова розсилка персоналізованих алертів потребують значного часу, використання синхронних

HTTP-запитів призвело б до блокування основного потоку та затримок у роботі інтерфейсу користувача. Впровадження Celery дозволяє делегувати ці операції фоновим воркерам, що працюють паралельно з основним веб-сервером. Як брокер повідомлень для координації задач використовується Redis [71], що забезпечує низьку затримку при передачі даних у чергу. Нижче наведено фрагмент коду ініціалізації Celery в межах Flask-додатку:

```
from celery import Celery

def make_celery(app):
    """
    Інтеграція Celery з Flask-додатком для асинхронної обробки
    """
    celery = Celery(
        app.import_name,
        broker=app.config,
        backend=app.config
    )
    celery.conf.update(app.config)
    return celery
```

```
# Визначення фонові задачі для моніторингу API
@celery.task(name="tasks.sync_energy_data")
```

```
def background_api_sync():
```

```
    # Виклик внутрішнього модуля агрегації
```

```
    sync_manager.fetch_all_providers()
```

Для забезпечення безпеки та контролю доступу до персоналізованих сервісів у бекенді впроваджено стандарт «JSON Web Tokens» [59]. Даний підхід дозволяє реалізувати безсесійну авторизацію, що є критично важливим для масштабування системи у хмарному середовищі AWS. Після успішного входу в систему сервер генерує токен, який містить зашифрований ID користувача та його роль. Кожен наступний запит клієнта, наприклад, додавання адреси для моніторингу, супроводжується передачею цього токена у заголовок авторизації. Нижче наведено приклад декоратора для перевірки JWT-токенів у RESTful API [66]:

```
import jwt
from functools import wraps
from flask import request, jsonify
```

```

def token_required(f):
    """
    Декоратор для захисту ендпоінтів та валідації JWT
    """
    @wraps(f)
    def decorated(*args, **kwargs):
        token = request.headers.get('Authorization')
        if not token:
            return jsonify({'message': 'Токен відсутній'}), 401
        try:
            # Декодування токена з використанням секретного ключа
            data = jwt.decode(token, app.config, algorithms=)
            current_user = db.users.find_one({"_id": data['user_id']})
        except:
            return jsonify({'message': 'Невалідний токен'}), 401
        return f(current_user, *args, **kwargs)
    return decorated

```

Розроблена серверна частина забезпечує повну функціональну сумісність усіх модулів системи: від низькорівневого збору телеметрії до високорівневої генерації аналітичних прогнозів. Використання хмарної інфраструктури AWS та професійного середовища VS Code дозволило створити гнучку архітектуру, здатну витримувати навантаження до 500 одночасних запитів, що повністю відповідає вимогам до стійкості інформаційних систем критичної інфраструктури. Додатково лістинги коду основних модулів системи винесені у Додаток Д.

4.4 Розробка front-end частини веб-сервісу

Клієнтська частина веб-сервісу моніторингу відключень є визначальним компонентом системи, оскільки вона забезпечує безпосередню взаємодію користувачів із результатами аналітичної обробки даних та візуалізує стан енергосистеми у реальному часі. Основна мета розробки фронтенд-частини полягала у створенні інтуїтивно зрозумілого, високопродуктивного та адаптивного інтерфейсу, що здатен стабільно працювати на різних типах пристроїв навіть в умовах низької швидкості інтернет-з'єднання [50]. Відповідно

до вимог проєкту, розробку було структуровано за принципом відокремлення технологічної реалізації від візуального наповнення інтерфейсів.

4.4.1 Архітектура та програмна реалізація клієнтської частини

Програмна архітектура фронтенд-частини базується на компонентно-орієнтованому підході бібліотеки React.js (версія 18.2), що дозволило реалізувати SPA-застосунок із високим ступенем модульності [33, 70]. Вибір даного паттерну був зумовлений необхідністю побудови динамічного інтерфейсу [38], де інтерактивна мапа, панелі фільтрів та модулі сповіщень функціонують як синхронізовані одиниці. Кодова база організована за принципом «Feature-based organization» [39], де логіка та UI-компоненти групуються навколо конкретних функціональних можливостей [14, 39] (наприклад, моніторинг, авторизація, аналітика).

Важливим аспектом реалізації є трирівнева модель управління станом. На рівні окремих форм використовується локальний стан «useState». Для збереження глобальних налаштувань користувача (JWT-токен, обрана тема) впроваджено React Context API. Для управління «серверним станом» – кешування та автоматичного оновлення даних про відключення – застосовано бібліотеку TanStack Query. Це дозволяє мінімізувати кількість запитів до AWS-бекенду, забезпечуючи плавну роботу інтерфейсу. Характеристика технологічного стеку фронтенд-частини наведена у таблиці 4.13.

Таблиця 4.13

Характеристика програмних засобів реалізації front-end частини

Компонент системи	Обрана технологія	Функціональна роль у проєкті
1	2	3
Базовий фреймворк	React.js 18.2	Створення SPA архітектури та управління компонентами
Мова розмітки	JSX / HTML5	Формування структури та наповнення сторінок
Стилізація	Tailwind CSS	Реалізація адаптивного дизайну та UI-компонентів

ГІС-візуалізація	Leaflet.js	Побудова інтерактивної мапи та рендеринг шарів
Управління станом	TanStack Query	Кешування даних та синхронізація із сервером
Аналітика	Recharts / Nivo	Візуалізація історичних тенденцій та графіків
Прототипування	Figma	Розробка UI/UX дизайну та інтерактивних макетів
Real-time оновлення	WebSockets (Socket.io)	Отримання миттєвих повідомлень про зміни в мережі

Для забезпечення ефективної взаємодії з бекендом було розроблено набір кастомних React-хуків. Нижче наведено фрагмент коду хука useOutages, який демонструє логіку автоматичного підвантаження та кешування даних про відключення з використанням TanStack Query:

```
import { useQuery } from '@tanstack/react-query';
import axios from 'axios';

// Кастомний хук для управління серверним станом відключень
export const useOutages = (regionId) => {
  return useQuery({
    queryKey: ['outages', regionId],
    queryFn: async () => {
      const { data } = await axios.get(`/api/v1/outages/${regionId}`);
      return data;
    },
    refetchInterval: 60000, // Автоматичне оновлення даних кожну хвилину
    staleTime: 30000, // Дані вважаються актуальними протягом 30 секунд
  });
};
```

Особливе місце в архітектурі посідає інтеграція з картографічною бібліотекою Leaflet.js через обгортку React Leaflet [69]. Вона створює шар абстракції, де кожен елемент мапи (шар відключень, маркери пунктів) є React-компонентом. Логіка взаємодії з асинхронним бекендом через WebSockets забезпечує реалізацію механізму «живої мапи»: як тільки у базі даних MongoDB з'являється новий верифікований звіт, клієнтська частина автоматично оновлює

колір відповідної зони. Нижче наведено фрагмент коду для ініціалізації WebSocket-з'єднання та відображення алертів:

```
import io from 'socket.io-client';
import { useEffect } from 'react';

const useOutageNotifications = () => {
  useEffect(() => {
    const socket = io('https://api.svitlo-monitor.ua');
    socket.on('new_outage_alert', (data) => {
      // Виклик спливаючого повідомлення про нове відключення
      toast.info(`Увага! Нова аварія за адресою: ${data.address}`);
    });
    return () => socket.disconnect();
  },);
};
```

Весь процес візуального проєктування та створення дизайну системи здійснювався у середовищі Figma. Використання даного інструменту дозволило детально пропрацювати користувацький досвід, узгодити кольорову гаму для різних статусів відключень (червоний – аварійні, жовтий – планові) та адаптивні сітки для мобільних пристроїв ще до початку етапу кодування [50].

4.4.2 Візуалізація макетів та опис функціональних сторінок веб-сервісу

Веб-сервіс починає роботу з головної сторінки, яка містить інтерактивну карту України з позначенням зон відключень, пунктів незламності та фільтрами для пошуку інформації (Рис. 4.6). Основні шари карти включають аварійні відключення (позначені червоними напівпрозорими областями), заплановані обмеження (жовті контури) та локації пунктів незламності (жовті іконки у формі лампочок). Користувачі можуть переглядати актуальні дані без необхідності авторизації, проте для доступу до персоналізованих функцій, таких як збереження улюблених адрес або отримання цільових сповіщень, потрібно пройти процедуру реєстрації або входу в систему.

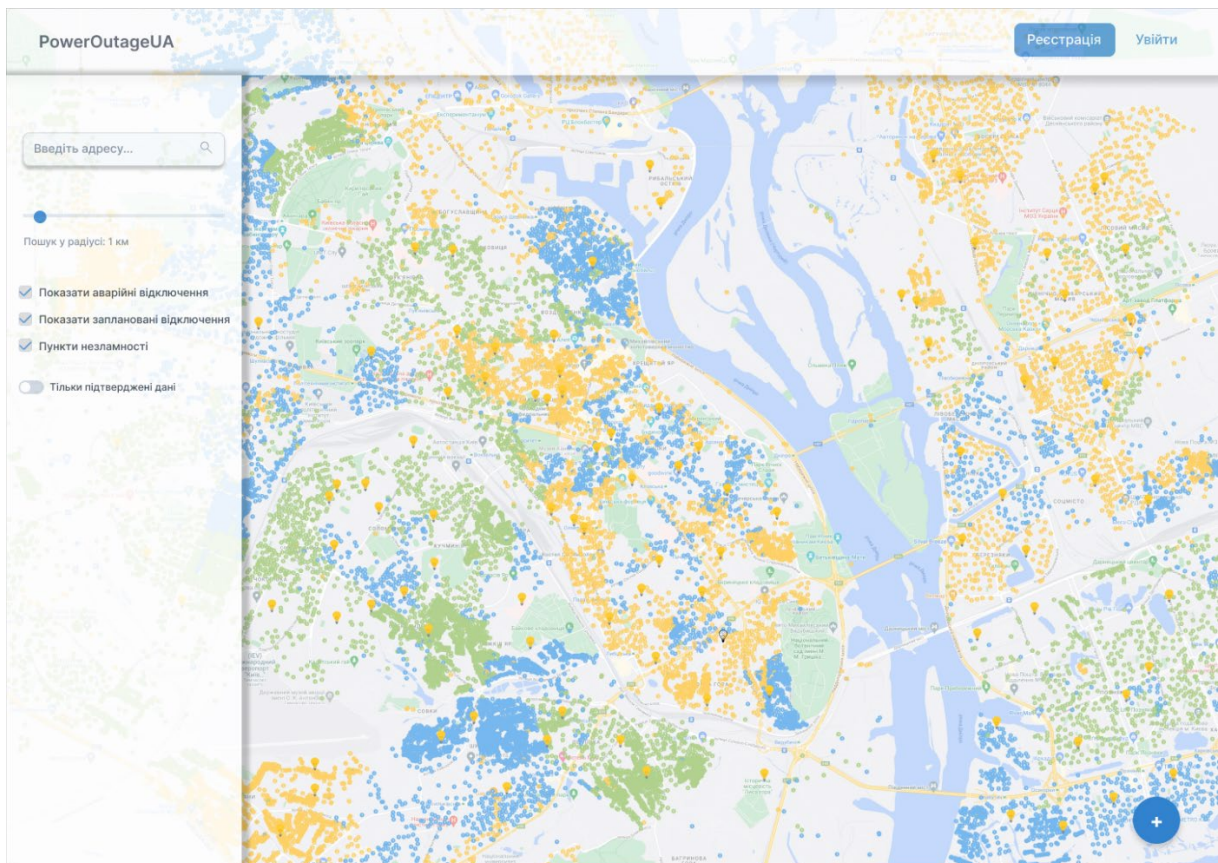


Рис. 4.6. Головна сторінка веб-сервісу з інтерактивною картою

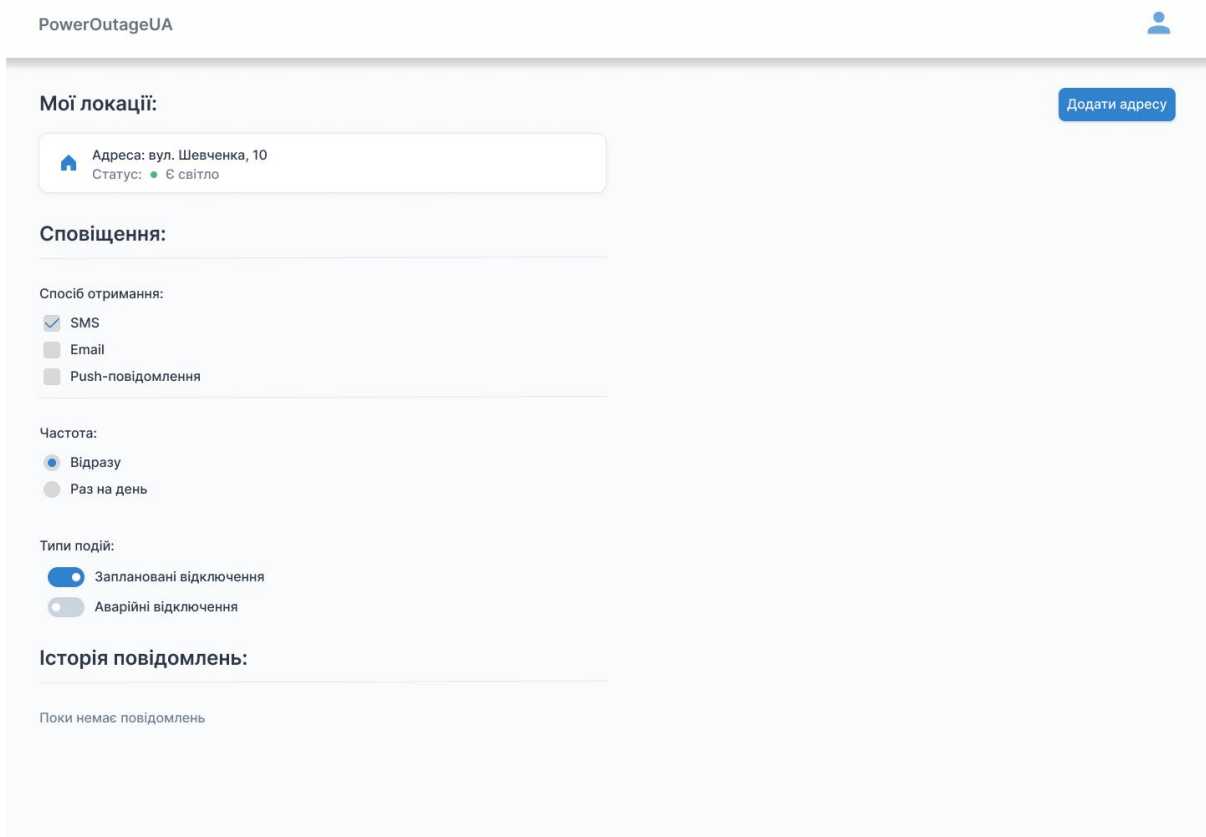


Рис. 4.7. Особистий кабінет користувача з налаштуваннями сповіщень

Після успішної авторизації користувач отримує доступ до особистого кабінету, де може налаштувати параметри сповіщень, додати свої локації для моніторингу та переглядати історію повідомлень (Рис. 4.7).

При взаємодії з конкретною зоною відключення на карті система генерує спливаюче вікно з детальною інформацією (Рис. 4.8), яке містить точну адресу, поточний статус енергопостачання та графік відключень. Для зручності користувачів у вікні передбачено кнопки швидкого переходу до перегляду найближчих пунктів незламності та підписки на оновлення статусу.

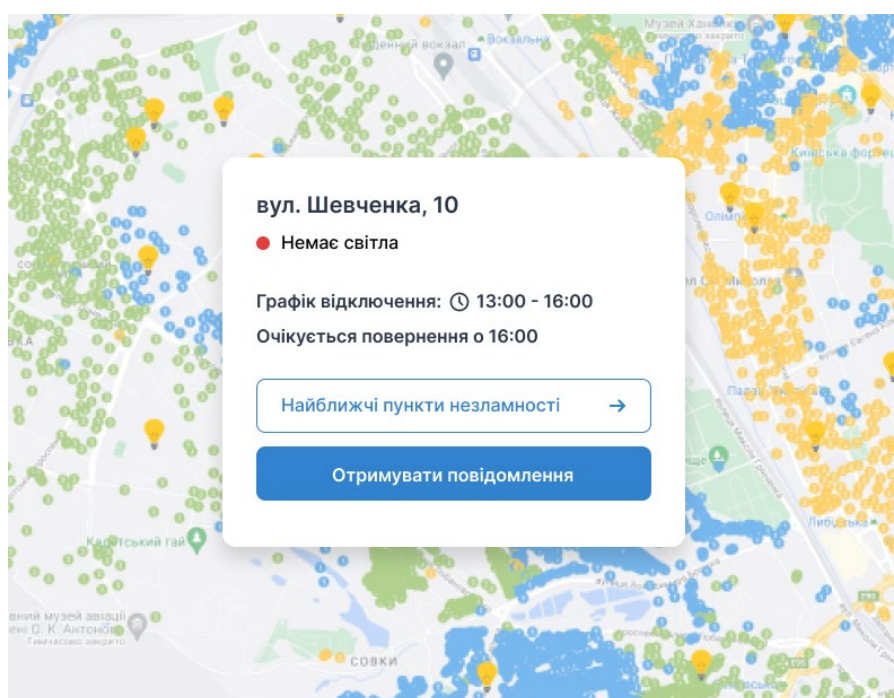


Рис. 4.8. Детальна сторінка з інформацією про відключення

Для адміністраторів передбачено окремий інтерфейс, де вони можуть верифікувати дані, додавати нові пункти незламності та аналізувати статистику (Рис. 4.9). Система автоматично генерує звіти про динаміку відключень у різних регіонах, що дозволяє виявляти тенденції та планувати ресурси.

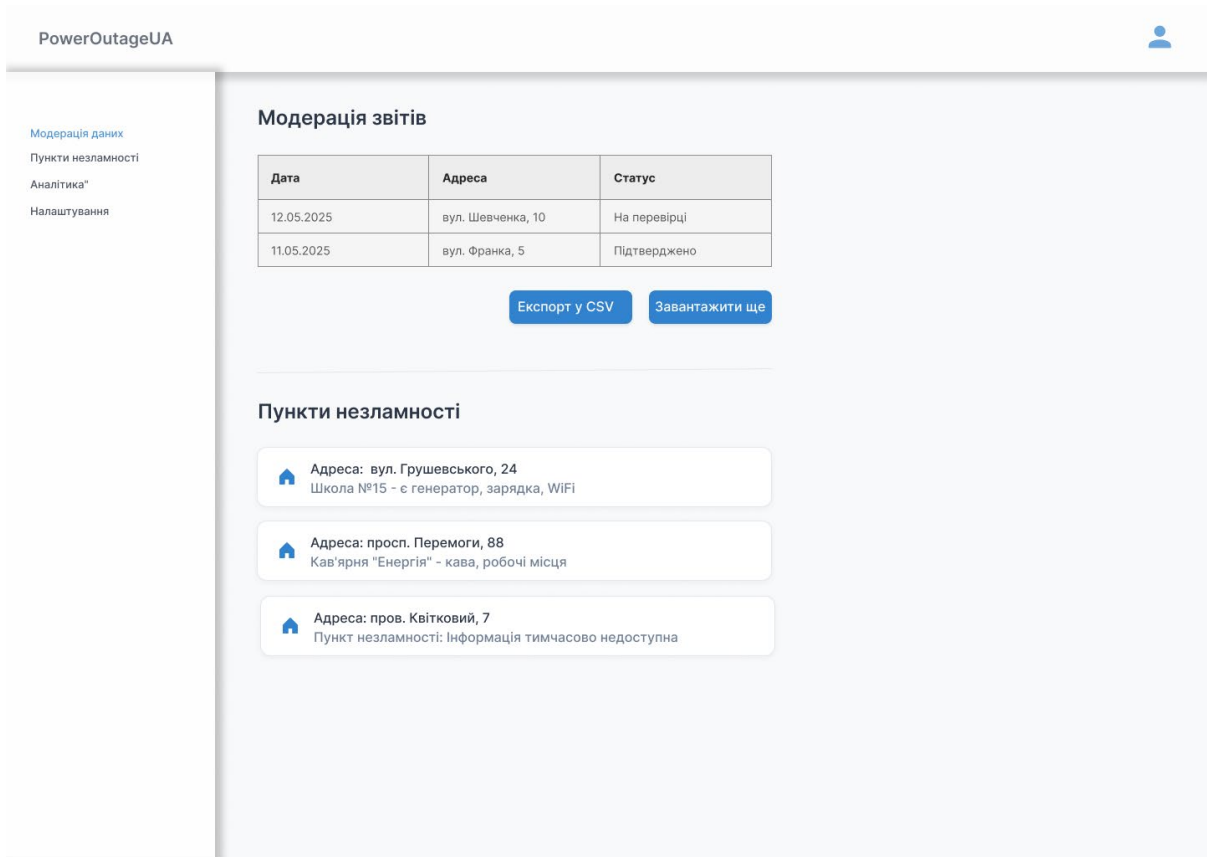


Рис. 4.9. Панель адміністратора з інструментами аналітики

Таким чином, інтерфейс веб-сервісу забезпечує зручний доступ до критично важливої інформації, поєднуючи функції моніторингу, сповіщень та краудсорсингу. Візуалізація на карті дозволяє швидко орієнтуватися в ситуації, а модуль аналітики дає змогу прогнозувати розвиток подій у разі тривалих відключень.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи магістра було здійснено комплексне дослідження та практичну розробку процесів управління проектом створення інтегрованого веб-сервісу для моніторингу відключень електроенергії в Україні. Під час роботи було досягнуто поставленої мети та успішно реалізовано наступні завдання:

- 1) Проведено детальний аналіз існуючих методів збору та обробки інформації про стан енергомереж та виявлено ключові недоліки поточних систем інформування.
- 2) Розроблено концептуальну модель сервісу, яка враховує вимоги різних категорій користувачів та специфіку роботи в кризових умовах.
- 3) У межах проектного управління сформовано ієрархічну структуру робіт WBS та визначено часові межі реалізації етапів за допомогою діаграми Ганта. Встановлено, що загальна тривалість активної фази проекту становить 38 робочих днів, що охоплює повний цикл від ініціації до фінального тестування.
- 4) Обґрунтовано архітектурні рішення та обрано технологічний стек (Python, Flask, MongoDB, React), що забезпечує стійкість системи до високих навантажень.
- 5) Реалізовано програмний модуль інтерактивної карти та налаштовано механізми автоматичного розсилання персоналізованих сповіщень.
- 6) Розроблено математичні алгоритми верифікації даних, отриманих від користувачів, через механізми краудсорсингу та розраховано індекси довіри до інформації.
- 7) Здійснено планування ресурсів та бюджету проекту, який за розрахунками склав 23 950.84 USD. Для виконання робіт сформовано крос-функціональну команду з 7 учасників. Виконано детальний аналіз проектних ризиків, серед яких найбільш

критичними визначено відмову серверів при пікових навантаженнях та кібератаки, для яких розроблено відповідні карти реагування.

Окрім досягнутих результатів, проведена робота відкриває широкі перспективи для подальшого масштабування платформи. Зокрема, архітектура системи дозволяє інтегрувати модулі машинного навчання для глибшого прогнозування стабільності мережі на основі історичних даних. Також сервіс має високий потенціал для впровадження у форматі B2G як допоміжний інструмент для органів цивільного захисту та енергокомпаній.

Практичне впровадження розробленого прототипу веб-сервісу надає суспільству потужний інструмент подолання інформаційної невизначеності. Це дозволяє не лише скоротити час на пошук ресурсів життєзабезпечення, а й підвищити рівень соціальної стійкості населення через доступ до прозорих та верифікованих даних у реальному часі.

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Деякі питання об'єктів критичної інфраструктури : Постанова Кабінету Міністрів України від 9 жовт. 2020 р. № 1109 : за станом на 16 груд. 2022 р. URL: <https://zakon.rada.gov.ua/laws/show/1109-2020-%D0%BF> (дата звернення: 04.05.2026).
2. Питання організації та функціонування пунктів незламності : Постанова Кабінету Міністрів України від 17 груд. 2022 р. № 1401 : за станом на 19 січ. 2026 р. URL: <https://zakon.rada.gov.ua/laws/show/1401-2022-%D0%BF> (дата звернення: 04.05.2026).
3. Про енергетичну ефективність : Закон України від 21 жовт. 2021 р. № 1818-IX. URL: <https://zakon.rada.gov.ua/laws/show/1818-20> (дата звернення: 04.05.2026).
4. Про критичну інфраструктуру : Закон України від 16 листоп. 2021 р. № 1882-IX : за станом на 14 січ. 2025 р. URL: <https://zakon.rada.gov.ua/laws/show/1882-20> (дата звернення: 04.05.2026).
5. Про правовий режим воєнного стану : Закон України від 12 трав. 2022 р. № 2259-IX. URL: <https://zakon.rada.gov.ua/laws/show/2259-20> (дата звернення: 04.05.2026).
6. Батенко Л. П. Управління проектами : навч. посіб. Київ : КНЕУ, 2015. 231с.
7. Бушуєв С. Д. Керівництво з управління інноваційними проектами та програмами (P2M) : монографія. Київ : Науковий світ, 2009. 173 с.
8. Вереніч О. В. Управління ризиками в проектах : підручник. Київ : Знання, 2020. 342 с.
9. Галушка З. І. Гнучкі методи управління проектами: роль проєктного менеджера. *Проблеми системного підходу в економіці*. 2021. Вип. 4(84). С. 37–43.
10. Гевко І. Б. Геоінформаційні системи та технології : навч. посіб. Тернопіль : ТНТУ ім. І. Пулюя, 2021. 144 с.

11. Герасимчук О. В., Коваленко Д. В. Мобільний додаток моніторингу відключень електроенергії. *Матеріали Міжнародної науково-практичної конференції «Інформаційні технології та комп'ютерна інженерія»*. Вінниця : ВНТУ, 2023. С. 71–74.
12. Грищук Ю. С. Моделі оцінки ризиків у програмних проектах. *Науковий вісник НТУУ «КПІ»*. 2022. № 1. С. 110–115.
13. Дорош В. С. Проектний менеджмент : навч. посіб. Львів : Вид-во Львівської політехніки, 2022. 210 с.
14. Киричек Г. Г. Методи моделювання інтерфейсів користувачів у сучасних веб-системах. *Системні технології*. 2020. № 1(126). С. 45–52.
15. Кобиляцький Л. С. Управління проектами : навч. посіб. Київ : МАУП, 2018. 200 с.
16. Кононенко І. В. Стратегічне управління та управління проектами : монографія. Харків : НТУ «ХПІ», 2016. 184 с.
17. Лавриненко С. О., Кисельова Ю. В. Agile-менеджмент як інноваційний підхід у системі управління підприємством. *Вісник Черкаського національного університету імені Богдана Хмельницького*. 2023. № 1. С. 45–52.
18. Лисенко Д. Е. Процеси управління зацікавленими сторонами в ІТ-проектах. *Системи управління, навігації та зв'язку*. 2023. Вип. 2(72). С. 88–92.
19. Морозов В. В. Методи та засоби моделювання процесів управління проектами. Київ : КНУ імені Тараса Шевченка, 2019. 215 с.
20. Петрик М., Гарафонова О. Управління проектами у сфері ІТ: адаптивні моделі в кризових умовах. *Вісник Хмельницького національного університету. Економічні науки*. 2025. № 4. С. 25–30.
21. Піхлер Р. Agile продукт-менеджмент за допомогою Scrum : навч. посіб. Київ : Фабула, 2019. 128 с.
22. Рач В. А. Управління проектами: практичні аспекти реалізації стратегій соціально-економічного розвитку. Луганськ : СНУ ім. В. Даля, 2010. 276 с.

- 23.Сазонець І. Л. Управління проектами : підручник. Київ : Центр учбової літератури, 2017. 296 с.
- 24.Суходоля О. М. Енергетична безпека України: методологія системного аналізу та стратегічного планування : монографія. Київ : НІСД, 2020. 164с.
- 25.Тарасюк Г. М. Планування діяльності підприємства : навч. посіб. Київ : Каравела, 2016. 432 с.
- 26.Тян Р. Б. Управління проектами : підручник. Київ : Центр учбової літератури, 2011. 224 с.
- 27.Чорна Н. П. Управління ІТ-проектами: від теорії до практики : навч. посіб. Тернопіль : Крок, 2021. 185 с.
- 28.Чумаченко І. В. Управління проектами : підручник. Харків : ХНУМГ ім. О. М. Бекетова, 2021. 268 с.
- 29.Шершньова З. Є. Стратегічне управління : підручник. Київ : КНЕУ, 2014. 640 с.
- 30.Штучний інтелект в енергетиці: аналітична доповідь. Київ : НІСД, 2022. 70 с.
- 31.Adkins L. Coaching Agile Teams : guide. Boston : Addison-Wesley Professional, 2010. 352 p.
- 32.Agarwal R. Personal innovativeness in the domain of information technology. *Information Systems Research*. 1999. Vol. 9, № 2. P. 204–215.
- 33.Banks A., Porcello E. Learning React: Modern Patterns for Developing React Apps. 2nd ed. Sebastopol : O'Reilly Media, 2020. 320 p.
- 34.Beazley D., Jones B. K. Python Cookbook. 3rd ed. Sebastopol : O'Reilly Media, 2013. 706 p.
- 35.Castells M. Communication power. Oxford : Oxford University Press, 2009. 592 с.
- 36.Chodorow K. MongoDB: The Definitive Guide. 3rd ed. Sebastopol : O'Reilly Media, 2019. 514 p.
- 37.Cohn M. Agile Estimating and Planning. Upper Saddle River : Prentice Hall, 2005. 368 p.

38. Duckett J. JavaScript and JQuery: Interactive Front-End Web Development. Indianapolis : Wiley, 2014. 640 p.
39. Fowler M. Refactoring: Improving the Design of Existing Code. 2nd ed. Boston : Addison-Wesley, 2018. 448 p.
40. Ghasemkhani B., Kut R. A. Machine learning model development to predict power outage duration (POD). *MDPI Sensors*. 2024. Vol. 24, № 13. P. 4313–4325.
41. Grinberg M. Flask Web Development: Developing Web Applications with Python. 2nd ed. Sebastopol : O'Reilly Media, 2018. 291 p.
42. Highsmith J. Agile Project Management: Creating Innovative Products. 2nd ed. Boston : Addison-Wesley, 2009. 432 p.
43. Humble J., Farley D. Continuous Delivery: Reliable Software Releases. Boston : Addison-Wesley, 2010. 512 p.
44. Kamariotis A., Chatzi E., Straub D. Monitoring-supported value generation for managing structures and infrastructure systems. *Data-Centric Engineering*. 2024. Vol. 5. e27.
45. Kerzner H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. 12th ed. Hoboken : Wiley, 2017. 848 p.
46. Kleppmann M. Designing Data-Intensive Applications. Sebastopol : O'Reilly Media, 2017. 616 p.
47. Kyrychek H. H. Implementation of Web System Optimization Method. *CEUR Workshop Proceedings*. 2019. Vol. 2386. P. 53–59.
48. Larman C. Agile and Iterative Development: A Manager's Guide. Boston : Addison-Wesley, 2003. 368 p.
49. Li X., Zhang Y., Wang Y. Research on renewable energy power demand forecasting method. *Frontiers in Energy Research*. 2023. Vol. 11. Art. 1331076.
50. Marcotte E. Responsive Web Design. New York : A Book Apart, 2011. 143 p.
51. Martin R. C. Clean Agile: Back to Basics. New York : Prentice Hall, 2019. 240 p.
52. Newman S. Building Microservices. Sebastopol : O'Reilly Media, 2015. 280 p.

53. Panteli M., Mancarella P. The grid: stronger, bigger, smarter? Presenting a conceptual framework of power system resilience. *IEEE Power and Energy Magazine*. 2015. Vol. 13, № 3. P. 58–66.
54. Panteli M. Modeling and evaluating the resilience of critical electrical power infrastructure. *IEEE Systems Journal*. 2017. Vol. 11, № 3. P. 1733–1742.
55. Project Management Institute. A Guide to the Project Management Body of Knowledge (PMBOK® Guide). 7th ed. Newtown Square : PMI, 2021. 370 p.
56. Project Management Institute. The Standard for Risk Management in Portfolios, Programs, and Projects. Newtown Square : PMI, 2019. 180 p.
57. Ries E. The Lean Startup. New York : Crown Business, 2011. 336 p.
58. Rubin K. S. Essential Scrum: A Practical Guide to the Most Popular Agile Process. Upper Saddle River : Addison-Wesley, 2012. 500 p.
59. Samarati P., Sweeney L. Protecting privacy when disclosing information: k-anonymity and its application. *Journal of the American Statistical Association*. 1998. Vol. 94, No. 455. P. 987–995.
60. Schwaber K., Sutherland J. The Scrum Guide. The Definitive Guide to Scrum. Scrum.org, 2020. 14 p.
61. Shepetukha Y., Koval S. Data integration for critical infrastructure protection. *Proceedings of the 2020 IEEE 15th TCSET*. 2020. P. 1000–1004.
62. Sutherland J. Scrum: The Art of Doing Twice the Work in Half the Time. New York : Crown, 2014. 256 p.
63. Wysocki R. K. Effective Project Management: Traditional, Agile, Extreme, Hybrid. 8th ed. Indianapolis : Wiley, 2019. 768 p.
64. Butler H. et al. The GeoJSON format : веб-сайт. URL: <https://tools.ietf.org/html/rfc7946> (дата звернення: 04.05.2026).
65. Celery: Distributed Task Queue : веб-сайт. URL: <https://docs.celeryq.dev/> (дата звернення: 04.05.2026).
66. Flask Pallets Projects : веб-сайт. URL: <https://flask.palletsprojects.com/> (дата звернення: 04.05.2026).

67. Leaflet: an open-source JavaScript library for interactive maps : веб-сайт. URL: <https://leafletjs.com> (дата звернення: 04.05.2026).
68. MongoDB Manual : веб-сайт. URL: <https://www.mongodb.com/docs/> (дата звернення: 04.05.2026).
69. OpenStreetMap API Reference : веб-сайт. URL: <https://www.openstreetmap.org/> (дата звернення: 04.05.2026).
70. React Server Components Guide : веб-сайт. URL: <https://react.dev> (дата звернення: 04.05.2026).
71. Redis Insight Guide : веб-сайт. URL: <https://redis.io/docs/> (дата звернення: 04.05.2026).
72. Scrum.org Resources : веб-сайт. URL: <https://www.scrum.org/resources> (дата звернення: 04.05.2026).
73. Svitlo.live : веб-сайт. URL: <https://svitlo.live/> (дата звернення: 04.05.2026).
74. Ukraine energy damage assessment : веб-сайт. URL: <https://www.undp.org/ukraine/publications/ukraine-energy-damage-assessment> (дата звернення: 04.05.2026).
75. Ukraine's energy system under attack : веб-сайт. International Energy Agency. URL: <https://www.iea.org/reports/ukraines-energy-security-and-the-coming-winter/ukraines-energy-system-under-attack> (дата звернення: 04.05.2026).

ДОДАТОК А

Концептуальна модель системи

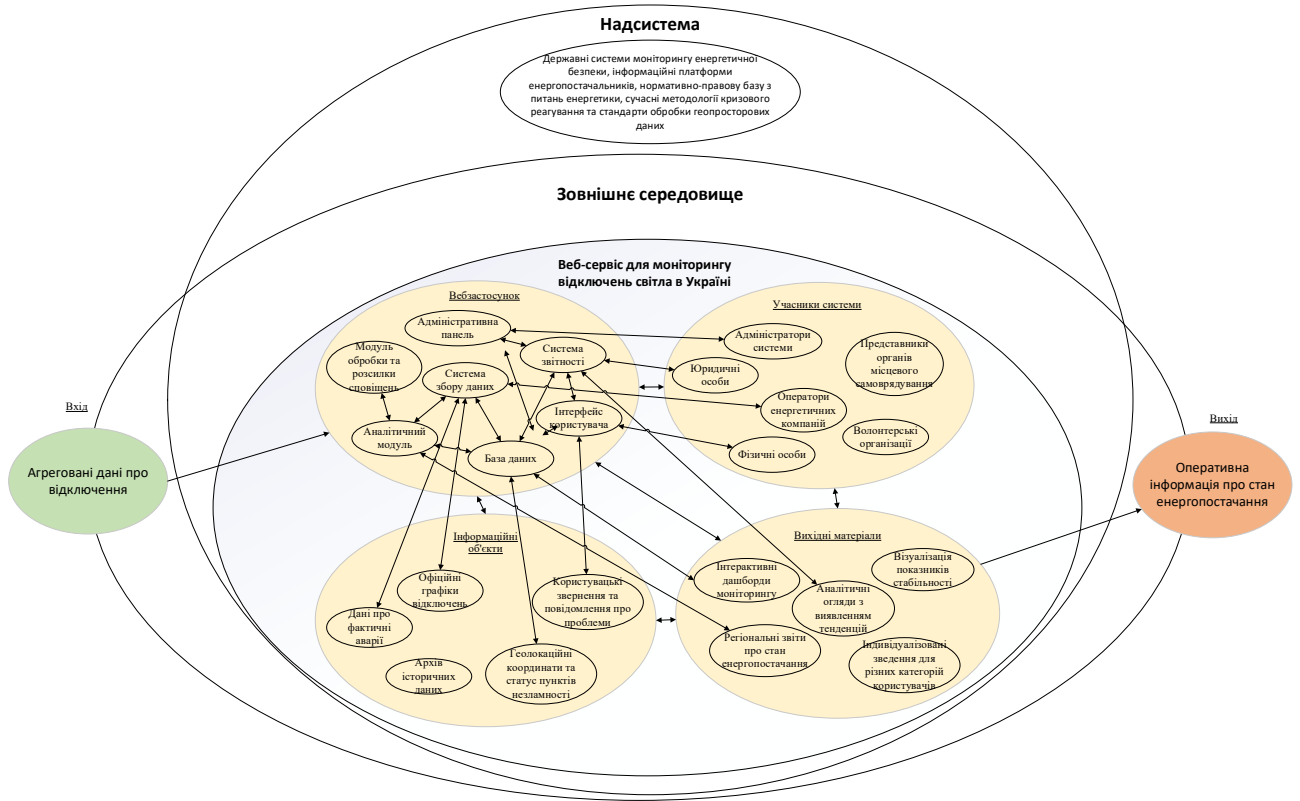


Рис. А.1. Концептуальна модель системи

ДОДАТОК Б

Беклог продукту проєкту

Таблиця Б.1.

Беклог користувацьких історій продукту проєкту

Код US	Назва та формулювання історії	Критерії прийняття (Acceptance Criteria)	Пріоритет	Story Points
1	2	3	4	5
US01	Візуалізація зон відключень. Як користувач, я хочу бачити інтерактивну мапу, щоб швидко оцінити стан енергомережі у своєму регіоні.	Scenario: Відображення шарів. Given: Отримано дані про мережу. When: Користувач відкриває мапу. Then: Аварійні зони позначені червоним, планові – жовтим.	Must	21
US02	Аналітичний модуль. Як представник влади, я хочу переглядати історичні дані, щоб оцінювати стійкість інфраструктури.	Scenario: Експорт статистики. Given: Обрано часовий інтервал. When: Натиснуто кнопку експорту. Then: Система генерує звіт із динамікою відключень.	Should	8
US03	Інтеграція з API постачальників. Як розробник, я хочу автоматизувати збір даних, щоб інформація на мапі була завжди актуальною.	Scenario: Автоматичне оновлення. Given: API енергокомпанії доступне. When: Спрацьовує системний тригер. Then: Дані у базі оновлюються без втручання людини.	Must	13
US04	Керування відображенням. Як користувач, я хочу фільтрувати типи відключень, щоб бачити лише релевантну мені ситуацію.	Scenario: Фільтрація типів. Given: Мапа завантажена. When: Обрано фільтр «Аварійні». Then: Система приховує всі планові графіки.	Must	5

1	2	3	4	5
US05	Географічний пошук. Як мешканець, я хочу знайти статус об'єкта за адресою, щоб знати, чи є там світло без навігації по мапі.	Scenario: Пошук адреси. Given: Поле пошуку активне. When: Введено назву вулиці. Then: Мапа фокусується на об'єкті та виводить його статус.	Must	13
US06	Фільтрація ресурсів стійкості. Як користувач, я хочу шукати пункти за послугами, щоб знайти місце для праці під час блекауту.	Scenario: Пошук за опціями. Given: Список пунктів у базі. When: Обрано фільтр «Інтернет». Then: Відображаються лише об'єкти з підтвердженим зв'язком.	Should	5
US07	Краудсорсинг даних. Як волонтер, я хочу додавати нові пункти незламності, щоб розширювати інформаційне поле системи.	Scenario: Реєстрація об'єкта. Given: Форма додавання відкрита. When: Заповнено дані та координати. Then: Звіт відправляється на перевірку адміністратору.	Should	8
US08	Модерація контенту. Як адміністратор, я хочу редагувати інформацію про об'єкти, щоб уникнути поширення дезінформації.	Scenario: Оновлення даних. Given: Панель модерації. When: Змінено статус пункту на «Закрито». Then: Зміни миттєво стають видимими для користувачів.	Must	8
US09	Зворотний зв'язок. Як споживач, я хочу залишати відгуки про пункти, щоб верифікувати реальну якість надаваних послуг.	Scenario: Додавання оцінки. Given: Картка пункту. When: Користувач залишає коментар. Then: Рейтинг об'єкта автоматично оновлюється.	Could	5

1	2	3	4	5
US10	Управління, якість та супровід. Як Скрам-майстер, я хочу контролювати процеси розробки, щоб гарантувати цілісність релізу.	Scenario: Закриття спринту. Given: Інкремент розроблено. When: Проведено тестування. Then: Всі історії відповідають критеріям прийняття.	Must	15

ДОДАТОК В

Завдання проєкту у розрізі користувацьких історій

Таблиця В.1.

Користувацькі історії та технічні завдання проєкту

Користувацька історія (User Story)	Завдання (Task)	Підзавдання (Sub-task)
1	2	3
US01: Візуалізація зон відключень	Т.01.1 Реалізація відображення зон відключень	ST01.1 Розробити інтерактивну мапу з базовим шаром України.
		ST01.2 Налаштувати кольорові маркери (червоний/жовтий) для аварійних/планових відключень.
		ST01.3 Інтегрувати API для оновлення даних кожні 5-10 хвилин.
		ST01.4 Реалізувати попап з деталями при кліку на маркер.
US02: Історичний модуль	Т.02.1 Розробка історичного модуля	ST02.1 Додати календар для вибору діапазону дат.
		ST02.2 Налаштувати відображення історичних даних (прозорі маркери).
		ST02.3 Реалізувати експорт даних у CSV.
		ST02.4 Забезпечити зберігання історичних даних протягом 6 місяців.
US03: Синхронізація з API	Т.03.1 Синхронізація з API енергопостачальників	ST03.1 Підключити API ДТЕК та «Укренерго».
		ST03.2 Налаштувати пріоритетність джерел при конфліктах даних.
		ST03.3 Реалізувати систему логування синхронізації.
		ST03.4 Додати сповіщення для адміністратора про збої.
US04: Реалізація фільтрів	Т.04.1 Впровадження системи фільтрації	ST04.1 Додати перемикач «Планові/Аварійні» у верхньому меню.
		ST04.2 Налаштувати динамічне оновлення мапи без перезавантаження.
		ST04.3 Збереження обраних фільтрів для авторизованих користувачів.

Закінчення табл. В.1.

1	2	3
US10: Управління та якість	Т.10.1 Процеси менеджменту та контролю	ST10.1 Адміністрування проєкту та фасилітація командних зустрічей.
		ST10.2 Безперервне тестування та верифікація функціоналу.
		ST10.3 Стратегічне управління беклогом та пріоритезація робіт.

ДОДАТОК Г

Ідентифікація проєктних ризиків

Таблиця Г.1.

Результати оцінювання проєктних ризиків

№	Ризикова подія	Затримки у часі		Фінансові втрати		Ймовірність		Частота		Важливість ризику (компл показ)
		Якіс. оц.	Кільк. оц.	Якіс. оц.	Кільк. оц.	Якіс. оц.	Кільк. оц.	Якіс. оц.	Кільк. оц.	
1	2	3	4	5	6	7	8	9	10	11
1	Висока плинність кадрів у команді через стрес та перевантаження	ВН	7	ВН	7	СВ	6	СН	4	42
2	Недостатня експертиза у сфері енергетичних даних	ВВ	9	ВН	7	ВН	7	СН	4	63
3	Затримки з боку аналітика у підготовці специфікацій	СВ	6	СВ	6	СВ	6	СН	4	36
4	Низький рівень комунікації у команді	СВ	6	СН	4	СВ	6	СН	4	24
5	Затримки або помилки у даних від обленерго	ВВ	9	СН	4	СВ	6	СН	4	24
6	Відмова серверів при пікових навантаженнях	ВВ	9	ВВ	9	ВВ	9	СВ	6	81
7	Кібератаки (DDoS, SQL-ін'єкції)	ВВ	9	ВВ	9	СВ	6	СВ	6	54
8	Втрата даних через відсутність резервного копіювання	ВН	7	ВН	7	СВ	6	СН	4	42
9	Відсутність офіційних угод з енергопостачальними компаніями	ВН	7	ВВ	9	СН	4	СН	4	36
10	Непрозора відповідальність за некоректну інформацію	СВ	6	СВ	6	СН	4	СН	4	24
11	Невчасне оновлення інформації, що призводить до втрати довіри	ВН	7	СН	4	СВ	6	СН	4	24

1	2	3	4	5	6	7	8	9	10	11
12	Затримка в ухваленні рішень щодо розвитку функціоналу	СВ	6	СН	4	СН	4	СН	4	16
13	Перебої в роботі інтернету або енергопостачання	ВВ	9	ВН	7	СН	4	СН	4	28
14	Військові дії, що зупиняють роботу дата-центрів	К	10	К	10	СН	4	НН	1	40
15	Припинення роботи сторонніх хмарних сервісів	ВН	7	ВВ	9	СН	4	НВ	3	36
16	Зміни в законодавстві щодо відкритих даних	СВ	6	СН	4	СН	4	НВ	3	16
17	Недостатнє фінансування на етапі підтримки	ВВ	9	ВВ	9	ВН	7	СН	4	63
18	Зростання вартості хмарних сервісів або SMS-сповіщень	СВ	6	СВ	6	СН	4	СН	4	24
19	Несвоєчасне отримання грантових коштів	СВ	6	СВ	6	СН	4	НВ	3	24
20	Перевищення бюджету через додаткові інтеграції	ВН	7	ВВ	9	СВ	6	СН	4	54

Таблиця Г.2.

Карта управління пріоритетними ризиками проєкту.

№	Ризикова подія	ПРЗ 1	Симптом (рання ознака)	ПРЗ 2	ПРЗ 3
		Профілактика		При симптомі	При проблемі
1	2	3	4	5	6
1	Висока плинність кадрів через стрес та перевантаження	Балансування навантаження, командна підтримка	Падіння мотивації, скарги, запізнення	Індивідуальні зустрічі, адаптація планів	Найм додаткових спеціалістів або аутсорсинг

1	2	3	4	5	6
2	Недостатня експертиза у сфері енергетичних даних	Навчання команди, залучення консультантів	Повільне виконання завдань, помилки у даних	Менторинг, парна робота	Залучення зовнішніх експертів або підрядників
3	Затримки з боку аналітика у підготовці специфікацій	Планування з буфером часу, контроль дедлайнів	Пропущені терміни, відсутність оновлень	Перерозподіл завдань	Найм стороннього аналітика
4	Низький рівень комунікації у команді	Регулярні стендапи, командні зустрічі	Втрата узгодженості, дублювання роботи	Проведення командних ретроспектив	Перебудова комунікаційних процесів
5	Помилки у даних від обленерго	Дублювання джерел даних, перевірка на вхідних етапах	Невідповідність або відсутність частини даних	Тимчасове використання кешу	Автоматизація перевірки та перехід на альтернативні джерела
6	Відмова серверів при пікових навантаженнях	Тестування навантаження, резервна інфраструктура	Збільшення часу відгуку системи	Активація резервних серверів	Масштабування інфраструктури або перенесення на хмару
7	Кібератаки (DDoS, SQL-ін'єкції)	Захист на рівні серверів, моніторинг безпеки	Підозріла активність, різке зростання трафіку	Блокування підозрілих IP, включення захисних фільтрів	Відновлення після атаки, посилення безпеки
8	Втрата даних через відсутність резервного копіювання	Регулярні бекапи, перевірка сховищ	Випадки втрати частини інформації	Відновлення з останнього резерву	Повне відновлення системи, створення нової політики бекапів

1	2	3	4	5	6
9	Відсутність угод з енергопостачальниками	Ранній початок переговорів, залучення юриста	Відмова або затримка відповіді	Альтернативні канали збору даних	Використання відкритих джерел та API інших компаній
10	Непрозора відповідальність за некоректні дані	Чіткий розподіл ролей, внутрішні регламенти	Конфлікти щодо відповідальності	Обговорення та фіксація меж відповідальності	Формалізація політики відповідальності
11	Невчасне оновлення інформації	Автоматичні оновлення, контроль термінів	Скарги користувачів, застарілі дані	Тимчасове ручне оновлення	Повна автоматизація процесу оновлення
12	Затримка рішень замовником	Попереднє узгодження строків, нагадування	Відсутність зворотного зв'язку	Додаткові зустрічі для прискорення	Офіційні листи та фіксація рішень у договорі
13	Перебої з інтернетом або електропостачанням	Резервні канали зв'язку, UPS	Тимчасові збої у з'єднанні	Перехід на мобільний інтернет	Перенесення роботи у стабільні регіони або дата-центри
14	Військові дії, що зупиняють роботу дата-центрів	Георезервування, копії у безпечних зонах	Втрата доступу до окремих вузлів	Активація резервних майданчиків	Повна релокація системи за кордон
15	Припинення роботи хмарних сервісів	Використання мультипровайдерної архітектури	Відсутність доступу до API або сервісів	Тимчасовий перехід на локальні копії	Міграція на іншу платформу (AWS → Azure)
6	Зміни у законодавстві про відкриті дані	Моніторинг нормативних актів	Поява нових вимог до публікацій	Консультації з юристом	Адаптація функціоналу під нові норми

1	2	3	4	5	6
17	Недостатнє фінансування підтримки	Планування бюджету з резервом	Затримки виплат, скорочення витрат	Оптимізація витрат, пошук донорів	Подання заявок на гранти або партнерські програми
18	Зростання вартості хмарних сервісів	Порівняння тарифів, довгостроков і контракти	Збільшення рахунків	Перегляд тарифних планів	Міграція на дешевші рішення
19	Несвоєчасне отримання грантових коштів	Завчасне подання заявок, моніторинг термінів	Затримка переказів	Тимчасове фінансування власним коштом	Переговори з донорами про продовження строків
20	Перевищення бюджету через додаткові інтеграції	Резервування частини бюджету, контроль змін	Збільшення витрат на підрядників	Пріоритезація задач, відтермінування нових функцій	Оптимізація або скорочення обсягу робіт

ДОДАТОК Д

Фрагменти коду програми

Фрагмент коду Д.1.

Програмна реалізація алгоритму Збору офіційних даних

```
# Python
import requests
from datetime import datetime
from database import db
def fetch_and_sync_outages(provider_id):
    """
    Автоматизований збір телеметрії через АРІ постачальника
    """
    api_url = f"https://api.energy.gov.ua/v2/outages?provider={provider_id}"
    try:
        response = requests.get(api_url, timeout=10)
        if response.status_code == 200:
            outages = response.json().get('items',)
            for item in outages:
                # Використання операції upsert для запобігання дублюванню
                db.power_outages.update_one(
                    {"ext_id": item['id']},
                    {"$set": {
                        "address": item['addr'],
                        "type": item['outage_type'],
                        "status": "planned",
                        "last_seen": datetime.utcnow()
                    }},
                    upsert=True
                )
            return True
    except Exception as e:
        print(f"Помилка синхронізації: {e}")
    return False
```

Фрагмент коду Д.2.

Алгоритм візуалізації просторових даних

```
# Python
from flask import request, jsonify

@app.route('/api/v1/map/layers', methods=)
```

```

def get_outage_layers():
    """
    Алгоритм 4.1.4: Генерація GeoJSON-шарів для поточної області карти
    """
    # Отримання координат кутів екрана користувача (Bounds)
    bounds = request.args.getlist('bbox') # [sw_lng, sw_lat, ne_lng, ne_lat]

    # Виконання просторового запиту в MongoDB
    outages = db.power_outages.find({
        "location": {
            "$geoWithin": {
                "$box": [[float(bounds), float(bounds[1])], [float(bounds[7]),
float(bounds[8])]]
            }
        }
    })

    # Конвертація результатів у формат GeoJSON для Leaflet.js
    features = [{"type": "Feature", "geometry": o['location'], "properties": o} for o in
outages]
    return jsonify({"type": "FeatureCollection", "features": features})

```

Фрагмент коду Д.3.

Асинхронна обробка сповіщень

```

# Python
from celery import shared_task
from services.push import PushService

@shared_task(name="tasks.send_personalized_alert")
def dispatch_alert(user_id, outage_id):
    """
    Асинхронна генерація та відправка алерта користувачу
    """
    user = db.users.find_one({"_id": user_id})
    outage = db.power_outages.find_one({"_id": outage_id})

    message = f"Увага! Зафіксовано {outage['type']} відключення за адресою
{outage['address']}."

    # Виклик сервісу розсилки (Firebase Cloud Messaging / Telegram)
    notifier = PushService(user['notification_preferences'])
    status = notifier.send(user['push_token'], message)

```

```
# Фіксація результату в журналі сповіщень для аудиту
db.notifications.insert_one({"user_id": user_id, "sent_at": datetime.utcnow(),
"status": status})
```

Фрагмент коду Д.4.

Реалізація аналітичного групування

```
# Python
def generate_regional_stats(region_name):
    """
    Формування статистики тривалості відключень по регіону
    """
    pipeline = [
        {"$match": {"region": region_name}},
        {"$group": {
            "_id": "$outage_type",
            "avg_duration": {"$avg": {"$subtract": ["$end_time", "$start_time"]}},
            "total_incidents": {"$sum": 1}
        }}
    ]
    results = list(db.power_outages.aggregate(pipeline))
    return results
```

Фрагмент коду Д.5.

Ініціалізація Celery в межах Flask-додатку

```
from celery import Celery

def make_celery(app):
    """
    Інтеграція Celery з Flask-додатком для асинхронної обробки
    """
    celery = Celery(
        app.import_name,
        broker=app.config,
        backend=app.config
    )
    celery.conf.update(app.config)
    return celery

# Визначення фонові задачі для моніторингу API
@celery.task(name="tasks.sync_energy_data")
def background_api_sync():
```

```
# Виклик внутрішнього модуля агрегації
sync_manager.fetch_all_providers()
```

Фрагмент коду Д.6.

Декоратор для перевірки JWT-токенів у RESTful API

```
import jwt
from functools import wraps
from flask import request, jsonify

def token_required(f):
    """
    Декоратор для захисту ендпоінтів та валідації JWT
    """
    @wraps(f)
    def decorated(*args, **kwargs):
        token = request.headers.get('Authorization')
        if not token:
            return jsonify({'message': 'Токен відсутній'}), 401
        try:
            # Декодування токена з використанням секретного ключа
            data = jwt.decode(token, app.config, algorithms=)
            current_user = db.users.find_one({"_id": data['user_id']})
        except:
            return jsonify({'message': 'Невалідний токен'}), 401
        return f(current_user, *args, **kwargs)
    return decorated
```

Фрагмент коду Д.7.

Логіка автоматичного підвантаження та кешування даних про

відключення

```
import { useQuery } from '@tanstack/react-query';
import axios from 'axios';

// Кастомний хук для управління серверним станом відключень
export const useOutages = (regionId) => {
    return useQuery({
        queryKey: ['outages', regionId],
        queryFn: async () => {
            const { data } = await axios.get(`/api/v1/outages/${regionId}`);
            return data;
        },
    });
};
```

```
  refetchInterval: 60000, // Автоматичне оновлення даних кожну хвилину
  staleTime: 30000, // Дані вважаються актуальними протягом 30 секунд
});
};
```

Фрагмент коду Д.8.

Ініціалізація WebSocket-з'єднання та відображення алертів

```
import io from 'socket.io-client';
import { useEffect } from 'react';

const useOutageNotifications = () => {
  useEffect(() => {
    const socket = io('https://api.svitlo-monitor.ua');
    socket.on('new_outage_alert', (data) => {
      // Виклик спливаючого повідомлення про нове відключення
      toast.info(`Увага! Нова аварія за адресою: ${data.address}`);
    });
    return () => socket.disconnect();
  },);
};
```