

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

**ПОГОДЖЕНО
Декан факультету**

**ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри інформаційних
систем і технологій**

_____ **Ігор БОЛБОТ**
(підпис)

_____ **Михайло ШВИДЕНКО**
(підпис)

«___» _____ 2026 р.

«___» _____ 2026 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

**на тему: «Інтелектуальна система моніторингу та прогнозування технічного
стану обладнання на основі IoT і машинного навчання»**

Спеціальність «126 Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні системи та технології»

Гарант освітньої програми
к.е.н., доцент

_____ **Максим МОКРІЄВ**
(підпис)

**Керівник бакалаврської
кваліфікаційної роботи**
к.е.н., доцент

_____ **Євгеній СТАРИЧЕНКО**
(підпис)

Виконав

_____ **Владислав ХАРІЧЕВ**
(підпис)

Київ-2026

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри інформаційних систем і
технологій

к.е.н., доцент _____ Михайло ШВИДЕНКО
(підпис)

«___» _____ 2026 р.

ЗАВДАННЯ
ДО ВИКОНАННЯ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
ЗДОБУВАЧУ

Харічеву Владиславу Яновичу

Спеціальність 126 – Інформаційні системи та технології

ОП «Інформаційні системи та технології»

Тема бакалаврської кваліфікаційної роботи «Інтелектуальна система моніторингу та прогнозування технічного стану обладнання на основі IoT і машинного навчання» затверджена наказом від «19» грудня 2025 р. № 3205 «С»

Термін подання завершеної роботи на кафедру «22» травня 2026 р.

Вихідні дані до бакалаврської кваліфікаційної роботи (проєкту):

Телеметричні дані промислового обладнання, параметри роботи IoT-датчиків, інформація про технічний стан обладнання та журнали подій; дані про температуру, вібрацію, тиск та інші експлуатаційні показники; алгоритми машинного навчання для аналізу аномалій і прогнозування відмов; засоби інтеграції з IoT-пристроями, MQTT/REST API та системами ERP; інструменти візуалізації та аналітики телеметричних даних.

Перелік питань, що підлягають дослідженню:

1. Аналіз предметної області моніторингу технічного стану промислового обладнання та технологій predictive maintenance.
2. Дослідження сучасних IoT-платформ і інформаційних систем моніторингу обладнання.
3. Проєктування інформаційного та програмного забезпечення інтелектуальної системи моніторингу технічного стану обладнання.

4. Розробка механізмів збору, збереження та оброблення телеметричних даних від IoT-пристроїв.
5. Реалізація алгоритмів машинного навчання для виявлення аномалій і прогнозування можливих відмов обладнання.
6. Розробка засобів візуалізації, аналітики та формування сповіщень про критичні стани обладнання.
7. Тестування функціональних можливостей системи та оцінювання ефективності прогнозування технічного стану обладнання.

Дата видачі завдання «19» грудня 2025 р.

**Керівник бакалаврської
кваліфікаційної роботи**

(підпис)

Євгеній СТАРИЧЕНКО

**Завдання взяв до
виконання**

(підпис)

Владислав ХАРІЧЕВ

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	4
ВСТУП	6
1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Опис предметної області	9
1.2 Огляд інформаційних джерел та існуючих рішень	12
1.3 Аналіз вимог програмної системи.....	18
1.4 Моделювання предметної області.....	21
1.5 Постановка завдання.....	25
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	29
2.1 Логічна модель даних інтелектуальної системи моніторингу та прогнозування технічного стану обладнання	29
2.2 Проектування класової структури та коопераційних сценаріїв інтелектуальної системи.....	33
2.3 Компонентна модель програмної архітектури системи.....	39
2.4 Пакетна структуризація програмного забезпечення системи	41
2.5 Висновки до другого розділу	44
3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ....	46
3.1 Вибір технологій та інструментальних засобів реалізації	46
3.2 Архітектура та розгортання інтелектуальної системи моніторингу обладнання	49
3.3 Фізична структура бази даних системи моніторингу обладнання.....	53
3.4 Алгоритмізація програмних модулів інтелектуальної системи	55
3.5 Висновки до третього розділу.....	60
4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ.....	62
4.1 Тестування програмних модулів інтелектуальної системи моніторингу технічного стану обладнання.....	62
4.2 Тестування інтерфейсних модулів та основних функцій системи	65
4.3 Розгортання системи, склад інсталяційного пакету	72
4.4 Висновки до четвертого розділу.....	76
ВИСНОВКИ.....	78
СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	80
ДОДАТОК А.....	83

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API – Application Programming Interface, програмний інтерфейс взаємодії між модулями системи.

БД – база даних.

CSV – Comma-Separated Values, формат табличного експорту даних.

CMMS – Computerized Maintenance Management System, комп'ютеризована система управління технічним обслуговуванням.

Docker – платформа контейнеризації програмних компонентів.

ERP – Enterprise Resource Planning, система планування ресурсів підприємства.

Git – система контролю версій програмного коду.

HTML – HyperText Markup Language, мова розмітки вебсторінок.

HTTP – HyperText Transfer Protocol, протокол передавання даних у вебсередовищі.

HTTPS – захищена версія протоколу HTTP.

IoT – Internet of Things, Інтернет речей.

IIoT – Industrial Internet of Things, промисловий Інтернет речей.

JSON – JavaScript Object Notation, формат обміну структурованими даними.

JWT – JSON Web Token, токен авторизації користувача.

ML – Machine Learning, машинне навчання.

MQTT – Message Queuing Telemetry Transport, протокол передавання телеметричних повідомлень.

OpenAPI – стандарт опису програмних REST API.

PLC – Programmable Logic Controller, програмований логічний контролер.

PostgreSQL – реляційна система керування базами даних.

REST – Representational State Transfer, архітектурний стиль побудови веб API.

REST API – програмний інтерфейс для обміну даними між клієнтською та серверною частинами системи.

RUL – Remaining Useful Life, залишковий ресурс роботи обладнання.

SCADA – Supervisory Control and Data Acquisition, система диспетчерського керування та збору даних.

SQL – Structured Query Language, мова структурованих запитів до бази даних.

TimescaleDB – розширення PostgreSQL для зберігання та оброблення часових рядів.

UI – User Interface, інтерфейс користувача.

UML – Unified Modeling Language, уніфікована мова моделювання.

Web UI – вебінтерфейс користувача.

APM – автоматизоване робоче місце.

ТО – технічне обслуговування.

IoT-шлюз – проміжний вузол для приймання, буферизації та передавання даних від датчиків до серверної частини системи.

ML-модель – програмна модель машинного навчання, що використовується для аналізу телеметричних даних і прогнозування технічного стану обладнання.

ВСТУП

Надійність та безперервність роботи технологічного обладнання є одним із ключових факторів ефективності виробництва. Відмови технічних систем призводять до зупинок виробничих процесів, зниження продуктивності підприємства та збільшення витрат на ремонт і обслуговування обладнання. Традиційні підходи до технічного обслуговування, які базуються на регламентних перевірках або періодичних оглядах обладнання, не забезпечують своєчасного виявлення потенційних несправностей і не дозволяють оперативно реагувати на зміну технічного стану машин та механізмів.

Розвиток технологій Інтернету речей (IoT) дозволяє реалізувати безперервний моніторинг параметрів роботи обладнання за допомогою мережі промислових датчиків. Такі датчики забезпечують збір телеметричних даних про ключові експлуатаційні параметри обладнання, зокрема температуру, вібрацію, навантаження, швидкість обертання та інші характеристики функціонування технічних систем. Отримані дані передаються через промислові комунікаційні мережі до інформаційної системи, де виконуються їх обробка, зберігання та подальший аналіз.

Застосування методів машинного навчання у системах моніторингу дозволяє автоматизувати процес аналізу великих обсягів телеметричних даних та виявляти закономірності у роботі обладнання. Алгоритми аналізу даних забезпечують виявлення аномальних режимів роботи технічних систем, оцінювання технічного стану обладнання та прогнозування ймовірності виникнення відмов. Використання таких підходів формує основу систем прогнозного технічного обслуговування (predictive maintenance), що дозволяє своєчасно планувати ремонтні роботи та зменшувати ризик аварійних зупинок виробництва.

Актуальність теми зумовлена необхідністю створення інтелектуальної інформаційної системи, яка забезпечуватиме автоматизований моніторинг

технічного стану обладнання, аналіз телеметричних даних та прогнозування можливих відмов на основі методів машинного навчання. Використання такої системи дозволить підвищити надійність роботи виробничого обладнання, зменшити кількість аварійних зупинок, оптимізувати витрати на технічне обслуговування та підвищити ефективність управління виробничими процесами.

Метою дослідження є удосконалення системи моніторингу та прогнозування технічного стану обладнання шляхом розробки програмного забезпечення на основі технологій IoT та машинного навчання, яка забезпечуватиме автоматизований збір телеметричних даних, їх аналіз, виявлення аномалій у роботі обладнання та прогнозування можливих відмов.

Об'єктом дослідження є процеси моніторингу технічного стану промислового обладнання та організації технічного обслуговування виробничих систем.

Предметом дослідження виступають методи та засоби проєктування інформаційних систем для збору, обробки та аналізу телеметричних даних обладнання із використанням технологій Інтернету речей та машинного навчання.

Для досягнення поставленої мети у роботі необхідно вирішити такі **завдання**:

1. Провести аналіз предметної області моніторингу технічного стану промислового обладнання та визначити основні процеси, що підлягають автоматизації.
2. Виконати аналіз сучасних інформаційних систем моніторингу обладнання та технологій predictive maintenance.
3. Сформулювати функціональні та нефункціональні вимоги до інтелектуальної системи моніторингу технічного стану обладнання.
4. Провести моделювання предметної області із використанням сучасних методів проєктування інформаційних систем.
5. Розробити структуру бази даних для збереження телеметричних даних та інформації про стан обладнання.

6. Реалізувати програмне забезпечення інтелектуальної системи моніторингу та прогнозування технічного стану обладнання.

7. Провести тестування розробленої системи та оцінити ефективність її використання для задач прогнозування відмов обладнання.

Практична значущість роботи полягає у створенні програмного рішення, яке може бути використане для автоматизованого контролю технічного стану обладнання на підприємствах різних галузей промисловості. Запропонована система дозволить здійснювати безперервний моніторинг параметрів роботи машин, виявляти потенційні несправності на ранніх етапах їх виникнення, прогнозувати можливі відмови обладнання та формувати рекомендації щодо проведення технічного обслуговування. Використання такої системи сприятиме підвищенню надійності експлуатації технічних систем, зменшенню витрат на ремонт та оптимізації виробничих процесів підприємства.

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

Традиційні підходи до технічного обслуговування обладнання базуються на планово-попереджувальних перевірках або періодичних регламентних оглядах. Такий підхід не забезпечує безперервного контролю параметрів роботи технічних систем і не дозволяє оперативно виявляти потенційні відхилення у функціонуванні обладнання. У результаті несправності часто виявляються вже після виникнення критичних збоїв, що значно ускладнює процес технічного обслуговування та підвищує витрати на ремонт.

Розвиток технологій Інтернету речей (IoT) дозволяє реалізувати безперервний контроль технічного стану обладнання за допомогою мережі промислових датчиків та пристроїв збору даних. IoT-датчики забезпечують реєстрацію технологічних параметрів роботи обладнання, таких як температура, вібрація, тиск, швидкість обертання, електричні параметри та інші експлуатаційні характеристики. Отримані телеметричні дані передаються через промислові комунікаційні канали до інформаційної системи, де виконуються їх зберігання, обробка та подальший аналіз.

Для аналізу великих обсягів телеметричних даних доцільно використовувати методи машинного навчання, які дозволяють автоматично виявляти закономірності у роботі технічних систем. Алгоритми аналізу даних забезпечують визначення аномальних режимів роботи обладнання, оцінювання технічного стану машин та прогнозування можливих відмов. Використання таких підходів формує основу систем прогнозного технічного обслуговування (predictive maintenance), що дозволяє перейти від реактивного ремонту до попереджувального управління технічним станом обладнання.

У межах предметної області інтелектуальної системи моніторингу та прогнозування технічного стану обладнання виділяються основні учасники

взаємодії, до яких належать оператор або технік, інженер з технічного обслуговування, адміністратор системи та аналітик даних. Оператор або технік здійснює контроль поточного стану обладнання та використовує систему для перегляду показників роботи машин. Інженер з технічного обслуговування аналізує технічний стан обладнання та приймає рішення щодо проведення ремонтних або профілактичних робіт. Адміністратор виконує функції налаштування та управління системою, а аналітик даних відповідає за розробку та навчання моделей машинного навчання для прогнозування відмов.

На рисунку 1.1 представлено контекстну модель взаємодії інтелектуальної системи моніторингу технічного стану обладнання з основними джерелами даних та зовнішніми інформаційними системами.

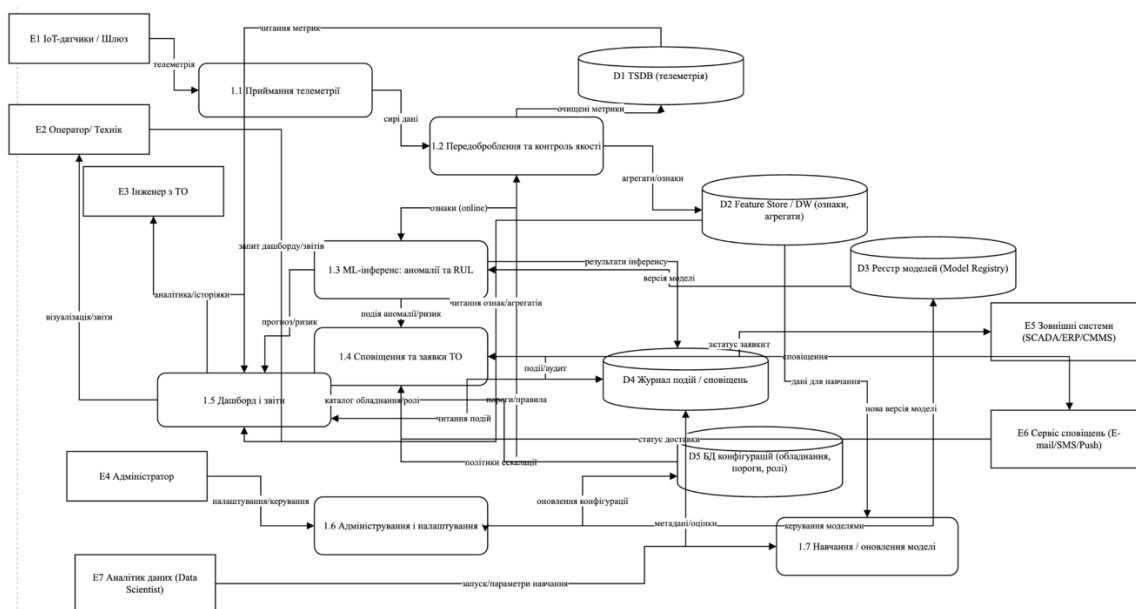


Рис. 1.1 – DFD-діаграма нульового рівня інтелектуальної системи моніторингу та прогнозування технічного стану обладнання

Як видно з рисунка 1.1, інформаційна система виступає центральним елементом обробки телеметричних даних, що надходять від мережі IoT-датчиків та пристроїв збору інформації. Система забезпечує приймання телеметрії, передоброблення та контроль якості даних, формування ознак для подальшого аналізу, а також виконання алгоритмів машинного навчання для виявлення аномалій і прогнозування залишкового ресурсу роботи обладнання. Результати

аналізу використовуються для формування сповіщень про потенційні несправності та створення заявок на проведення технічного обслуговування.

Для забезпечення ефективного функціонування системи доцільно виділити основні процеси предметної області, які підлягають автоматизації. До таких процесів належать збір телеметричних даних, їх передоброблення та зберігання, аналітична обробка інформації із застосуванням моделей машинного навчання, формування сповіщень про відхилення у роботі обладнання, а також візуалізація результатів аналізу у вигляді аналітичних панелей та звітів.

Основні процеси предметної області інтелектуальної системи моніторингу технічного стану обладнання наведено у таблиці 1.1.

Таблиця 1.1

Основні процеси системи моніторингу технічного стану обладнання

№	Процес	Опис
1	Збір телеметричних даних	Отримання даних від IoT-датчиків та пристроїв збору інформації
2	Передоброблення даних	Фільтрація, нормалізація та контроль якості телеметричних даних
3	Зберігання телеметрії	Збереження даних у базах часових рядів для подальшого аналізу
4	Формування ознак	Обчислення агрегованих параметрів та характеристик роботи обладнання
5	Виявлення аномалій	Аналіз даних для визначення відхилень від нормального режиму роботи
6	Прогнозування відмов	Оцінювання ймовірності виникнення несправностей та залишкового ресурсу обладнання
7	Формування сповіщень	Автоматичне створення повідомлень про можливі відмови обладнання
8	Аналітика та звітність	Візуалізація даних моніторингу у вигляді дашбордів і аналітичних звітів

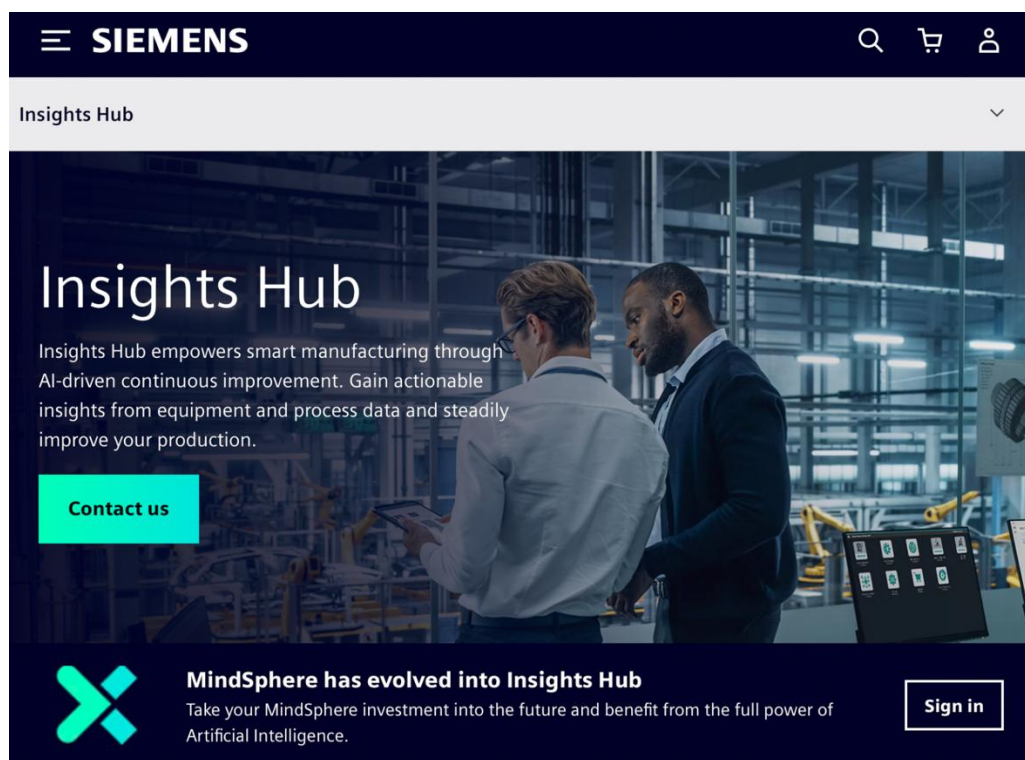
Автоматизація зазначених процесів дозволяє забезпечити безперервний контроль технічного стану обладнання, своєчасно виявляти потенційні несправності та підвищувати ефективність управління виробничими системами. Використання інтелектуальної системи моніторингу сприяє зменшенню кількості аварійних відмов обладнання, оптимізації витрат на технічне обслуговування та підвищенню надійності експлуатації виробничих установок.

1.2 Огляд інформаційних джерел та існуючих рішень

Розвиток цифрових технологій у промисловості сприяв появі великої кількості програмних платформ, призначених для моніторингу технічного стану обладнання, збору телеметричних даних та прогнозування відмов виробничих систем. Такі рішення використовують технології Industrial Internet of Things (IIoT), аналітику великих даних та алгоритми машинного навчання для забезпечення безперервного контролю параметрів роботи обладнання. Використання спеціалізованих інформаційних платформ дозволяє автоматизувати процеси збору телеметрії, аналізу експлуатаційних характеристик машин та формування рекомендацій щодо технічного обслуговування обладнання.

На сучасному ринку програмного забезпечення існує значна кількість платформ для реалізації систем predictive maintenance та промислового моніторингу. Найбільш відомими серед них є Siemens MindSphere (Insights Hub), IBM Maximo Application Suite, GE Digital Predix, Microsoft Azure IoT та PTC ThingWorx. Зазначені рішення відрізняються архітектурою системи, функціональними можливостями, способами інтеграції з обладнанням та методами обробки даних.

На рисунку 1.2 представлено інтерфейс платформи Siemens MindSphere (Insights Hub).



Why Insights Hub?

Deliver business value by adopting a continuous improvement process and achieve higher equipment availability, production performance, quality rate and sustainability, while optimizing your operating costs.



Explore data

Connect your assets to the cloud, collect and explore your data, and strategically develop your IoT capabilities.



Discover insights

Unlock a world of previously unattainable insights: use intelligent analytic tools to better understand and improve processes.



Рис. 1.2 – Інтерфейс платформи Siemens Insights Hub

Дана система є хмарною платформою Industrial IoT, яка забезпечує підключення промислового обладнання до цифрової інфраструктури підприємства. Платформа підтримує збір телеметричних даних з різних типів промислових пристроїв, їх обробку та аналітику у режимі реального часу. Основними функціями системи є моніторинг стану обладнання, аналіз продуктивності виробничих процесів та виявлення відхилень у роботі технічних систем.

На рисунку 1.3 показано приклад роботи системи IBM Maximo Application Suite.

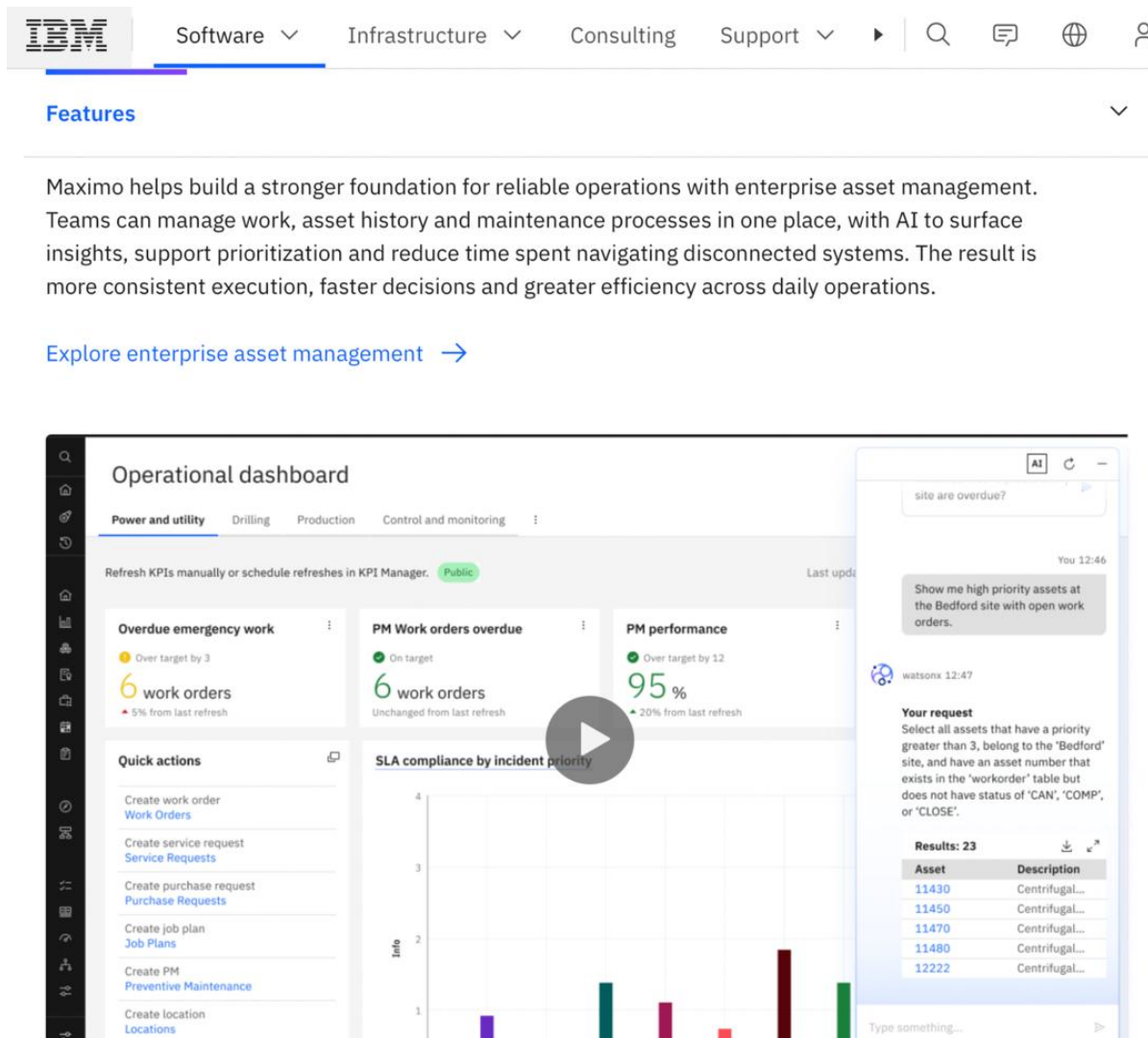


Рис. 1.3 – Інтерфейс системи IBM Maximo

IBM Maximo є комплексною платформою управління виробничими активами підприємства (Enterprise Asset Management). Система забезпечує управління життєвим циклом обладнання, моніторинг технічного стану машин та автоматизацію процесів технічного обслуговування. Вона інтегрує інструменти аналітики даних та алгоритми штучного інтелекту для виявлення несправностей і оптимізації процесів ремонту та обслуговування виробничого обладнання.

На рисунку 1.4 представлено приклад аналітичної панелі системи GE Digital Predix, яка використовується для аналізу телеметричних даних виробничого обладнання.



Рис. 1.4 – Аналітична панель системи GE Digital Predix

Дана платформа орієнтована на обробку великих обсягів даних, отриманих від промислових датчиків. Вона забезпечує візуалізацію технічних параметрів обладнання, аналіз експлуатаційних показників та прогнозування можливих відмов. Використання аналітичних інструментів дозволяє підвищити ефективність контролю виробничих процесів та своєчасно реагувати на зміни у роботі технічних систем.

На рисунку 1.5 представлено приклад архітектури платформи Microsoft Azure IoT, яка широко використовується для побудови систем моніторингу обладнання на основі хмарних технологій.

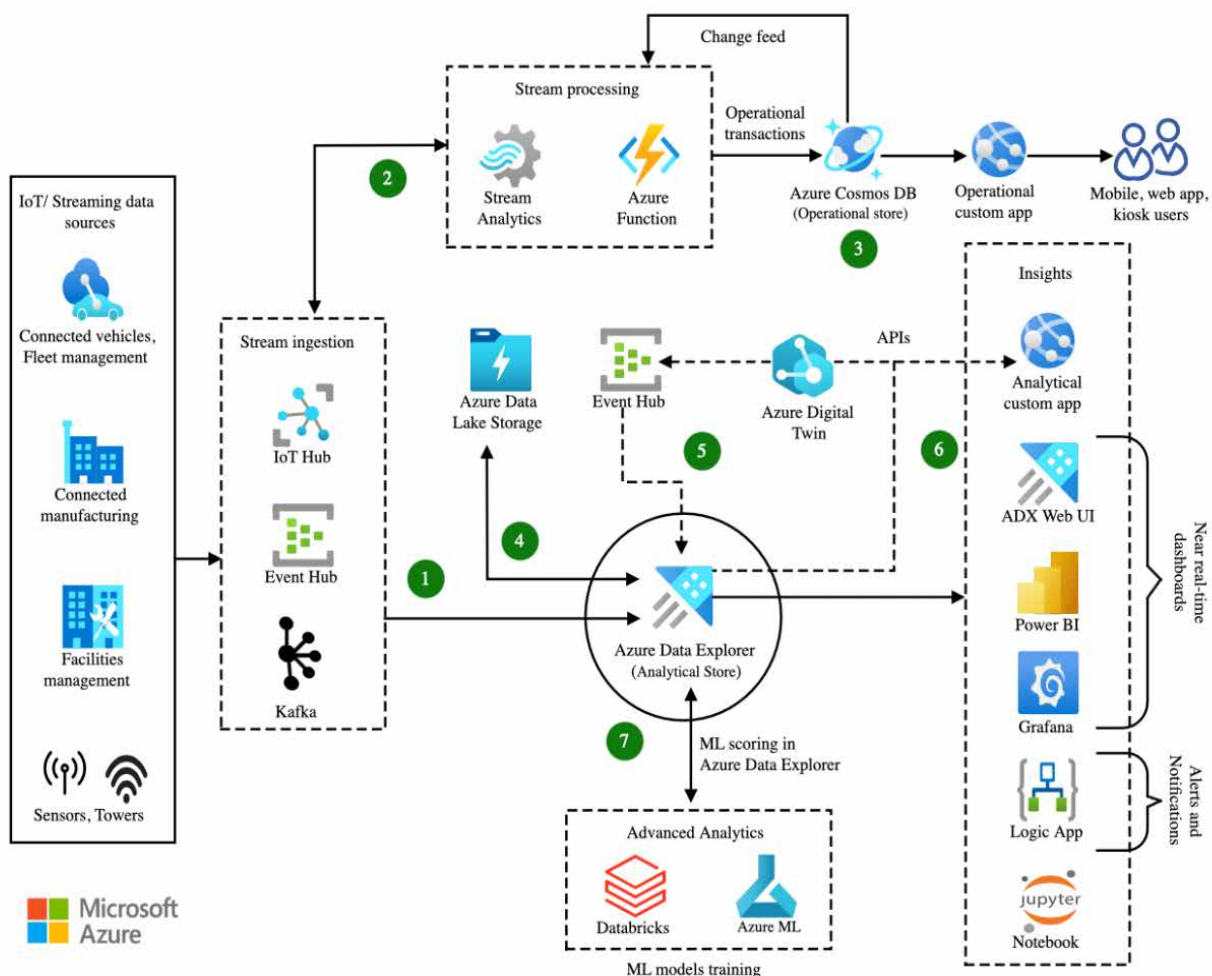


Рис. 1.5 – Архітектура платформи Microsoft Azure IoT

Платформа Azure IoT забезпечує підключення IoT-пристроїв до хмарної інфраструктури, збір телеметричних даних та їх подальшу обробку із використанням інструментів аналітики та машинного навчання. Система підтримує масштабовану обробку потокових даних, інтеграцію з сервісами аналітики та побудову моделей прогнозування технічного стану обладнання.

На рисунку 1.6 представлено інтерфейс платформи PTC ThingWorx, яка використовується для розробки Industrial IoT-застосунків.

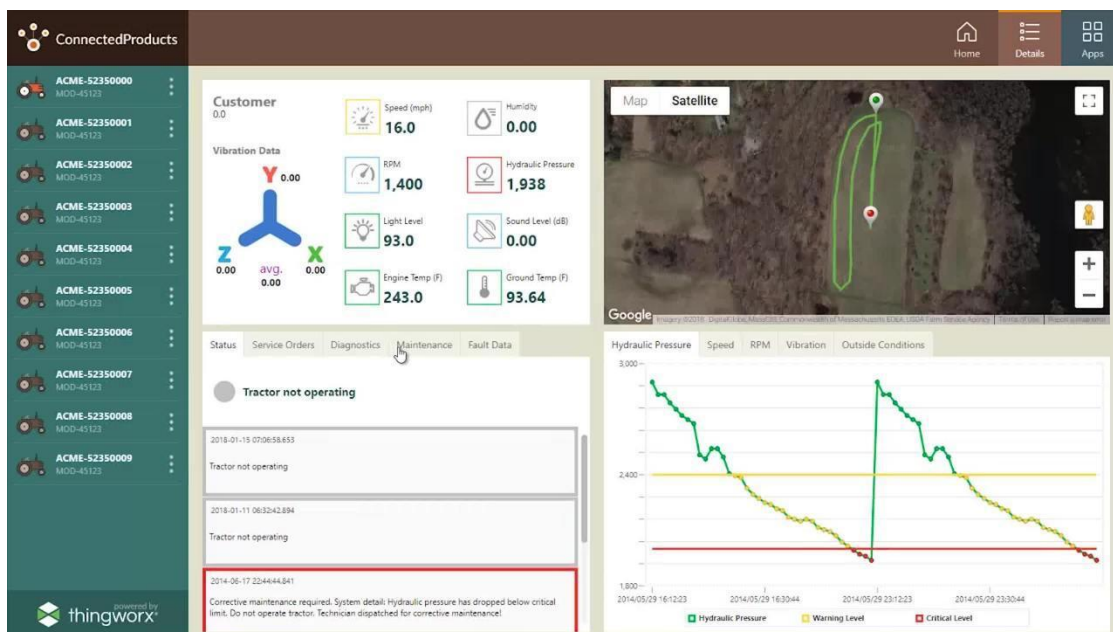


Рис. 1.6 – Інтерфейс платформи PTC ThingWorx

Дана платформа призначена для створення інтелектуальних систем моніторингу обладнання та аналізу телеметричних даних. ThingWorx забезпечує інтеграцію IoT-пристроїв, збір та обробку даних, візуалізацію параметрів роботи обладнання та створення аналітичних панелей для контролю технічного стану виробничих систем.

Для більш наочного аналізу можливостей зазначених платформ було проведено їх порівняння за основними функціональними характеристиками, що наведено у таблиці 1.2.

Таблиця 1.2

Порівняння існуючих систем моніторингу технічного стану обладнання

Система	Збір телеметрії	Аналіз даних	ML-аналітика	Візуалізація	Інтеграція з IoT	Наша система
Siemens MindSphere	✓	✓	✓	✓	✓	✓
IBM Maximo	✓	✓	✓	✓	✓	✓
GE Digital Predix	✓	✓	✓	✓	✓	✓
Microsoft Azure IoT	✓	✓	✓	✓	✓	✓
PTC ThingWorx	✓	✓	✓	✓	✓	✓
Розроблювана система	✓	✓	✓	✓	✓	✓

Проведений аналіз існуючих інформаційних платформ показує, що сучасні системи моніторингу обладнання орієнтовані на використання технологій Industrial IoT, хмарних обчислень та методів машинного навчання для аналізу телеметричних даних. Вони забезпечують збір та обробку великих обсягів даних, візуалізацію параметрів роботи обладнання та підтримку процесів прогнозного технічного обслуговування.

Разом із тим більшість існуючих рішень мають певні обмеження, зокрема складність інтеграції з різними типами обладнання, високу вартість ліцензійного програмного забезпечення та залежність від хмарної інфраструктури постачальника. Крім того, деякі системи потребують складного налаштування та адаптації до специфіки конкретного підприємства. Це обумовлює доцільність розроблення спеціалізованої інтелектуальної системи моніторингу та прогнозування технічного стану обладнання, яка буде орієнтована на інтеграцію IoT-датчиків, використання алгоритмів машинного навчання та забезпечення ефективного контролю виробничих процесів.

1.3 Аналіз вимог програмної системи

Аналіз вимог є важливим етапом розроблення програмних систем, оскільки саме на цьому етапі визначаються функціональні можливості майбутнього програмного продукту, його технічні характеристики, обмеження та очікування користувачів. Результати аналізу вимог формують основу для побудови архітектури системи, вибору технологій реалізації та визначення основних компонентів програмного забезпечення. Чітке формулювання вимог дозволяє забезпечити узгодженість між функціональними можливостями системи та задачами, які вона повинна вирішувати.

Для інтелектуальної системи моніторингу та прогнозування технічного стану обладнання аналіз вимог має особливе значення, оскільки така система повинна забезпечувати обробку значних обсягів телеметричних даних, аналіз технічних параметрів роботи обладнання та підтримку процесів прийняття

рішень щодо технічного обслуговування. Основними функціональними завданнями системи є збір телеметричних даних від IoT-датчиків, обробка та зберігання отриманої інформації, аналіз даних із застосуванням методів машинного навчання, виявлення аномалій у роботі обладнання та прогнозування можливих відмов.

У процесі аналізу предметної області було визначено основні категорії вимог до програмної системи, які включають функціональні вимоги, нефункціональні вимоги та вимоги до безпеки інформації. Така класифікація дозволяє систематизувати характеристики системи та визначити її основні функціональні можливості.

Функціональні вимоги визначають перелік основних операцій та сервісів, які повинна забезпечувати система моніторингу технічного стану обладнання. Вони описують зовнішню поведінку програмного забезпечення та функції, доступні користувачам системи.

Таблиця 1.3

Функціональні вимоги до системи моніторингу технічного стану обладнання

№	Вимога	Опис функціональності
1	Реєстрація та автентифікація користувачів	Створення облікових записів та доступ користувачів до системи
2	Реєстрація обладнання	Додавання інформації про технічні пристрої та їх параметри
3	Збір телеметричних даних	Отримання даних від IoT-датчиків та пристроїв збору інформації
4	Передоброблення даних	Фільтрація, нормалізація та перевірка якості телеметричних даних
5	Аналіз даних	Обробка телеметричних параметрів для визначення стану обладнання
6	Виявлення аномалій	Визначення відхилень у роботі технічних систем
7	Прогнозування відмов	Оцінювання ймовірності виникнення несправностей та залишкового ресурсу обладнання
8	Візуалізація даних	Відображення інформації про стан обладнання у вигляді дашбордів та графіків
9	Формування сповіщень	Автоматичне інформування про можливі несправності
10	Адміністрування системи	Керування користувачами, обладнанням та налаштуваннями системи

Реалізація зазначених функцій дозволяє забезпечити автоматизований контроль технічного стану обладнання, оперативне виявлення потенційних несправностей та підтримку процесів технічного обслуговування.

Нефункціональні вимоги визначають якісні характеристики програмної системи, які впливають на ефективність її використання, продуктивність та надійність роботи. Вони визначають технічні параметри функціонування системи та вимоги до її експлуатації.

Таблиця 1.4

Нефункціональні вимоги до програмної системи

№	Категорія	Опис вимог
1	Продуктивність	Система має забезпечувати обробку поточкових телеметричних даних у режимі реального часу
2	Масштабованість	Можливість підключення великої кількості IoT-пристроїв
3	Надійність	Стійкість системи до збоїв та збереження даних під час помилок
4	Зручність використання	Наявність зрозумілого інтерфейсу для моніторингу параметрів обладнання
5	Сумісність	Підтримка інтеграції з промисловими системами (SCADA, ERP, CMMS)
6	Портативність	Можливість використання системи на різних операційних системах та платформах
7	Розширюваність	Можливість додавання нових модулів аналітики та алгоритмів машинного навчання

Дотримання зазначених нефункціональних вимог дозволяє забезпечити стабільну роботу системи моніторингу, ефективну обробку великих обсягів даних та зручну взаємодію користувачів із програмним забезпеченням.

Оскільки система працює з технічними даними виробничого обладнання та може бути інтегрована з іншими інформаційними системами підприємства, важливим аспектом є забезпечення належного рівня інформаційної безпеки. Система повинна забезпечувати захист доступу до даних, контроль дій користувачів та запобігання несанкціонованому використанню інформації.

Таблиця 1.5

Вимоги до безпеки програмної системи

№	Вимога	Опис
1	Аутентифікація користувачів	Доступ до системи після перевірки облікових даних користувача
2	Контроль доступу	Розмежування прав доступу між різними ролями користувачів
3	Захист даних	Зберігання телеметричних даних у захищених базах даних
4	Журналювання подій	Фіксація дій користувачів та системних подій
5	Захист від несанкціонованого доступу	Використання механізмів контролю доступу до критичних функцій системи

Проведений аналіз вимог дозволив визначити основні функціональні можливості, технічні характеристики та вимоги до безпеки інтелектуальної системи моніторингу та прогнозування технічного стану обладнання. Сформований набір вимог є основою для подальшого етапу проектування архітектури системи, визначення її програмних компонентів та реалізації функціональних модулів.

1.4 Моделювання предметної області

У межах моделювання інтелектуальної системи моніторингу та прогнозування технічного стану обладнання було побудовано кілька типів UML-діаграм, які відображають різні аспекти функціонування системи. Зокрема, було розроблено діаграму варіантів використання, діаграму послідовності та діаграму діяльності. Використання цих моделей дозволяє описати взаємодію користувачів із системою, порядок виконання основних операцій та логіку обробки телеметричних даних.

На рисунку 1.4 представлено діаграму варіантів використання інтелектуальної системи моніторингу та прогнозування технічного стану обладнання.

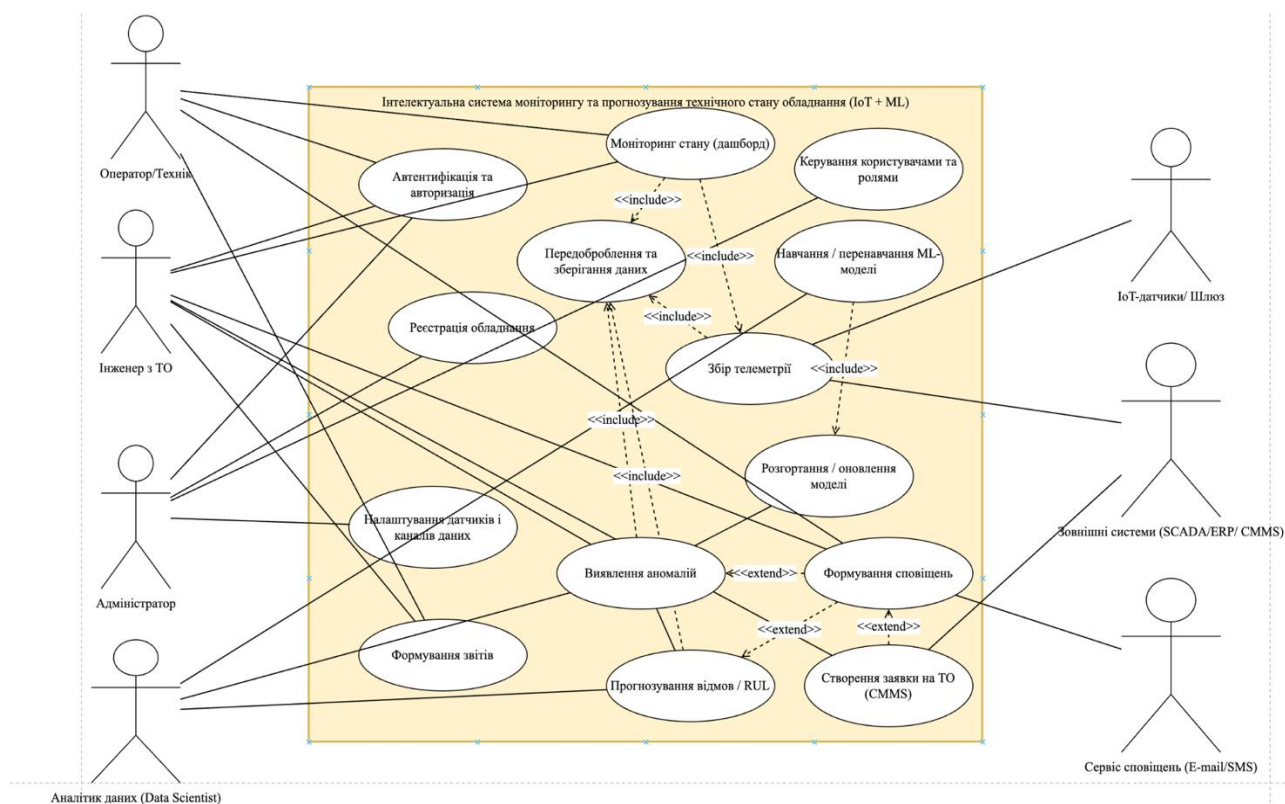


Рис. 1.4 – Діаграма варіантів використання системи моніторингу та прогнозування технічного стану обладнання

Як видно з рисунка 1.4, у системі визначено декілька основних акторів: оператор або технік, інженер з технічного обслуговування, адміністратор системи та аналітик даних (Data Scientist). Окрім користувачів, система взаємодіє із зовнішніми компонентами, такими як IoT-датчики або шлюзи, зовнішні виробничі системи (SCADA, ERP, CMMS) та сервіси сповіщень (E-mail, SMS).

Оператор або технік використовує систему для моніторингу поточного стану обладнання, перегляду показників роботи технічних систем та отримання сповіщень про можливі відхилення у функціонуванні обладнання. Інженер з технічного обслуговування аналізує отримані дані, оцінює технічний стан обладнання та приймає рішення щодо проведення ремонтних або профілактичних робіт.

Адміністратор системи відповідає за налаштування системних параметрів, керування користувачами, конфігурацію датчиків та каналів передачі даних. У свою чергу аналітик даних займається розробкою, навчанням та оновленням

моделей машинного навчання, які використовуються для аналізу телеметричних даних і прогнозування відмов обладнання.

Для більш детального відображення взаємодії між компонентами системи було побудовано діаграму послідовності, представлену на рисунку 1.5.

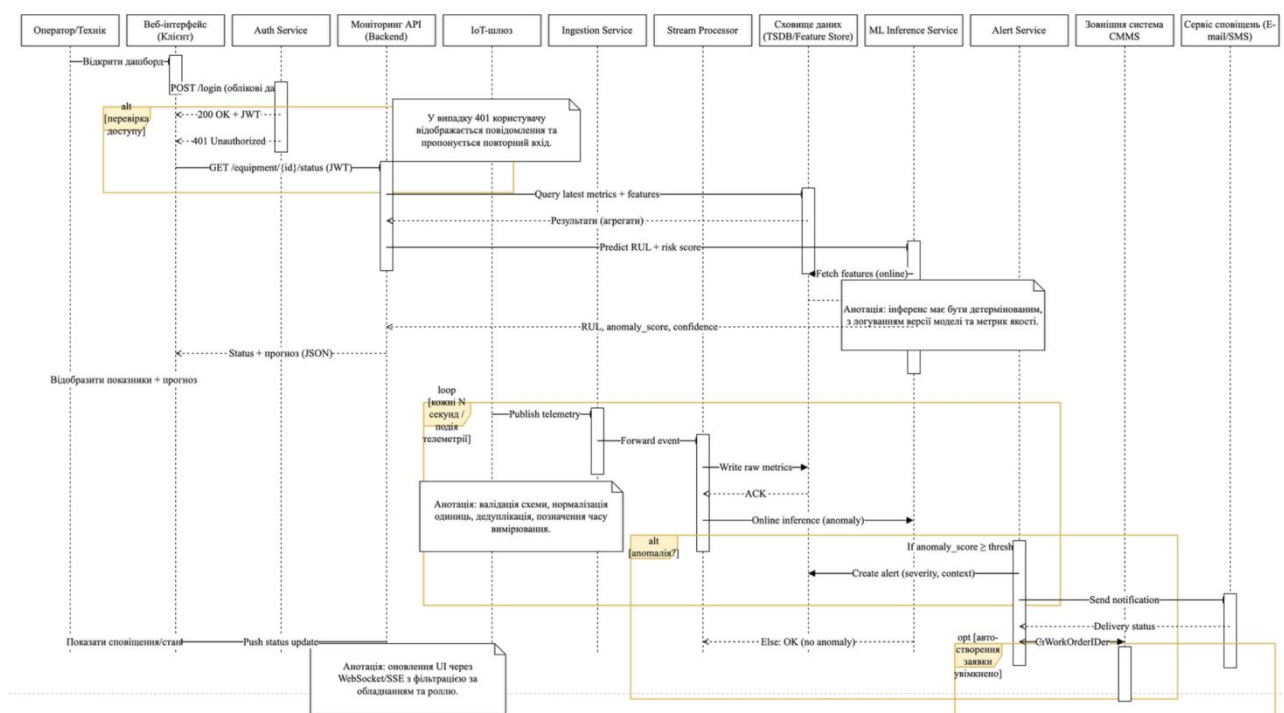


Рис. 1.5 – Діаграма послідовності взаємодії компонентів системи

Дана діаграма демонструє порядок обробки телеметричних даних у системі моніторингу. Процес починається з автентифікації користувача через веб-інтерфейс системи. Після успішного входу користувач отримує доступ до інформації про технічний стан обладнання та може переглядати відповідні аналітичні показники.

Паралельно система здійснює безперервний прийом телеметричних даних від IoT-датчиків через шлюз збору даних. Отримані дані передаються до сервісу обробки поточкових даних, де виконуються процедури валідації, нормалізації та підготовки інформації до подальшого аналізу. Після цього дані зберігаються у спеціалізованому сховищі телеметричних даних.

Наступним етапом є виконання ML-інференсу, під час якого система використовує моделі машинного навчання для оцінювання технічного стану обладнання. Результатом роботи алгоритмів є визначення рівня аномальності

роботи системи, оцінка ризику відмови обладнання та прогнозування залишкового ресурсу роботи.

У випадку виявлення відхилень система формує відповідні сповіщення та передає їх користувачам через сервіси повідомлень. Крім того, за потреби система може автоматично створювати заявки на проведення технічного обслуговування у зовнішніх системах управління виробничими процесами.

Для відображення логіки виконання основного процесу обробки телеметричних даних було побудовано діаграму діяльності, представлену на рисунку 1.6.

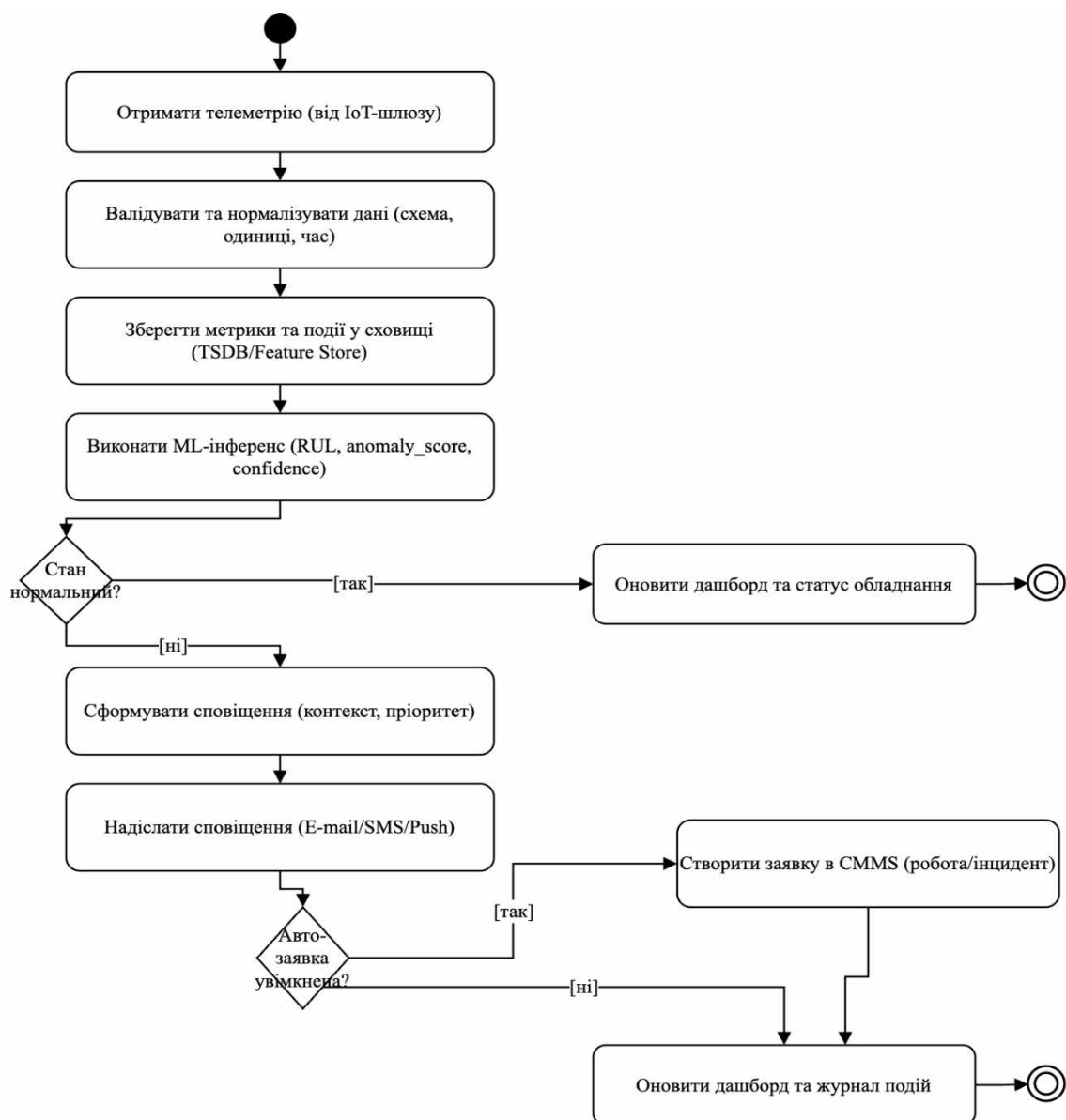


Рис. 1.6 – Діаграма діяльності системи моніторингу технічного стану обладнання

Діаграма діяльності відображає послідовність виконання основних операцій у системі моніторингу. Процес починається з отримання телеметричних даних від IoT-шлюзу. Далі виконується перевірка коректності отриманої інформації, нормалізація одиниць вимірювання та синхронізація часових міток даних.

Після цього оброблені дані зберігаються у спеціалізованому сховищі телеметричних показників, яке використовується для подальшого аналізу. На основі накопичених даних система виконує інференс моделей машинного навчання, що дозволяє визначити технічний стан обладнання, виявити аномальні режими роботи та оцінити ризик виникнення несправностей.

У випадку нормального функціонування обладнання система оновлює інформацію на аналітичних панелях моніторингу. Якщо ж виявлено потенційні відхилення або аномалії, система формує повідомлення для відповідальних користувачів та, за необхідності, створює заявку на проведення технічного обслуговування обладнання.

Побудовані UML-моделі дозволяють отримати цілісне уявлення про функціонування інтелектуальної системи моніторингу технічного стану обладнання. Діаграма варіантів використання визначає ролі користувачів та основні функції системи, діаграма послідовності демонструє порядок взаємодії між компонентами програмного забезпечення, а діаграма діяльності відображає логіку обробки телеметричних даних та формування управлінських рішень. Отримані моделі є основою для подальшого проектування архітектури системи та реалізації її програмних компонентів.

1.5 Постановка завдання

Розроблювана інтелектуальна система моніторингу та прогнозування технічного стану обладнання орієнтована на використання у виробничому середовищі, де одночасно взаємодіють декілька категорій користувачів та технічних компонентів. До основних учасників взаємодії належать оператор або

технік, інженер з технічного обслуговування, адміністратор системи та аналітик даних. Кожна з цих ролей виконує визначені функції під час роботи із системою: оператор здійснює моніторинг показників обладнання, інженер аналізує технічний стан та приймає рішення щодо обслуговування, адміністратор відповідає за конфігурацію системи та керування користувачами, а аналітик даних працює з алгоритмами машинного навчання та моделями прогнозування.

Основою функціонування системи є обробка телеметричних даних, що надходять від IoT-датчиків та шлюзів збору інформації, встановлених на обладнанні. Дані про параметри роботи технічних систем передаються до програмної платформи, де відбувається їх перевірка, нормалізація, збереження та подальший аналіз. На основі отриманої інформації формується оцінка технічного стану обладнання, визначаються можливі відхилення у роботі систем та виконується прогнозування ризику виникнення відмов.

У межах реалізації поставленого завдання система реалізує такі основні функціональні можливості:

1. Автентифікація та керування доступом користувачів з урахуванням ролей та рівнів доступу.
2. Реєстрація обладнання та датчиків, а також налаштування параметрів збору телеметричних даних.
3. Приймання телеметричних даних від IoT-пристроїв через шлюзи або канали передачі даних.
4. Передоброблення інформації, що включає перевірку структури даних, нормалізацію одиниць вимірювання та синхронізацію часових міток.
5. Збереження телеметричних показників у спеціалізованому сховищі даних для подальшого аналізу.
6. Аналітичну обробку даних, включаючи визначення аномалій у роботі обладнання.
7. Прогнозування технічного стану обладнання, зокрема оцінювання залишкового ресурсу роботи (RUL) та ризику відмови.

8. Формування повідомлень про відхилення у роботі обладнання та критичні події.

9. Відображення технічних параметрів у вигляді графіків, дашбордів та аналітичних панелей.

Окрім базових функцій, система містить додаткові можливості, що розширюють аналітичний потенціал платформи та спрощують процес експлуатації обладнання:

1. Відображення статистичних показників роботи обладнання у режимі реального часу.

2. Формування аналітичних звітів щодо технічного стану обладнання за визначений період.

3. Автоматичне оновлення інформації на аналітичних панелях моніторингу.

4. Фіксацію подій та змін у системі у вигляді журналу для подальшого аналізу.

5. Інтеграцію із зовнішніми виробничими системами підприємства (SCADA, ERP, CMMS).

Під час розроблення програмної системи використовується підхід, орієнтований на сучасні технології обробки даних, потокову аналітику та алгоритми машинного навчання. Телеметричні дані, отримані від датчиків, зберігаються у структурованому сховищі даних, що забезпечує їх цілісність, доступність та можливість подальшого аналізу.

Процес створення системи включає основні етапи життєвого циклу програмного забезпечення: моделювання предметної області, проектування архітектури програмної системи, вибір та реалізацію алгоритмів машинного навчання для аналізу технічного стану обладнання, програмну реалізацію функціональних модулів, тестування та підготовку системи до впровадження.

Результатом виконання зазначених етапів стає створення програмної системи, що забезпечує автоматизований моніторинг технічного стану обладнання, оперативне виявлення аномалій у роботі технічних систем та підтримку процесів прогностного технічного обслуговування на основі аналізу телеметричних даних.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Логічна модель даних інтелектуальної системи моніторингу та прогнозування технічного стану обладнання

Логічна модель даних визначає структуру основних інформаційних об'єктів інтелектуальної системи моніторингу та прогнозування технічного стану обладнання, а також взаємозв'язки між ними. Побудова такої моделі є важливим етапом проектування програмного забезпечення, оскільки вона дозволяє формалізувати предметну область, визначити склад даних, необхідних для роботи системи, та забезпечити узгоджене зберігання інформації про обладнання, датчики, телеметричні показники, результати аналітичної обробки, сповіщення і заявки на технічне обслуговування.

Логічна модель даних сформована з урахуванням принципів реляційного проектування та нормалізації даних. У моделі передбачено використання первинних ключів для однозначної ідентифікації кожного запису та зовнішніх ключів для встановлення зв'язків між сутностями. Такий підхід забезпечує цілісність даних, зменшує дублювання інформації та створює основу для подальшого переходу до фізичної моделі бази даних.

На рисунку 2.1 представлено логічну модель даних інтелектуальної системи моніторингу та прогнозування технічного стану обладнання. Діаграма відображає основні сутності системи, їх атрибути та зв'язки, які забезпечують збереження інформації про користувачів, обладнання, датчики, телеметрію, результати аналізу, сповіщення та заявки на технічне обслуговування.

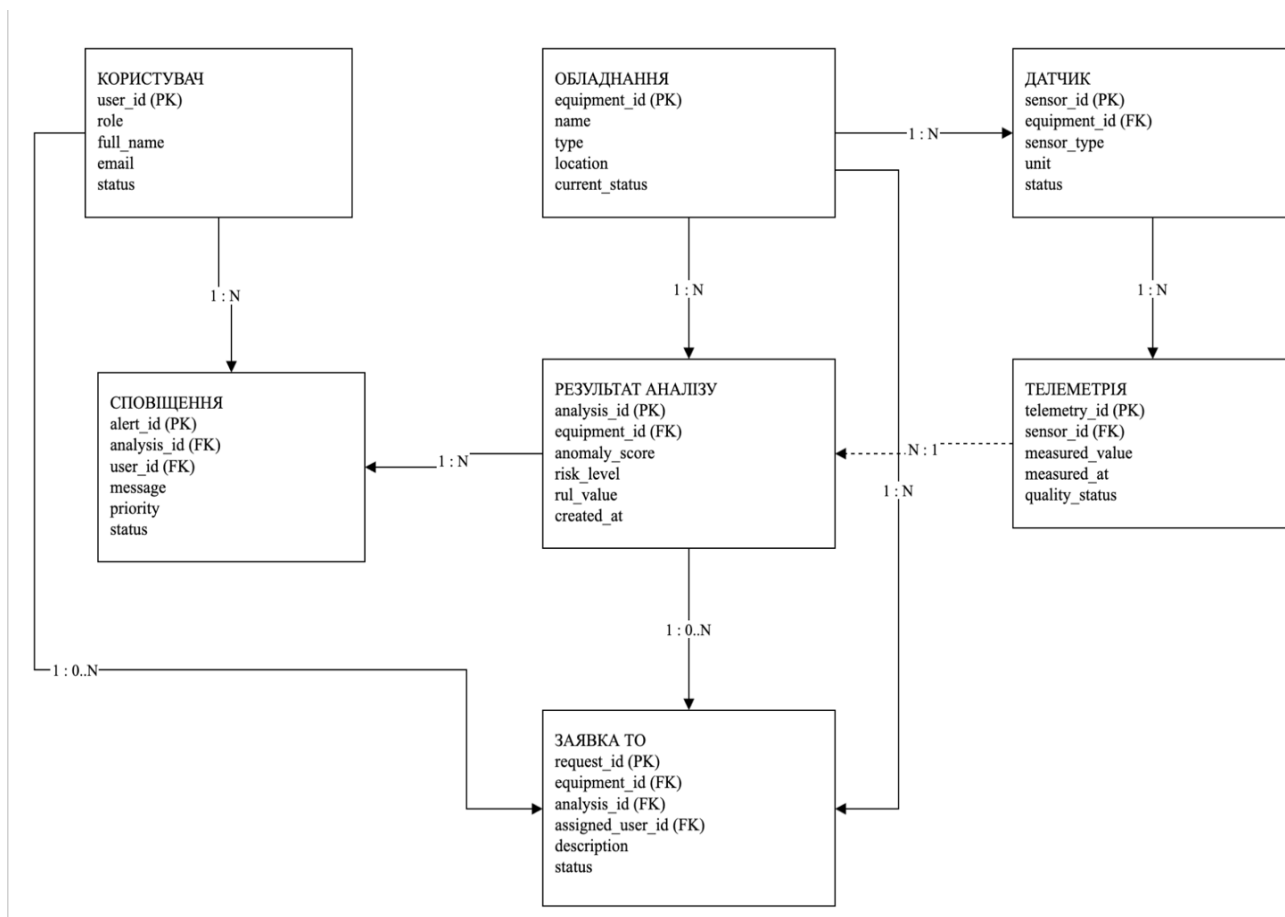


Рис. 2.1 – Логічна модель даних інтелектуальної системи моніторингу та прогнозування технічного стану обладнання

У межах логічної моделі центральною сутністю є «Обладнання», оскільки саме навколо неї формується основний потік даних системи. Для кожної одиниці обладнання зберігаються ідентифікатор, назва, тип, місце встановлення та поточний технічний стан. До обладнання можуть бути підключені декілька датчиків, які вимірюють експлуатаційні параметри, зокрема температуру, вібрацію, тиск, струм, швидкість обертання або інші показники, важливі для оцінювання технічного стану.

Сутність «Датчик» описує пристрої збору телеметричних даних. Кожен датчик належить до певної одиниці обладнання та має власний тип, одиницю вимірювання і статус роботи. Зв'язок між обладнанням і датчиками має тип «один-до-багатьох», оскільки одна одиниця обладнання може мати декілька датчиків, але кожен датчик у межах моделі прив'язаний до конкретного обладнання.

Для збереження виміряних параметрів використовується сутність «Телеметрія». Вона містить значення, отримані з датчиків, час вимірювання та ознаку якості даних. Така структура дозволяє накопичувати історію роботи обладнання та використовувати її для подальшого аналізу. Один датчик може формувати багато телеметричних записів, тому між сутностями «Датчик» і «Телеметрія» встановлено зв'язок типу «один-до-багатьох».

Сутність «Результат аналізу» призначена для збереження результатів обробки телеметричних даних алгоритмами машинного навчання. У ній фіксуються оцінка аномальності, рівень ризику відмови, прогнозований залишковий ресурс обладнання та дата формування результату. Результат аналізу пов'язаний з обладнанням, оскільки саме для конкретної одиниці обладнання визначається технічний стан. Також результат аналізу формується на основі телеметричних даних, що дозволяє пов'язати первинні вимірювання з аналітичними висновками системи.

У разі виявлення небезпечних відхилень система формує сповіщення. Сутність «Сповіщення» містить повідомлення для користувача, пріоритет, статус і посилання на відповідний результат аналізу. Вона пов'язана з користувачем, який отримує повідомлення, та з результатом аналізу, на основі якого це повідомлення було створено. Такий зв'язок дозволяє простежити причину формування сповіщення та відповідальну особу, яка має відреагувати на подію.

Для підтримки процесів технічного обслуговування у моделі передбачено сутність «Заявка ТО». Вона використовується для реєстрації робіт, пов'язаних із перевіркою, ремонтом або профілактичним обслуговуванням обладнання. Заявка містить посилання на обладнання, результат аналізу, відповідального користувача, опис проблеми та поточний статус виконання. Наявність цієї сутності дозволяє перейти від простого виявлення аномалій до організованого процесу реагування на технічні відхилення.

Основні сутності логічної моделі даних та їх призначення наведено у таблиці 2.1.

Таблиця 2.1

Основні сутності логічної моделі даних інтелектуальної системи моніторингу обладнання

Сутність	Призначення
Користувач	Зберігання даних про операторів, інженерів, адміністраторів та інших користувачів системи
Обладнання	Опис одиниць промислового обладнання, для яких здійснюється моніторинг технічного стану
Датчик	Зберігання інформації про IoT-датчики, підключені до обладнання
Телеметрія	Накопичення вимірних параметрів роботи обладнання з часовими мітками
Результат аналізу	Збереження результатів ML-аналізу, оцінки аномальності, ризику відмови та залишкового ресурсу
Сповіщення	Фіксація повідомлень для користувачів про виявлені відхилення або критичні події
Заявка ТО	Реєстрація заявок на технічне обслуговування, ремонт або профілактичну перевірку обладнання

Запропонована логічна модель даних забезпечує чітке розмежування інформаційних сутностей відповідно до функціональної логіки системи. Дані про обладнання та датчики формують базовий рівень моделі, телеметричні записи забезпечують накопичення первинної інформації, а результати аналізу, сповіщення та заявки на технічне обслуговування відображають аналітичний і управлінський рівні роботи системи.

Використання зв'язків типу «один-до-багатьох» дозволяє коректно моделювати реальні процеси експлуатації обладнання. Одна одиниця обладнання може мати багато датчиків, один датчик може формувати багато телеметричних записів, одне обладнання може мати багато результатів аналізу, а кожен результат аналізу може бути підставою для створення сповіщень або заявок на технічне обслуговування. Завдяки цьому модель підтримує накопичення історичних даних, аналіз динаміки технічного стану та контроль реагування на виявлені відхилення.

Отже, сформована логічна модель даних є основою для подальшого проєктування фізичної структури бази даних інтелектуальної системи моніторингу та прогнозування технічного стану обладнання. Вона забезпечує узгоджене зберігання телеметричних і аналітичних даних, підтримує роботу модулів машинного навчання, механізмів сповіщення та процесів технічного обслуговування, що є необхідним для реалізації програмної системи прогнозного обслуговування обладнання.

2.2 Проєктування класової структури та коопераційних сценаріїв інтелектуальної системи

Проєктування програмної структури інтелектуальної системи моніторингу та прогнозування технічного стану обладнання передбачає формалізацію основних програмних об'єктів, які забезпечують збір телеметричних даних, їх збереження, аналітичне опрацювання, виявлення відхилень, формування сповіщень і створення заявок на технічне обслуговування. Для відображення статичної структури програмного забезпечення використано UML-діаграму класів, яка дозволяє визначити склад класів, їхні атрибути, операції та взаємозв'язки між ними.

Діаграма класів інтелектуальної системи моніторингу та прогнозування технічного стану обладнання представлена на рисунку 2.2. Вона відображає основні програмні сутності, що беруть участь у процесі контролю стану обладнання, оброблення телеметрії та підтримки прогнозного технічного обслуговування.

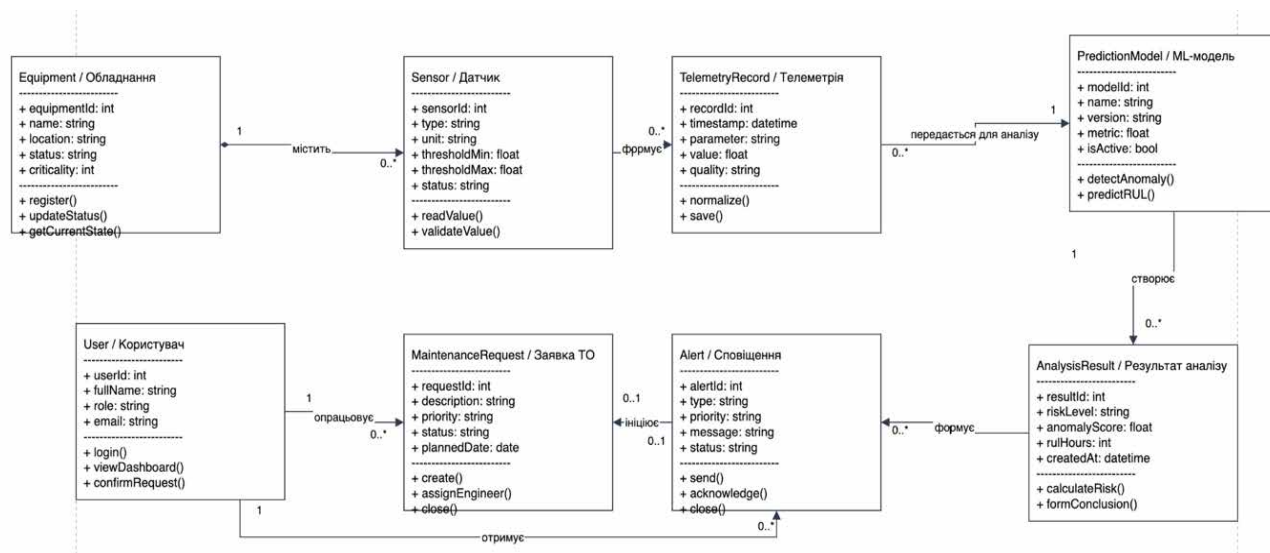


Рис. 2.2 – UML-діаграма класів інтелектуальної системи моніторингу та прогнозування технічного стану обладнання

У моделі виділено основні класи предметної області: Equipment, Sensor, TelemetryRecord, PredictionModel, AnalysisResult, Alert, User та MaintenanceRequest. Така структура відповідає логіці роботи системи, у якій обладнання оснащується датчиками, датчики формують телеметричні записи, телеметрія передається до модуля машинного навчання, а результат аналізу використовується для створення сповіщень і заявок на технічне обслуговування.

Клас Equipment описує одиницю промислового обладнання, для якої здійснюється моніторинг технічного стану. Він містить атрибути ідентифікатора обладнання, назви, місця розташування, поточного статусу та рівня критичності. Операції цього класу забезпечують реєстрацію обладнання, оновлення його стану та отримання поточних параметрів роботи. Зв'язок між класами Equipment і Sensor показує, що одна одиниця обладнання може містити багато датчиків, але кожен датчик належить до конкретного обладнання.

Клас Sensor представляє IoT-датчик, що використовується для вимірювання експлуатаційних параметрів обладнання. До його атрибутів належать ідентифікатор датчика, тип, одиниця вимірювання, мінімальне та максимальне допустиме значення, а також статус роботи. Операції readValue() і validateValue() забезпечують зчитування показника та перевірку його

коректності. Датчик формує множину записів телеметрії, що відображено зв'язком між класами `Sensor` і `TelemetryRecord`.

Клас `TelemetryRecord` призначений для зберігання окремого телеметричного запису. Він містить часову мітку, назву параметра, вимірне значення та ознаку якості даних. Операції `normalize()` і `save()` використовуються для приведення значень до узгодженого формату та подальшого збереження у сховищі телеметрії. Саме записи телеметрії є основою для подальшого машинного аналізу технічного стану обладнання.

Клас `PredictionModel` описує ML-модель, яка виконує виявлення аномалій та прогнозування залишкового ресурсу обладнання. У класі передбачено атрибути ідентифікатора моделі, назви, версії, метрики якості та статусу активності. Операції `detectAnomaly()` і `predictRUL()` забезпечують визначення аномальних режимів роботи та розрахунок прогнозованого залишкового ресурсу. Модель використовує множину телеметричних записів і формує результати аналізу.

Клас `AnalysisResult` зберігає результат аналітичної обробки даних. Він містить рівень ризику, значення показника аномальності, прогнозований залишковий ресурс у годинах і дату створення результату. Операції `calculateRisk()` та `formConclusion()` використовуються для визначення технічного стану обладнання і формування висновку щодо подальших дій. Якщо результат аналізу вказує на підвищений ризик відмови, система формує сповіщення.

Клас `Alert` описує повідомлення про виявлене відхилення або критичну подію. Він містить тип сповіщення, пріоритет, текст повідомлення та статус обробки. Операції `send()`, `acknowledge()` і `close()` забезпечують надсилання повідомлення користувачу, підтвердження його перегляду та закриття після реагування. Сповіщення може ініціювати створення заявки на технічне обслуговування.

Клас `MaintenanceRequest` використовується для представлення заявки на технічне обслуговування або ремонт обладнання. Він містить опис проблеми, пріоритет, статус і заплановану дату виконання робіт. Операції `create()`,

`assignEngineer()` і `close()` забезпечують створення заявки, призначення відповідального інженера та завершення обробки. Клас `User` описує користувача системи, який може переглядати дашборд, отримувати сповіщення та підтверджувати виконання заявки.

Для деталізації динамічних аспектів роботи системи побудовано три діаграми кооперації. Вони демонструють взаємодію об'єктів під час виконання основних сценаріїв: збору телеметрії, машинного аналізу та реагування на критичні події.

Перша діаграма кооперації, наведена на рисунку 2.3, відображає процес збору та збереження телеметричних даних від IoT-датчика.



Рис. 2.3 – Діаграма кооперації збору та збереження телеметричних даних

У цьому сценарії об'єкт `sensor: IoTSensor` зчитує поточні параметри роботи обладнання та передає їх до об'єкта `gateway: IoTGateway`. IoT-шлюз формує пакет даних і передає його у вигляді об'єкта `record: TelemetryRecord`. Після створення телеметричного запису дані зберігаються в об'єкті `storage: TelemetryStorage` як часовий ряд. Така кооперація відображає базовий процес надходження первинної інформації до системи моніторингу.

Друга діаграма кооперації представлена на рисунку 2.4 і описує процес ML-аналізу телеметричних даних та фіксації виявленої аномалії.

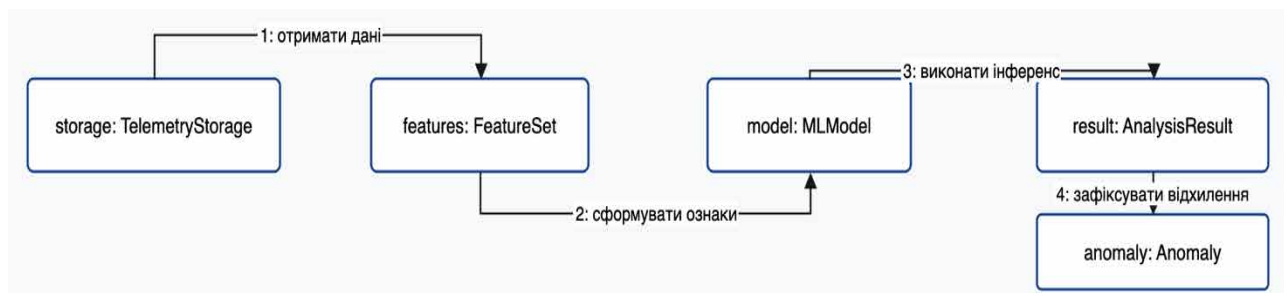


Рис. 2.4 – Діаграма кооперації ML-аналізу телеметричних даних

У межах цього сценарію об'єкт `storage: TelemetryStorage` передає накопичені дані для формування набору ознак `features: FeatureSet`. Після цього сформовані ознаки надходять до об'єкта `model: MLModel`, який виконує інференс і формує об'єкт `result: AnalysisResult`. Якщо результат аналізу свідчить про відхилення від нормального режиму роботи, створюється об'єкт `anomaly: Anomaly`, у якому фіксується факт виявлення аномалії. Такий сценарій демонструє перехід від сирих телеметричних даних до аналітичного висновку про технічний стан обладнання.

Третя діаграма кооперації, наведена на рисунку 2.5, описує процес формування сповіщення та створення заявки на технічне обслуговування.

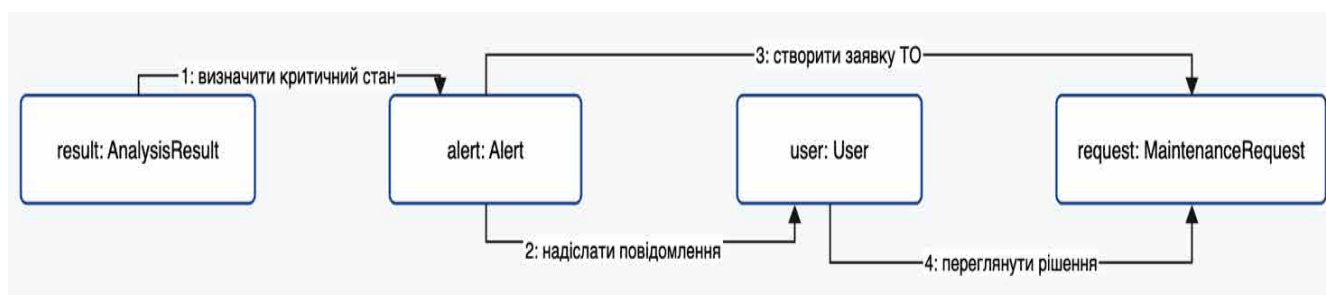


Рис. 2.5 – Діаграма кооперації формування сповіщення та заявки на технічне обслуговування

У цьому сценарії об'єкт `result: AnalysisResult` визначає критичність технічного стану та передає інформацію об'єкту `alert: Alert`. Після цього сповіщення надсилається користувачу `user: User`, який переглядає повідомлення та приймає рішення щодо подальших дій. У разі необхідності система створює об'єкт `request: MaintenanceRequest`, що представляє заявку на технічне обслуговування. Заявка містить дані про проблему, пріоритет, відповідального виконавця та статус виконання робіт.

Основні програмні класи системи та їх призначення узагальнено у таблиці 2.2.

Таблиця 2.2

Основні класи програмної моделі інтелектуальної системи

Клас	Призначення
Equipment	Представлення одиниці промислового обладнання, для якої здійснюється моніторинг технічного стану
Sensor	Опис IoT-датчика, що вимірює параметри роботи обладнання
TelemetryRecord	Зберігання окремого телеметричного запису з часовою міткою та значенням параметра
PredictionModel	Реалізація ML-моделі для виявлення аномалій і прогнозування залишкового ресурсу
AnalysisResult	Збереження результату аналітичної обробки, рівня ризику та прогнозу RUL
Alert	Формування та надсилання сповіщення про відхилення або критичну подію
User	Представлення користувача системи, який переглядає дашборди, отримує сповіщення та опрацьовує заявки
MaintenanceRequest	Реєстрація заявки на технічне обслуговування або ремонт обладнання

Побудована діаграма класів дозволяє формалізувати статичну структуру програмного забезпечення, а діаграми кооперації уточнюють порядок взаємодії об'єктів у ключових сценаріях роботи системи. Запропонована модель охоплює повний цикл функціонування інтелектуальної системи: від зчитування телеметрії з обладнання до формування аналітичного результату, сповіщення користувача та створення заявки на технічне обслуговування.

Отже, класова структура та коопераційні сценарії створюють основу для подальшого проєктування компонентної архітектури системи. Вони забезпечують узгоджене представлення програмних об'єктів, спрощують реалізацію модулів моніторингу, ML-аналізу та технічного обслуговування, а також дозволяють забезпечити логічну цілісність програмного забезпечення інтелектуальної системи прогнозного обслуговування обладнання.

2.3 Компонентна модель програмної архітектури системи

Проектування програмної архітектури інтелектуальної системи моніторингу та прогнозування технічного стану обладнання передбачає визначення основних компонентів програмного забезпечення та зв'язків між ними. Для цього використано UML-діаграму компонентів, яка дозволяє відобразити поділ системи на клієнтський рівень, серверний рівень, рівень даних та зовнішні інтеграційні сервіси.

Діаграма компонентів інтелектуальної системи моніторингу та прогнозування технічного стану обладнання представлена на рисунку 2.6.

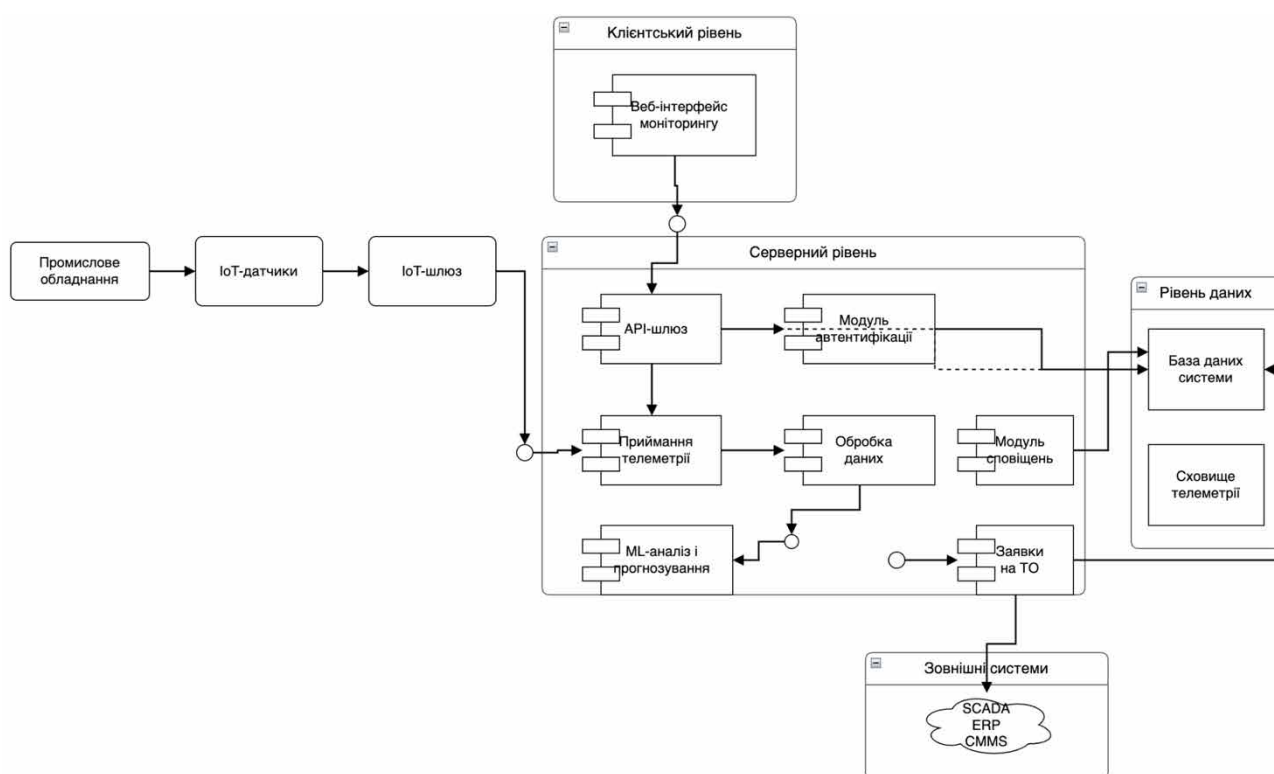


Рис. 2.6 – UML-діаграма компонентів інтелектуальної системи моніторингу та прогнозування технічного стану обладнання

Архітектура системи побудована за модульним принципом. Первинні дані надходять від промислового обладнання через IoT-датчики та IoT-шлюз. На серверному рівні вони приймаються, проходять попередню обробку, передаються до модуля ML-аналізу та використовуються для оцінювання технічного стану обладнання. Результати аналізу відображаються у веб-

інтерфейсі, зберігаються у базі даних і можуть бути підставою для формування сповіщень або заявок на технічне обслуговування.

Клієнтський рівень представлений веб-інтерфейсом моніторингу, через який користувачі переглядають показники обладнання, аналітичні результати, сповіщення та заявки. Серверний рівень містить API-шлюз, модуль автентифікації, компонент приймання телеметрії, модуль обробки даних, ML-модуль, модуль сповіщень і компонент заявок на технічне обслуговування. Рівень даних включає базу даних системи та сховище телеметрії, що забезпечують збереження користувацьких, технічних, аналітичних і часових даних.

Зовнішні системи SCADA, ERP і CMMS використовуються для інтеграції системи моніторингу з виробничою інфраструктурою підприємства. Це дозволяє не лише фіксувати технічні відхилення, а й передавати інформацію для подальшого планування ремонтних або профілактичних робіт.

Основні компоненти архітектури системи та їх призначення узагальнено в таблиці 2.3.

Таблиця 2.3

Основні компоненти архітектури інтелектуальної системи моніторингу обладнання

Компонент	Призначення
Промислове обладнання	Об'єкт моніторингу, з якого отримуються експлуатаційні параметри
IoT-датчики	Збір показників температури, вібрації, тиску, струму та інших технічних параметрів
IoT-шлюз	Передавання телеметрії від датчиків до серверної частини
Веб-інтерфейс моніторингу	Відображення стану обладнання, графіків, сповіщень і заявок
API-шлюз	Координація запитів між інтерфейсом і серверними модулями
Модуль автентифікації	Перевірка користувачів і керування правами доступу
Приймання телеметрії	Отримання та первинна перевірка телеметричних повідомлень
Обробка даних	Нормалізація, фільтрація та підготовка даних до аналізу

Продовження таблиці 2.3

ML-аналіз і прогнозування	Виявлення аномалій, оцінювання ризику та прогнозування RUL
Модуль сповіщень	Формування повідомлень про відхилення та критичні події
Заявки на ТО	Створення і супровід заявок на технічне обслуговування
База даних системи	Зберігання користувачів, обладнання, результатів аналізу, сповіщень і заявок
Сховище телеметрії	Накопичення часових рядів телеметричних показників
SCADA / ERP / CMMS	Зовнішні системи для інтеграції виробничих процесів

Запропонована компонентна архітектура забезпечує чіткий розподіл функцій між окремими модулями та підтримує масштабування системи. Завдяки поділу на клієнтський, серверний, аналітичний і інформаційний рівні система може стабільно обробляти телеметричні дані, виконувати прогнозування технічного стану обладнання та підтримувати процеси технічного обслуговування.

Отже, побудована діаграма компонентів формалізує архітектурну структуру інтелектуальної системи моніторингу та прогнозування технічного стану обладнання. Вона створює основу для подальшої реалізації програмних модулів, організації взаємодії між ними та інтеграції системи з виробничими сервісами підприємства.

2.4 Пакетна структуризація програмного забезпечення системи

Для логічної організації програмного забезпечення інтелектуальної системи моніторингу та прогнозування технічного стану обладнання використано UML-діаграму пакетів. Така діаграма дозволяє відобразити поділ системи на окремі функціональні частини, визначити їх призначення та показати залежності між основними програмними модулями.

Діаграма пакетів інтелектуальної системи моніторингу та прогнозування технічного стану обладнання наведена на рисунку 2.7.

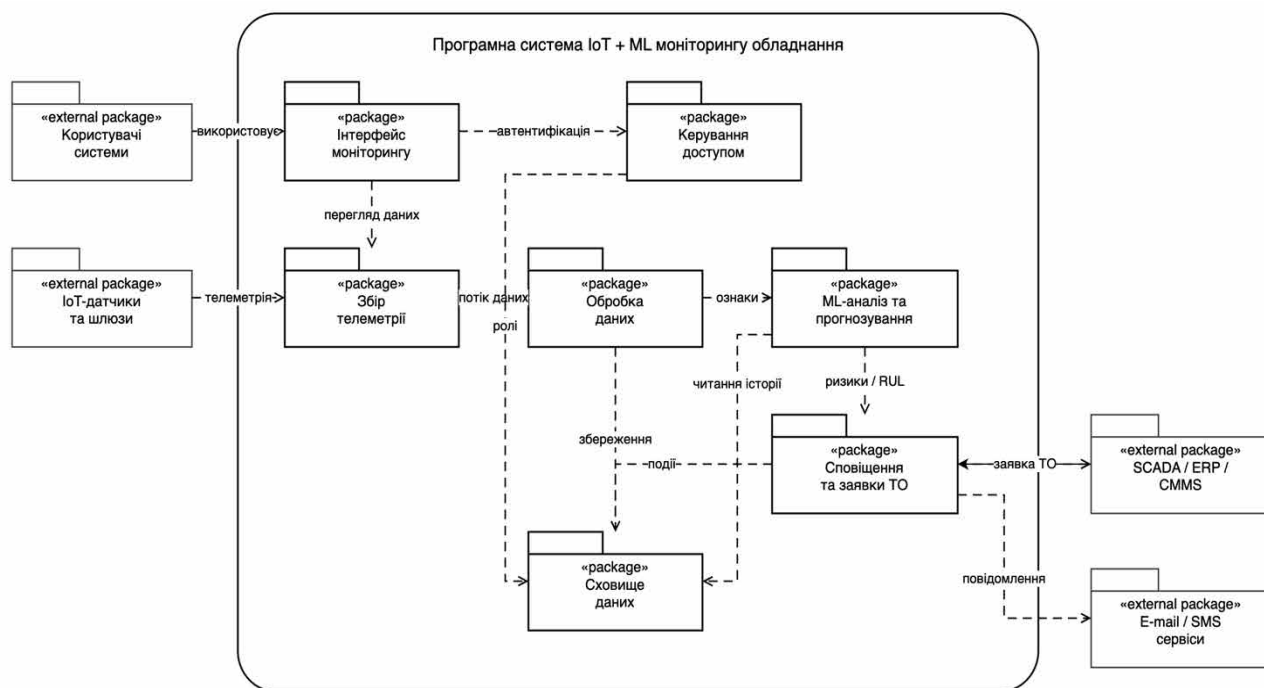


Рис. 2.7 – UML-діаграма пакетів інтелектуальної системи моніторингу та прогнозування технічного стану обладнання

У запропонованій пакетній структурі програмне забезпечення поділено на декілька логічних підсистем. Пакет інтерфейсу моніторингу забезпечує взаємодію користувача із системою та відображення даних про стан обладнання. Пакет керування доступом відповідає за автентифікацію користувачів і розмежування прав відповідно до ролей оператора, інженера, адміністратора та аналітика даних.

Основну функціональність системи реалізують пакети збору телеметрії, обробки даних, ML-аналізу та прогнозування. Вони забезпечують приймання показників від IoT-датчиків, перевірку й нормалізацію телеметричних даних, формування ознак, виявлення аномалій і прогнозування залишкового ресурсу обладнання. Пакет сповіщень та заявок на ТО використовується для інформування відповідальних користувачів і створення заявок на технічне обслуговування у разі виявлення критичних відхилень.

Окремо виділено пакет сховища даних, який забезпечує доступ до бази даних системи та сховища телеметрії. Зовнішні пакети IoT-датчиків і шлюзів, а

також SCADA, ERP і CMMS відображають інтеграцію системи з виробничим обладнанням та інформаційною інфраструктурою підприємства.

Основні пакети програмної архітектури системи та їх призначення наведено у таблиці 2.4.

Таблиця 2.4

Основні пакети програмної архітектури системи

Пакет	Призначення
Інтерфейс моніторингу	Забезпечує відображення стану обладнання, сповіщень, графіків і заявок
Керування доступом	Виконує автентифікацію користувачів і контроль ролей
Збір телеметрії	Забезпечує приймання даних від IoT-датчиків і шлюзів
Обробка даних	Виконує фільтрацію, нормалізацію та підготовку телеметрії до аналізу
ML-аналіз та прогнозування	Виявляє аномалії, оцінює ризик відмови та прогнозує залишковий ресурс
Сповіщення та заявки ТО	Формує повідомлення і заявки на технічне обслуговування
Сховище даних	Забезпечує збереження користувацьких, технічних, аналітичних і телеметричних даних
IoT-датчики та шлюзи	Передають первинні показники роботи обладнання до системи
SCADA / ERP / CMMS	Забезпечують інтеграцію з виробничими та сервісними системами підприємства

Запропонована пакетна структура забезпечує модульність програмного забезпечення та спрощує його подальше розширення. Поділ системи на окремі пакети дозволяє незалежно розвивати інтерфейс, модулі збору телеметрії, аналітичну частину, механізми сповіщення та інтеграцію із зовнішніми системами.

Отже, побудована UML-діаграма пакетів формалізує логічну організацію програмного забезпечення інтелектуальної системи моніторингу та прогнозування технічного стану обладнання. Вона створює основу для подальшої реалізації модулів системи та забезпечує узгоджену структуру програмного коду.

2.5 Висновки до другого розділу

У другому розділі кваліфікаційної роботи виконано проєктування інформаційного та програмного забезпечення інтелектуальної системи моніторингу та прогнозування технічного стану обладнання. Основну увагу приділено формуванню логічної структури даних, визначенню програмних класів, побудові компонентної архітектури та пакетній організації програмного забезпечення.

У межах розділу розроблено логічну модель даних, яка визначає основні інформаційні сутності системи: користувачів, обладнання, датчики, телеметрію, результати аналізу, сповіщення та заявки на технічне обслуговування. Запропонована структура даних забезпечує цілісне зберігання інформації про технічний стан обладнання, історію вимірювань і результати аналітичного опрацювання.

Для формалізації програмної структури побудовано UML-діаграму класів і діаграми кооперації. Вони відображають основні об'єкти системи та типові сценарії їх взаємодії під час збору телеметрії, виконання ML-аналізу, формування сповіщень і створення заявок на технічне обслуговування.

Компонентна модель системи дозволила визначити основні архітектурні частини програмного забезпечення: клієнтський рівень, серверний рівень, рівень даних та зовнішні інтеграційні сервіси. Такий підхід забезпечує зрозумілу організацію системи, підтримує масштабування та створює умови для стабільної обробки телеметричних даних у режимі постійного моніторингу.

Також у розділі сформовано пакетну структуру програмного забезпечення, що відображає логічний поділ системи на функціональні підсистеми. Виділені пакети інтерфейсу моніторингу, керування доступом, збору телеметрії, обробки даних, ML-аналізу, сповіщень, заявок на ТО та сховища даних забезпечують модульність і зручність подальшої реалізації.

Отже, у другому розділі сформовано комплексну проєктну основу інтелектуальної системи моніторингу та прогнозування технічного стану

обладнання. Отримані моделі визначають структуру даних, програмні об'єкти, компоненти та пакети системи, що створює підґрунтя для подальшої реалізації програмного забезпечення, його тестування та впровадження у процеси прогностного технічного обслуговування.

3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір технологій та інструментальних засобів реалізації

Розроблення інтелектуальної системи моніторингу та прогнозування технічного стану обладнання потребує обґрунтованого вибору технологій, які забезпечують стабільне приймання телеметричних даних, їх оброблення, зберігання, аналітичне опрацювання та прогнозування можливих відмов. Оскільки система працює з потоковими даними від IoT-датчиків і використовує алгоритми машинного навчання, технологічний стек повинен підтримувати асинхронну обробку запитів, інтеграцію з промисловими пристроями, роботу з часовими рядами та можливість подальшого масштабування.

Під час вибору інструментальних засобів враховано функціональні вимоги до системи, зокрема збирання телеметрії, передоброблення даних, виявлення аномалій, прогнозування залишкового ресурсу обладнання, формування сповіщень і створення заявок на технічне обслуговування. Також враховано нефункціональні вимоги: продуктивність, надійність, безпеку, розширюваність і зручність подальшого супроводу програмного забезпечення.

Основні технології та інструментальні засоби, доцільні для реалізації системи, наведено в таблиці 3.1.

Таблиця 3.1

Вибір технологій та інструментальних засобів реалізації

№	Компонент системи	Технологія / інструмент	Призначення
1	Мова програмування серверної частини	Python	Реалізація логіки оброблення даних, ML-аналізу та інтеграційних сервісів
2	Серверний API	FastAPI	Побудова REST API для взаємодії між веб-інтерфейсом, модулями системи та зовнішніми сервісами

Продовження таблиці 3.1

3	Клієнтський веб-інтерфейс	React + TypeScript	Реалізація дашбордів моніторингу, форм заявок, сповіщень і аналітичних панелей
4	Протокол передавання телеметрії	MQTT	Передавання даних від IoT-шлюзу до серверної частини системи
5	MQTT-брокер	Eclipse Mosquitto	Організація обміну повідомленнями між IoT-пристроями та сервером
6	Основна база даних	PostgreSQL	Зберігання користувачів, обладнання, датчиків, заявок, сповіщень і результатів аналізу
7	Сховище часових рядів	TimescaleDB	Накопичення та швидке опрацювання телеметричних даних за часовими мітками
8	Засоби машинного навчання	scikit-learn, pandas, NumPy	Передоброблення даних, формування ознак, виявлення аномалій і прогнозування стану обладнання
9	Механізм автентифікації	JWT	Захищений доступ користувачів до функцій системи з урахуванням ролей
10	Контейнеризація	Docker	Розгортання компонентів системи в ізольованому та відтворюваному середовищі
11	Документування API	OpenAPI / Swagger	Опис REST-інтерфейсів і спрощення тестування серверних методів
12	Контроль версій	Git	Відстеження змін програмного коду та організація процесу розроблення

Основною мовою реалізації серверної частини обрано Python, оскільки вона має розвинену екосистему бібліотек для оброблення даних, машинного навчання та роботи з мережевими сервісами. Python доцільно використовувати для реалізації модулів приймання телеметрії, передоброблення даних, ML-аналізу, формування прогнозів і взаємодії з базами даних.

Для побудови серверного API доцільно застосувати FastAPI. Цей фреймворк підтримує швидку обробку запитів, асинхронну взаємодію та автоматичне формування документації OpenAPI. Використання FastAPI дозволяє реалізувати окремі програмні інтерфейси для роботи з користувачами, обладнанням, датчиками, телеметричними записами, результатами аналізу, сповіщеннями та заявками на технічне обслуговування.

Клієнтську частину системи доцільно реалізувати з використанням React і TypeScript. Такий підхід забезпечує створення зручного веб-інтерфейсу для операторів, інженерів з технічного обслуговування, адміністраторів і аналітиків даних. Через інтерфейс користувач може переглядати поточний стан обладнання, аналізувати графіки телеметрії, отримувати сповіщення про відхилення та працювати із заявками на технічне обслуговування.

Для передавання телеметричних даних від IoT-пристроїв до серверної частини системи обрано протокол MQTT. Він є придатним для обміну короткими повідомленнями між датчиками, шлюзами та серверними сервісами, оскільки забезпечує низьке навантаження на мережу та підтримує модель публікації-підписки. Як MQTT-брокер може використовуватися Eclipse Mosquitto, що дозволяє організувати приймання даних від різних типів IoT-шлюзів.

Для зберігання основних структурованих даних системи доцільно використовувати PostgreSQL. У цій базі можуть зберігатися дані про користувачів, ролі, обладнання, датчики, результати аналізу, сповіщення та заявки на технічне обслуговування. Для телеметричних даних, які мають характер часових рядів, доцільним є використання TimescaleDB, що розширює можливості PostgreSQL і забезпечує ефективну роботу з великими обсягами вимірювань.

Для реалізації аналітичної частини системи використовуються бібліотеки pandas, NumPy та scikit-learn. Вони забезпечують підготовку телеметричних даних, очищення некоректних записів, нормалізацію показників, формування ознак, виявлення аномалій і розрахунок прогностичних показників. У межах системи ці засоби можуть використовуватися для оцінювання рівня ризику відмови обладнання, визначення аномальних режимів роботи та прогнозування залишкового ресурсу.

Безпека доступу до системи забезпечується за допомогою JWT-токенів. Такий механізм дозволяє виконувати автентифікацію користувачів, перевіряти їхні права та розмежовувати функціональні можливості відповідно до ролей. Це є важливим для системи, у якій різні користувачі виконують різні дії: оператор

переглядає показники, інженер працює із заявками, адміністратор керує налаштуваннями, а аналітик даних відповідає за ML-моделі.

Для розгортання програмного забезпечення доцільно використовувати Docker. Контейнеризація дозволяє запускати серверну частину, базу даних, MQTT-брокер, сховище телеметрії та інші компоненти в ізольованому середовищі. Це спрощує встановлення системи, зменшує залежність від конкретної операційної системи та забезпечує однаковість середовища розроблення, тестування й експлуатації.

Документування API доцільно виконувати за допомогою OpenAPI та Swagger. Це дозволяє описати всі основні запити до серверної частини, спростити тестування програмних інтерфейсів і забезпечити зручність подальшої інтеграції системи з SCADA, ERP або CMMS. Для контролю версій програмного коду використовується Git, що дає змогу фіксувати зміни, відстежувати етапи розроблення та підтримувати стабільність програмного проєкту.

Отже, обраний технологічний стек забезпечує реалізацію повного циклу роботи інтелектуальної системи моніторингу та прогнозування технічного стану обладнання: від приймання телеметричних даних до їх аналітичного опрацювання, прогнозування відмов, формування сповіщень і створення заявок на технічне обслуговування. Використання Python, FastAPI, MQTT, PostgreSQL, TimescaleDB, бібліотек машинного навчання та Docker створює надійну технологічну основу для реалізації масштабованої та розширюваної системи прогнозного технічного обслуговування.

3.2 Архітектура та розгортання інтелектуальної системи моніторингу обладнання

Архітектура інтелектуальної системи моніторингу та прогнозування технічного стану обладнання побудована з урахуванням багаторівневої організації програмного забезпечення, оброблення поточкових телеметричних

даних та інтеграції з виробничою інфраструктурою підприємства. Основною метою архітектурного проектування є забезпечення стабільного приймання даних від IoT-датчиків, їх подальшого опрацювання, виконання ML-аналізу, формування прогнозних результатів і підтримки процесів технічного обслуговування.

Під час розроблення архітектури враховано необхідність розмежування функцій між фізичним рівнем обладнання, IoT-шлюзом, серверною частиною, сховищами даних і зовнішніми інформаційними системами. Такий підхід дозволяє підвищити надійність системи, спростити її масштабування та забезпечити незалежну модернізацію окремих компонентів без порушення загальної логіки роботи програмного забезпечення.

Загальна архітектура розгортання інтелектуальної системи моніторингу та прогнозування технічного стану обладнання наведена на рисунку 3.1.

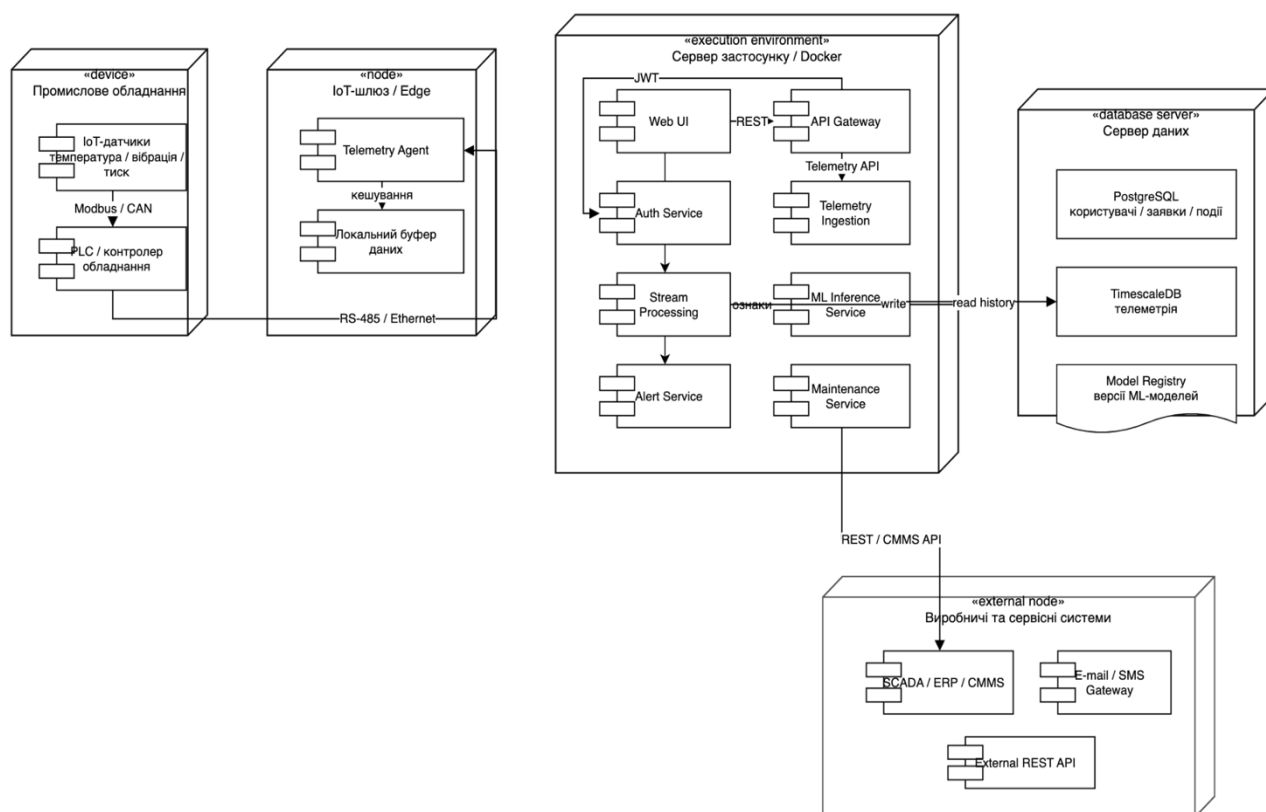


Рис. 3.1 – Архітектура розгортання інтелектуальної системи моніторингу та прогнозування технічного стану обладнання

Подана архітектура відображає взаємодію між промисловим обладнанням, IoT-шлюзом, сервером застосунку, сервером даних і зовнішніми сервісами підприємства. Первинні дані формуються на рівні обладнання за допомогою IoT-датчиків, після чого передаються до шлюзу, де виконується попереднє приймання та буферизація телеметрії. Далі інформація надходить до серверної частини, у якій реалізовано основні модулі оброблення, аналізу та реагування на технічні відхилення.

Сервер застосунку виступає центральним елементом архітектури. У його межах функціонують веб-інтерфейс, API Gateway, сервіс автентифікації, модуль приймання телеметрії, потокове оброблення даних, сервіс машинного аналізу, модуль сповіщень і сервіс технічного обслуговування. Така структура дозволяє організувати повний цикл роботи системи: від отримання показників до формування аналітичного результату та створення заявки на технічне обслуговування.

Для зберігання даних використовується окремий сервер даних. У базі PostgreSQL зберігаються відомості про користувачів, обладнання, заявки, події та результати оброблення. Для телеметричних показників доцільно використовувати TimescaleDB, оскільки вона орієнтована на роботу з часовими рядами та забезпечує ефективне зберігання великої кількості вимірювань. Окремо передбачено Model Registry, який використовується для збереження версій ML-моделей і керування їх використанням у процесі прогнозування.

Важливою особливістю архітектури є наявність зовнішнього інтеграційного рівня. Система може взаємодіяти з SCADA, ERP і CMMS, що дозволяє передавати інформацію про критичні стани обладнання, формувати заявки на ремонт і синхронізувати дані з виробничими сервісами підприємства. Для надсилання повідомлень також передбачено інтеграцію з E-mail або SMS Gateway.

Основні рівні архітектури системи та їх призначення наведено у таблиці 3.2.

Таблиця 3.2

Основні рівні архітектури інтелектуальної системи моніторингу обладнання

Рівень архітектури	Призначення
Рівень промислового обладнання	Формування первинних показників технічного стану за допомогою IoT-датчиків
IoT-шлюз / Edge-рівень	Приймання, локальне буферизування та передавання телеметричних даних
Сервер застосунку	Оброблення запитів, приймання телеметрії, ML-аналіз, сповіщення та заявки на ТО
Сервер даних	Зберігання користувацьких, технічних, телеметричних і аналітичних даних
Зовнішні системи	Інтеграція з SCADA, ERP, CMMS та сервісами повідомлень

Запропонована архітектура забезпечує розподіл функцій між окремими вузлами системи та підтримує надійне опрацювання телеметричних потоків. Використання IoT-шлюзу дає змогу зменшити залежність від стабільності мережевого з'єднання, сервер застосунку забезпечує виконання основної логіки системи, а окремий рівень даних дозволяє ефективно зберігати як структуровану інформацію, так і часові ряди вимірювань.

Застосування контейнеризованого середовища виконання Docker є доцільним для розгортання серверних компонентів, оскільки воно забезпечує ізоляцію сервісів, однаковість середовища розроблення і експлуатації, а також спрощує масштабування системи. За потреби окремі сервіси, зокрема модуль приймання телеметрії або ML-сервіс, можуть бути винесені в окремі контейнери для незалежного оновлення та балансування навантаження.

Архітектура інтелектуальної системи моніторингу та прогнозування технічного стану обладнання забезпечує узгоджену взаємодію між фізичним рівнем збору даних, серверною обробкою, сховищами інформації та зовнішніми виробничими системами. Запропонована структура є достатньо гнучкою для подальшого розширення, підтримує інтеграцію з промисловими сервісами та створює основу для реалізації програмного забезпечення прогнозного технічного обслуговування.

3.3 Фізична структура бази даних системи моніторингу обладнання

Фізична модель даних інтелектуальної системи моніторингу та прогнозування технічного стану обладнання визначає реальну структуру таблиць бази даних, типи полів, первинні та зовнішні ключі, а також зв'язки між інформаційними об'єктами. На відміну від логічної моделі, фізична модель орієнтована безпосередньо на подальшу реалізацію бази даних у середовищі PostgreSQL із можливістю використання TimescaleDB для ефективного зберігання телеметричних часових рядів.

Фізична модель даних системи наведена на рисунку 3.2.

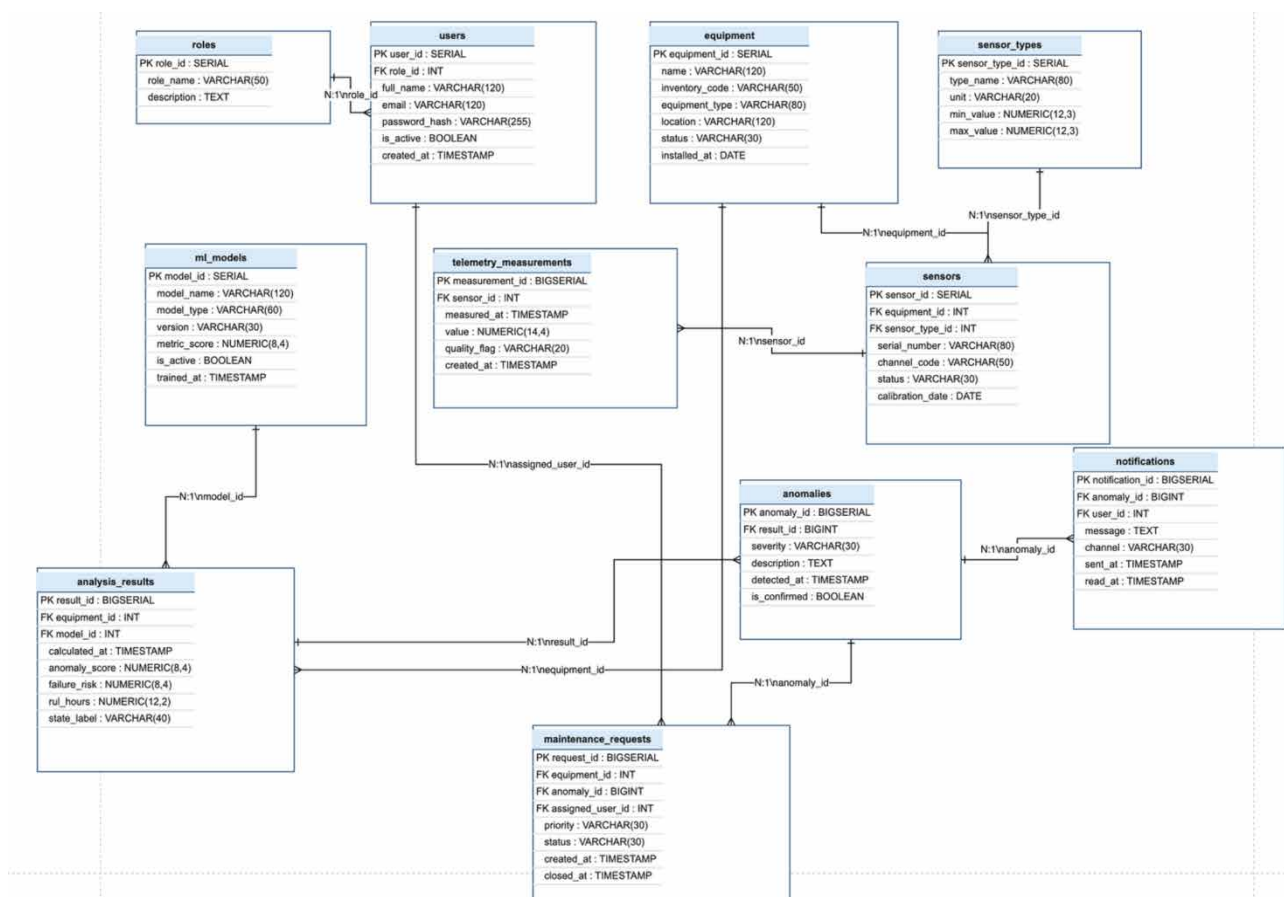


Рис. 3.2 – Фізична модель даних інтелектуальної системи моніторингу та прогнозування технічного стану обладнання

У структурі бази даних передбачено таблиці для зберігання користувачів, ролей, обладнання, типів датчиків, датчиків, телеметричних вимірювань, ML-

моделей, результатів аналізу, аномалій, сповіщень і заявок на технічне обслуговування. Такий склад таблиць відповідає основним процесам системи: реєстрації обладнання, збору телеметрії, аналітичному опрацюванню даних, виявленню відхилень і організації технічного обслуговування.

Основою моделі є таблиці `equipment`, `sensors` і `telemetry_measurements`, які забезпечують збереження інформації про обладнання, підключені до нього датчики та отримані вимірювання. Для аналітичної частини використано таблиці `ml_models`, `analysis_results` і `anomalies`, у яких зберігаються відомості про ML-моделі, результати прогнозування, рівень ризику та зафіксовані аномальні стани. Таблиці `notifications` і `maintenance_requests` забезпечують реагування на виявлені відхилення шляхом формування повідомлень і заявок на технічне обслуговування.

Основні таблиці фізичної моделі даних наведено в таблиці 3.3.

Таблиця 3.3

Основні таблиці фізичної моделі даних системи

Таблиця	Призначення
<code>roles</code>	Зберігання ролей користувачів системи
<code>users</code>	Зберігання облікових записів користувачів
<code>equipment</code>	Опис одиниць промислового обладнання
<code>sensor_types</code>	Зберігання типів датчиків і одиниць вимірювання
<code>sensors</code>	Реєстрація датчиків, підключених до обладнання
<code>telemetry_measurements</code>	Збереження телеметричних вимірювань із часовими мітками
<code>ml_models</code>	Зберігання інформації про ML-моделі та їх версії
<code>analysis_results</code>	Фіксація результатів аналізу технічного стану обладнання
<code>anomalies</code>	Збереження виявлених аномалій і рівня критичності
<code>notifications</code>	Формування повідомлень для відповідальних користувачів
<code>maintenance_requests</code>	Реєстрація заявок на технічне обслуговування або ремонт

У моделі використано первинні ключі типу `SERIAL` або `BIGSERIAL`, що забезпечують унікальну ідентифікацію записів. Зовнішні ключі застосовано для зв'язку користувачів із ролями, обладнання з датчиками, датчиків із телеметричними вимірюваннями, результатів аналізу з ML-моделями та

обладнанням, а також заявок і сповіщень із відповідними аномаліями. Це дозволяє підтримувати цілісність даних і запобігати появі неузгоджених записів у базі.

Для телеметричних даних передбачено окрему таблицю `telemetry_measurements`, оскільки такі записи формуються постійно та можуть накопичуватися у великому обсязі. Наявність полів `measured_at`, `value` і `quality_flag` дає змогу зберігати часові ряди вимірювань, контролювати якість даних і використовувати ці записи для подальшого ML-аналізу.

Фізична модель також підтримує процес прогностного технічного обслуговування. Результати аналізу зберігаються в таблиці `analysis_results`, де фіксуються `anomaly_score`, `failure_risk`, `rul_hours` і `state_label`. У разі виявлення небезпечного стану формується запис у таблиці `anomalies`, після чого система може створити сповіщення або заявку на технічне обслуговування.

Отже, розроблена фізична модель даних забезпечує структуроване зберігання технічної, телеметричної та аналітичної інформації. Вона є основою для реалізації бази даних системи, підтримує зв'язок між обладнанням, датчиками, результатами ML-аналізу та заявками на обслуговування, а також створює умови для подальшого масштабування системи при збільшенні кількості обладнання і телеметричних даних.

3.4 Алгоритмізація програмних модулів інтелектуальної системи

Алгоритмізація програмних модулів є важливим етапом реалізації інтелектуальної системи моніторингу та прогнозування технічного стану обладнання, оскільки вона визначає послідовність виконання основних операцій: збирання телеметричних даних, їх попереднє оброблення, аналітичне оцінювання, виявлення аномалій та формування реакції системи на критичні стани. Для опису логіки роботи ключових модулів розроблено блок-схеми

алгоритмів, які відображають основні сценарії функціонування системи без деталізації внутрішньої програмної реалізації.

У межах системи виділено три основні алгоритмічні процеси: алгоритм збору та передоброблення телеметрії, алгоритм ML-аналізу та прогнозування технічного стану обладнання, а також алгоритм формування сповіщень і заявок на технічне обслуговування. Такий поділ відповідає загальній архітектурі програмного забезпечення та забезпечує послідовний перехід від первинних вимірювань до управлінського рішення щодо обслуговування обладнання.

На рисунку 3.3 наведено блок-схему алгоритму збору та передоброблення телеметричних даних.

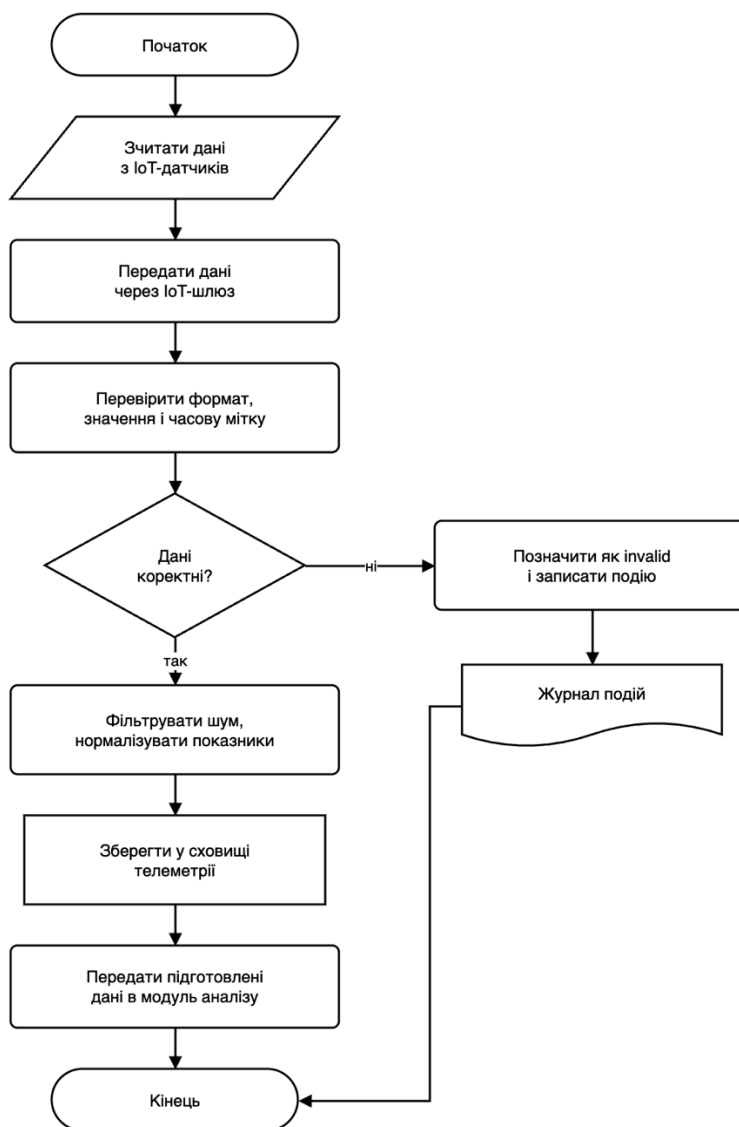


Рис. 3.3 – Блок-схема алгоритму збору та передоброблення телеметрії

Алгоритм збору телеметрії забезпечує приймання даних з IoT-датчиків, їх передавання через IoT-шлюз, перевірку коректності та підготовку до подальшого аналізу. У разі виявлення некоректних значень система фіксує подію в журналі, що дозволяє контролювати якість вхідних даних. Якщо дані відповідають встановленим вимогам, вони проходять фільтрацію, нормалізацію та зберігаються у сховищі телеметрії.

Другий алгоритм описує процес аналітичного опрацювання підготовлених даних. Його блок-схема наведена на рисунку 3.4.

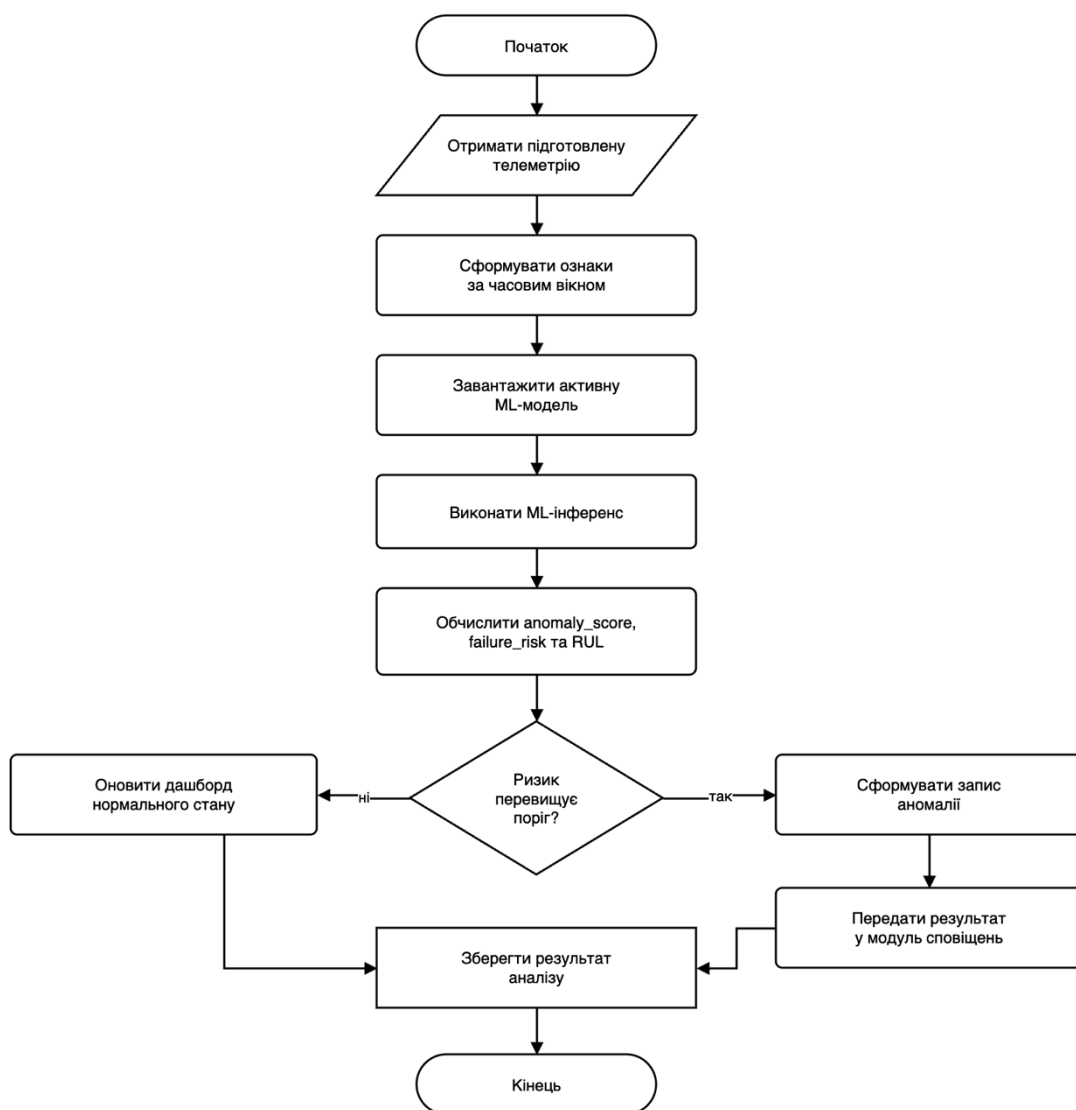


Рис. 3.4 – Блок-схема алгоритму ML-аналізу та прогнозування технічного стану обладнання

Алгоритм ML-аналізу передбачає отримання підготовленої телеметрії, формування ознак за часовим вікном, завантаження активної ML-моделі та

виконання інференсу. У результаті система обчислює значення *anomaly_score*, *failure_risk* і *RUL*, які використовуються для визначення технічного стану обладнання. Якщо рівень ризику не перевищує встановлений поріг, система оновлює дашборд нормального стану. У разі перевищення порогу формується запис аномалії та результат передається до модуля сповіщень.

На рисунку 3.5 представлено блок-схему алгоритму формування сповіщень і заявки на технічне обслуговування.

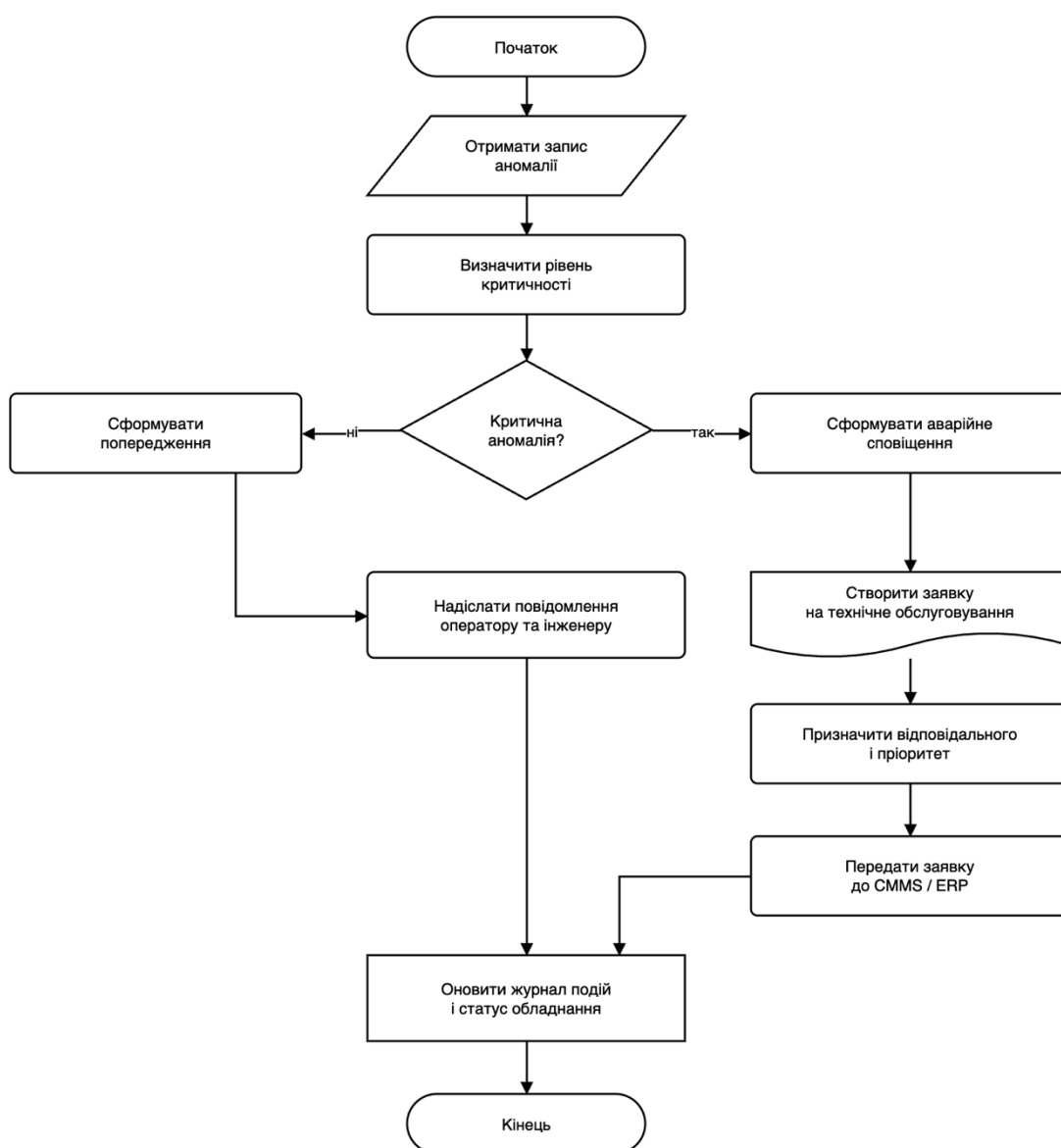


Рис. 3.5 – Блок-схема алгоритму формування сповіщень і заявки на технічне обслуговування

Цей алгоритм забезпечує реагування системи на виявлену аномалію. Після визначення рівня критичності формується відповідне повідомлення для

оператора та інженера. Якщо аномалія має критичний характер, система додатково створює заявку на технічне обслуговування, призначає відповідального виконавця, встановлює пріоритет і передає заявку до зовнішньої системи CMMS або ERP. Після виконання цих дій оновлюється журнал подій і статус обладнання.

Основні алгоритмічні модулі системи та їх призначення наведено в таблиці 3.4.

Таблиця 3.4

Основні алгоритмічні модулі інтелектуальної системи

Алгоритмічний модуль	Призначення
Збір та передоброблення телеметрії	Отримання даних від IoT-датчиків, перевірка коректності, фільтрація та збереження показників
ML-аналіз і прогнозування	Формування ознак, виконання інференсу, оцінювання ризику відмови та прогнозування RUL
Сповіщення та заявка на ТО	Формування повідомлень, створення заявки на обслуговування та оновлення статусу обладнання

Запропоновані алгоритми охоплюють основний цикл роботи інтелектуальної системи: від отримання первинних даних до формування практичного рішення щодо технічного обслуговування. Їх використання дозволяє забезпечити послідовну обробку телеметрії, підвищити достовірність аналітичних результатів і своєчасно реагувати на відхилення у роботі обладнання.

Отже, алгоритмізація програмних модулів створює основу для реалізації ключових функцій системи. Розроблені блок-схеми формалізують логіку збору даних, машинного аналізу та реагування на критичні події, що забезпечує узгоджену роботу програмних компонентів інтелектуальної системи моніторингу та прогнозування технічного стану обладнання.

3.5 Висновки до третього розділу

У третьому розділі кваліфікаційної роботи виконано проєктування та обґрунтування основних рішень щодо реалізації інтелектуальної системи моніторингу та прогнозування технічного стану обладнання. Розглянуто технологічну основу системи, архітектуру її розгортання, фізичну структуру бази даних та алгоритми функціонування ключових програмних модулів.

У процесі вибору технологій визначено інструментальні засоби, які забезпечують приймання телеметричних даних, їх оброблення, зберігання та аналітичне опрацювання. Для серверної частини обґрунтовано використання Python і FastAPI, для передавання телеметрії – MQTT та Eclipse Mosquitto, для зберігання структурованих даних – PostgreSQL, а для часових рядів телеметрії – TimescaleDB. Для реалізації аналітичної частини передбачено використання бібліотек машинного навчання, що дозволяють виконувати передоброблення даних, виявлення аномалій та прогнозування залишкового ресурсу обладнання.

У межах архітектурного проєктування сформовано структуру розгортання системи, яка охоплює рівень промислового обладнання, IoT-шлюз, сервер застосунку, сервер баз даних та зовнішні виробничі системи. Запропонована архітектура забезпечує розподіл функцій між окремими вузлами, підтримує стабільне передавання телеметричних даних і створює умови для інтеграції з SCADA, ERP та CMMS. Такий підхід є доцільним для систем прогнозного технічного обслуговування, оскільки дозволяє поєднати моніторинг обладнання, аналітичну обробку та організацію ремонтних дій у межах єдиного програмного комплексу.

Розроблена фізична модель даних визначає структуру таблиць, типи полів, первинні та зовнішні ключі, необхідні для реалізації бази даних системи. У моделі передбачено зберігання інформації про користувачів, ролі, обладнання, типи датчиків, телеметричні вимірювання, ML-моделі, результати аналізу, аномалії, сповіщення та заявки на технічне обслуговування. Така структура

забезпечує цілісне зберігання технічної, телеметричної й аналітичної інформації та підтримує подальшу реалізацію функцій прогнозування відмов.

У пункті 3.4 виконано алгоритмізацію основних програмних модулів системи. Розроблені блок-схеми описують послідовність збору та передоброблення телеметрії, виконання ML-аналізу і прогнозування, а також формування сповіщень і заявок на технічне обслуговування. Запропоновані алгоритми відображають повний цикл роботи системи: від отримання даних з IoT-датчиків до прийняття рішення щодо реагування на виявлену аномалію.

Отже, у третьому розділі сформовано практичну основу для реалізації інтелектуальної системи моніторингу та прогнозування технічного стану обладнання. Обґрунтований технологічний стек, архітектура розгортання, фізична модель бази даних і алгоритми роботи програмних модулів забезпечують узгоджену реалізацію функцій збору телеметрії, машинного аналізу, виявлення відхилень, сповіщення користувачів і підтримки процесів технічного обслуговування.

4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ

4.1 Тестування програмних модулів інтелектуальної системи моніторингу технічного стану обладнання

Тестування програмних модулів інтелектуальної системи моніторингу та прогнозування технічного стану обладнання спрямоване на перевірку коректності роботи основних підсистем, що забезпечують приймання телеметричних даних, їх попереднє оброблення, зберігання, ML-аналіз, формування сповіщень і створення заявок на технічне обслуговування. Оскільки система працює з потоками даних від IoT-датчиків і виконує аналітичне оцінювання технічного стану обладнання, тестування охоплює як окремі програмні модулі, так і взаємодію між ними.

Основною метою тестування є підтвердження того, що система стабільно приймає телеметрію від IoT-шлюзу, коректно перевіряє та нормалізує дані, зберігає вимірювання у сховищі часових рядів, виконує аналіз показників і правильно реагує на виявлені відхилення. Окремо перевіряється робота модуля автентифікації, доступ користувачів відповідно до ролей, збереження результатів аналізу, формування сповіщень і створення заявок на технічне обслуговування.

Під час тестування враховано функціональні, інтеграційні, модульні та навантажувальні перевірки. Такий підхід дозволяє оцінити не лише правильність виконання окремих операцій, а й загальну працездатність системи в умовах безперервного надходження телеметричних даних.

Загальний план тестування програмних модулів інтелектуальної системи наведено у таблиці 4.1.

Таблиця 4.1

План тестування основних програмних модулів системи

№	Модуль системи	Мета тестування	Очікуваний результат
1	Модуль автентифікації	Перевірка входу користувача та обмеження доступу за ролями	Користувач отримує доступ лише до дозволених функцій
2	Модуль реєстрації обладнання	Перевірка додавання, редагування та перегляду даних про обладнання	Дані обладнання коректно зберігаються і відображаються
3	Модуль IoT-датчиків	Перевірка прив'язки датчиків до обладнання та контролю їх статусу	Датчики коректно пов'язані з відповідними одиницями обладнання
4	Модуль приймання телеметрії	Перевірка отримання даних від IoT-шлюзу	Телеметричні записи приймаються без втрати структури
5	Модуль передоброблення даних	Перевірка фільтрації, нормалізації та контролю якості показників	Некоректні записи відхиляються, коректні передаються далі
6	Сховище телеметрії	Перевірка збереження часових рядів вимірювань	Дані зберігаються з часовими мітками та доступні для аналізу
7	Модуль ML-аналізу	Перевірка виявлення аномалій і розрахунку ризику відмови	Формуються значення <code>anomaly_score</code> , <code>failure_risk</code> і <code>RUL</code>
8	Модуль сповіщень	Перевірка створення повідомлень про відхилення	Користувач отримує сповіщення з правильним пріоритетом
9	Модуль заявок на ТО	Перевірка створення заявки після критичної події	Заявка створюється з прив'язкою до обладнання та аномалії
10	Журнал подій	Перевірка фіксації дій користувачів і системних подій	Події записуються у журнал без втрати даних
11	API системи	Перевірка коректності відповідей серверної частини	API повертає правильні коди стану та структуру JSON
12	Інтеграція з CMMS / ERP	Перевірка передавання заявки до зовнішньої системи	Дані заявки передаються у встановленому форматі

У процесі модульного тестування перевіряється робота окремих компонентів без урахування складної взаємодії з іншими частинами системи.

Для модуля автентифікації перевіряється правильність оброблення облікових даних, формування токена доступу та заборона виконання дій для користувачів без відповідних прав. Для модуля телеметрії контролюється приймання повідомлень, перевірка структури даних, оброблення некоректних значень і запис результатів у сховище.

Інтеграційне тестування зосереджене на перевірці взаємодії між IoT-шлюзом, серверною частиною, базою даних і ML-модулем. У межах такого тестування аналізується повний шлях проходження даних: від отримання показників датчика до формування результату аналізу. Це дозволяє переконатися, що дані не втрачаються під час передавання, коректно зберігаються та можуть бути використані для подальшого прогнозування.

Функціональне тестування охоплює основні сценарії роботи користувачів. Оператор переглядає поточний стан обладнання та сповіщення, інженер опрацьовує заявки на технічне обслуговування, адміністратор керує користувачами та обладнанням, а аналітик даних контролює результати роботи ML-моделей. Такий підхід дає змогу перевірити відповідність реалізованих функцій вимогам, визначеним на етапі системного аналізу.

Для перевірки стабільності роботи системи також передбачено навантажувальне тестування. Воно виконується шляхом імітації одночасного надходження великої кількості телеметричних записів від декількох датчиків. Під час цього оцінюється час оброблення повідомлень, швидкість запису даних у сховище, стабільність API та здатність ML-модуля своєчасно формувати результати аналізу.

Результати тестування дозволяють оцінити коректність реалізації програмних модулів і готовність системи до подальшого експериментального використання. Особлива увага приділяється стабільності оброблення телеметричних даних, правильності виявлення аномалій і своєчасності формування сповіщень, оскільки саме ці функції є ключовими для системи прогнозного технічного обслуговування.

Отже, тестування програмних модулів інтелектуальної системи моніторингу та прогнозування технічного стану обладнання підтверджує узгоджену роботу основних підсистем. Запропонований план тестування охоплює критичні компоненти системи та створює основу для подальшого оцінювання її ефективності, надійності та придатності до використання у виробничому середовищі.

4.2 Тестування інтерфейсних модулів та основних функцій системи

Тестування інтелектуальної системи моніторингу та прогнозування технічного стану обладнання виконувалося з метою перевірки працездатності основних інтерфейсних модулів, коректності оброблення телеметричних даних, стабільності формування аналітичних показників, роботи заявок на технічне обслуговування та функцій адміністрування. Основна увага приділялася перевірці типових сценаріїв взаємодії користувача із системою: входу до облікового запису, перегляду панелі моніторингу, додавання обладнання, внесення телеметрії, створення заявки на ТО, формування звітів і перегляду журналу дій.

Тестування проводилося в умовах демонстраційного наповнення системи, що містило перелік обладнання, контрольні телеметричні показники, розраховані health-score, рівні ризику, журнал технічного обслуговування та дані аудиту. Такий підхід дозволив перевірити не лише окремі екрани інтерфейсу, а й логіку переходу даних між модулями системи.

На рисунку 4.1 наведено екран входу до системи.

**IoT
MONITOR
ML**

Інтелектуальний моніторинг обладнання:
телеметрія сенсорів, health-score, прогноз
ризиків відмови, заявки ТО та звіти.

чорно-білий мінімалізм

SQLite + IoT telemetry

ML прогноз технічного стану

Вхід до системи

Демо: admin@example.com / admin12345 або
engineer@example.com / engineer123

admin@example.com

.....

Увійти

Швидка реєстрація

ПІБ

Email

Пароль

operator

Відділ

Створити акаунт

Рис. 4.1 – Екран авторизації користувача в системі

Під час тестування модуля авторизації перевірялося введення коректних облікових даних адміністратора та інженера, оброблення пароля, створення користувацької сесії та перехід до головної панелі моніторингу. Також перевірялася форма швидкої реєстрації з вибором ролі користувача. У результаті встановлено, що система коректно приймає облікові дані, розмежовує ролі користувачів і відкриває доступ до відповідних функцій.

На рисунку 4.2 представлено панель моніторингу технічного стану обладнання.

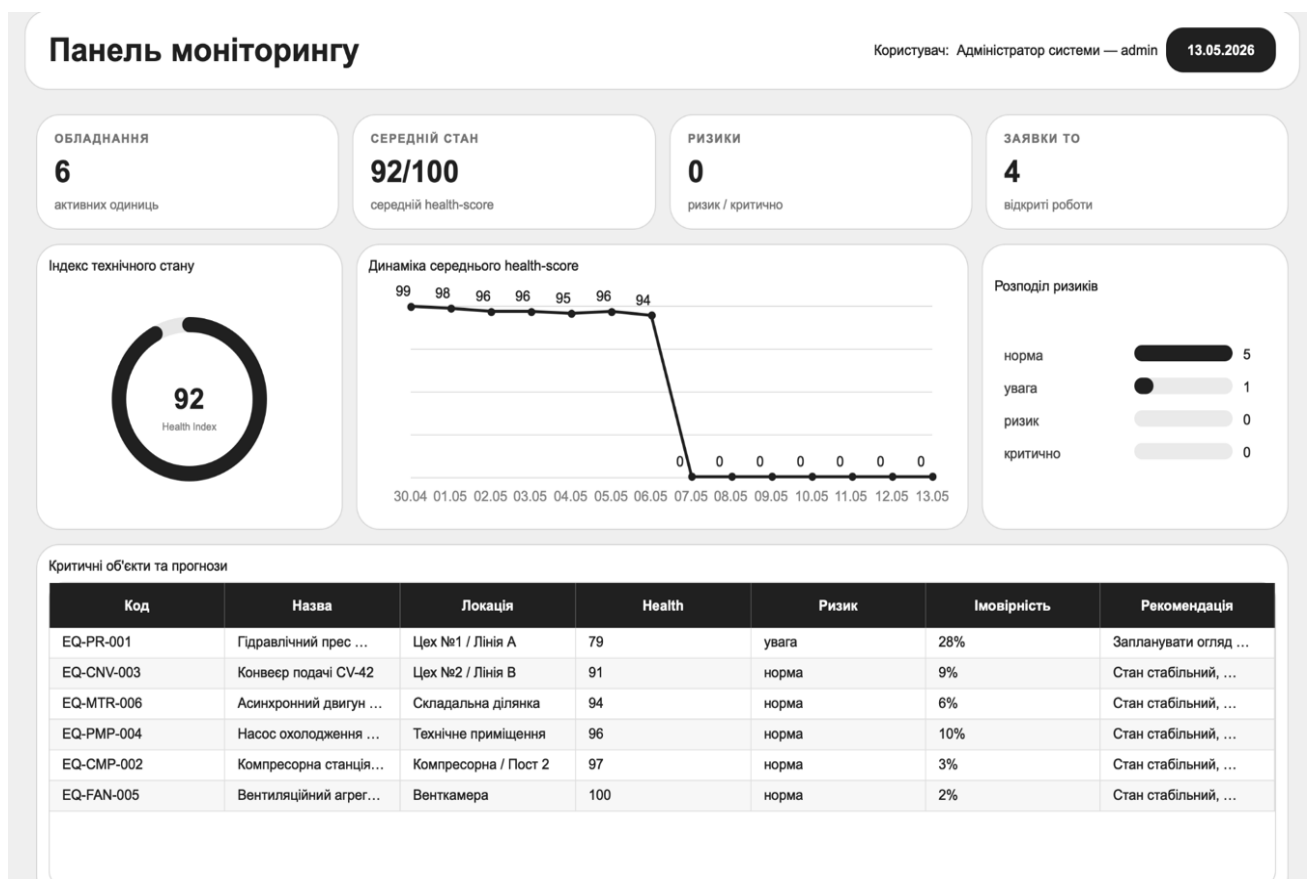


Рис. 4.2 – Панель моніторингу технічного стану обладнання

Панель моніторингу використовувалася для перевірки відображення агрегованих показників: кількості активного обладнання, середнього health-score, кількості ризиків і відкритих заявок на технічне обслуговування. Додатково перевірено побудову графіка динаміки технічного стану, розподіл ризиків і таблицю критичних об'єктів. Результати тестування показали, що дані на панелі оновлюються узгоджено з телеметричними записами та аналітичними результатами.

На рисунку 4.3 наведено інтерфейс реєстру обладнання.

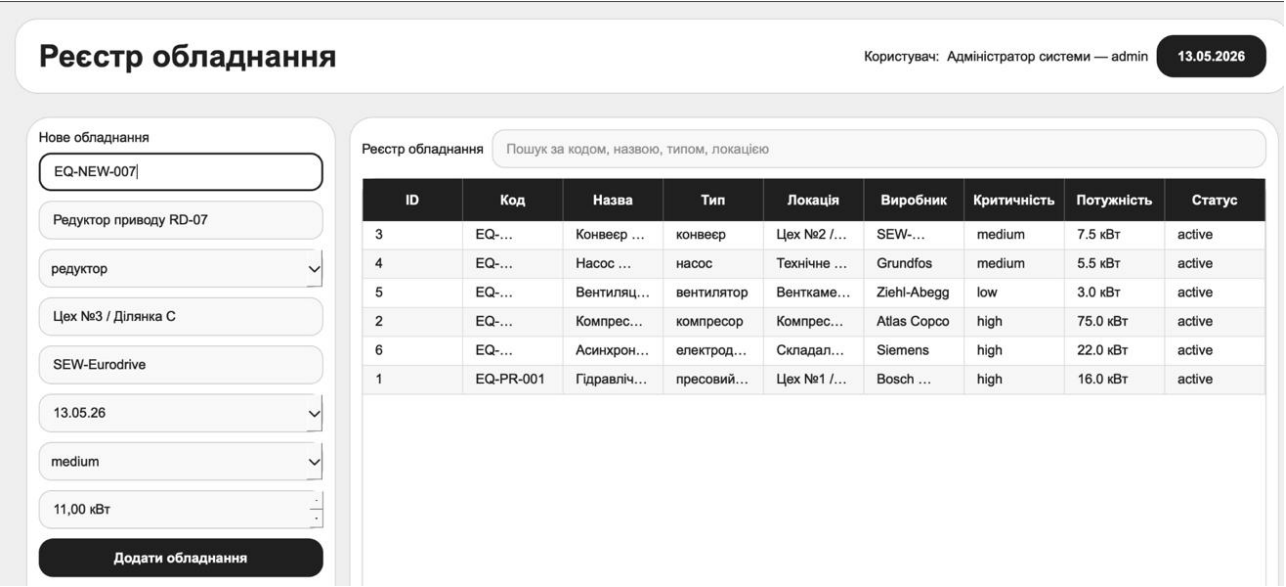


Рис. 4.3 – Інтерфейс реєстру обладнання

У модулі реєстру обладнання перевірялося додавання нової одиниці обладнання, заповнення коду, назви, типу, локації, виробника, дати введення, рівня критичності та потужності. Також тестувалася таблиця обладнання з можливістю пошуку за кодом, назвою, типом і локацією. Перевірка підтвердила, що система коректно зберігає нові записи, відображає їх у реєстрі та підтримує фільтрацію даних.

На рисунку 4.4 показано модуль роботи з IoT-телеметрією.

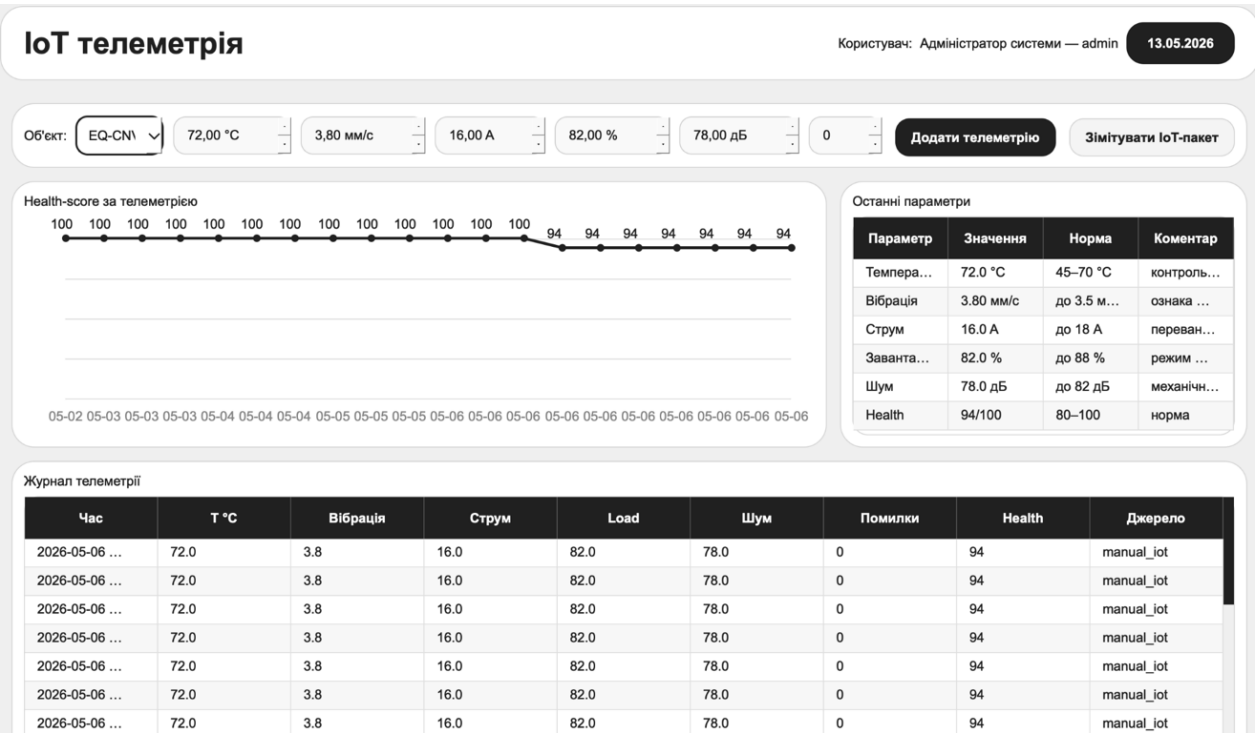


Рис. 4.4 – Інтерфейс введення та перегляду IoT-телеметрії

Тестування модуля телеметрії включало внесення параметрів обладнання, зокрема температури, вібрації, струму, навантаження, шуму та кількості помилок. Після додавання IoT-пакета система формувала запис у журналі телеметрії, оновлювала графік health-score та таблицю останніх параметрів. Перевірка показала, що система правильно обробляє введені значення, зберігає часові записи та відображає результат оцінювання технічного стану обладнання.

На рисунку 4.5 наведено модуль технічного обслуговування.

Технічне обслуговування Користувач: Адміністратор системи — admin 13.05.2026

EQ-CNV-003 — Конвеєр подачі CV-42 Позапланова вібродіагностика вузла high 14.05.26 Інженер діагностики Створити заявку

Журнал технічного обслуговування

ID	Код	Обладнання	Робота	Priority	Статус	План	Виконавець	Причина
1	EQ-PMP-004	Насос ...	Перевірити ...	high	open	2026-05-07	Інженер ...	Підвищена ...
2	EQ-PR-001	Гідралічний ...	Планова ...	high	open	2026-05-08	Інженер ...	Зростання ...
4	EQ-CMP-002	Компресорна ...	Очищення ...	medium	open	2026-05-11	Оператор ...	Зниження зап...
3	EQ-CNV-003	Конвеєр подачі...	Контроль ...	medium	in_progress	2026-05-09	Сервісна служба	Періодичне ТО

Рис. 4.5 – Інтерфейс створення та перегляду заявок на технічне обслуговування

У цьому модулі перевірялося створення заявки на технічне обслуговування для вибраного обладнання, зазначення виду роботи, пріоритету, планової дати та відповідального виконавця. Після створення заявка відображалася в журналі технічного обслуговування із зазначенням статусу, обладнання, типу роботи та причини. Результати тестування підтвердили, що модуль коректно підтримує процес планування ремонтних і профілактичних дій.

На рисунку 4.6 представлено модуль звітів та експорту.

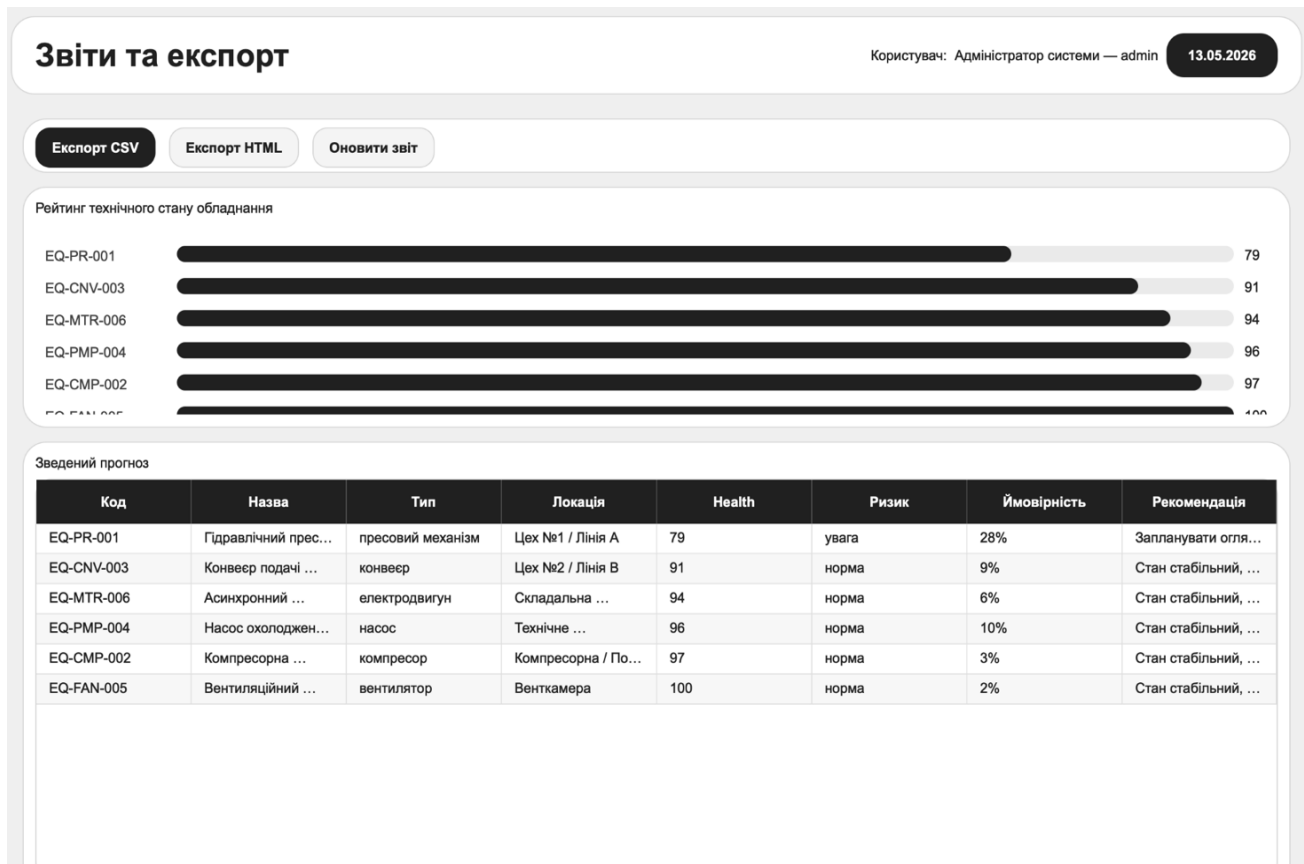


Рис. 4.6 – Інтерфейс формування звітів і експорту результатів

Модуль звітності тестувався шляхом оновлення зведеного прогнозу, перегляду рейтингу технічного стану обладнання та експорту результатів у CSV і HTML. Перевірено, що система формує узагальнену таблицю з кодом обладнання, назвою, типом, локацією, health-score, рівнем ризику, ймовірністю відмови та рекомендацією. Це дозволяє використовувати результати моніторингу для подальшого аналізу технічного стану обладнання.

На рисунку 4.7 наведено інтерфейс адміністрування системи.

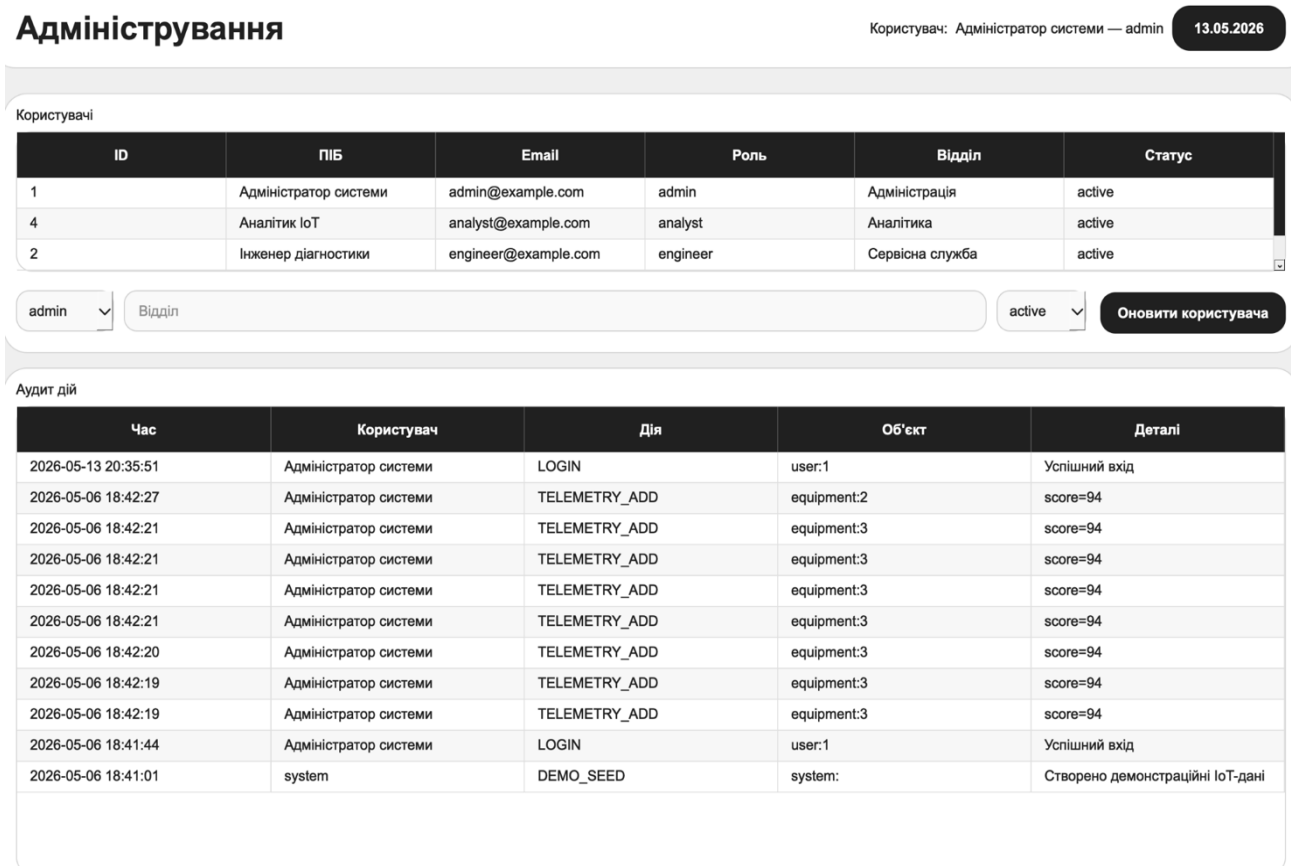


Рис. 4.7 – Інтерфейс адміністрування користувачів і аудиту дій

Під час тестування модуля адміністрування перевірялося відображення користувачів, зміна ролі, оновлення відділу та статусу облікового запису. Окремо перевірено журнал аудиту, у якому фіксуються входи користувачів, додавання телеметрії та системні події. Наявність аудиту дозволяє контролювати дії користувачів і забезпечує простежуваність основних операцій у системі.

Результати функціонального тестування інтерфейсних модулів наведено у таблиці 4.2.

Таблиця 4.2

Результати тестування основних функцій системи				
№	Об'єкт тестування	Перевірена дія	Очікуваний результат	Результат
1	Авторизація	Вхід адміністратора до системи	Відкриття панелі моніторингу	Виконано
2	Реєстрація користувача	Створення нового облікового запису	Дані користувача збережено	Виконано
3	Панель моніторингу	Перегляд health-score та ризиків	Показники відображаються коректно	Виконано

Продовження таблиці 4.2

4	Реєстр обладнання	Додавання нової одиниці обладнання	Запис додано до таблиці	Виконано
5	Пошук обладнання	Фільтрація за кодом або назвою	Відображаються релевантні записи	Виконано
6	IoT-телеметрія	Додавання телеметричного пакета	Дані збережено в журналі телеметрії	Виконано
7	Health-score	Перерахунок стану після внесення телеметрії	Оновлено графік і таблицю параметрів	Виконано
8	Технічне обслуговування	Створення заявки на ТО	Заявка з'являється у журналі робіт	Виконано
9	Звіти	Формування зведеного прогнозу	Дані відображаються у звіті	Виконано
10	Експорт	Експорт результатів у CSV або HTML	Файл експорту формується системою	Виконано
11	Адміністрування	Оновлення ролі та статусу користувача	Зміни збережено	Виконано
12	Аудит дій	Фіксація входу та додавання телеметрії	Події записуються до журналу	Виконано

За результатами тестування встановлено, що інтерфейсні модулі системи працюють узгоджено з основними функціональними вимогами. Система забезпечує вхід користувачів, ведення реєстру обладнання, приймання телеметрії, відображення стану обладнання, створення заявок на технічне обслуговування, формування звітів і контроль дій через журнал аудиту.

Отже, проведене тестування підтвердило працездатність основних модулів інтелектуальної системи моніторингу та прогнозування технічного стану обладнання. Отримані результати свідчать про коректну реалізацію ключових сценаріїв роботи та готовність системи до подальшого оцінювання ефективності оброблення телеметричних даних і прогнозування технічного стану обладнання.

4.3 Розгортання системи, склад інсталяційного пакету

Розгортання інтелектуальної системи моніторингу та прогнозування технічного стану обладнання передбачає підготовку програмного середовища, запуск серверних компонентів, підключення бази даних, налаштування

приймання IoT-телеметрії та перевірку доступу користувачів до веб-інтерфейсу. Основна мета цього етапу полягає в забезпеченні стабільної роботи системи в умовах наближених до виробничого середовища, де дані надходять від обладнання через IoT-шлюз і використовуються для аналітичного оцінювання технічного стану.

Система розгортається за багаторівневою схемою, що включає робоче місце оператора або інженера, промислове обладнання з датчиками, IoT-шлюз, сервер застосунку, сервер баз даних та зовнішні виробничі сервіси. Такий підхід дозволяє розділити функції збору даних, оброблення телеметрії, зберігання результатів, ML-аналізу та формування заявок на технічне обслуговування.

На рисунку 4.8 наведено діаграму розгортання інтелектуальної системи моніторингу та прогнозування технічного стану обладнання.

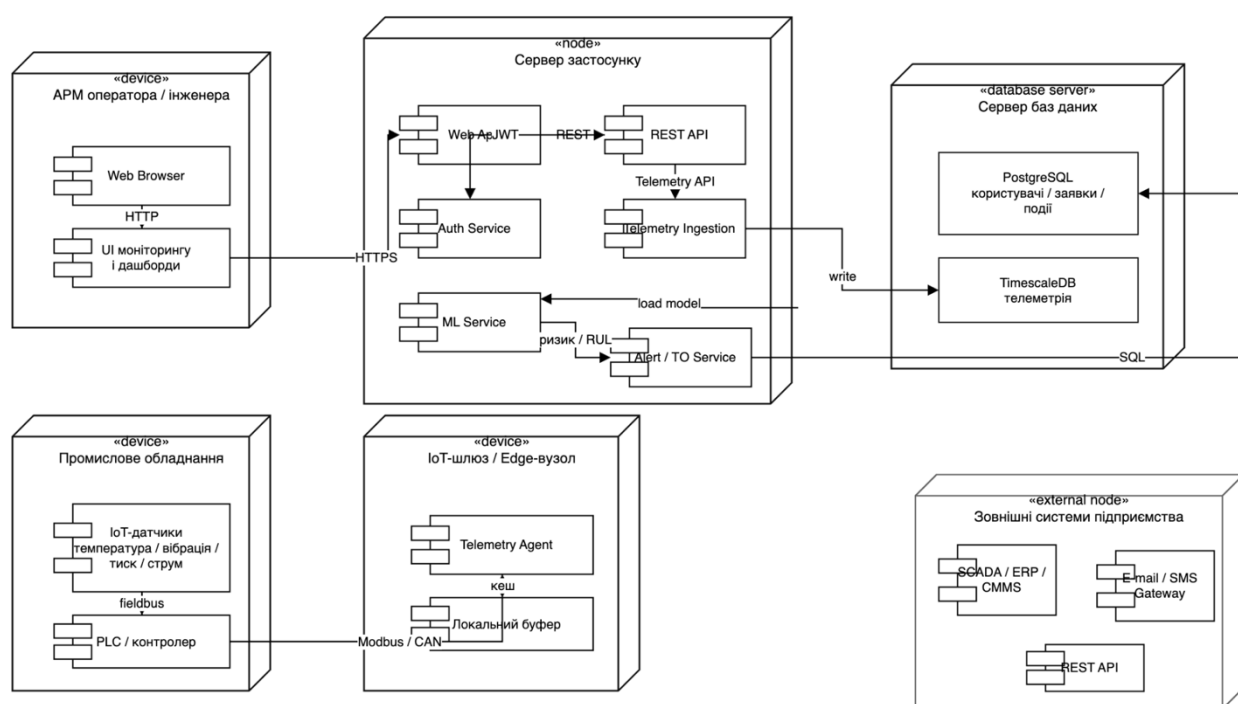


Рис. 4.8 – Діаграма розгортання інтелектуальної системи моніторингу та прогнозування технічного стану обладнання

Діаграма відображає основні вузли розгортання системи: АРМ оператора або інженера, промислове обладнання, IoT-шлюз, сервер застосунку, сервер баз даних і зовнішні системи підприємства. Сервер застосунку забезпечує роботу

веб-інтерфейсу, REST API, сервісу автентифікації, модуля приймання телеметрії, ML-сервісу та модуля сповіщень. Сервер баз даних використовується для зберігання структурованих даних, телеметрії та інформації про результати прогнозування.

Для розгортання системи доцільно використовувати контейнеризоване середовище Docker. Це дозволяє запускати серверні компоненти, базу даних, MQTT-брокер і допоміжні сервіси в ізольованому середовищі. Контейнеризація спрощує встановлення системи, зменшує залежність від конфігурації операційної системи та забезпечує однакові умови роботи під час розроблення, тестування і впровадження.

Перед запуском системи необхідно виконати підготовку середовища: встановити Docker, Docker Compose, налаштувати змінні середовища, створити базу даних, виконати міграції таблиць і перевірити доступність мережових портів для API, веб-інтерфейсу та IoT-шлюзу. Після цього запускаються серверні сервіси, перевіряється підключення до PostgreSQL і TimescaleDB, а також виконується тестове передавання телеметричного пакета.

Послідовність розгортання системи наведено у таблиці 4.3.

Таблиця 4.3

Основні етапи розгортання системи

№	Етап	Зміст робіт	Результат
1	Підготовка середовища	Встановлення Docker, Docker Compose та залежностей	Підготовлене середовище запуску
2	Налаштування конфігурації	Заповнення змінних середовища, портів, параметрів БД та API	Сформований конфігураційний файл
3	Запуск бази даних	Розгортання PostgreSQL і TimescaleDB	Доступне сховище даних
4	Ініціалізація структури БД	Створення таблиць, індексів і початкових довідників	База готова до роботи
5	Запуск серверної частини	Розгортання REST API, сервісів телеметрії, ML-аналізу та сповіщень	Серверні модулі активні
6	Запуск клієнтської частини	Відкриття веб-інтерфейсу оператора або інженера	Доступний інтерфейс системи
7	Перевірка телеметрії	Надсилання тестового IoT-пакета через шлюз	Дані приймаються і зберігаються

Інсталяційний пакет системи має містити всі необхідні файли для запуску програмного комплексу. До його складу входять вихідний код клієнтської та серверної частини, конфігураційні файли, SQL-скрипти створення бази даних, Docker-файли, документація API, тестові дані та інструкція користувача. Такий склад дозволяє швидко відновити систему на новому сервері або робочому середовищі.

Склад інсталяційного пакету наведено у таблиці 4.4.

Таблиця 4.4

Склад інсталяційного пакету системи

Компонент пакету	Призначення
frontend/	Файли клієнтського веб-інтерфейсу моніторингу
backend/	Серверна частина системи, REST API та бізнес-логіка
ml_service/	Модулі ML-аналізу, прогнозування RUL і виявлення аномалій
iot_gateway/	Компонент приймання або імітації IoT-телеметрії
database/	SQL-скрипти створення таблиць, індексів і початкових даних
docker-compose.yml	Опис запуску контейнерів системи
Dockerfile	Файли складання контейнерів клієнтської та серверної частини
.env.example	Приклад конфігурації змінних середовища
docs/	Документація API, інструкції встановлення та опис структури системи
tests/	Тестові сценарії для перевірки основних модулів
seed_data/	Демонстраційні дані обладнання, користувачів і телеметрії
README.md	Коротка інструкція з розгортання і запуску системи

Після розпакування інсталяційного пакету користувач або адміністратор виконує налаштування конфігураційного файлу, запускає контейнери та перевіряє доступність веб-інтерфейсу. Для первинного тестування можуть використовуватися демонстраційні облікові записи адміністратора, інженера та оператора, а також тестові дані обладнання і телеметрії. Це дозволяє швидко перевірити працездатність системи без підключення реальних промислових датчиків.

Окрему увагу під час розгортання потрібно приділити захисту доступу до системи. Паролі користувачів повинні зберігатися у хешованому вигляді,

конфігураційні ключі не мають передаватися у відкритому вигляді, а доступ до API повинен здійснюватися через авторизовані запити. Для виробничого використання доцільно застосовувати HTTPS, обмеження доступу до адміністративних функцій і резервне копіювання бази даних.

Отже, розгортання інтелектуальної системи моніторингу та прогнозування технічного стану обладнання передбачає запуск взаємопов'язаних програмних компонентів, що забезпечують збір телеметрії, зберігання даних, ML-аналіз, сповіщення та підтримку технічного обслуговування. Сформований інсталяційний пакет забезпечує повний набір файлів для встановлення, тестового запуску та подальшого впровадження системи в інформаційне середовище підприємства.

4.4 Висновки до четвертого розділу

У четвертому розділі кваліфікаційної роботи виконано тестування та оцінювання працездатності інтелектуальної системи моніторингу та прогнозування технічного стану обладнання. Основну увагу приділено перевірці програмних модулів, інтерфейсних форм, коректності оброблення телеметричних даних, стабільності формування аналітичних показників, роботи сповіщень і заявок на технічне обслуговування.

У процесі тестування програмних модулів перевірено основні функціональні складові системи: автентифікацію користувачів, реєстрацію обладнання, роботу з IoT-датчиками, приймання телеметрії, передоброблення даних, збереження часових рядів, ML-аналіз, формування сповіщень, створення заявок на ТО та ведення журналу подій. Результати перевірки показали, що основні модулі працюють узгоджено та забезпечують виконання функцій, визначених на етапі аналізу вимог.

Під час тестування інтерфейсних модулів перевірено екран авторизації, панель моніторингу, реєстр обладнання, модуль введення телеметрії, журнал технічного обслуговування, звіти та адміністративний розділ. Система коректно

відображає інформацію про обладнання, обчислює health-score, фіксує телеметричні записи, формує перелік ризиків, створює заявки на технічне обслуговування та зберігає дії користувачів у журналі аудиту.

Окремо розглянуто процес розгортання системи та склад інсталяційного пакету. Визначено, що для стабільного запуску програмного комплексу необхідні клієнтська частина, серверний застосунок, модуль ML-аналізу, компонент приймання IoT-телеметрії, база даних, конфігураційні файли, Docker-файли, тестові дані та документація. Запропонована структура інсталяційного пакету дозволяє швидко розгорнути систему, перевірити її працездатність і підготувати до подальшого використання.

Результати тестування підтвердили, що розроблена система забезпечує повний цикл роботи з технічними даними: від приймання показників обладнання до аналітичного оцінювання стану, формування прогнозу ризику, створення сповіщень і організації технічного обслуговування. Особливо важливим є те, що система підтримує не лише збереження телеметрії, а й її практичне використання для прийняття рішень щодо профілактики можливих відмов.

Отже, у четвертому розділі підтверджено працездатність основних модулів інтелектуальної системи моніторингу та прогнозування технічного стану обладнання. Проведене тестування засвідчило відповідність системи поставленим вимогам, коректність роботи інтерфейсів, стабільність оброблення даних і готовність програмного забезпечення до подальшого впровадження в умовах виробничого середовища.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальне практичне завдання розроблення інтелектуальної системи моніторингу та прогнозування технічного стану обладнання. Актуальність роботи зумовлена потребою підприємств у безперервному контролі параметрів роботи технічних систем, своєчасному виявленні відхилень, зменшенні кількості аварійних зупинок і переході від планово-регламентного обслуговування до прогнозного технічного обслуговування на основі даних.

У першому розділі виконано системний аналіз предметної області. Розглянуто особливості моніторингу промислового обладнання, роль IoT-датчиків у зборі телеметричних даних і доцільність застосування методів машинного навчання для виявлення аномалій та прогнозування відмов. Проаналізовано сучасні рішення у сфері Industrial IoT і predictive maintenance, зокрема Siemens Insights Hub, IBM Maximo, GE Digital Predix, Microsoft Azure IoT і PTC ThingWorx. На основі аналізу визначено основні вимоги до системи, її функціональні можливості, ролі користувачів і процеси, що підлягають автоматизації.

У другому розділі розроблено проєктні моделі інформаційного та програмного забезпечення системи. Побудовано логічну модель даних, яка охоплює користувачів, обладнання, датчики, телеметрію, результати аналізу, сповіщення та заявки на технічне обслуговування. Розроблено UML-діаграму класів і коопераційні сценарії, що відображають взаємодію основних програмних об'єктів під час збору телеметрії, ML-аналізу та реагування на критичні події. Також сформовано компонентну і пакетну структуру системи, що забезпечує модульність, масштабованість і зручність подальшої реалізації.

У третьому розділі обґрунтовано технологічні та архітектурні рішення для реалізації програмного забезпечення. Для серверної частини доцільним визначено використання Python і FastAPI, для передавання телеметрії –

MQTT, для зберігання структурованих даних — PostgreSQL, а для часових рядів телеметрії — TimescaleDB. Для аналітичної частини передбачено використання бібліотек машинного навчання, що дають змогу виконувати передоброблення даних, виявлення аномалій, оцінювання ризику відмови та прогнозування залишкового ресурсу обладнання. Розроблено фізичну модель бази даних і алгоритми роботи ключових модулів системи.

У четвертому розділі проведено тестування програмних модулів та інтерфейсних компонентів системи. Перевірено авторизацію користувачів, роботу панелі моніторингу, реєстр обладнання, модуль IoT-телеметрії, формування заявок на технічне обслуговування, звіти, експорт даних і журнал аудиту. Результати тестування підтвердили, що система коректно обробляє телеметричні дані, формує показники технічного стану, відображає ризики, створює сповіщення та підтримує процеси технічного обслуговування.

Практичним результатом роботи є програмне рішення, яке забезпечує повний цикл роботи з технічними даними: від отримання показників обладнання через IoT-датчики до аналітичного оцінювання стану, прогнозування ризику відмови, формування рекомендацій і створення заявок на технічне обслуговування. Запропонована система може бути використана як основа для впровадження прогнозного технічного обслуговування на підприємствах, де важливими є надійність обладнання, контроль технічного стану та мінімізація простоїв.

Отже, поставлену мету кваліфікаційної роботи досягнуто. Розроблена інтелектуальна система моніторингу та прогнозування технічного стану обладнання забезпечує автоматизований збір телеметрії, аналітичну обробку даних, виявлення аномальних режимів, прогнозування можливих відмов і підтримку прийняття рішень щодо технічного обслуговування. Використання такої системи сприяє підвищенню надійності експлуатації обладнання, оптимізації ремонтних робіт і підвищенню ефективності виробничих процесів.

СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ISO 17359:2018. Condition monitoring and diagnostics of machines. General guidelines. Geneva : International Organization for Standardization, 2018.
URL: <https://www.iso.org/standard/71194.html>
2. ISO 13374-1:2003. Condition monitoring and diagnostics of machines. Data processing, communication and presentation. Part 1: General guidelines. Geneva : International Organization for Standardization, 2003.
URL: <https://www.iso.org/ru/standard/21832.html>
3. ISO 55000:2024. Asset management. Vocabulary, overview and principles. Geneva : International Organization for Standardization, 2024.
URL: <https://www.iso.org/standard/83053.html>
4. ISO/IEC 27001:2022. Information security, cybersecurity and privacy protection. Information security management systems. Requirements. Geneva : International Organization for Standardization, 2022.
URL: <https://www.iso.org/standard/27001>
5. ISA/IEC 62443 Series of Standards. Industrial automation and control systems security. URL: <https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards>
6. MQTT Version 5.0. OASIS Standard. 2019. URL: <https://www.oasis-open.org/standard/mqtt-v5-0-os/>
7. Eclipse Mosquitto. An open source MQTT broker.
URL: <https://mosquitto.org/>
8. Fielding R. T. Architectural Styles and the Design of Network-based Software Architectures : doctoral dissertation. University of California, Irvine, 2000.
URL: <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>

9. Jones M., Bradley J., Sakimura N. RFC 7519: JSON Web Token (JWT).
Internet Engineering Task Force, 2015.
URL: <https://datatracker.ietf.org/doc/html/rfc7519>
10. OpenAPI Specification. Version 3.1.0. OpenAPI Initiative.
URL: <https://swagger.io/specification/>
11. Python 3.14.5 Documentation. Python Software Foundation.
URL: <https://docs.python.org/>
12. FastAPI Documentation. URL: <https://fastapi.tiangolo.com/>
13. PostgreSQL Documentation. The PostgreSQL Global Development Group. URL: <https://www.postgresql.org/docs/>
14. TimescaleDB Documentation. Tiger Data.
URL: <https://www.tigerdata.com/docs>
15. Docker Documentation. Docker Inc. URL: <https://docs.docker.com/>
16. React Documentation. Meta Open Source. URL: <https://react.dev/>
17. The TypeScript Handbook. Microsoft.
URL: <https://www.typescriptlang.org/docs/handbook/intro.html>
18. NumPy Documentation. NumPy Developers.
URL: <https://numpy.org/doc/stable/>
19. pandas Documentation. The pandas development team.
URL: <https://pandas.pydata.org/docs/>
20. scikit-learn User Guide. scikit-learn developers. URL: https://scikit-learn.org/stable/user_guide.html
21. SQLAlchemy Documentation. SQLAlchemy authors.
URL: <https://docs.sqlalchemy.org/>
22. OWASP Top Ten Web Application Security Risks. OWASP Foundation.
URL: <https://owasp.org/www-project-top-ten/>
23. Siemens Insights Hub. Industrial IoT application suite. Siemens.
URL: <https://www.siemens.com/en-us/products/insights-hub/>
24. IBM Maximo Application Suite. IBM.
URL: <https://www.ibm.com/products/maximo>

25. Azure IoT Documentation. Microsoft Learn.
URL: <https://learn.microsoft.com/en-us/azure/iot/>
26. ThingWorx Industrial IoT Software. PTC.
URL: <https://www.ptc.com/en/products/thingworx>
27. Çınar Z. M., Nuhu A. A., Zeeshan Q., Korhan O., Asmael M., Safaei B. Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0. Sustainability. 2020. Vol. 12, No. 19. Article 8211. DOI: <https://doi.org/10.3390/su12198211>
28. Ferreira C., Gonçalves G. Remaining Useful Life prediction and challenges: A literature review on the use of Machine Learning Methods. Journal of Manufacturing Systems. 2022. Vol. 63. P. 550–562. DOI: <https://doi.org/10.1016/j.jmsy.2022.05.010>
29. Elkateb S., Métwalli A., Shendy A., Abu-Elanien A. E. B. Machine learning and IoT-based predictive maintenance approach for industrial applications. Alexandria Engineering Journal. 2024. Vol. 88. P. 298–309. DOI: <https://doi.org/10.1016/j.aej.2023.12.065>
30. Wu F., Wu Q., Tan Y., Xu X. Remaining Useful Life Prediction Based on Deep Learning: A Survey. Sensors. 2024. Vol. 24, No. 11. Article 3454. DOI: <https://doi.org/10.3390/s24113454>
31. Hamasha M. M., Albedoor Q., Hamasha S., Ali H., Qamar A., Berrah F. A comprehensive framework for IoT-driven predictive maintenance: Leveraging AI and edge computing for enhanced equipment reliability. Journal of Applied Engineering Science. 2025. Vol. 23, No. 3. P. 471–486. DOI: <https://doi.org/10.5937/jaes0-57002>

ДОДАТОК А

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ

І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

Декларація про академічну доброчесність та використання ШІ

ДЕКЛАРАЦІЯ ЗДОБУВАЧА

Я, Харічев Владислав Янович, здобувач(ка) НУБіП України, усвідомлюючи відповідальність за порушення академічної доброчесності, цією заявою підтверджую, що:

1. Моя бакалаврська кваліфікаційна робота на тему «Інтелектуальна система моніторингу та прогнозування технічного стану обладнання на основі IoT і машинного навчання» є результатом моєї особистої праці.

2. Усі запозичення з текстів інших авторів мають відповідні посилання згідно з чинними стандартами.

3. Щодо використання інструментів ШІ зазначаю, що (обрати необхідне): ☒ інструменти генеративного ШІ не використовувалися.

☒ інструменти ШІ (ChatGPT) були використані для:

- ☐ ☐ перекладу іншомовних джерел;
- ☐ ☒ стилістичного редагування та перевірки граматики; ■
- ☐ ☒ допомоги у написанні/відлагодженні програмного коду;
- ☐ ☐ інше: _____.

Я розумію, що надання недостовірної інформації може призвести до скасування результатів захисту та відрахування з числа здобувачів НУБіП України.

«22» травня 2026 р.

_____ **Владислав ХАРІЧЕВ**