

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## **ПОЯСНЮВАЛЬНА ЗАПИСКА**

до кваліфікаційної роботи освітнього ступеня «бакалавр»  
за спеціальністю 121 «Інженерія програмного забезпечення»  
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Вебсистема управління технічним  
обслуговуванням для виробництва корму для риб»**

Виконав студент групи ПЗ-22-4  
КОЗАЧОК Денис Петрович

Керівник роботи:  
ВАКАЛЮК Тетяна Анатоліївна

Рецензент:  
ЄФРЕМОВ Юрій Миколайович

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**«ЗАТВЕРДЖУЮ»**

Зав. кафедри інженерії  
програмного забезпечення  
Тетяна Вакалюк  
**«28» лютого 2026 р.**

**ЗАВДАННЯ**  
**на кваліфікаційну роботу**

Здобувач вищої освіти: **КОЗАЧОК Денис Петрович**

Керівник роботи: **ВАКАЛЮК Тетяна Анатоліївна**

Тема роботи: **«Вебсистема управління технічним обслуговуванням для виробництва корму для риб»,**

затверджена Наказом закладу вищої освіти від **«27» лютого 2026 р., №92/с**

Вихідні дані для роботи: **об'єктом дослідження є процеси управління технічним обслуговуванням обладнання на підприємстві з виробництва корму для риб; предметом дослідження є методи та інструментальні засоби розробки вебсистеми управління технічним обслуговуванням, призначеної для обліку, планування та контролю технічного стану обладнання.**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

| Розділ | Консультант | Завдання видав | Завдання прийняв |
|--------|-------------|----------------|------------------|
| 1      | Вакалюк Т.А | 27.02.2024р.   | 27.02.2024р.     |
| 2      | Вакалюк Т.А | 24.04.2024р.   | 24.04.2024р.     |
| 3      | Вакалюк Т.А | 02.05.2024р.   | 02.05.2024р.     |

## РЕФЕРАТ

Пояснювальна записка до бакалаврської кваліфікаційної роботи на тему:  
«Веб-система управління технічним обслуговуванням (CMMS) для виробництва корму для риб» містить 119 сторінок, 68 рисунків, 30 використаних джерел та 2 додаток.

Метою роботи є створення вебсистеми (CMMS) для автоматизації процесів обліку виробничих потужностей, планування планово-попереджувальних ремонтів (ППР), реєстрації та контролю виконання заявок на обслуговування обладнання підприємства з виготовлення кормів для риб.

У роботі проаналізовано принципи побудови сучасних CMMS-систем і розроблено програмне рішення на базі PHP-фреймворку Laravel 11, СКБД MySQL та Docker. Реалізовано рольову модель доступу (адміністратор, інженер, технік) та механізм логування дій користувачів (Audit Log).

Система містить модулі обліку обладнання (з автоматичним розрахунком лінійної амортизації та поточної вартості активів), планування профілактичних робіт за календарем, ведення заявок на ремонт із розподілом витрат на оплату праці та запчастини, а також складського обліку деталей із контролем лімітів та списанням. Інтерфейс підтримує генерацію аналітичних звітів щодо простоїв обладнання та фінансових витрат на ремонт із можливістю експорту в PDF.

Запропоноване рішення дозволяє мінімізувати аварійні зупинки конвеєрних ліній, оптимізувати складські запаси деталей та підвищити прозорість витрат на ТО.

**КЛЮЧОВІ СЛОВА:** CMMS, ТЕХНІЧНЕ ОБСЛУГОВУВАННЯ, ОБЛАДНАННЯ, ПЛАНУВАННЯ ТО, ЗАЯВКА НА РЕМОНТ, СКЛАДСЬКИЙ ОБЛІК, АНАЛІЗ ПРОСТОїв, АМОРТИЗАЦІЯ АКТИВів, LARAVEL.

|           |      |               |        |      |                                                                                     |      |      |         |
|-----------|------|---------------|--------|------|-------------------------------------------------------------------------------------|------|------|---------|
|           |      |               |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ                                                            |      |      |         |
| Змн.      | Арк. | № докум.      | Підпис | Дата | Вебсистема управління<br>технічним обслуговуванням<br>для виробництва корму для риб | Лім. | Арк. | Аркушів |
| Розроб.   |      | Козачок Д. П. |        |      |                                                                                     |      | 2    | 120     |
| Перевір.  |      | Вакалюк Т.А.  |        |      |                                                                                     |      |      |         |
| Керівник  |      |               |        |      |                                                                                     |      |      |         |
| Н. контр. |      | Тичина О.П.   |        |      |                                                                                     |      |      |         |
| Зав. каф. |      | Вакалюк Т.А.  |        |      | Житомирська політехніка,<br>група ІПЗ-22-4                                          |      |      |         |

## ABSTRACT

Explanatory note to the bachelor's qualification work on the topic:

"Web-based Computerized Maintenance Management System (CMMS) for fish feed production" contains 119 pages, 68 figures, 30 sources used, and 2 appendix.

The purpose of the work is to develop a web-based computerized maintenance management system (CMMS) to automate equipment asset tracking, schedule preventive maintenance (PM), register and control work orders for fish feed production facilities.

The study analyzes the architecture of modern CMMS platforms and implements a software solution based on the Laravel 11 PHP framework, MySQL DBMS, and Docker. The system integrates role-based access control (admin, engineer, technician) and user action logging (Audit Log).

The software features modules for equipment management (with automatic straight-line depreciation and current book value calculation), calendar-based preventive maintenance scheduling, work order tracking with labor and material cost division, and inventory management with minimum limit alerts. The user interface supports generating analytical reports on equipment downtime and maintenance costs with PDF export capabilities.

The proposed solution helps minimize unexpected conveyor line shutdowns, optimize spare parts inventory, and improve maintenance cost transparency.

**KEYWORDS:** CMMS, MAINTENANCE, EQUIPMENT, PREVENTIVE MAINTENANCE, WORK ORDER, INVENTORY MANAGEMENT, DOWNTIME ANALYSIS, ASSET DEPRECIATION, LARAVEL.

|      |      |               |        |      |                         |      |
|------|------|---------------|--------|------|-------------------------|------|
|      |      | Козачок Д. П. |        |      | ПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                         |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                         | 4    |

## ЗМІСТ

|                                                                                                                                    |    |
|------------------------------------------------------------------------------------------------------------------------------------|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....                                                                                                    | 7  |
| ВСТУП .....                                                                                                                        | 8  |
| РОЗДІЛ 1.АНАЛІЗ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ                                                                              | 10 |
| 1.1    Постановка задачі .....                                                                                                     | 10 |
| 1.2    Аналіз аналогів програмного продукту .....                                                                                  | 12 |
| 1.3    Вибір архітектури системи .....                                                                                             | 16 |
| 1.4    Обґрунтування вибору інструментальних засобів та вимоги до<br>апаратного забезпечення .....                                 | 19 |
| Висновок до першого розділу .....                                                                                                  | 22 |
| РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБСИСТЕМИ УПРАВЛІННЯ ТЕХНІЧНИМ<br>ОБСЛУГОВУВАННЯМ ОБЛАДНАННЯ .....                                         | 23 |
| 2.1    Визначення варіантів використання .....                                                                                     | 23 |
| 2.2    Розробка бази даних системи .....                                                                                           | 32 |
| 2.3    Проєктування та реалізація алгоритмів роботи системи.....                                                                   | 46 |
| 2.4    Реалізація функціоналу системи управління технічним<br>обслуговуванням.....                                                 | 55 |
| Висновки до другого розділу.....                                                                                                   | 60 |
| РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБСИСТЕМОЮ<br>УПРАВЛІННЯ ТЕХНІЧНИМ ОБСЛУГОВУВАННЯМ ДЛЯ ВИРОБНИЦТВА<br>КОРМУ ДЛЯ РИБ ..... | 61 |
| 3.1    Порядок встановлення та налаштування параметрів платформи .....                                                             | 61 |
| 3.2    Структура інтерфейсу веб-системою контролю за технічним<br>обслуговуванням на виробництві корму для риб .....               | 64 |
| 3.3    Тестування системи.....                                                                                                     | 83 |
| Висновки до третього розділу .....                                                                                                 | 91 |

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 5    |

|                          |     |
|--------------------------|-----|
| ВИСНОВКИ .....           | 92  |
| ДЖЕРЕЛА ІНФОРМАЦІЇ ..... | 93  |
| ДОДАТКИ.....             | 96  |
| Додаток А.....           | 97  |
| Додаток Б .....          | 117 |

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 6    |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

**API** - прикладний програмний інтерфейс

**БД** - база даних

**ПЗ** - програмне забезпечення

**СУБД** - система управління базами даних

**MVC** - схема розділення даних додатків і керуючої логіки на три окремі компоненти Model-View-Controller

**CRUD** - основні операції, що використовуються для управління даними (Create, Read, Update, Delete)

**HTTP / HTTPS** - протокол передавання гіпертексту

**JSON** - текстовий формат обміну даними

**JWT** - відкритий стандарт для створення токенів доступу (використовується для автентифікації)

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 7    |

## ВСТУП

**Актуальність теми.** В умовах розвитку промисловості та автоматизації виробничих процесів зростає потреба в ефективному управлінні технічним обслуговуванням обладнання. Особливо це актуально для підприємств із виробництва кормів для риб, де безперервність роботи технологічних ліній безпосередньо впливає на обсяги виробництва, якість продукції та економічні показники підприємства.

Несправності обладнання, несвоєчасне проведення профілактичних робіт та відсутність систематизованого обліку ремонтів можуть призводити до простоїв виробництва, збільшення витрат на ремонт та зниження ефективності роботи підприємства. Тому виникає потреба у впровадженні інформаційних систем, що дозволяють автоматизувати процеси планування технічного обслуговування, обліку ремонтних робіт, контролю стану обладнання та управління запасами запчастин.

Існуючі програмні рішення не завжди враховують специфіку виробництва кормів, особливості роботи технологічних ліній та потреби персоналу підприємств. Це зумовлює актуальність розробки спеціалізованої веб-системи управління технічним обслуговуванням (CMMS), орієнтованої на використання сучасних архітектурних підходів та інструментальних засобів.

**Метою** кваліфікаційної роботи є аналіз предметної області та проєктування веб-системи управління технічним обслуговуванням обладнання на підприємстві з виробництва корму для риб.

Для досягнення поставленої мети в процесі виконання кваліфікаційної роботи необхідно вирішити такі завдання:

- дослідити існуючі програмні аналоги та визначити їхні переваги й недоліки;
- провести аналіз предметної області технічного обслуговування виробничого обладнання;
- сформулювати вимоги до веб-системи CMMS;
- обґрунтувати вибір архітектури системи;

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 8    |

- визначити інструментальні засоби реалізації програмного продукту та вимоги до апаратного забезпечення;
- виконати проєктування структури системи з використанням UML-діаграм.

**Об’єктом дослідження** є процеси технічного обслуговування та ремонту виробничого обладнання із застосуванням інформаційних технологій.

**Предметом дослідження** є методи та засоби проєктування веб-систем управління технічним обслуговуванням обладнання підприємств.

**Практичне значення** одержаних результатів полягає у створенні вебплатформи, яка забезпечує централізоване управління процесами технічного обслуговування та ремонтів (CMMS) виробничого обладнання і може бути використана промисловими підприємствами, інженерними службами та технічними відділами у сфері виготовлення кормів для риби та інших галузях харчової промисловості.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 9    |

# РОЗДІЛ 1. АНАЛІЗ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

## 1.1 Постановка задачі

Активний розвиток інформаційних технологій сприяє їх широкому застосуванню для автоматизації виробничих процесів на промислових підприємствах. Використання сучасних програмних систем дозволяє підвищити ефективність експлуатації обладнання, скоротити час на виконання технічного обслуговування, зменшити кількість аварійних зупинок, а також забезпечити централізований облік і обробку виробничих даних.

Під час дослідження предметної області в межах виконання кваліфікаційної роботи було встановлено, що на підприємствах із виробництва корму для риб значна частина процесів технічного обслуговування обладнання здійснюється з використанням паперової документації або розрізнених програмних засобів. Такий підхід ускладнює контроль виконання робіт, планування профілактичних заходів, облік використаних матеріалів.

Аналіз існуючих програмних рішень показує, що багато з них не повною мірою відповідають сучасним вимогам виробничих підприємств. Основними проблемами є обмежений функціонал, складність адаптації до специфіки конкретного виробництва, труднощі інтеграції з іншими інформаційними системами, а також недостатній рівень масштабованості та захисту даних. Часто такі системи орієнтовані на загальні сценарії використання та потребують значних доопрацювань для застосування у виробництві кормів.

У зв'язку з цим виникає потреба у створенні спеціалізованої веб-системи управління технічним обслуговуванням (CMMS), яка забезпечить комплексну автоматизацію процесів обліку обладнання, планування технічних робіт, контролю виконання ремонтів та управління запасами запчастин. Така система повинна мати зручний користувацький інтерфейс, підтримувати сучасну архітектуру програмного забезпечення, забезпечувати можливість

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 10   |

подальшого розширення функціоналу та відповідати вимогам безпеки й надійності.

Основною метою розробки є створення та впровадження спеціалізованої вебсистеми управління технічним обслуговуванням (CMMS), яка б забезпечувала надійну та всебічну автоматизацію процесів підтримання працездатності виробничого обладнання, підвищувала продуктивність інженерно-технічного персоналу підприємства з виготовлення кормів для риб, а також гарантувала структуроване збереження й обробку інформації про поточний стан потужностей і виконані сервісні роботи.

Для успішного досягнення цієї мети було вирішено комплекс взаємопов'язаних науково-практичних завдань. Спершу виконано детальний аналіз специфіки експлуатації та регламентного обслуговування конвеєрних ліній на підприємствах з виготовлення кормів для риб, де кожна незапланована зупинка критично впливає на технологічний процес. На основі проведеного аналізу сформовано детальний перелік функціональних та нефункціональних вимог до проєктованої системи, що охоплюють модулі моніторингу обладнання, автоматичного планування ППР, обліку ремонтних витрат та складського контролю. Додатково здійснено критичне порівняння існуючих комерційних CMMS-рішень, що дозволило чітко виділити їхні недоліки, зокрема високу вартість впровадження та складність адаптації під специфічні потреби вітчизняного виробництва.

Важливим кроком стало наукове обґрунтування вибору архітектурних рішень для побудови системи, що базуються на сучасному PHP-фреймворку Laravel 11, СКБД MySQL та контейнеризації середовища розробки за допомогою Docker. Це дозволило забезпечити високий рівень масштабованості, безпеки даних через детальну ролєву модель та гнучкість інтеграції нових модулів.

Кінцевим результатом постановки задачі є повне концептуальне та технічне обґрунтування розробки вебсистеми, яке поєднало теоретичні моделі розподілу прав користувачів і планування ремонтів із практичною реалізацією

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 11   |

функціоналу, що безпосередньо спрямований на зниження аварійності та підвищення прозорості витрат на ТО.

## 1.2 Аналіз аналогів програмного продукту

З метою обґрунтування доцільності розробки власної веб-системи управління технічним обслуговуванням було проведено детальний аналіз існуючих програмних рішень на світовому ринку. Сучасні системи класу CMMS (Computerized Maintenance Management System) пропонують широкий спектр інструментів для автоматизації сервісних процесів, контролю стану активів та координації ремонтних бригад. Для проведення порівняльного аналізу було обрано найбільш поширені хмарні та промислові платформи, що лідирують у сегменті автоматизації технічного обслуговування та ремонтів, а саме Fiix, UpKeep та eMaint CMMS.

Першим розглянутим аналогом є хмарна веб-платформа Fiix, розроблена корпорацією Rockwell Automation [14]. Цей програмний продукт позиціонується як інтелектуальна система управління активами, що використовує інструменти штучного інтелекту для прогнозування поломок. Основним функціональним фокусом Fiix є ведення детального реєстру обладнання, автоматичне планування регламентних робіт за заданими інтервалами часу та контроль за виконанням нарядів. До переваг цієї платформи можна віднести високий ступінь зручності інтерфейсу користувача, наявність функціонального мобільного додатка для техніків, який дозволяє працювати безпосередньо біля обладнання, а також розвинений модуль обліку запасних частин та інтеграцію із зовнішніми ERP-системами. Однак використання Fiix на вітчизняних підприємствах малого та середнього бізнесу суттєво обмежується низкою недоліків. Головними серед них є висока вартість передплати за кожного окремого користувача, складність адаптації стандартних робочих процесів платформи під специфічні потреби локального виробництва, а також надлишкова складність аналітичних інструментів,

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 12   |

більшість із яких залишаються незадіяними на невеликих лініях із виготовлення кормів.

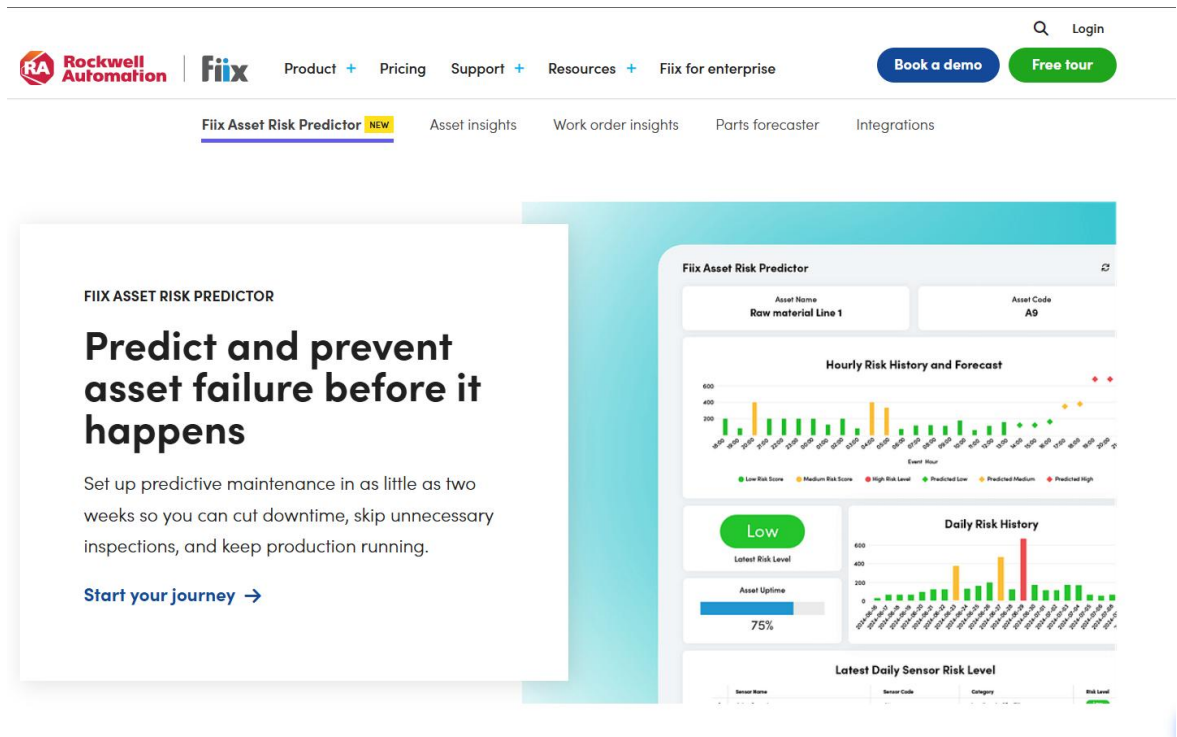


Рис.1.1 Інтерфейс платформи Fiix

Другим вагомим конкурентом у цьому сегменті є система UpKeep, яка орієнтована на мобільне використання та швидку координацію персоналу в режимі реального часу [15]. UpKeep спрощує комунікацію між операторами обладнання та ремонтними бригадами завдяки спрощеному процесу створення заявок на ремонт безпосередньо зі смартфонів. Функціональні можливості системи включають миттєву фіксацію несправностей, диспетчеризацію завдань, ведення базової історії обслуговування активів та формування простих аналітичних звітів щодо продуктивності техніків. Ключовою перевагою UpKeep є його простота та мінімальний час, необхідний для навчання персоналу роботі з інтерфейсом.

Водночас спрощеність системи є її основним мінусом, оскільки платформа пропонує обмежені можливості для глибокої кастомізації логіки переходів статусів замовлень-нарядів, не дозволяє налаштовувати специфічні правила розрахунку амортизації обладнання та вимагає значних фінансових

витрат при переході на корпоративні тарифи, які містять розширені функції звітності.

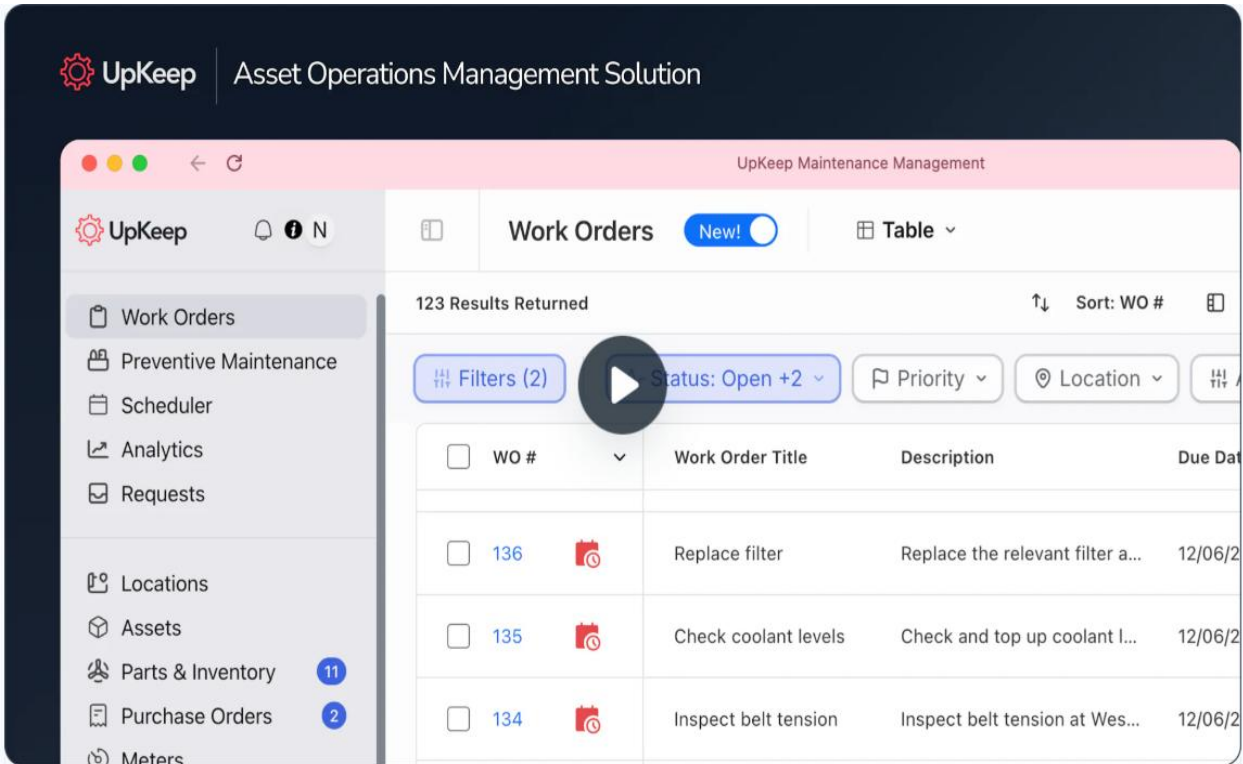


Рис. 1.2 Інтерфейс платформи UpKeep

Третім аналогом є eMaint CMMS від компанії Fluke Reliability, яка являє собою класичне індустріальне рішення для великих промислових об'єктів [16]. Ця система відрізняється глибокою функціональністю в галузі прогнозованого обслуговування, планування капітальних ремонтів та інтеграції з датчиками стану обладнання (IoT). Платформа eMaint надає потужні інструменти для ведення детальної ієрархії активів, управління закупівлями та запасами на багатьох складах, формування складних регуляторних звітів і гнучкого налаштування бізнес-правил під вимоги замовника. Беззаперечною перевагою eMaint є можливість повної кастомізації інтерфейсу та баз даних під будь-яку галузь промисловості.

Проте зворотною стороною такої універсальності є надзвичайна складність процесу впровадження та адміністрування системи. Користувачам потрібне тривале навчання для опанування заплутаного інтерфейсу, а початкове налаштування платформи вимагає залучення сертифікованих

консультантів, що робить це рішення економічно та організаційно недоступним для підприємств середнього масштабу.

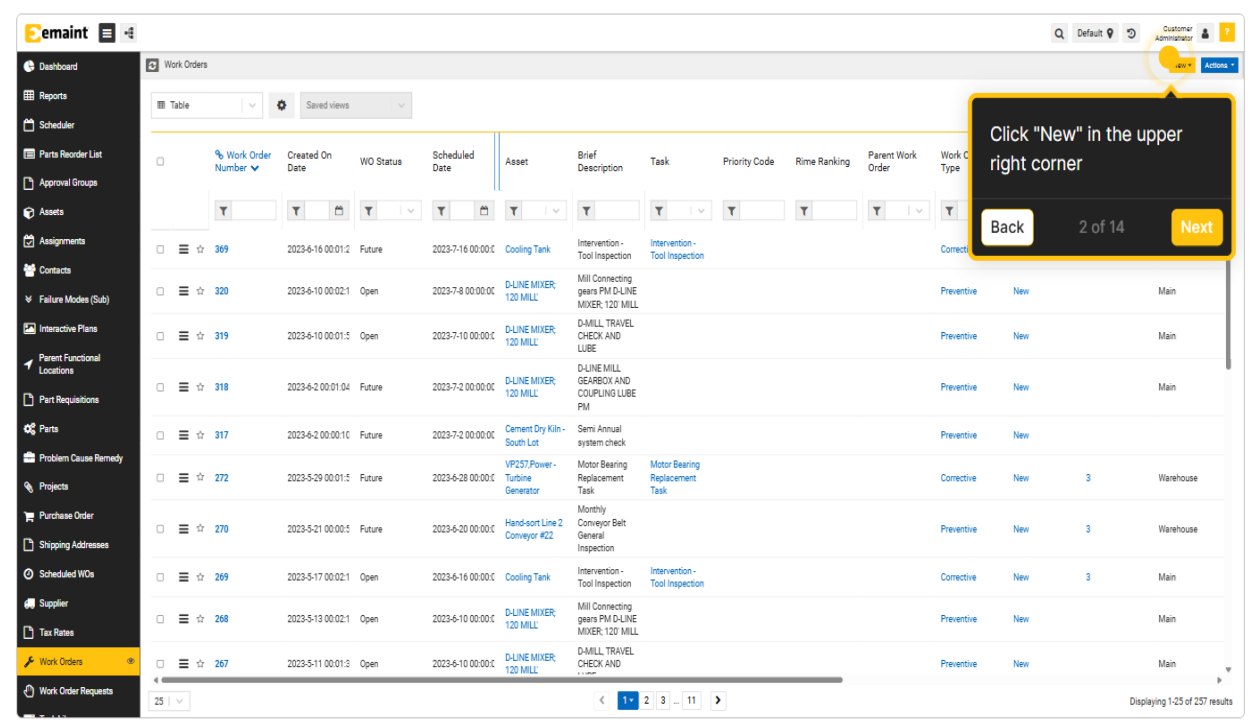


Рис. 1.3 Інтерфейс платформи eMaint CMMS

Проведений порівняльний аналіз програмних рішень Fiix, UpKeep та eMaint CMMS підтвердив, що сучасний ринок пропонує переважно універсальні комерційні SaaS-платформи. Вони володіють широким інструментарієм, але характеризуються значною вартістю ліцензій, залежністю від закордонних хмарних серверів, відсутністю локальної підтримки та складністю інтеграції зі специфічними фінансовими моделями обліку (зокрема, розрахунком заробітної плати ремонтних бригад за пріоритетами та автоматичним капіталізованим перерахунком залишкової вартості обладнання після ремонту). Для оптимізації процесів обслуговування на підприємстві з виробництва кормів для риб виникає потреба у створенні легкої, локально розгорнутої веб-системи, яка б поєднувала необхідний мінімум технічних функцій (реєстр активів, заявки на ремонт, графіки ТО, облік запчастин) із гнучким модулем фінансового обліку та аудиту дій користувачів, повністю адаптованим під структуру конкретного вітчизняного підприємства.

### 1.3 Вибір архітектури системи

Під час проєктування вебсистеми управління технічним обслуговуванням обладнання одним із ключових етапів є вибір архітектури програмного продукту. Саме архітектурні рішення визначають продуктивність, масштабованість, зручність супроводу та можливість подальшого розвитку. Розроблювана CMMS (Computerized Maintenance Management System) орієнтована на автоматизацію управління активами, планування технічного обслуговування, облік ремонтів та контроль використання запасних частин. Це вимагає стабільної архітектурної основи, здатної підтримувати поступове розширення функціональних можливостей. Неправильний вибір архітектури на початковому етапі може суттєво ускладнити підтримку системи та збільшити витрати на її подальшу модернізацію.

У процесі аналізу було розглянуто декілька архітектурних підходів: монолітну та багаторівневу клієнт-серверну архітектури. Монолітна архітектура характеризується відносною простотою реалізації, централізованою структурою та зручністю розгортання. Поряд із перевагами, вона має певні обмеження щодо незалежного масштабування окремих компонентів. Однак для проєктів початкового та середнього рівнів складності такий підхід забезпечує високу швидкість розробки та легкість у супроводі.

Враховуючи поточні технічні вимоги, обсяг проєкту та часові рамки, для реалізації програмного продукту було обрано монолітну архітектуру. Такий підхід дозволяє зосередити всю бізнес-логіку в межах одного серверного застосунку, що суттєво спрощує процеси розгортання, тестування та подальшого супроводу.

Розробка серверної частини виконується на базі мови програмування PHP, яка є стандартом для створення динамічних вебзастосунків та має розвинену екосистему інструментів. Для збереження структурованих даних застосовується реляційна система керування базами даних (СКБД) MySQL, що

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 16   |

забезпечує цілісність інформації, підтримку транзакцій та стабільну роботу під навантаженням. Структурну схему монолітної архітектури проєкту наведено на рисунку 1.4.

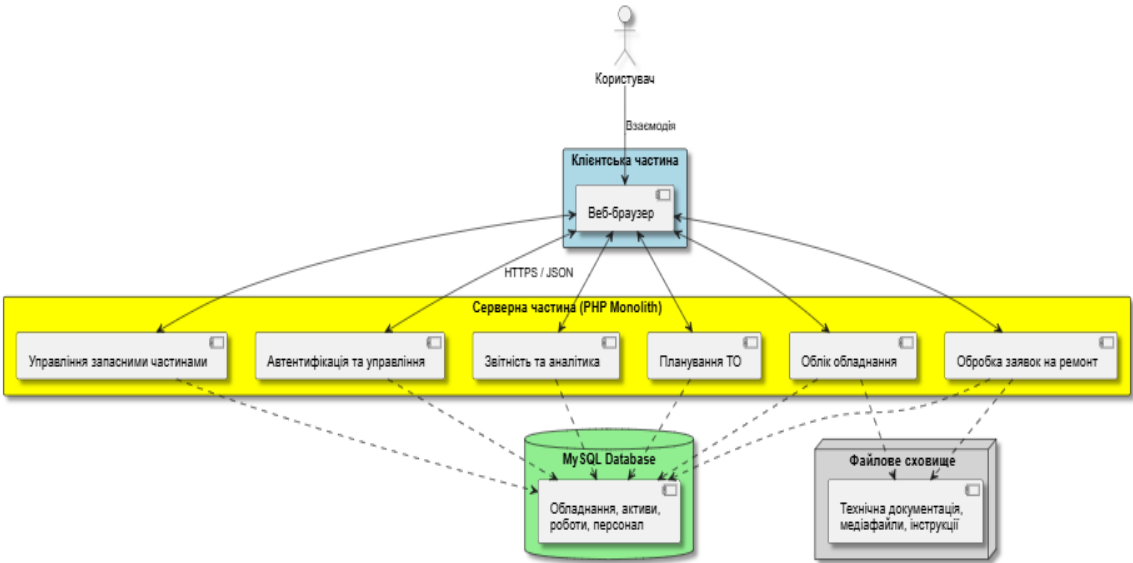


Рис. 1.4 Спрощена схема монолітної архітектури CMMS-системи

Наведена схема ілюструє структуру монолітної архітектури, де користувач взаємодіє із системою через веббраузер. Обмін даними між клієнтською та серверною частинами здійснюється за протоколом HTTPS із використанням формату JSON. Серверна частина об’єднує ключові функціональні модулі: автентифікації, управління користувачами, обліку обладнання, планування ТО, обробки заявок на ремонт, управління складом запасних частин, а також модулі звітності та аналітики.

Дані зберігаються в СКБД MySQL, де агрегується інформація про активи, історію робіт, персонал та матеріальні витрати. Для збереження технічної документації та медіафайлів використовується файлова система сервера, що дозволяє оптимізувати роботу бази даних та підвищити загальну продуктивність.

Для наочного представлення взаємодії користувача з програмним продуктом на рисунку 1.5 наведено спрощену схему роботи системи.

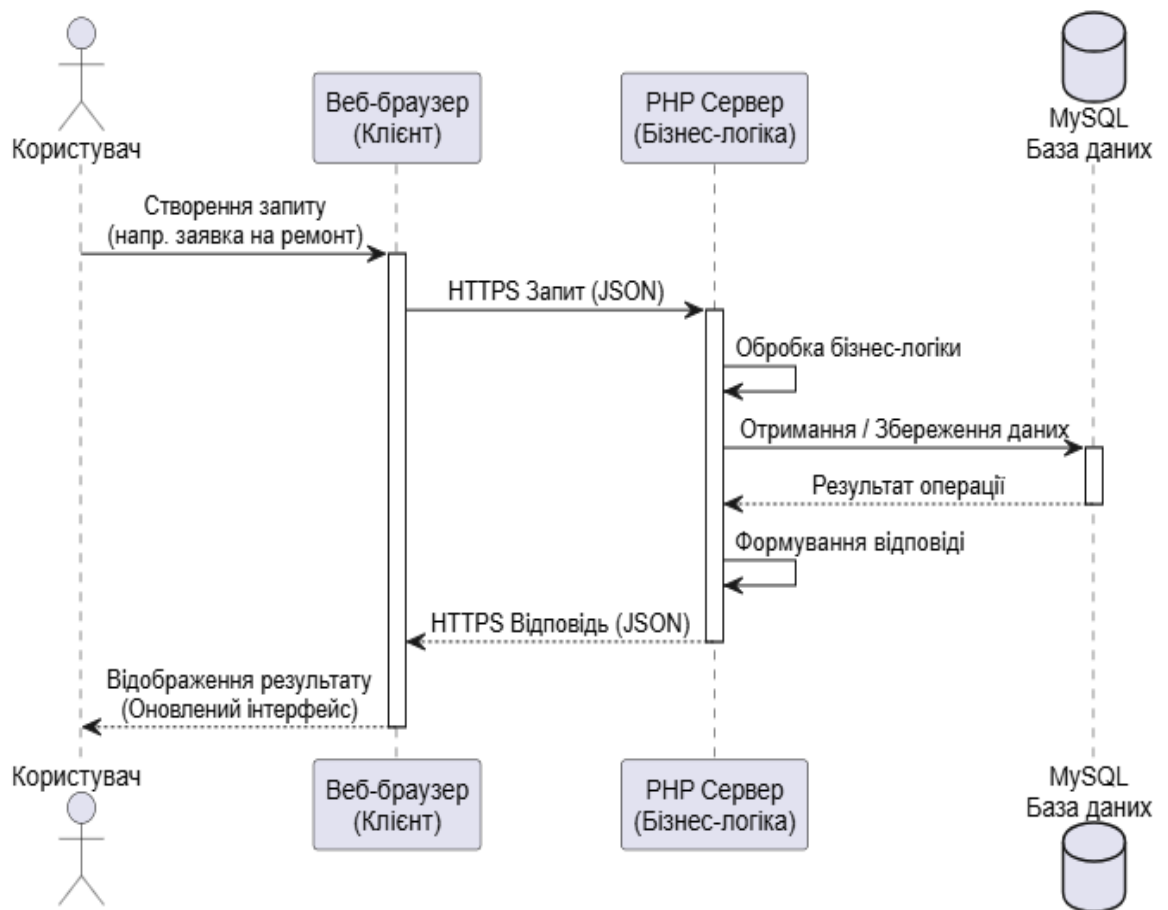


Рис. 1.5 Спрощена схема взаємодії користувача з CMMS-системою

Наведена схема демонструє типовий сценарій роботи: користувач створює заявку на ремонт або переглядає план технічного обслуговування через веб-інтерфейс. Клієнтська частина формує HTTPS-запит та передає його на сервер. Серверна частина обробляє запит, виконує необхідні операції бізнес-логіки та звертається до СКБД MySQL. Після обробки формується відповідь, яка повертається користувачу та відображається в інтерфейсі системи.

Запропонована структура демонструє централізований характер обробки запитів, що забезпечує узгодженість операцій, спрощує контроль бізнес-процесів та підвищує керованість програмного продукту. Таким чином, використання монолітної архітектури в поєднанні з технологіями PHP та MySQL є оптимальним рішенням на поточному етапі розробки CMMS-системи.

## 1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Для проектування, розробки та подальшого впровадження вебсистеми управління технічним обслуговуванням (CMMS) на підприємстві з виготовлення кормів для риб було обрано сучасний, збалансований та технологічно виправданий стек інструментів. При виборі програмних засобів ключовими критеріями виступали: швидкість розробки та прототипування, стабільність функціонування під навантаженням, безпека зберігання конфіденційних комерційних даних, простота інтеграції сторонніх бібліотек, а також кросплатформеність клієнтської частини.

Центральним елементом серверної архітектури системи виступає мова програмування PHP версії 8.2 у поєднанні з передовим веб-фреймворком Laravel 11. Вибір PHP 8.2 зумовлений суттєвим підвищенням продуктивності інтерпретатора порівняно з попередніми версіями, підтримкою строгої типізації, readonly-властивостей класів, покращеного синтаксису конструкцій та механізмів обробки помилок. Laravel 11 є лідером серед сучасних PHP-фреймворків завдяки чіткій реалізації архітектурного шаблону MVC (Model-View-Controller). Це дозволяє розробнику ізолювати бізнес-логіку системи від інтерфейсу користувача та механізмів безпосереднього збереження даних, спрощуючи підтримку, тестування та масштабування кодової бази.

Важливим аргументом на користь Laravel стало використання вбудованої системи об'єктно-реляційного відображення ORM Eloquent. Завдяки Eloquent взаємодія з таблицями бази даних відбувається на рівні об'єктів мови PHP. Це дозволяє повністю відмовитися від написання сирих SQL-запитів, що автоматично захищає додаток від найбільш поширених вразливостей, зокрема SQL-ін'єкцій (SQL Injection), оскільки ORM використовує параметризовані запити (PDO) під капотом.

Звіти про прості обладнання, графіки планово-попереджувальних ремонтів та фінансові відомості щодо залишкової вартості та амортизації

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 19   |

активів повинні мати можливість експорту в друковані формати. Для реалізації цієї вимоги обрано бібліотеку DomPDF (у форматизації пакета Barryvdh Laravel-DomPDF). Вона дозволяє рендерити складні адаптивні HTML-шаблони з використанням CSS-стилів безпосередньо у бінарний потік PDF-документа, забезпечуючи коректне відображення кирилиці та генерацію структурованих таблиць.

Для довгострокового збереження структурованої інформації (даних про верстати, конвеєри, персонал, склади запчастин, фінансові транзакції та журнали дій) обрано реляційну систему керування базами даних MySQL. Вона повністю відповідає вимогам стандарту ACID (атомарність, узгодженість, ізолюваність, довговічність), підтримує роботу із зовнішніми ключами для забезпечення цілісності зв'язків та транзакції, що виключає ситуації часткового збереження даних (наприклад, коли кількість деталей на складі зменшилась, але запис про використання в заявці не створився через збій). Крім того, MySQL оптимізована для роботи під керуванням ОС Linux у Docker-контейнерах та має високу швидкість обробки індексованих запитів.

Інтерфейс користувача CMMS-системи спроектовано як адаптивний вебзастосунок. Базовим інструментом візуалізації та стилізації обрано CSS-фреймворк Bootstrap 5. Його використання дозволяє швидко будувати інтерфейси за сітковою моделлю (Grid System), гарантуючи коректне відображення всіх елементів системи як на широких моніторах інженерів у диспетчерській, так і на сенсорних екранах смартфонів чи планшетів техніків безпосередньо біля конвеєрних ліній. Bootstrap 5 не потребує використання jQuery, що полегшує загальну вагу сторінок та прискорює їх рендеринг.

Візуалізація статистичних даних, таких як співвідношення аварійних та планових ремонтів, графік витрат на обслуговування по місяцях та тривалість простоїв конкретних одиниць обладнання, покладена на бібліотеку Chart.js. Вона використовує HTML5 Canvas для малювання плоских та кругових діаграм, підтримує анімацію при завантаженні та адаптивно масштабується під розміри батьківського контейнера.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 20   |

Збірка клієнтських скриптів та стилів здійснюється за допомогою інструменту Vite. Він забезпечує миттєве гаряче перезавантаження модулів (Hot Module Replacement) під час розробки та високий рівень стиснення і мініфікації коду при підготовці проекту до продакшн-розгортання, що значно прискорює первинне завантаження веб-сторінок користувачами із повільним мобільним інтернетом.

Таблиця 1.1.

**Таблиця системних вимог**

| Компонент системи                   | Мінімальні технічні вимоги                              | Призначення та опис                                                                   |
|-------------------------------------|---------------------------------------------------------|---------------------------------------------------------------------------------------|
| Серверна частина (Процесор)         | Чотириядерний процесор із тактовою частотою від 2.0 ГГц | Забезпечення швидкої обробки запитів та виконання черг завдань.                       |
| Серверна частина (ОЗУ)              | Не менше 8 ГБ оперативної пам'яті                       | Забезпечення стабільної спільної роботи вебсервера, СУБД MySQL та фреймворку Laravel. |
| Серверна частина (Накопичувач)      | Швидкісний дисковий простір типу SSD                    | Зберігання реляційної бази даних, файлів журналів аудиту та завантажених медіафайлів. |
| Серверна частина (Програмна основа) | PHP версії 8.2+ та СУБД MySQL версії 8.x                | Сумісність із кодовою базою та підтримка сучасних функцій безпеки.                    |
| Клієнтська частина (Пристрій)       | Персональний комп'ютер, ноутбук або мобільний пристрій  | Перегляд інтерфейсу та взаємодія із CMMS-системою.                                    |
| Клієнтська частина (Браузер)        | Сучасний веббраузер із підтримкою HTML5, CSS3 та ES6+   | Коректне відображення адаптивного UI-шаблону та виконання JavaScript-скриптів.        |
| Мережеве забезпечення               | Стабільне Інтернет-з'єднання або локальна мережа        | Обмін даними між клієнтськими пристроями та сервером підприємства.                    |

Використання обраних функціональних модулів CMMS-системи гарантує її стабільну роботу під навантаженням. Застосування обраного стеку технологій забезпечує ефективну реалізацію всіх функціональних можливостей системи та закладає фундамент для подальшого масштабування, підвищення продуктивності й відмовостійкості програмного продукту.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 21   |

## Висновок до першого розділу

У першому розділі було здійснено детальний аналіз процесів використання сучасних інформаційних технологій у сфері управління технічним обслуговуванням та ремонтами обладнання на промислових підприємствах із виробництва кормів для риб. Було досліджено особливості експлуатації виробничих ліній, сформульовано загальну постановку задачі, визначено ключові функціональні та нефункціональні вимоги до майбутньої системи, а також науково та практично обґрунтовано доцільність її створення.

Проведений порівняльний аналіз найбільш поширених комерційних аналогів (Fiix, UpKeep та eMaint CMMS) показав, що наявні ринкові рішення мають суттєві обмеження для використання на підприємствах середнього та малого бізнесу. Вони відрізняються надлишковою складністю інтерфейсу, високою вартістю ліцензування, відсутністю гнучких вітчизняних моделей фінансового розрахунку витрат на ремонт та жорсткою прив'язкою до хмарної інфраструктури розробників, що безпосередньо підтверджує необхідність створення власного спеціалізованого програмного продукту.

Також у розділі було проведено порівняльний аналіз архітектурних рішень і обґрунтовано вибір монолітної архітектури веб-додатка як найбільш оптимальної з точки зору швидкості розробки, простоти обслуговування та надійності. Визначено стек технологій, що базується на мові програмування PHP, фреймворці Laravel 11 та реляційній системі управління базами даних MySQL, а також детально описано вимоги до системного та апаратного забезпечення серверної та клієнтської частин. Отримані результати аналізу та обрані технологічні рішення слугують теоретичним та практичним підґрунтям для подальшого проектування бази даних, розробки алгоритмів та програмної реалізації веб-системи в межах дипломної роботи.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 22   |

## РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБСИСТЕМИ УПРАВЛІННЯ ТЕХНІЧНИМ ОБСЛУГОВУВАННЯМ ОБЛАДНАННЯ

### 2.1 Визначення варіантів використання

Для забезпечення безперебійного функціонування виробничих потужностей підприємства з виготовлення кормів для риб та мінімізації незапланованих зупинок конвеєрних ліній було визначено функціональні вимоги до проекрованої CMMS-системи. Основною метою розробки є автоматизація процесів реєстрації несправностей, планування регламентних робіт, контролю складських запасів запасних частин, а також розрахунку витрат на технічне обслуговування та ремонт обладнання.

Система базується на рольовій моделі доступу, що дозволяє чітко розмежувати права користувачів та забезпечити безпеку даних. Для моделювання взаємодії з системою було виділено чотири ключові актори, якими є адміністратор, інженер, технік та оператор. Кожна роль має чітко окреслену сферу відповідальності та набір дозволів у межах бізнес-процесів підприємства. Оператори виконують функції первинного моніторингу роботи обладнання на виробничих ділянках, переглядають його поточний стан, оперативно фіксують випадки раптових відмов та створюють первинні заявки на ремонт. Технічні фахівці безпосередньо виконують відновлювальні роботи, отримують призначені їм наряди на ремонт, змінюють їхні статуси у реальному часі, списують використані під час ремонту деталі зі складу та фіксують завершення робіт із внесенням фактично витраченого часу. Інженери здійснюють комплексне керування життєвим циклом обладнання та планування, створюють графіки планово-запобіжних робіт, ведуть реєстр активів, контролюють залишки запчастин, управляють фінансовими параметрами системи та аналізують причини і тривалість простоїв за допомогою аналітичного модуля з можливістю експорту звітів у формат PDF. Адміністратори забезпечують технічну підтримку працездатності платформи,

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 23   |

керують обліковими записами користувачів та мають повний доступ до журналу аудиту для контролю дій персоналу.

З точки зору функціональної структури система забезпечує наскрізний інформаційний потік. Процес взаємодії розпочинається з проходження автентифікації користувача, після чого він отримує доступ до головної інформаційної панелі. На цій панелі в реальному часі виводяться ключові показники ефективності, графіки розподілу витрат на обслуговування, кількість активних простоїв та інтерактивний календар найближчих регламентних робіт. Це дозволяє керівництву та інженерам оперативно оцінювати поточний стан обслуговування активів.

Модуль обліку обладнання дозволяє зберігати детальні технічні паспорти кожної одиниці техніки, включаючи інвентарні номери, місце встановлення, поточний експлуатаційний статус, амортизаційну інформацію та накопичену історію ремонтних втручань. При виникненні несправності система супроводжує життєвий цикл заявки від створення оператором до фінального закриття техніком. Під час виконання ремонту здійснюється списання запчастин через надсилання запиту до складського модуля, що безпосередньо впливає на розрахунок собівартості робіт. Система автоматично прораховує фінансові витрати на кожне робоче замовлення, об'єднуючи вартість витрачених матеріалів та заробітну плату виконавця, розраховану на основі встановлених годинних ставок, штрафних коефіцієнтів за перевищення планового часу та бонусів за дострокове завершення робіт.

Важливою складовою системи є модуль обліку простоїв, який автоматично фіксує часові інтервали непрацездатності техніки при зміні статусу наряду на «В роботі» та закриває їх при завершенні ремонту, що дозволяє інженерам аналізувати сумарний час простою обладнання. Для попередження аварійних зупинок система підтримує автоматичну генерацію планових нарядів за допомогою консольної команди на основі налаштованих графіків обслуговування, а також надсилає сповіщення про наближення термінів ТО. Вся бізнес-логіка покривається модулем звітності, який на основі

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 24   |

накопичених даних генерує аналітичні звіти щодо витрат на амортизацію, структури простоїв, фінансових витрат та продуктивності праці персоналу. Для забезпечення прозорості та відповідності стандартам безпеки будь-які модифікації даних логуються у спеціалізованому модулі аудиту, який зберігає інформацію про автора змін, тип операції, час та детальний опис оновлених значень.

Об'єктом комп'ютеризації є процес управління технічним обслуговуванням і ремонтом обладнання на підприємстві. У традиційному підході значна частина операцій виконується вручну, що призводить до зниження ефективності управління технічними ресурсами. Пошук інформації про попередні ремонти здійснюється за допомогою паперових журналів, відсутня єдина система обліку простоїв обладнання, а дані про використання запчастин зберігаються фрагментарно. Крім того, існує ризик несвоєчасного виконання регламентних робіт з технічного обслуговування, а документація зберігається у неструктурованому вигляді.

Розроблювана веб-система забезпечує автоматизацію зазначених процесів шляхом централізованого зберігання та обробки інформації. Основними сутностями є користувачі, виробниче обладнання, заявки на ремонт, плани технічного обслуговування, історія виконаних ремонтних робіт, запаси деталей та журнал дій користувачів. Такий підхід дозволяє забезпечити цілісність даних, підвищити контроль за виконанням робіт та покращити процес прийняття управлінських рішень.

Функціональні можливості системи охоплюють управління користувачами, облік обладнання, організацію ремонтних робіт, планування технічного обслуговування, матеріальний облік та формування звітності. Система забезпечує реєстрацію та авторизацію користувачів, розмежування прав доступу відповідно до їх ролей за допомогою Spatie Laravel-Permission, а також ведення журналу дій для аудиту операцій. Для обліку обладнання передбачено перегляд переліку активів, додавання нових одиниць техніки,

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 25   |

редагування їхніх характеристик, а також пошук і фільтрацію інформації за різними критеріями.

Підсистема управління ремонтами дозволяє створювати заявки на ремонт, відстежувати стан їх виконання, призначати відповідальних осіб та накопичувати інформацію про всі проведені роботи. Для підтримки профілактичного обслуговування реалізовано механізми планування графіків технічного обслуговування, розрахунку наступної дати ТО та автоматичного нагадування про наближення запланованих робіт. Матеріальний облік забезпечує фіксацію використання запчастин під час виконання ремонтних операцій та дозволяє контролювати витрати матеріальних ресурсів і залишків на складі. Особливе значення має підсистема звітності та аналітики, яка надає можливість формувати звіти щодо виконаних робіт, аналізувати причини простоїв обладнання, а також відображати ключові показники діяльності на інформаційній панелі керування.

До нефункціональних вимог системи належать забезпечення високої продуктивності, надійності, безпеки, масштабованості, зручності використання та можливості подальшого розвитку. Система повинна забезпечувати швидке створення заявок, ефективний пошук даних та стабільну роботу навіть при збільшенні кількості обладнання і накопиченні значних обсягів інформації. Надійність досягається шляхом збереження історії ремонтів у стійкому сховищі даних та організації регулярного резервного копіювання бази даних.

Безпека системи забезпечується використанням сучасних механізмів автентифікації та авторизації, реалізованих засобами пакету Laravel Breeze, а також чітким розмежуванням доступу до функціональних можливостей відповідно до ролі користувача за допомогою проміжних шарів (Middleware). Додатково передбачено захист персональних даних та контроль доступу до адміністративних функцій.

Архітектура програмного забезпечення побудована за модульним принципом і включає окремі підсистеми автентифікації, управління

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 26   |

обладнанням, ремонтами, технічним обслуговуванням, запасами та аналітикою. Такий підхід забезпечує масштабованість рішення та можливість інтеграції нових виробничих ліній або додаткових функціональних модулів у майбутньому. Для підвищення зручності роботи персоналу передбачено інтуїтивно зрозумілий адаптивний інтерфейс користувача та наочне відображення статистичних показників на панелі керування. Використання фреймворку Laravel 11 забезпечує зручність супроводу програмного коду, спрощує внесення змін та створює умови для подальшого розширення функціональних можливостей системи.



Рис. 2.1. Діаграма варіантів використання системи

Вимоги користувачів та адміністратора знаходять своє відображення у ключових сценаріях використання веб-системи, які охоплюють повний життєвий цикл експлуатації та обслуговування активів підприємства. Процес взаємодії розпочинається зі сценарію автентифікації та авторизації, що забезпечує гнучке розмежування прав доступу на основі ролей користувача. Наступним є сценарій обліку та перегляду обладнання, який дозволяє інженерному складу вести актуальний реєстр активів заводу, вносити нові агрегати та редагувати технічні характеристики існуючих. Сценарій планування профілактичного технічного обслуговування (ТО) дає змогу інженерам формувати регламентні графіки робіт, на основі яких система автоматично генерує нагадування та повідомлення для технічного персоналу.

Для безпосереднього реагування на відмови обладнання реалізовано сценарій створення заявок на ремонт та відстеження статусів їх виконання. У межах цього процесу оператор або диспетчер фіксує виникнення несправності, створюючи нове робоче замовлення, а призначений технік змінює його поточний стан у міру виконання відновлювальних операцій. Процес ремонту нерозривно пов'язаний зі сценарієм обліку списання запасних частин, який фіксує витрати матеріалів безпосередньо у картці відповідного замовлення. Інформація про незаплановані зупинки ліній обробляється у сценарії моніторингу простоїв, що дозволяє накопичувати дані для розрахунку загальної ефективності обладнання та формувати аналітичні звіти. Контроль дій користувачів та прозорість функціонування платформи забезпечуються через сценарій ведення журналу аудиту, який реєструє будь-які зміни в базі даних. Вся узагальнена інформація візуалізується на інтерактивній інформаційній панелі (дашборді), надаючи керівництву наочний огляд ключових показників у реальному часі.

На основі визначених бізнес-сценаріїв було спроектовано об'єктно-орієнтовану модель системи, яка базується на дев'яти ключових сутностях (класах моделей), що реалізовані у програмному коді додатку:

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 28   |

- User представляє користувачів системи з рольовим доступом та містить персональні дані працівників;
- Equipment описує одиницю виробничого обладнання заводу (наприклад, екструдер чи гранулятор), її характеристики, амортизаційні параметри та поточний технічний стан;
- WorkOrder представляє робоче замовлення (наряд на ремонт або регламентне ТО) із зазначенням термінів, пріоритету, вартості робіт та відповідальних осіб;
- MaintenanceSchedule визначає графік планово-запобіжних робіт для запобігання відмовам техніки;
- SparePart описує номенклатуру та кількість запасних частин, наявних на складі підприємства;
- SparePartUsage виступає сполучною сутністю, яка реєструє фактичне використання певної кількості деталей під час виконання конкретного ремонту;
- DowntimeLog фіксує часові інтервали вимушеного простою обладнання та причини його зупинки;
- AuditLog відповідає за ведення журналу дій користувачів системи, фіксуючи колишні та нові значення модифікованих полів;
- Setting зберігає глобальні конфігураційні параметри системи, зокрема фінансові ставки та коефіцієнти розрахунку вартості праці та нарахування премій з штрафами.

Усі об'єкти системи пов'язані між собою логічними зв'язками, які відображають реальну взаємодію сутностей. Між об'єктом обладнання (Equipment) та його робочими замовленнями (WorkOrder), плановими графіками обслуговування (MaintenanceSchedule) і журналами простоїв (DowntimeLog) реалізовано зв'язок типу агрегація. Це відношення типу «один-до-багатьох» означає, що замовлення, графіки чи записи простоїв логічно належать до конкретної одиниці техніки, проте у разі їх видалення сам об'єкт

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 29   |

обладнання продовжує існувати в базі даних, це забезпечує надійність програми та її гнучкість до будь яких сценаріїв.

Зв'язок типу композиція застосовано між класом робочого замовлення (WorkOrder) та записами про використання деталей (SparePartUsage). Це суворий тип зв'язку, оскільки запис про списання запчастини не може існувати без контексту конкретного виконаного ремонту. У разі видалення робочого замовлення всі пов'язані з ним записи використання деталей автоматично знищуються за каскадним принципом.

Зв'язок асоціація встановлено між користувачами (User) та робочими замовленнями (WorkOrder), де один користувач виступає автором запиту, а інший - його виконавцем. Також асоціацією пов'язані сутності користувачів із логами аудиту (AuditLog), оскільки кожен запис у журналі безпеки повинен посылатися на конкретного працівника, який здійснив дію, та сутність використаних деталей (SparePartUsage) із самим складським каталогом запасних частин (SparePart).

Для підвищення типізації та надійності обробки даних у системі використовуються перерахування (enumeration) для визначення статусів та рівнів доступу. Зокрема, перерахування ролей користувачів визначає права працівників як адміністраторів, інженерів, техніків чи операторів. Стан виконання наряду регламентується перерахуванням статусів робочого замовлення, які приймають значення: нова (new), в роботі (in\_progress), завершена (completed), відхилена (rejected) та призупинена (on\_hold).

Також для замовлень-нарядів на ремонт визначено перерахування пріоритетів виконання, до яких належать низький (low), середній (medium), високий (high) та критичний (critical). Поточний фізичний та експлуатаційний стан виробничих потужностей підприємства безперервно контролюється через перерахування статусів обладнання, що відображає такі ключові фази життєвого циклу техніки, як активна експлуатація (active), перебування в ремонті (in\_repair), виведення з експлуатації або списання (decommissioned), а також вимушений простій (idle) у разі зупинки лінії, що дозволяє

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 30   |

автоматизувати моніторинг готовності обладнання та збирати точну статистику його завантаженості

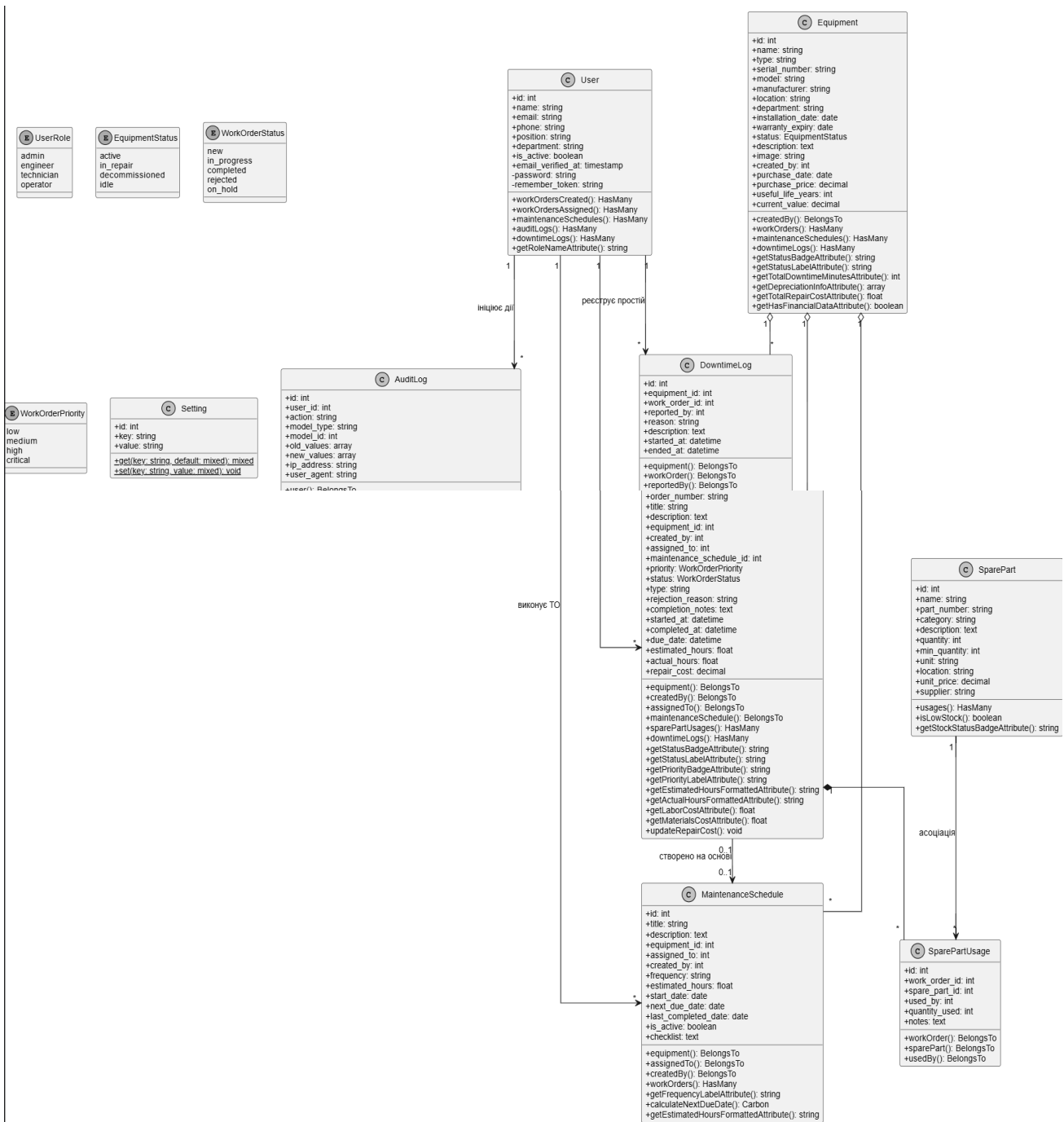


Рис. 2.2. Діаграма класів доменних моделей

Представлена діаграма класів доменних моделей відображає архітектурний каркас об'єктно-орієнтованої структури CMMS-системи та демонструє принципи взаємодії її компонентів. В основі цієї моделі лежить концепція забезпечення цілісності та простежуваності всіх процесів технічного обслуговування.

## 2.2 Розробка бази даних системи

Для забезпечення збереження та обробки даних у веб-системі контролю за технічним обслуговуванням на виробництві корму для риб було спроектовано реляційну базу даних, структура якої відображає основні бізнес-процеси платформи. База даних реалізує зберігання інформації про персонал підприємства, виробниче обладнання, заявки на ремонт, плани регламентного обслуговування, складський облік запасних частин, фінансові витрати, фіксацію простоїв та журнал дій користувачів. Також підтримується зберігання даних, пов'язаних із автентифікацією, розмежуванням прав доступу та системними налаштуваннями.

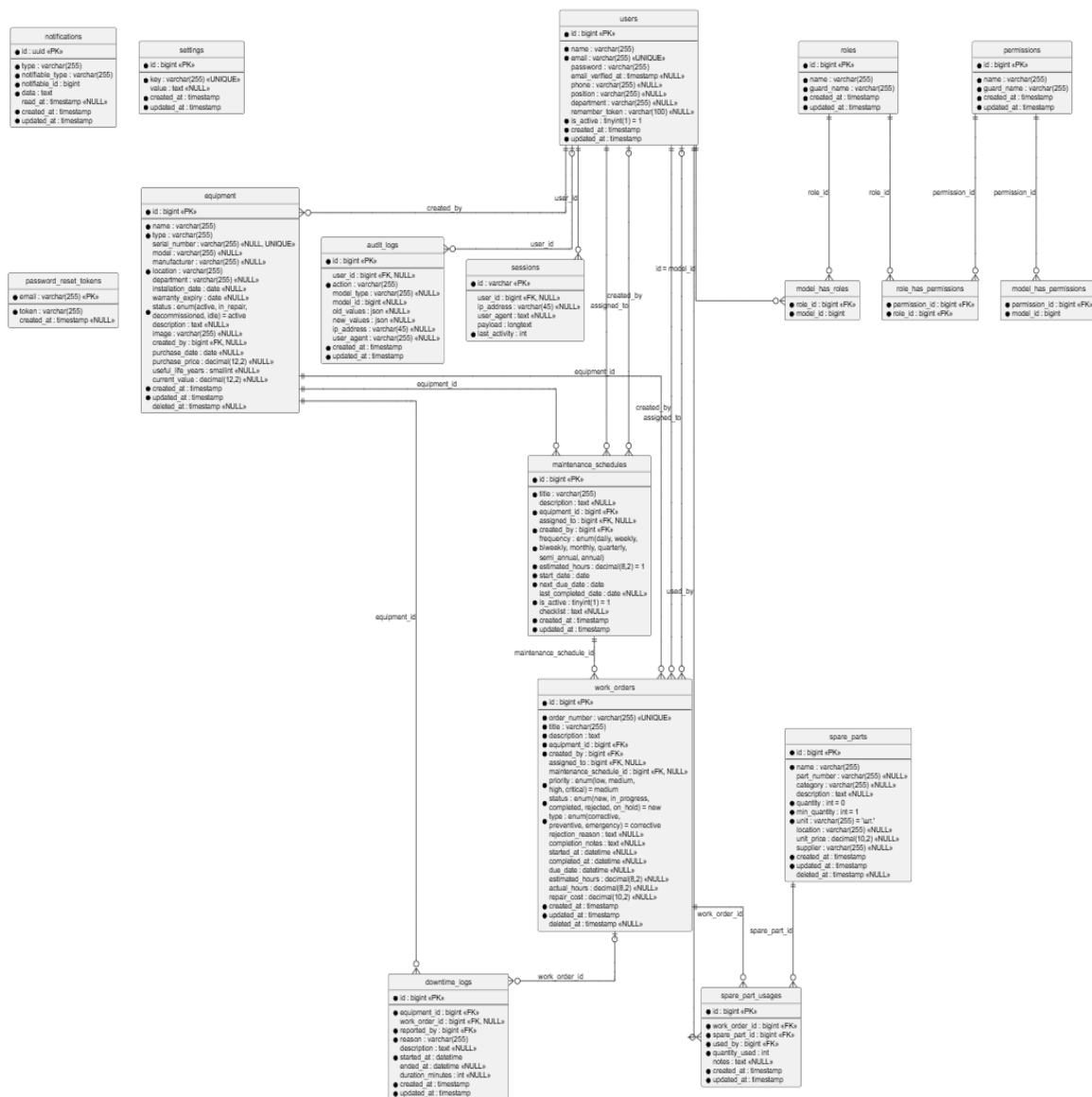


Рис. 2.3. Структура таблиць бази даних

|      |      |               |        |      |                          |
|------|------|---------------|--------|------|--------------------------|
|      |      | Козачок Д. П. |        |      | Арк.                     |
|      |      | Вакалюк Т.А   |        |      | ПІЗ.КР.Б – 121 – 26 – ПЗ |
| Змн. | Арк. | № докум.      | Підпис | Дата | 32                       |

Ядром бази даних є таблиця `users`, де зберігаються основні відомості для ідентифікації та авторизації співробітників підприємства: ім'я, адреса електронної пошти та захищений хеш паролю. Додаткові відомості у вигляді посади (`position`), підрозділу (`department`) та контактного телефону (`phone`) дозволяють чітко ідентифікувати відповідального виконавця при призначенні на ремонтні роботи. Поле `is_active` забезпечує можливість блокування доступу працівника без видалення його облікового запису з бази даних. Система керування доступом реалізована на базі пакету `Spatie Permission` через проміжні таблиці ролей і дозволів (`roles`, `permissions`, `model_has_roles`, `model_has_permissions`), де кожному користувачу призначається рівень повноважень, що визначає доступні функції системи.

Центральною сутністю предметної галузі є таблиця `equipment`, яка зберігає реєстр усіх виробничих активів підприємства. Вона містить ідентифікаційні характеристики кожного агрегату (найменування, тип, серійний номер, модель, виробник), його фізичне розміщення (`location`, `department`) та поточний експлуатаційний статус. Для підтримки фінансового обліку зносу активів до таблиці додано поля дати придбання (`purchase_date`), початкової вартості (`purchase_price`), нормативного строку корисного використання (`useful_life_years`) та розрахункової поточної залишкової вартості (`current_value`), що автоматично перераховується після кожного завершеного ремонту.

Основним операційним модулем системи виступає таблиця `work_orders`, яка фіксує всі ремонтні та регламентні роботи. Кожна заявка має унікальний номер (`order_number`), прив'язана до конкретного обладнання та має два ключових зовнішніх посилання на таблицю `users` - автора заявки (`created_by`) та призначеного виконавця (`assigned_to`). Поля перерахувань визначають тип роботи (`corrective`, `preventive`, `emergency`), пріоритет (`low`, `medium`, `high`, `critical`) та поточний статус виконання (`new`, `in_progress`, `completed`, `rejected`, `on_hold`). Для фінансового розрахунку у таблиці зберігаються планові та фактичні трудовитрати (`estimated_hours`, `actual_hours`), а загальна вартість ремонту

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 33   |

(repair\_cost) обраховується автоматично як сума вартості витраченої праці та матеріалів.

Планування профілактичних робіт реалізовано через таблицю maintenance\_schedules. Кожен запис у ній визначає регламент обслуговування конкретного агрегату із вказанням виконавця, частоти виконання (frequency), дати початку, а також автоматично обчислюваної наступної дати виконання (next\_due\_date). Поле checklist у форматі тексту дозволяє зберігати перелік кроків для виконання регламентних робіт.

Взаємодія між ремонтними роботами та складськими запасами підтримується парою таблиць spare\_parts та spare\_part\_usages. Таблиця spare\_parts зберігає номенклатурний каталог деталей із зазначенням одиниці вимірювання, поточного залишку, мінімального порогу (min\_quantity), одиничної вартості та постачальника. Таблиця spare\_part\_usages виступає проміжною сутністю, яка фіксує факт списання конкретної кількості деталей на конкретне замовлення, тим самим зменшуючи складський залишок та збільшуючи матеріальну складову вартості ремонту.

Таблиця downtime\_logs реєструє часові інтервали вимушених зупинок обладнання із зазначенням причини (reason), часу початку та завершення простою. Зв'язок із таблицею work\_orders через зовнішній ключ дозволяє визначити, під час виконання якого саме ремонту агрегат перебував у нероботоздатному стані.

Таблиця audit\_logs забезпечує повну прозорість операцій у системі, зберігаючи дані про автора дії, тип операції, модифіковану модель та порівняльний знімок змінених значень (old\_values, new\_values) у форматі JSON. Таблиця notifications зберігає сповіщення, що адресуються конкретним користувачам через поліморфний зв'язок. Таблиця settings слугує гнучким сховищем глобальних конфігураційних параметрів системи у форматі «ключ–значення», зокрема фінансових тарифних ставок та коефіцієнтів пріоритетів для розрахунку заробітної плати виконавців.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 34   |

Таблиця users (Користувачі)

| Назва             | Тип даних | PK | FK | Опис                                          |
|-------------------|-----------|----|----|-----------------------------------------------|
| id                | BIGINT    | +  |    | Унікальний ідентифікатор користувача.         |
| name              | VARCHAR   |    |    | Прізвище та ім'я працівника.                  |
| email             | VARCHAR   |    |    | Електронна пошта (унікальна).                 |
| password          | VARCHAR   |    |    | Хеш пароля для автентифікації.                |
| email_verified_at | TIMESTAMP |    |    | Дата підтвердження електронної пошти.         |
| phone             | VARCHAR   |    |    | Контактний номер телефону.                    |
| position          | VARCHAR   |    |    | Посада співробітника на заводі.               |
| department        | VARCHAR   |    |    | Цех або відділ, до якого належить користувач. |
| is_active         | BOOLEAN   |    |    | Статус активності облікового запису.          |
| remember_token    | VARCHAR   |    |    | Токен для функції «запам'ятати мене».         |

Для розмежування повноважень та забезпечення безпеки доступу до модулів системи (наприклад, заборони техніку редагувати фінансові

налаштування або списки обладнання) використовується рольова модель. Системні ролі користувачів описує табл. 2.2.

Таблиця 2.2.

**Таблиця roles (Ролі користувачів)**

| Назва      | Тип даних | PK | FK | Опис                                                |
|------------|-----------|----|----|-----------------------------------------------------|
| id         | BIGINT    | +  |    | Унікальний ідентифікатор ролі.                      |
| name       | VARCHAR   |    |    | Назва ролі (admin, engineer, technician, operator). |
| guard_name | VARCHAR   |    |    | Назва гарду автентифікації (web).                   |

Кожна роль надає певний набір прав на виконання операцій (дії з обладнанням, створення заявок тощо). Детальний перелік можливих системних дозволів та прав доступу наведено в табл. 2.3.

Таблиця 2.3.

**Таблиця permissions (Дозволи)**

| Назва      | Тип даних | PK | FK | Опис                                                              |
|------------|-----------|----|----|-------------------------------------------------------------------|
| id         | BIGINT    | +  |    | Унікальний ідентифікатор дозволу.                                 |
| name       | VARCHAR   |    |    | Системна назва дозволу (equipment.view, work-orders.create тощо). |
| guard_name | VARCHAR   |    |    | Назва гарду автентифікації (web).                                 |

Для забезпечення гнучкого зв'язку між конкретним обліковим записом працівника та його роллю використовується проміжна таблиця зв'язку багато-до-багатьох, структура якої наведена в табл. 2.4.

Таблиця 2.4.

**Таблиця model\_has\_roles (Призначення ролей користувачам)**

| Назва      | Тип даних | PK | FK | Опис                          |
|------------|-----------|----|----|-------------------------------|
| role_id    | BIGINT    | +  | +  | Посилання на таблицю roles.   |
| model_type | VARCHAR   | +  |    | Тип моделі (App\Models\User). |

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 36   |

продовж. табл. 2.4.

|          |        |   |  |                                                   |
|----------|--------|---|--|---------------------------------------------------|
| model_id | BIGINT | + |  | Ідентифікатор користувача, якому призначено роль. |
|----------|--------|---|--|---------------------------------------------------|

Аналогічно, кожна роль повинна мати чітко закріплені за нею дозволи. Цей зв'язок реалізується через окрему проміжну таблицю, структуру якої відображено в табл. 2.5.

Таблиця 2.5.

**Таблиця role\_has\_permissions (Дозволи ролей)**

| Назва         | Тип даних | PK | FK | Опис                              |
|---------------|-----------|----|----|-----------------------------------|
| permission_id | BIGINT    | +  | +  | Посилання на таблицю permissions. |
| role_id       | BIGINT    | +  | +  | Посилання на таблицю roles.       |

Окрім групових дозволів через ролі, архітектура Spatie передбачає можливість надання індивідуальних прав безпосередньо окремому користувачу. Специфікацію таблиці персональних прав наведено в табл. 2.6.

Таблиця 2.6.

**Таблиця model\_has\_permissions (Особисті дозволи користувачів)**

| Назва         | Тип даних | PK | FK | Опис                              |
|---------------|-----------|----|----|-----------------------------------|
| permission_id | BIGINT    | +  | +  | Посилання на таблицю permissions. |
| model_type    | VARCHAR   | +  |    | Тип моделі (App\Models\User).     |
| model_id      | BIGINT    | +  |    | Ідентифікатор користувача.        |

Авторизовані користувачі взаємодіють з основним матеріальним активом підприємства - виробничим обладнанням. Облік агрегатів, верстатів та конвеєрних ліній, а також розрахунок їх поточного стану й амортизації ведеться у таблиці, описаній у табл. 2.7.

Таблиця equipment (Обладнання)

| Назва             | Тип даних     | РК | FK | Опис                                                     |
|-------------------|---------------|----|----|----------------------------------------------------------|
| id                | BIGINT        | +  |    | Унікальний ідентифікатор одиниці обладнання.             |
| name              | VARCHAR       |    |    | Найменування обладнання.                                 |
| type              | VARCHAR       |    |    | Тип або категорія обладнання.                            |
| serial_number     | VARCHAR       |    |    | Серійний номер (унікальний).                             |
| model             | VARCHAR       |    |    | Модель обладнання.                                       |
| manufacturer      | VARCHAR       |    |    | Виробник обладнання.                                     |
| location          | VARCHAR       |    |    | Місце розташування агрегату в цеху.                      |
| department        | VARCHAR       |    |    | Виробничий цех або відділ.                               |
| installation_date | DATE          |    |    | Дата введення в експлуатацію.                            |
| warranty_expiry   | DATE          |    |    | Дата закінчення гарантійного строку.                     |
| status            | ENUM          |    |    | Поточний стан (active, in_repair, decommissioned, idle). |
| description       | TEXT          |    |    | Технічний опис обладнання.                               |
| image             | VARCHAR       |    |    | Шлях до фото обладнання.                                 |
| created_by        | BIGINT        |    | +  | Посилання на users - хто додав запис.                    |
| purchase_date     | DATE          |    |    | Дата придбання обладнання.                               |
| purchase_price    | DECIMAL(12,2) |    |    | Початкова вартість придбання.                            |
| useful_life_years | SMALLINT      |    |    | Нормативний строк корисного використання (роки).         |
| current_value     | DECIMAL(12,2) |    |    | Розрахункова залишкова вартість.                         |
| deleted_at        | TIMESTAMP     |    |    | Мітка м'якого видалення.                                 |

Для підтримання обладнання у працездатному стані та автоматичного запобігання поломкам інженери розробляють плани планово-попереджувальних ремонтів (ППР). Структуру графіків періодичного обслуговування наведено в табл. 2.8.

Таблиця 2.8.

**Таблиця maintenance\_schedules (Плани технічного обслуговування)**

| Назва               | Тип даних    | РК | FK | Опис                                                                |
|---------------------|--------------|----|----|---------------------------------------------------------------------|
| id                  | BIGINT       | +  |    | Унікальний ідентифікатор плану ТО.                                  |
| title               | VARCHAR      |    |    | Назва планового регламенту.                                         |
| description         | TEXT         |    |    | Детальний опис виду технічного обслуговування.                      |
| equipment_id        | BIGINT       |    | +  | Посилання на equipment - обладнання для обслуговування.             |
| assigned_to         | BIGINT       |    | +  | Посилання на users - відповідальний виконавець.                     |
| created_by          | BIGINT       |    | +  | Посилання на users - автор графіка.                                 |
| frequency           | ENUM         |    |    | Частота виконання (daily, weekly, monthly, quarterly, annual тощо). |
| estimated_hours     | DECIMAL(8,2) |    |    | Планові трудовитрати на виконання ТО.                               |
| start_date          | DATE         |    |    | Дата початку дії регламенту.                                        |
| next_due_date       | DATE         |    |    | Дата наступного запланованого виконання.                            |
| last_completed_date | DATE         |    |    | Дата останнього виконаного ТО.                                      |
| is_active           | BOOLEAN      |    |    | Ознака активності графіка.                                          |
| checklist           | TEXT         |    |    | Перелік кроків для виконання ТО.                                    |

На основі планів ТО або в разі раптової відмови техніки створюються та виконуються наряди на роботу. Структура сутності, яка координує всі ремонтні роботи та фіксує витрати часу й коштів, представлена в табл. 2.9.

Таблиця 2.9.

**Таблиця work\_orders (Заявки на ремонт)**

| Назва                   | Тип даних | PK | FK | Опис                                                              |
|-------------------------|-----------|----|----|-------------------------------------------------------------------|
| id                      | BIGINT    | +  |    | Унікальний ідентифікатор заявки.                                  |
| order_number            | VARCHAR   |    |    | Унікальний номер наряду (WO-XXXXXX).                              |
| title                   | VARCHAR   |    |    | Короткий заголовок заявки.                                        |
| description             | TEXT      |    |    | Детальний опис несправності або робіт.                            |
| equipment_id            | BIGINT    |    | +  | Посилання на equipment - обладнання, що ремонтується.             |
| created_by              | BIGINT    |    | +  | Посилання на users - ініціатор заявки.                            |
| assigned_to             | BIGINT    |    | +  | Посилання на users - призначений виконавець.                      |
| maintenance_schedule_id | BIGINT    |    | +  | Посилання на maintenance_schedules - якщо наряд плановий.         |
| priority                | ENUM      |    |    | Пріоритет виконання (low, medium, high, critical).                |
| status                  | ENUM      |    |    | Поточний статус (new, in_progress, completed, rejected, on_hold). |
| type                    | ENUM      |    |    | Тип роботи (corrective, preventive, emergency).                   |
| rejection_reason        | TEXT      |    |    | Причина відхилення заявки.                                        |
| completion_notes        | TEXT      |    |    | Примітки щодо виконаних робіт.                                    |
| started_at              | DATETIME  |    |    | Фактичний час початку виконання.                                  |
| completed_at            | DATETIME  |    |    | Фактичний час завершення.                                         |
| due_date                | DATETIME  |    |    | Планова дата виконання.                                           |

продовж. табл. 2.9.

|                 |               |  |  |                                                    |
|-----------------|---------------|--|--|----------------------------------------------------|
| estimated_hours | DECIMAL(8,2)  |  |  | Планові трудовитрати у годинах.                    |
| actual_hours    | DECIMAL(8,2)  |  |  | Фактичні трудовитрати у годинах.                   |
| repair_cost     | DECIMAL(10,2) |  |  | Автоматично розрахована загальна вартість ремонту. |
| deleted_at      | TIMESTAM<br>P |  |  | Мітка м'якого видалення.                           |

Під час виконання ремонту технік часто використовує матеріали зі складу. Для ведення детальної номенклатури деталей та моніторингу їх критичних залишків призначена таблиця, наведена в табл. 2.10.

Таблиця 2.10.

**Таблиця spare\_parts (Запасні частини)**

| Назва        | Тип даних     | PK | FK | Опис                                           |
|--------------|---------------|----|----|------------------------------------------------|
| id           | BIGINT        | +  |    | Унікальний ідентифікатор запчастини.           |
| name         | VARCHAR       |    |    | Найменування запасної частини.                 |
| part_number  | VARCHAR       |    |    | Артикул або номер деталі.                      |
| category     | VARCHAR       |    |    | Категорія                                      |
| description  | TEXT          |    |    | Технічний опис деталі.                         |
| quantity     | INT           |    |    | Поточна кількість одиниць на складі.           |
| min_quantity | INT           |    |    | Мінімально допустимий залишок (поріг сигналу). |
| unit         | VARCHAR       |    |    | Одиниця вимірювання (шт., кг, м тощо).         |
| location     | VARCHAR       |    |    | Місце зберігання на складі.                    |
| unit_price   | DECIMAL(10,2) |    |    | Вартість однієї одиниці.                       |
| supplier     | VARCHAR       |    |    | Постачальник запасної частини.                 |
| deleted_at   | TIMESTAMP     |    |    | Мітка м'якого видалення.                       |

Для того щоб пов'язати списання деталей зі складу з конкретною заявкою на ремонт і розрахувати собівартість матеріалів, використовується транзакційна таблиця списань, представлена в табл. 2.11.

Таблиця 2.11.

**Таблиця spare\_part\_usages (Списання запчастин)**

| Назва         | Тип даних | PK | FK | Опис                                                              |
|---------------|-----------|----|----|-------------------------------------------------------------------|
| id            | BIGINT    | +  |    | Унікальний ідентифікатор запису списання.                         |
| work_order_id | BIGINT    |    | +  | Посилання на work_orders - наряд, у межах якого витрачено деталь. |
| spare_part_id | BIGINT    |    | +  | Посилання на spare_parts - яку саме деталь було витрачено.        |
| used_by       | BIGINT    |    | +  | Посилання на users - виконавець, що списав деталь.                |
| quantity_used | INT       |    |    | Кількість витрачених одиниць.                                     |
| notes         | TEXT      |    |    | Додаткові примітки щодо використання.                             |

Разом із витратами матеріалів, критично важливо аналізувати та мінімізувати простої виробничих ліній. Для фіксації часу зупинки обладнання та розрахунку загальної ефективності активів розроблено журнал простоїв, описаний у табл. 2.12.

Таблиця 2.12.

**Таблиця downtime\_logs (Журнал простоїв)**

| Назва        | Тип даних | PK | FK | Опис                                                 |
|--------------|-----------|----|----|------------------------------------------------------|
| id           | BIGINT    | +  |    | Унікальний ідентифікатор запису простою.             |
| equipment_id | BIGINT    |    | +  | Посилання на equipment - обладнання, що простоювало. |

продовж. табл. 2.12.

|                  |          |  |   |                                                       |
|------------------|----------|--|---|-------------------------------------------------------|
| work_order_id    | BIGINT   |  | + | Посилання на work_orders - пов'язаний ремонт.         |
| reported_by      | BIGINT   |  | + | Посилання на users хто зафіксував простій.            |
| reason           | VARCHAR  |  |   | Причина зупинки (механічна відмова, планове ТО тощо). |
| description      | TEXT     |  |   | Детальний опис обставин простою.                      |
| started_at       | DATETIME |  |   | Час початку простою.                                  |
| ended_at         | DATETIME |  |   | Час завершення простою.                               |
| duration_minutes | INT      |  |   | Тривалість простою у хвиликах.                        |

З метою підвищення безпеки та контролю за діями користувачів (наприклад, для відстеження, хто саме змінив статус заявки чи списав деталь), система здійснює автоматичне логування всіх подій. Дані аудиту записуються в окрему таблицю, яку наведено в табл. 2.13.

Таблиця 2.13.

Таблиця audit\_logs (Журнал аудиту дій)

| Назва      | Тип даних | РК | FK | Опис                                      |
|------------|-----------|----|----|-------------------------------------------|
| id         | BIGINT    | +  |    | Унікальний ідентифікатор запису.          |
| user_id    | BIGINT    |    | +  | Посилання на users хто виконав дію.       |
| action     | VARCHAR   |    |    | Тип операції (created, updated, deleted). |
| model_type | VARCHAR   |    |    | Клас моделі, над якою виконано дію.       |

продовж. табл. 2.13.

|            |         |  |  |                                           |
|------------|---------|--|--|-------------------------------------------|
| model_id   | BIGINT  |  |  | Ідентифікатор зміненого запису.           |
| old_values | JSON    |  |  | Значення полів до зміни.                  |
| new_values | JSON    |  |  | Значення полів після зміни.               |
| ip_address | VARCHAR |  |  | ІР-адреса користувача під час дії.        |
| user_agent | VARCHAR |  |  | Інформація про браузер та ОС користувача. |

Окрім пасивного журналювання дій користувачів, система повинна активно повідомляти інженерів та техніків про призначення нових заявок, дефіцит запчастин чи прострочені ремонти. Специфікацію таблиці сповіщень наведено в табл. 2.14.

Таблиця 2.14.

**Таблиця notifications (Системні сповіщення)**

| Назва           | Тип даних | PK | FK | Опис                                        |
|-----------------|-----------|----|----|---------------------------------------------|
| id              | UUID      | +  |    | Унікальний ідентифікатор сповіщення (UUID). |
| type            | VARCHAR   |    |    | Клас-обробник типу сповіщення.              |
| notifiable_type | VARCHAR   |    |    | Тип адресата (App\Models\User).             |
| notifiable_id   | BIGINT    |    |    | Ідентифікатор адресата сповіщення.          |
| data            | TEXT      |    |    | Вміст сповіщення у форматі JSON.            |
| read_at         | TIMESTAMP |    |    | Дата та час прочитання сповіщення.          |

Для забезпечення гнучкості системи без необхідності редагування коду адміністратор може налаштовувати загальні константи (наприклад, ставку оплати праці техніка за годину для розрахунку вартості ремонтів). Збереження цих параметрів здійснюється у таблиці, описаній у табл. 2.15.

Таблиця 2.15.

**Таблиця settings (Системні налаштування)**

| Назва | Тип даних | PK | FK | Опис                                                   |
|-------|-----------|----|----|--------------------------------------------------------|
| id    | BIGINT    | +  |    | Унікальний ідентифікатор параметра.                    |
| key   | VARCHAR   |    |    | Унікальний ключ налаштування (finance_base_rate тощо). |
| value | TEXT      |    |    | Значення параметра налаштування.                       |

При тривалій роботі з вебсистемою сесії користувачів повинні бути захищені та мати часові обмеження для запобігання викраденню авторизаційних даних. Для цього використовується таблиця сесій, представлена в табл. 2.16.

Таблиця 2.16.

**Таблиця sessions (Сесії користувачів)**

| Назва         | Тип даних | PK | FK | Опис                                       |
|---------------|-----------|----|----|--------------------------------------------|
| id            | VARCHAR   | +  |    | Унікальний ідентифікатор сесії.            |
| user_id       | BIGINT    |    | +  | Посилання на users - власник сесії.        |
| ip_address    | VARCHAR   |    |    | IP-адреса користувача.                     |
| user_agent    | TEXT      |    |    | Дані браузера та операційної системи.      |
| payload       | LONGTEXT  |    |    | Зашифровані дані сесії.                    |
| last_activity | INT       |    |    | Час останньої активності (unix timestamp). |

Нарешті, у разі втрати доступу до облікового запису користувач має можливість безпечно відновити пароль за допомогою тимчасових токенів, що надсилаються на підтверджену електронну пошту. Опис відповідної таблиці наведено в табл. 2.17.

Таблиця 2.17.

**Таблиця password\_reset\_tokens (Токени скидання паролю)**

| Назва      | Тип даних | PK | FK | Опис                                           |
|------------|-----------|----|----|------------------------------------------------|
| email      | VARCHAR   | +  |    | Email-адреса, для якої запиту скидання паролю. |
| token      | VARCHAR   |    |    | Хеш одноразового токена для скидання.          |
| created_at | TIMESTAMP |    |    | Час створення токена.                          |

Таблиця 2.17. використовується для реалізації механізму відновлення доступу до облікового запису. Після надсилання запиту на скидання пароля система генерує унікальний одноразовий токен, який зберігається у вигляді хешу та прив'язується до відповідної електронної адреси.

## 2.3 Проєктування та реалізація алгоритмів роботи системи

Для формалізованого опису логіки роботи ключових функцій розробленої CMMS-системи було побудовано діаграму активності, яка відображає покроковий алгоритм виконання найбільш частого сценарію - реєстрації несправності та створення заявки на ремонт.

Процес ініціюється користувачем, який виявив несправність виробничого обладнання. Першим автоматичним кроком системи є перевірка наявності активної сесії автентифікації. Якщо користувач не пройшов авторизацію, Laravel-middleware перехоплює запит та перенаправляє його на сторінку входу, після чого процес завершується. За умови успішної автентифікації система виконує другу перевірку - наявність у поточного користувача дозволу work-orders.create. Даний дозвіл призначено ролям

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 46   |

оператора, техника, інженера та адміністратора, тому лише ці категорії персоналу можуть реєструвати нові заявки. У разі відсутності необхідного дозволу система повертає відповідь з кодом HTTP 403 та відображає повідомлення про відмову у доступі.

За наявності відповідних прав система завантажує форму створення заявки. При цьому перелік доступного обладнання автоматично фільтрується - до нього не потрапляють списані активи зі статусом decommissioned. Список можливих виконавців також обмежено: до нього входять виключно співробітники з роллю engineer або technician. Користувач заповнює форму, вказуючи заголовок та розгорнутий опис несправності, тип, рівень пріоритету, одиницю несправного обладнання, а також за потреби - відповідального виконавця та плановий термін виконання.

Після підтвердження форми система виконує серверну валідацію всіх обов'язкових полів. Якщо дані не пройшли перевірку, форма повертається користувачеві з детальними повідомленнями про кожну виявлену помилку. При успішній валідації система зберігає новий об'єкт WorkOrder у таблиці work\_orders зі статусом new, автоматично присвоює заявці унікальний ідентифікаційний номер у форматі WO-XXXXXX та фіксує подію у журналі аудиту через метод AuditLog::record(). Якщо при створенні заявки було призначено відповідального виконавця, система автоматично надсилає йому персональне системне сповіщення типу WorkOrderAssigned. Процес завершується перенаправленням користувача на сторінку деталей щойно створеної заявки.

Діаграма активності рис для управління планово-запобіжними роботами (ПЗР) демонструє логіку роботи системи під час налаштування та супроводу регламентних робіт для обладнання підприємства.

Процес починається із переходу користувача до модуля «Графіки ТО», де система пропонує вибір між переглядом списку наявних регламентів та конфігурацією графіків. Під час перегляду користувачам доступна гнучка

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 47   |

фільтрація за обладнанням, періодичністю обслуговування та статусом активності графіка.

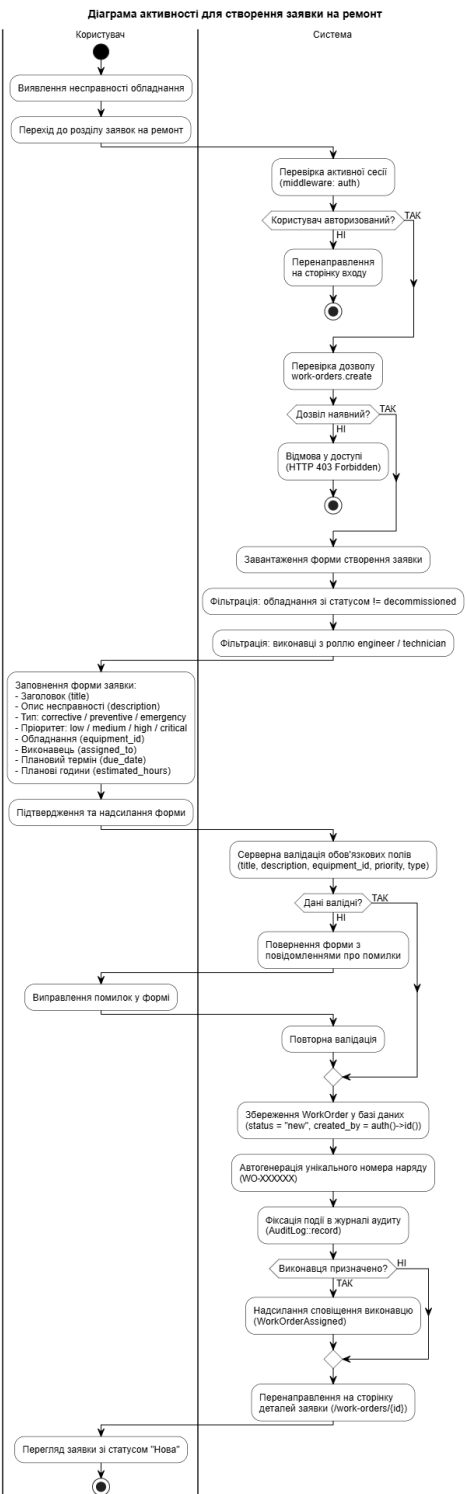


Рис. 2.4. Діаграма активності для створення заявки на ремонт

Якщо ініціюється створення або редагування графіка, система першочергово перевіряє права доступу користувача. До операцій модифікації допускаються лише користувачі з ролями адміністратора або інженера.

Спроба доступу з іншою роллю блокується із відображенням помилки авторизації. Після успішної перевірки прав для створення нового регламенту інженер обирає обладнання зі списку активних одиниць, призначає відповідального виконавця та встановлює обов'язкові параметри, серед яких назва графіка, дата початку, очікувані трудовитрати та періодичність проведення робіт. Валідація даних виконується на стороні сервера, і у разі виявлення помилок користувачеві пропонується скоригувати введену інформацію.

При успішній реєстрації графіка дата першого обслуговування автоматично прирівнюється до дати початку, а статус за замовчуванням встановлюється як активний. Після запису регламенту до бази даних система фіксує подію в журналі аудиту. Процес редагування дозволяє коригувати параметри вже створеного регламенту. Якщо за графіком ще не було виконано жодного наряду, зміна дати початку автоматично оновлює очікувану дату наступного ТО. У випадку, коли обслуговування вже проводилося, а інженер змінює періодичність циклу, система автоматично перераховує наступну дату виконання робіт на основі дати останнього фактичного закриття наряду та нової частоти. Будь-які зміни в параметрах графіків також підлягають обов'язковому логуванню в журналі аудиту.

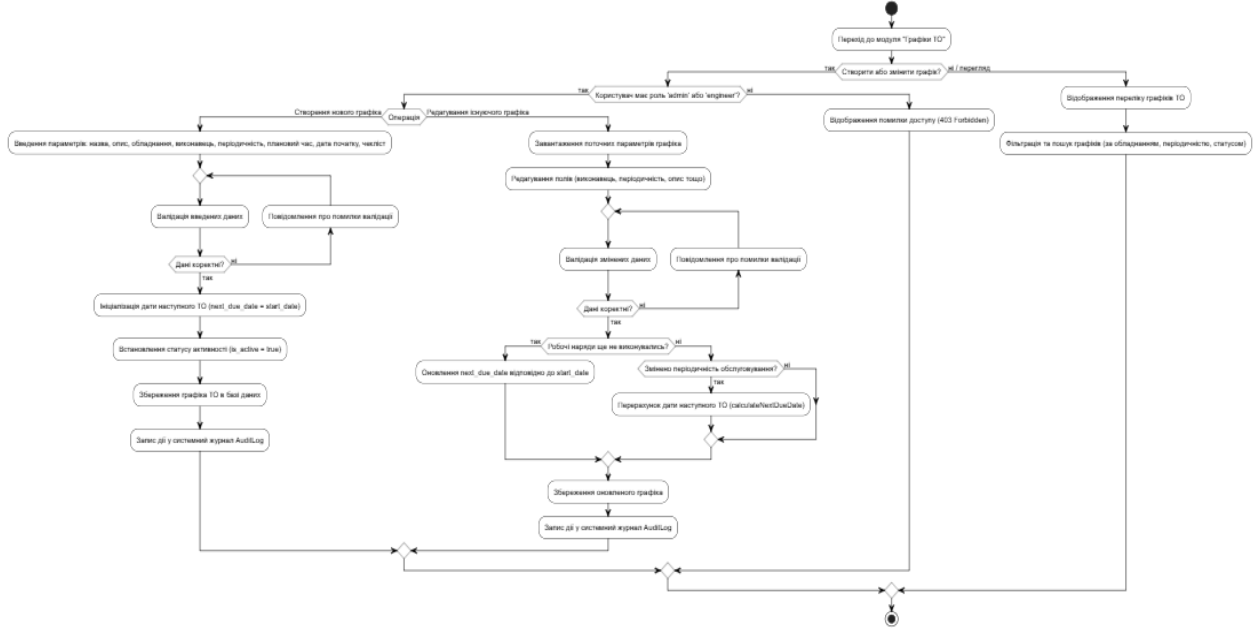


Рис. 2.5. Діаграма активності для управління графіками ТО

Діаграма життєвого циклу замовлення на ремонт наочно ілюструє взаємодію між ініціатором робіт та виконавцем у межах CMMS-системи. Процес починається у зоні відповідальності ініціатора, в ролі якого може виступати оператор або інженер. Вони виявляють несправність, обирають відповідну одиницю обладнання та реєструють заявку, заповнюючи форму з метаданими, де зазначають назву, опис проблеми, рівень пріоритету та тип робіт. Після збереження заявка автоматично отримує початковий статус «New» та реєструється в базі даних. Далі ініціатор або система призначає відповідального виконавця, що автоматично тригерить відправку email-сповіщення призначеному техніку.

З цього моменту процес переходить у зону відповідальності виконавця. Технік отримує сповіщення, переглядає деталі заявки та приймає її в роботу, переводячи статус у положення «In Progress». Система автоматично фіксує точний час початку ремонту, що дозволяє надалі аналізувати ефективність роботи. У ході виконання ремонту виконавець може фіксувати простій обладнання через лог простоїв та реєструвати фактичне списання використаних запчастин зі складу. Якщо під час робіт виникає потреба в очікуванні деталей чи додаткових погодженнях, заявка може тимчасово перейти у статус «On Hold».

Після завершення робіт виконавець вказує фактично витрачений час, заповнює звітні нотатки та змінює статус на «Completed». Система автоматично закриває всі пов'язані активні записи про простій, фіксуючи кінцевий час. Якщо виконана заявка була пов'язана із планово-запобіжними роботами, система автоматично оновлює дату останнього виконання регламенту та розраховує дату наступного обслуговування. У разі неможливості виконання ремонту заявка може бути переведена у статус «Rejected» із обов'язковим зазначенням причини відхилення. Будь-які переходи між статусами супроводжуються автоматичним записом до журналу AuditLog для забезпечення повної прозорості процесу.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 50   |



ремонті), а також створює новий запис у таблиці DowntimeLog, фіксуючи початок простою.

У ході проведення ремонту технік через інтерфейс заявки надсилає запит на списання необхідних витратних деталей, вказуючи конкретну запчастину та потрібну кількість. Цей запит обробляється методом writeOff у контролері SparePartController, який на першому етапі виконує серверну валідацію отриманих даних. Контролер перевіряє наявність запчастини на складі через запит до моделі SparePart. Якщо необхідна кількість є на складі, система зменшує поточний баланс запчастини, створює новий запис про використання у моделі SparePartUsage та реєструє подію в системному журналі AuditLog. У разі дефіциту деталей на складі процес списання переривається, а система повертає виконавцю повідомлення про помилку.

Після успішного виконання технічних робіт та повного усунення виявлених несправностей, виконавець через інтерфейс користувача надсилає фінальний запит на переведення ремонтної заявки у статус завершеної (completed). Цей запит спрямовується до WorkOrderController, який оновлює поточний статус наряду, фіксує точний час закриття та ініціює збереження змін у базі даних, що знову активує роботу спостерігача моделі WorkOrderObserver. На цьому етапі бізнес-логіки спостерігач автоматично переводить пов'язану одиницю обладнання з ремонту в активний робочий стан (active), перевіряє наявність відкритого запису простою у журналі DowntimeLog та закриває його, проставляючи час завершення простою й автоматично обчислюючи загальну тривалість відмови в хвилинах.

Наприкінці цієї операції контролер звертається до сервісного шару системи для аналізу пов'язаних планів профілактичного обслуговування; якщо для даного агрегату було налаштовано графік планово-запобіжних робіт, дата наступного огляду автоматично зміщується вперед на величину встановленої періодичності.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 52   |

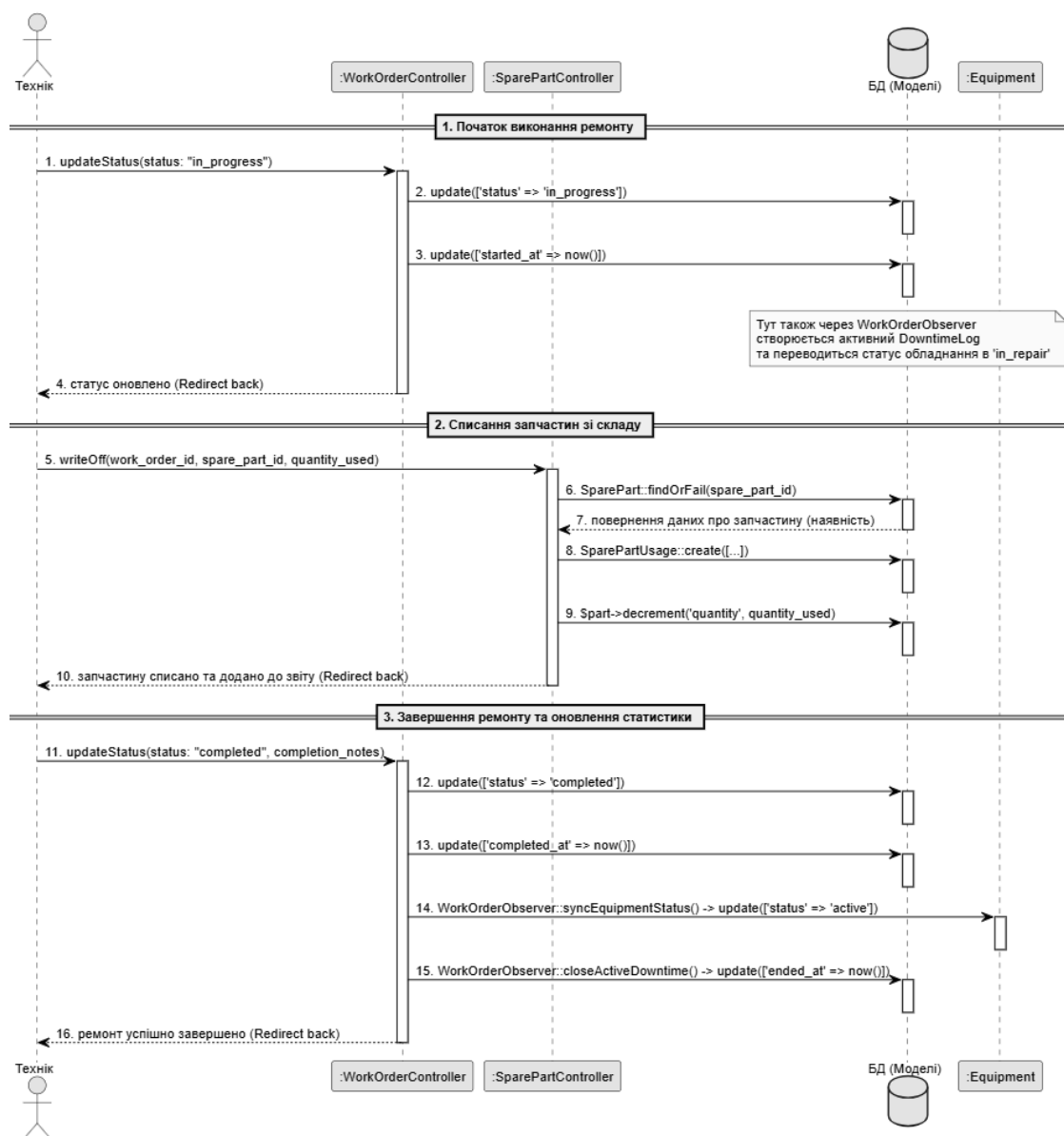


Рис. 2.7. Діаграма послідовності для виконання ремонту

Діаграма комунікації відображає структурні зв'язки між об'єктами під час нарахування витрат на обслуговування. Процес починається із запиту від інженера чи техника на зміну статусу заявки, що обробляється у WorkOrderController. Після оновлення статусу та внесення фактичних годин, модель WorkOrder автоматично ініціює перерахунок витрат через метод updateRepairCost. Взаємодія відбувається шляхом послідовного звернення до зв'язаних об'єктів для обчислення вартості праці та матеріалів. Для розрахунку матеріальних витрат модель звертається до сутності SparePartUsage, яка надає інформацію про кількість використаних деталей, та до сутності SparePart, звідки отримується актуальна ціна за одиницю. Розрахунок вартості праці

базується на фактично витраченому часі та конфігураційних налаштуваннях тарифних ставок і коефіцієнтів пріоритету з моделі Setting. Отримана сумарна вартість ремонту записується в полі repair\_cost моделі WorkOrder. Відразу після цього викликається сервіс амортизації DepreciationService, який виконує перерахунок залишкової вартості обладнання з урахуванням вартості проведеного ремонту та накопиченого зносу, оновлюючи поле current\_value безпосередньо в моделі Equipment.

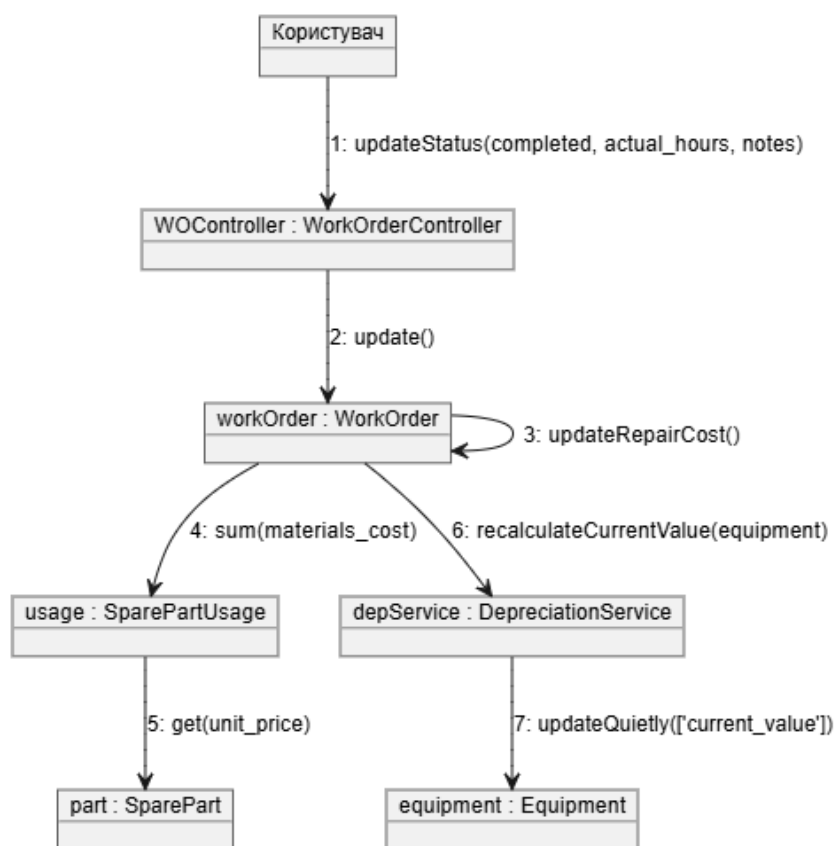


Рис. 2.8. Діаграма комунікації для обліку витрат

Діаграма кооперації описує взаємодію між системними компонентами під час автоматичної генерації та закриття нарядів на планове технічне обслуговування. Процес починається із запуску консольної команди GenerateMaintenanceTasks, яка здійснює перевірку активних графіків у базі даних, у яких настала дата чергового обслуговування. На основі знайдених записів у моделі MaintenanceSchedule команда створює нові заявки на ремонт WorkOrder зі статусом «new» та типом «preventive», а також відправляє сповіщення відповідальному техніку. Після проведення перевірки інженер чи

технік через інтерфейс надсилає запит на завершення робіт. Контролер WorkOrderController змінює статус наряду на «completed», що активує обробник подій WorkOrderObserver.

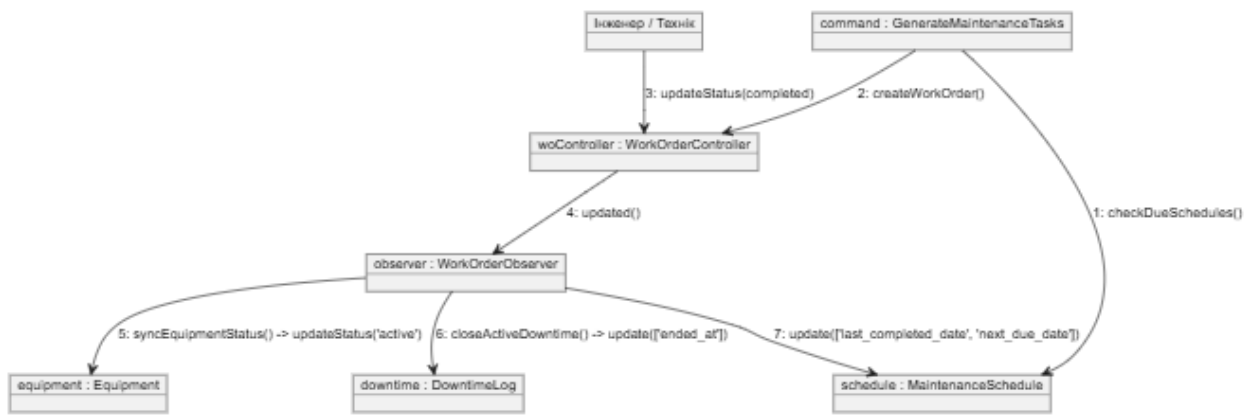


Рис. 2.9. Діаграма кооперацій для планового ТО

Також реалізований спостерігач який перевіряє стан зв'язаних сутностей і виконує синхронізацію: оновлює статус обладнання Equipment на «active», закриває активний простій у моделі DowntimeLog, встановлюючи час завершення, та звертається до моделі MaintenanceSchedule для фіксації дати завершення ТО та розрахунку наступного планового візиту на основі заданого циклу обслуговування.

### 2.4 Реалізація функціоналу системи управління технічним обслуговуванням

Ключовим елементом у логіці функціонування CMMS-платформи виступає процес формування, координації та супроводу замовлень-нарядів на виконання робіт (Work Orders). Цей механізм слугує основною сполучною ланкою між ключовими компонентами системи: виробничими активами (обладнанням), технічним персоналом (інженерами та техніками-виконавцями), складом запасних частин та аналітичним модулем оцінки ефективності. З огляду на це, логіку опрацювання даних та актуалізації технічних і фінансових показників системи варто детально проаналізувати на прикладі процедури створення, зміни статусів замовлень, списання матеріалів та розрахунку зносу обладнання.

На рівні серверної частини взаємодія користувача з системою забезпечується через контролери, розроблені на базі веб-фреймворку Laravel 11. Спеціалізовані методи обробки HTTP-запитів призначені для прийому вхідних даних, їхньої суворої фільтрації, перевірки відповідності бізнес-правилам та збереження інформації у реляційній базі даних MySQL через об'єктно-реляційне відображення ORM Eloquent.

Програмна реалізація механізму реєстрації нових заявок на ремонт ґрунтується на багаторівневій валідації вхідних потоків даних за допомогою спеціалізованого класу запитів `WorkOrderRequest`. Процес обробки кожного нового звернення починається із перевірки структури запиту на рівні сервера. Система контролює наявність обов'язкового заголовка, детального опису виявленої несправності та обов'язково верифікує ідентифікатор обладнання (`equipment_id`), перевіряючи його фізичне існування в базі даних та поточний статус роботи.

Окремо аналізується відповідність переданого рівня пріоритету виконання (низький, середній, високий, критичний) та типу наряду (аварійний, плановий, запобіжний). Після успішного проходження валідації контролер `WorkOrderController` ініціює транзакцію створення нового запису в таблиці `work_orders`. Під час цього процесу система автоматично призначає унікальний номер наряду за маскою `WO-XXXXXX` (де `XXXXXX` - унікальний генератор), фіксує поточний час створення, а також ідентифікатор авторизованого користувача, який подав заявку (`created_by`).

У разі, якщо під час створення заявки відразу призначається виконавець (`assigned_to`), система автоматично ініціює подію, яка через механізм черг надсилає техніку сповіщення на електронну пошту та внутрішній інтерфейс додатку (`NewUserRegistered / WorkOrderAssigned`), що дозволяє оперативно реагувати на критичні несправності конвеєрних ліній.

Особливої уваги заслуговує реалізація життєвого циклу замовлення, яка автоматизована за допомогою спостерігача моделей `WorkOrderObserver`. Цей механізм відстежує стан моделі `WorkOrder` на рівні подій бази даних (`updating`,

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 56   |

updated, saved). Коли статус заявки змінюється на «В роботі» (in\_progress), програмний модуль автоматично фіксує мітку часу фактичного початку робіт у полі started\_at, а статус відповідної одиниці обладнання в таблиці equipment автоматично переводиться у стан «В ремонті» (in\_repair). Це запобігає можливості призначення інших планових робіт на цей агрегат у той самий час.

При встановленні статусу «Завершено» (completed):

1. Фіксується час завершення робіт у полі completed\_at.
2. Обчислюється фактичний час ремонту в годинах (actual\_hours), який порівнюється з плановим (estimated\_hours).
3. Статус обладнання автоматично повертається у стан «Активне» (active) або переводиться у «Простій» (idle), залежно від внесених техніком приміток.
4. Автоматично розраховується вартість наряду (repair\_cost) за формулою: 
$$\text{Вартість ремонту} = (\text{Фактичні години} \times \text{Ставка праці інженера}) + \text{Вартість списаних деталей}$$
 Погодинна ставка праці підтягується з таблиці глобальних конфігурацій settings за ключем finance\_base\_rate.

У ході виконання ремонту технік через інтерфейс заявки надсилає запит на списання витратних деталей, вказуючи конкретну запчастину та потрібну кількість. Цей запит обробляється методом writeOff у контролері SparePartController.

Програмна логіка методу реалізована за таким алгоритмом:

1. Здійснюється блокування запису запчастини в базі даних для запобігання стану конкуренції запитів (Race Condition).
2. Перевіряється поточна кількість деталі на складі (quantity в таблиці spare\_parts). Якщо наявний залишок менший за запитувану кількість, транзакція скасовується, а система повертає повідомлення про помилку дефіциту.
3. Якщо деталі є в наявності, система зменшує баланс складу, створює новий запис про використання у моделі SparePartUsage (із прив'язкою

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 57   |

до `work_order_id`) та автоматично додає вартість деталей (`quantity_used`  $\times$  `unit_price`) до загального бюджету матеріальних витрат заявки.

- Після списання система перевіряє, чи не впав залишок деталі нижче критичного ліміту `min_quantity`. Якщо ліміт перевищено, додаток автоматично генерує та надсилає інженеру сповіщення про необхідність закупівлі запчастин.

Кожна одиниця обладнання має свій фінансовий життєвий цикл. У системі реалізовано сервіс `DepreciationService`, який автоматично розраховує лінійний знос активів. Під час створення або редагування обладнання вказуються параметри початкової вартості (`purchase_price`) та терміну корисного використання в роках (`useful_life_years`).

Сервіс `DepreciationService` обчислює суму річної та місячної амортизації. За допомогою періодичного фонового завдання система перераховує поточну залишкову вартість агрегатів (`current_value`), віднімаючи накопичений знос від початкової ціни придбання. Це дозволяє менеджменту підприємства оцінювати доцільність подальшого ремонту застарілого обладнання: якщо вартість чергового ремонту в таблиці `work_orders` перевищує залишкову вартість обладнання в таблиці `equipment`, система сигналізує про необхідність його списання (переведення в статус `decommissioned`).

Забезпечення безпеки інформаційних потоків та розмежування повноважень реалізовано через впровадження спеціальних посередників (`Middleware`) та рольову модель доступу (`RBAC`). Будь-яка дія користувача проходить перевірку на наявність прав. Наприклад, звичайний технік має доступ лише до перегляду своїх заявок та списання деталей, тоді як видалення записів обладнання, редагування складських цін або зміна налаштувань амортизації дозволені виключно користувачам із ролями `admin` або `engineer`.

Усі критичні операції модифікації даних, які охоплюють процеси створення, оновлення та видалення інформаційних сутностей, автоматично фіксуються в системному журналі аудиту через модель `AuditLog`. При

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 58   |

виконанні кожної такої дії спеціалізований спостерігач моделей перехоплює подію та детально записує в базу даних ідентифікатор користувача (`user_id`), який безпосередньо виконав цю операцію, а також тип виконаної дії (`action`), що може набувати значень створення, оновлення чи видалення запису. Крім того, у журналі фіксується системна назва зміненої моделі разом із її унікальним первинним ключем, що дозволяє точно ідентифікувати об'єкт втручання. Для детального аналізу внесених змін система зберігає стан усіх полів об'єкта до моменту модифікації (`old_values`) та їхні нові значення після успішного збереження (`new_values`) у структурованому форматі JSON.

Такий підхід забезпечує абсолютну прозорість дій персоналу, спрощує розслідування інцидентів та унеможливорює несанкціоноване видалення чи спотворення технічної або фінансової звітності підприємства.

Важливою складовою частиною розробленого функціоналу CMMS-системи є автоматизація планування регламентного обслуговування, що реалізується через взаємодію моделей `MaintenanceSchedule` (Плани обслуговування) та `WorkOrder` (Заявки на ремонт). Інженер з експлуатації створює картку регламентного планово-попереджувального ремонту, де вказує періодичність проведення робіт (`frequency` - щодня, щотижня, щомісяця, щоквартально, щорічно), орієнтовні трудовитрати та перелік обов'язкових кроків (чек-лист у форматі JSON).

Під час проведення ремонту та після успішного закриття відповідної заявки виконавець фіксує час запуску обладнання в полі `ended_at`. Система на рівні подій моделі автоматично розраховує тривалість простою у хвилинах (`duration_minutes`) та оновлює сумарний показник часу відмов для аналітичного модуля. Ці дані безпосередньо використовуються в контролері `ReportController` для формування інтерактивних графіків `Chart.js` та експорту звітів, дозволяючи менеджменту підприємства оцінювати коефіцієнт загальної ефективності обладнання (OEE - Overall Equipment Effectiveness) та виявляти найбільш проблемні вузли конвеєра, які потребують модернізації або заміни.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 59   |

## Висновки до другого розділу

Було проведено ґрунтовний аналіз предметної сфери управління технічним обслуговуванням обладнання на прикладі заводу з виробництва кормів для риб. Визначено функціональні вимоги до CMMS-системи, а також ідентифіковано основні ролі користувачів (Оператор, Технік, Інженер, Адміністратор) та їхні сценарії взаємодії з програмним продуктом.

Для візуалізації архітектури було розроблено діаграми прецедентів та класів, що дозволило структурувати ключові сутності системи (Обладнання: Equipment, Заявка на ремонт: WorkOrder, Графік ТО: MaintenanceSchedule, Запчастини: SparePart) та встановити логічні зв'язки між ними.

Проект охопив розробку комплексної реляційної моделі бази даних. Спроектована структура забезпечує цілісність даних про технічний стан активів, історію ремонтів та рух складських запасів. Використання нормалізованих таблиць дозволило мінімізувати дублювання інформації та оптимізувати виконання аналітичних запитів щодо простоїв лінії.

Здійснено детальне моделювання динаміки роботи системи. За допомогою діаграм діяльності та послідовності було описано критичні бізнес-процеси: створення аварійних заявок, автоматичне планування регламентних робіт та алгоритм синхронізації між раптовими ремонтами і плановим обслуговуванням. Це заклало надійний фундамент для програмної реалізації без логічних суперечностей.

На етапі імплементації було розроблено веб-систему на базі сучасного фреймворку Laravel 11 та мови програмування PHP 8.2. Реалізовано механізми авторизації та розмежування прав доступу на основі ролей (RBAC), впроваджено систему аудиту дій користувачів та автоматизовані сповіщення. Застосування архітектурного підходу MVC та Eloquent ORM забезпечило високу швидкість розробки та легкість подальшої підтримки коду.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 60   |

# РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБСИСТЕМОЮ УПРАВЛІННЯ ТЕХНІЧНИМ ОБСЛУГОВУВАННЯМ ДЛЯ ВИРОБНИЦТВА КОРМУ ДЛЯ РИБ

## 3.1 Порядок встановлення та налаштування параметрів платформи

Встановлення та налаштування параметрів системи передбачає підготовку серверного середовища, розгортання компонентів серверної та клієнтської частин, а також конфігурування мережових зв'язків між ізольованими сервісами.

Для опису процесу фізичного розміщення компонентів системи та їх взаємодії в процесі виконання було розроблено діаграму розгортання (рис. 3.1).

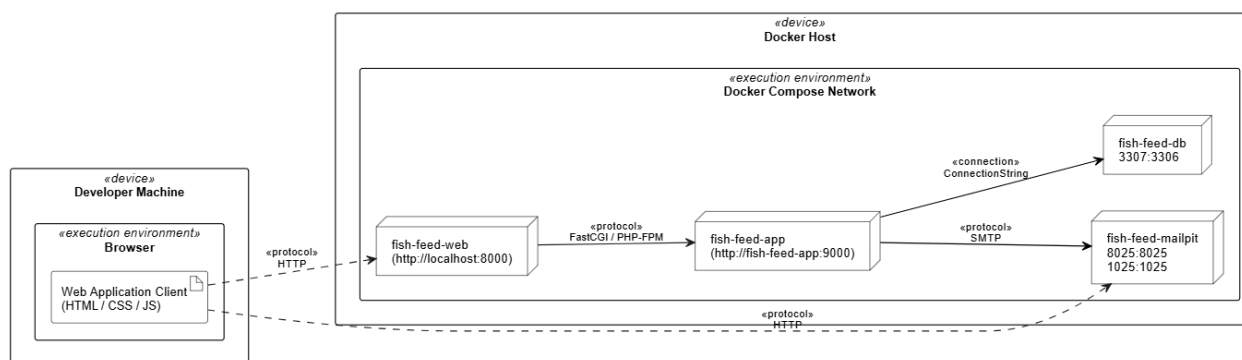


Рис. 3.1. Діаграма розгортання CMMS системи

Для забезпечення швидкого, надійного та стандартизованого розгортання вебсистеми управління технічним обслуговуванням (CMMS) на виробництві корму для риб застосовується технологія контейнеризації на базі платформи Docker та інструменту оркестрації Docker Compose. Цей підхід дозволяє повністю ізолювати програмні компоненти від конфігурації хост-машини, вирішити проблему сумісності залежностей та гарантувати ідентичність поведінки системи під час локальної розробки, тестування та фінального промислового запуску на сервері підприємства. Фізична та логічна архітектура платформи розділена на два ключові рівні, що взаємодіють між собою: робочу станцію користувача та серверне Docker-середовище.

Клієнтська робоча станція представляє собою кінцевий пристрій користувача, яким може виступати персональний комп'ютер інженера, монітор оператора в диспетчерській або мобільний телефон чергового техніка безпосередньо у виробничому цеху. На цьому рівні взаємодія із системою здійснюється через сучасний веббраузер, який завантажує статичні ресурси фронтенд-частини системи, зокрема розмітку HTML, стилі Bootstrap 5 та інтерактивні сценарії JavaScript, оптимізовані за допомогою інструменту Vite. Клієнтська частина відповідає за рендеринг календаря ППР за допомогою бібліотеки FullCalendar, відображення діаграм Chart.js та надсилання асинхронних HTTPS-запитів до серверного API для отримання динамічних даних без перезавантаження сторінок.

Серверна інфраструктура розгортається усередині ізолюваного мережевого простору Docker Compose, де кожен функціональний вузол працює у власному контейнері. Першою точкою входу для зовнішніх запитів є вебсервер Nginx. Працюючи в окремому контейнері, Nginx приймає вхідні HTTP/HTTPS-запити через зовнішній порт 8000 (або стандартні порти 80/443 у виробничому режимі), обробляє статичні файли (стилі, JS-скрипти, зображення завантаженого обладнання з папки Storage) та перенаправляє динамічні запити до інтерпретатора PHP. Зв'язок між вебсервером і контейнером додатку реалізовано через високопродуктивний протокол FastCGI на порту 9000.

Ядром системи є контейнер додатку, побудований на базі образу PHP 8.2-FPM. У цьому середовищі виконується весь програмний код системи, написаний на базі фреймворку Laravel 11. Тут зосереджена основна бізнес-логіка, яка обробляє заявки на ремонт, виконує обчислення лінійної амортизації через сервіс DepreciationService, перевіряє права доступу користувачів за допомогою Spatie-пакета та запускає автоматичні команди за розкладом.

Для роботи з даними використовується реляційна система керування базами даних MySQL 8.0, виділена в окремий контейнер. У ньому створюється

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 62   |

база даних системи, а для запобігання втраті накопиченої інформації, журналів аудиту, логів простоїв та облікових записів при перезапуску чи оновленні контейнера підключається постійне дискове сховище Docker Volume, що монтується до каталогу даних MySQL. У режимі розробки порт 3307 прокидається на локальну машину розробника, що дає змогу адмініструвати базу за допомогою зовнішніх клієнтів, таких як DBeaver чи phpMyAdmin.

Для налагодження поштових сповіщень (наприклад, повідомлень про нові заявки чи дефіцит деталей на складі) у локальному середовищі використовується додатковий контейнер із сервісом Mailpit. Він перехоплює всі SMTP-запити від додатку на порту 1025 і надає зручний веб-інтерфейс на порту 8025 для візуального тестування надісланих повідомлень без ризику відправити реальний лист клієнту. Взаємодія між усіма компонентами Docker-мережі побудована на стандартних мережевих протоколах TCP/IP, де PHP-сервер підключається до бази даних через драйвер PDO MySQL, а Nginx здійснює проксіювання запитів користувачів.

Перед початком експлуатації системи конфігурується файл оточення .env, який визначає параметри з'єднання з базою даних, налаштування автентифікації, рівень деталізації системних логів та режим роботи програми. Під час першого запуску системи автоматично викликається утиліта міграцій Laravel, яка створює схему таблиць у базі даних (від користувачів до списання запчастин), а також завантажуються початкові seed-дані для ролей доступу.

У виробничому середовищі (Production) схема розгортання адаптується до жорстких вимог безпеки та продуктивності: відключається налагоджувальний інтерфейс Mailpit, надсилання пошти переводиться на корпоративні SMTP-сервери із шифруванням TLS, а параметр налагодження APP\_DEBUG переводиться у положення false, що приховує детальні стеки помилок від сторонніх осіб. Для досягнення максимальної швидкості відгуку у промисловому режимі виконуються консольні команди оптимізації Laravel для кешування конфігураційного файлу, маршрутів та Blade-шаблонів, що

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 63   |

суттєво зменшує навантаження на процесор сервера та знижує час відповіді до мінімальних показників.

### 3.2 Структура інтерфейсу веб-системою контролю за технічним обслуговуванням на виробництві корму для риб

Клієнтська частина веб-системи реалізована у вигляді адаптивного веб-інтерфейсу, що виконується у веб-переглядачі користувача. Доступ до функціональності системи є суворо розмежованим відповідно до ролей користувачів (адміністратор, диспетчер, технічний фахівець). Взаємодія з платформою розпочинається з авторизації користувача.

Нижче наведено опис структури інтерфейсу користувача та ключових екранів розробленої веб-системи.

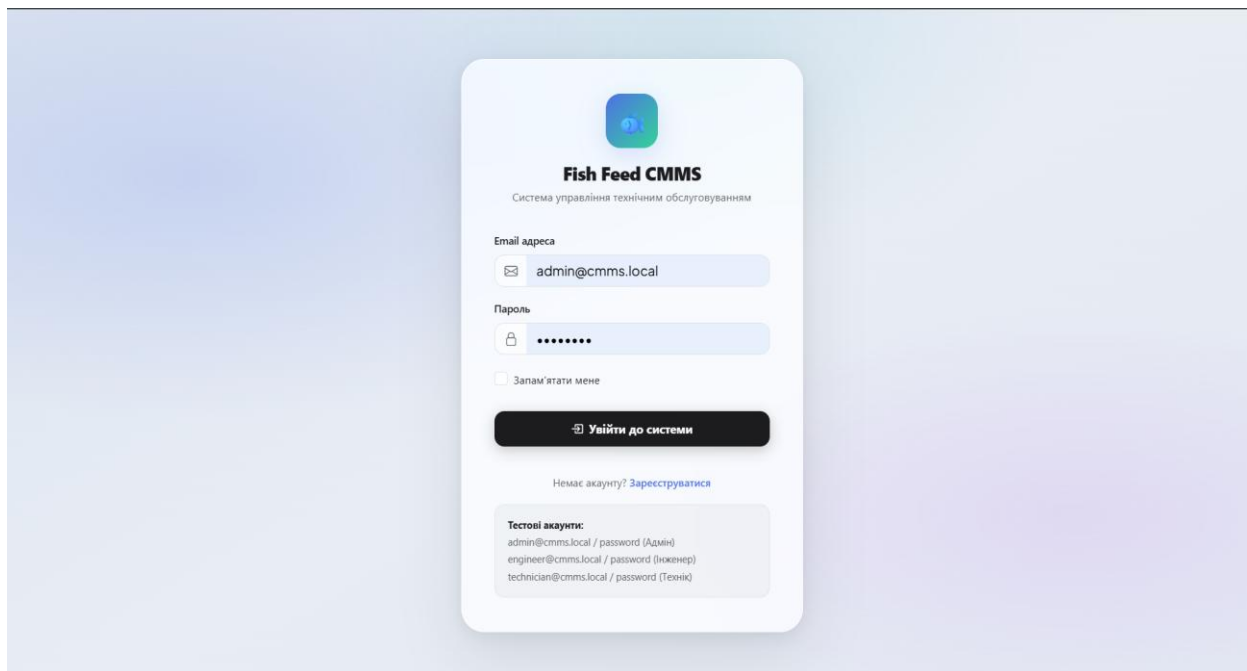


Рис 3.2 Сторінка входу до облікового запису

Для створення нового профілю користувача або реєстрації нового співробітника призначено спеціалізовану форму реєстрації (рис. 3.3), де вказуються його ім'я, адреса пошти, пароль та роль у системі.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 64   |

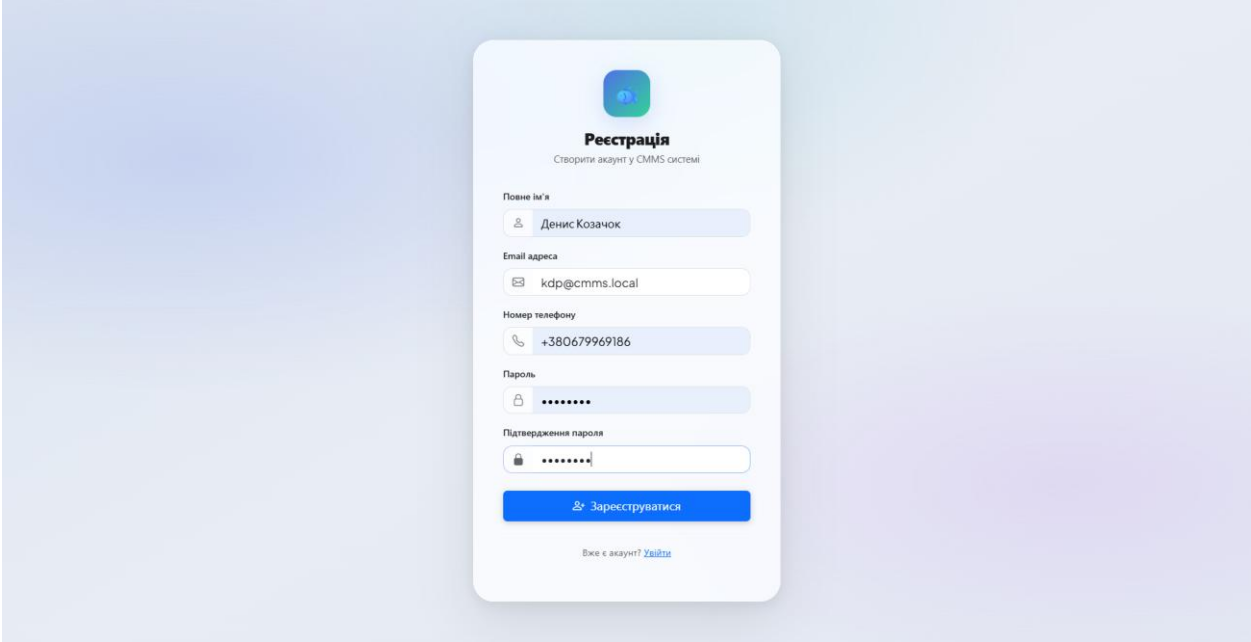


Рис 3.3 Сторінка реєстрації нового облікового запису

Після успішного входу користувач перенаправляється на головну сторінку системи (рис. 3.4). Вона виступає в ролі інформаційного дашборду, на якому відображається загальна кількість активного обладнання, відкриті заявки та поточні завдання.

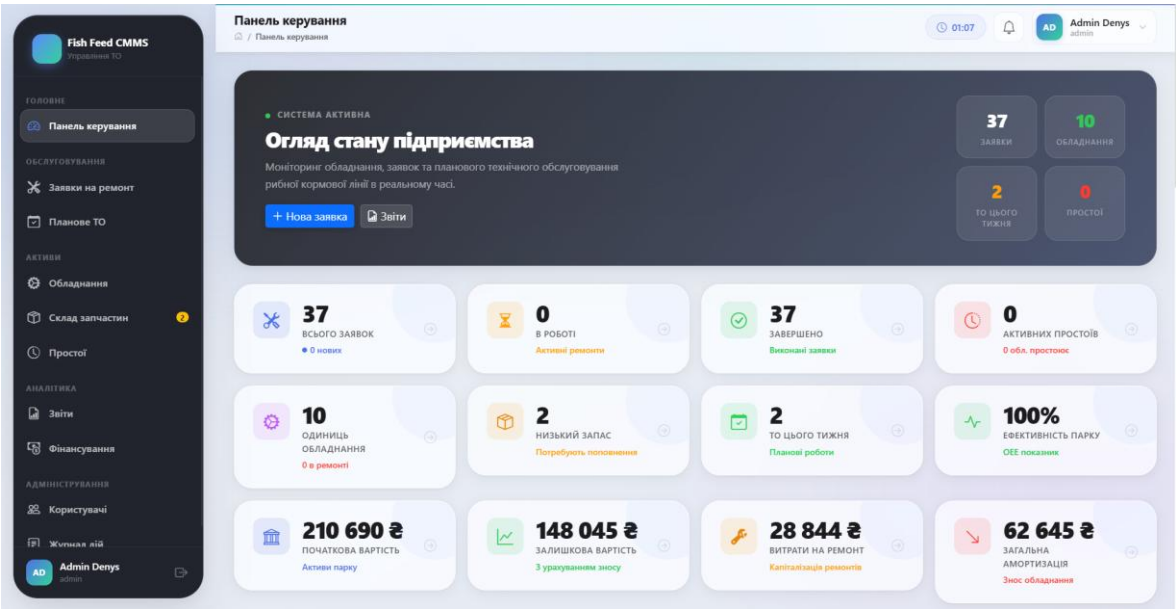


Рис 3.4 Головна сторінка після входження в систему

Для оперативного візуального моніторингу ефективності роботи технічної служби на головній сторінці розміщено інтерактивні графіки показників ефективності (рис. 3.5), які відображають розподіл витрат, середній час ремонту та кількість простоїв.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 65   |

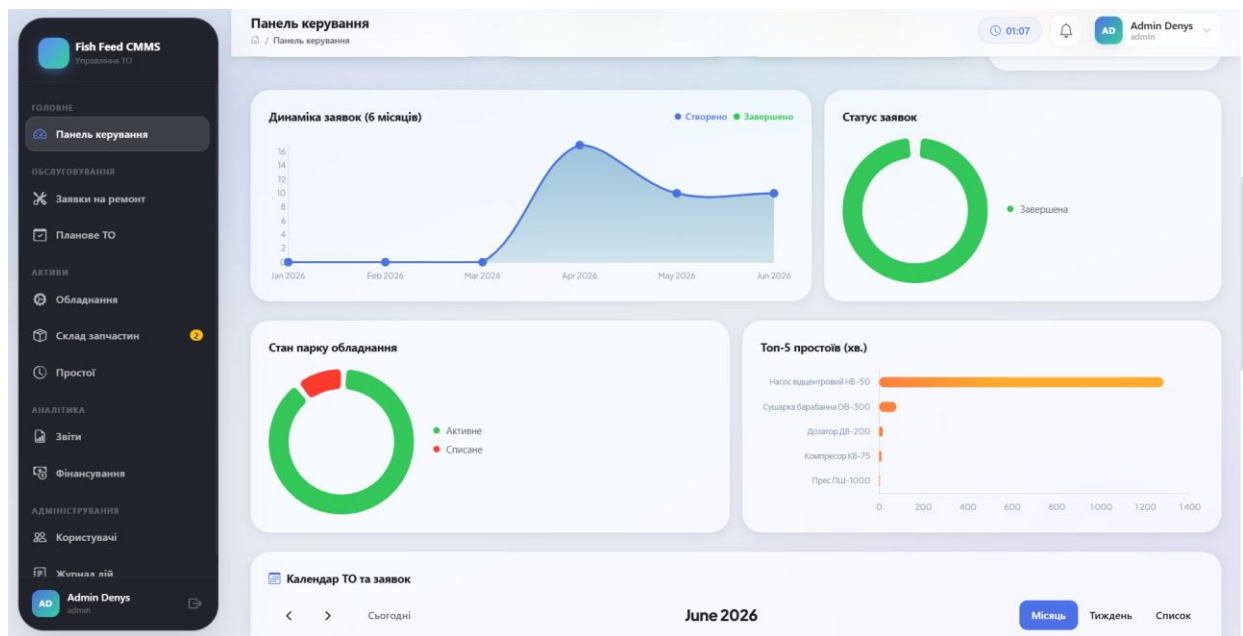


Рис 3.5 Графіки показників ефективності на головній сторінці

Також на головній панелі інтегровано календар планових технічних обслуговувань (рис. 3.6), що дозволяє візуально контролювати графік проведення регламентних робіт на тиждень або місяць вперед.

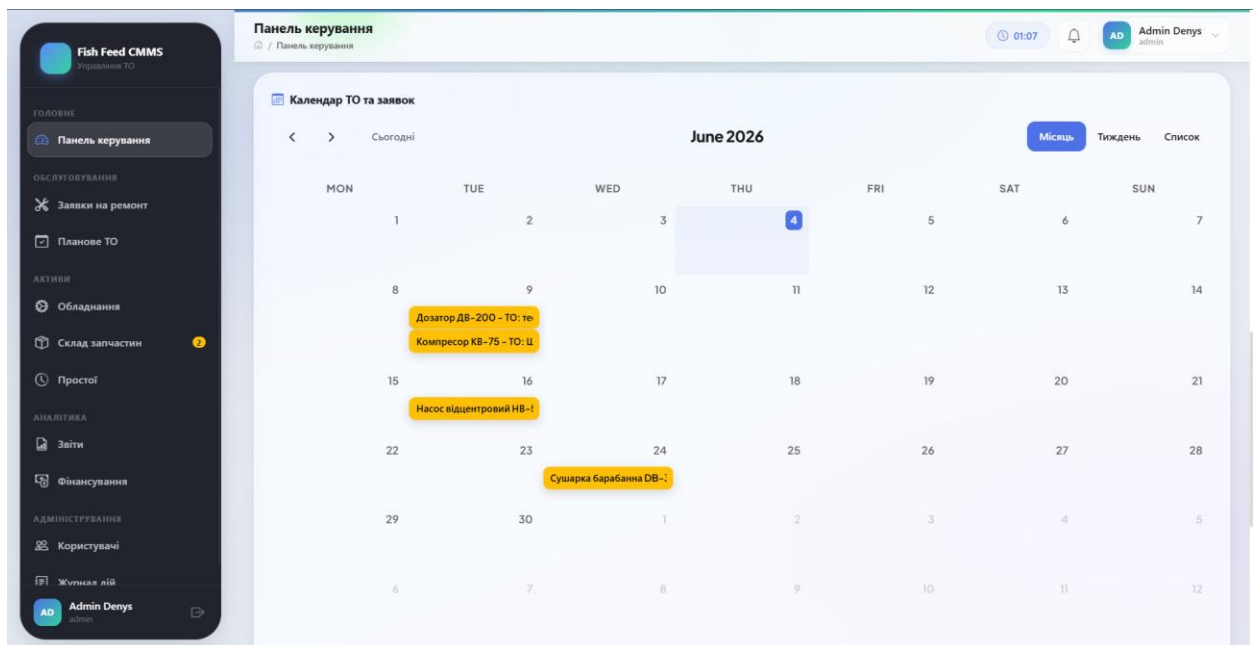


Рис 3.6 Календар планових ТО на головній сторінці

Керування особистими даними авторизованого працівника здійснюється на сторінці облікового запису (рис. 3.7), де користувач може оновити власну персональну інформацію, змінити пароль або завершити сесію.

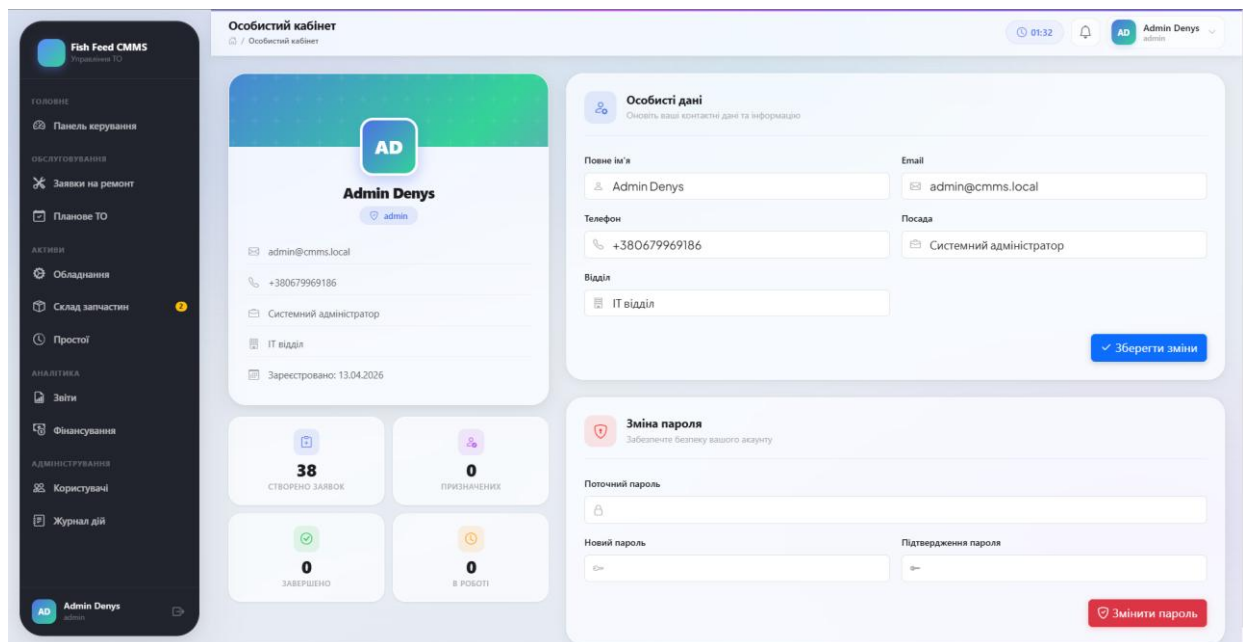


Рис 3.7 Сторінка облікового запису

Важливою частиною інтерфейсу є система миттєвих нотифікацій (рис. 3.8), яка відображає випадючий список останніх повідомлень про критичні події (наприклад, реєстрацію нового простою чи брак запчастин на складі).

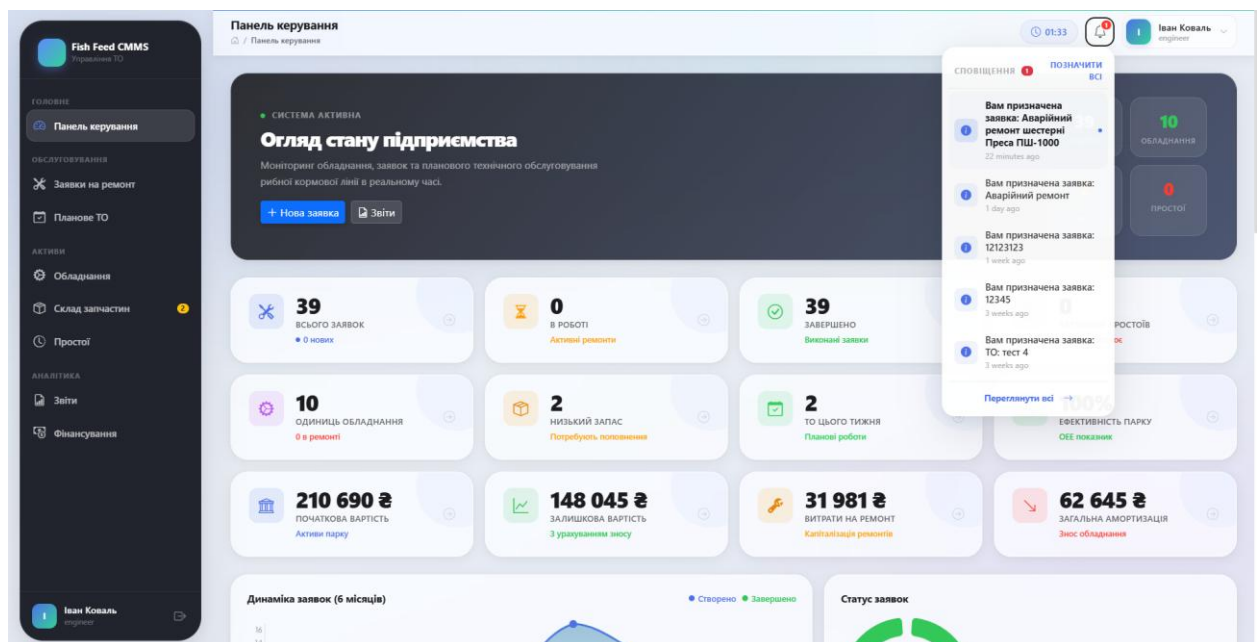


Рис 3.8 Система нотифікацій

Для детального ознайомлення зі списком усіх попереджень користувач може перейти на сторінку повної інформації про нотифікації (рис. 3.9).

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 67   |

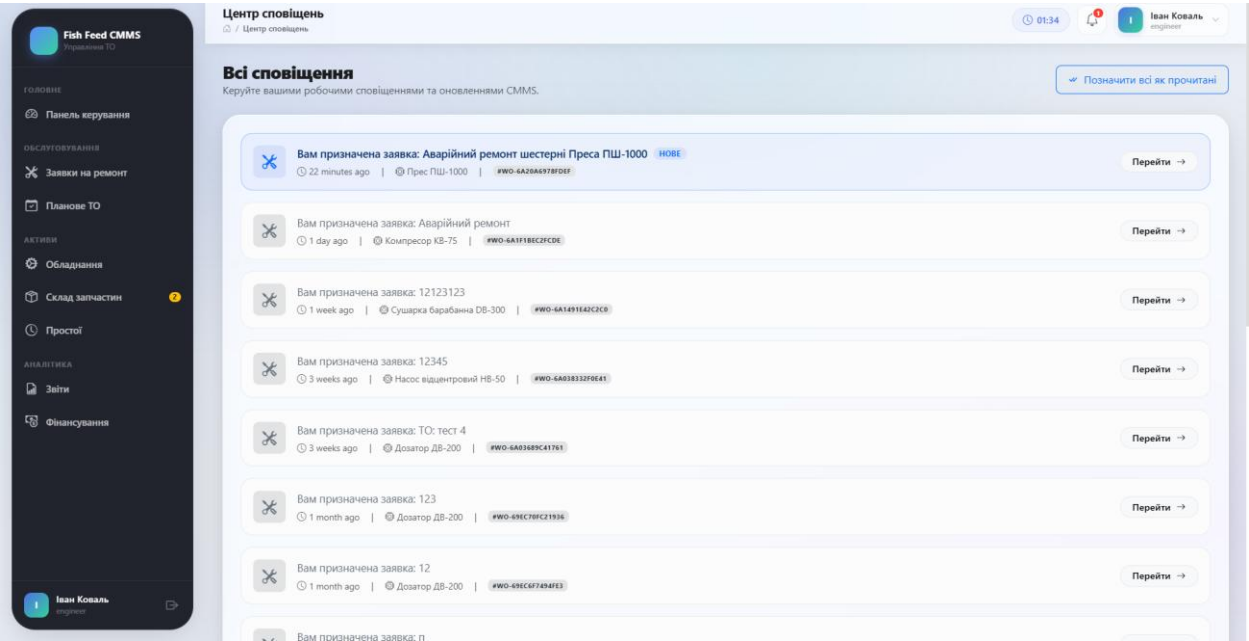


Рис 3.9 Сторінка повної інформації про нотифікації

Основним робочим простором для технічного персоналу є сторінка заявок на ремонт (рис. 3.10), яка містить перелік усіх зареєстрованих запитів на ремонтні роботи із зазначенням їх статусу та пріоритету.

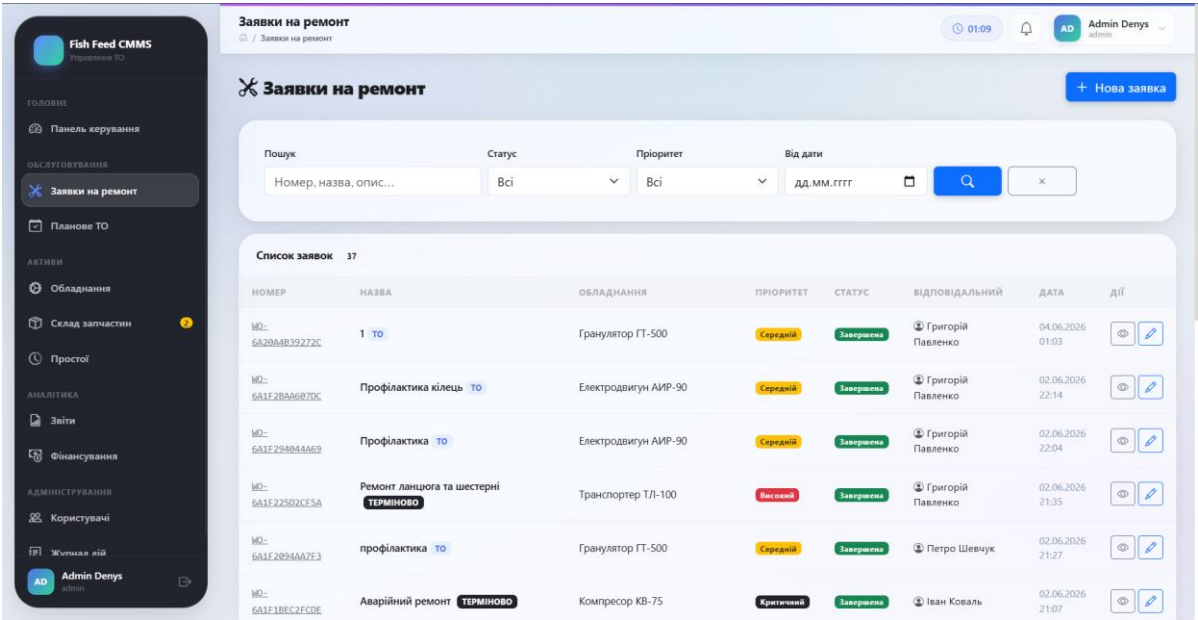


Рис 3.10 Сторінка заявок на ремонт

Створення нової непланової заявки при виявленні несправності здійснюється за допомогою форми, наведеної на рис. 3.11. Оператор або диспетчер вказує назву проблеми, обирає обладнання, призначає виконавця та вказує ступінь терміновості.

Fish Feed CMMS

Управління ТО

ГОЛОВНЕ

Панель керування

ОБСЛУГОВУВАННЯ

Заявки на ремонт

Планове ТО

АКТИВИ

Обладнання

Склад запчастин

Простої

АНАЛІТИКА

Зайти

Фінансування

АДМІНІСТРУВАННЯ

Користувачі

Життєвий цикл

Admin Denys

Нова заявка

Заявки / Нова заявка

Створити заявку на ремонт

Назва заявки \*

Аварійний ремонт шестерні Преса ПШ-1000

Обладнання \*

Прес ПШ-1000 (Цех №3)

Пріоритет \*

Критичний

Тип \*

Аварійний

Опис проблеми \*

Аварійний ремонт шестерні Преса ПШ-1000, вийшла з ладу шестерня, можливо пошкодила цілісність конструкції

Призначити відповідального

Іван Коваль (engineer)

Дедлайн

05.06.2026 03:00

Оцінка часу (годин) \*

3:00

Хвилини не можуть бути більше 59

Створити заявку

Скасувати

ПІОРИТЕТИ

Низький

Планові роботи, не впливає на виробництво

Середній

Часткове зниження продуктивності

Високий

Значний вплив на виробництво

Критичний

Зупинка виробництва, аварія

Рис 3.11 Процес створення звичайної заявки на ремонт

Після збереження даних система автоматично реєструє запит та перенаправляє користувача до списку, де відображається створена нова заявка (рис. 3.12).

Fish Feed CMMS

Управління ТО

ГОЛОВНЕ

Панель керування

ОБСЛУГОВУВАННЯ

Заявки на ремонт

Планове ТО

АКТИВИ

Обладнання

Склад запчастин

Простої

АНАЛІТИКА

Зайти

Фінансування

АДМІНІСТРУВАННЯ

Користувачі

Життєвий цикл

Admin Denys

Заявка WO-6A20A6978FDEF

Заявки / WO-6A20A6978FDEF

Статус заявки оновлено!

Аварійний ремонт шестерні Преса ПШ-1000

Критичний

В роботі

Редагувати

Видалити

ДЕТАЛІ ЗАЯВКИ

Обладнання

Прес ПШ-1000

Тип ремонту

Аварійний

Опис проблеми

Аварійний ремонт шестерні Преса ПШ-1000, вийшла з ладу шестерня, можливо пошкодила цілісність конструкції

Вартість ремонту

0.00 ₴ (запчастини: 0.00 ₴ + робота: 0.00 ₴)

Використані запчастини

Списати

ЗАПЧАСТИНА

КІЛЬКІСТЬ

ЦІНА/ШТ

РАЗОМ

КИМ

ДАТА

Запчастин не використано

ОНОВИТИ СТАТУС

Новий статус

В роботі

Оновити статус

ІНФОРМАЦІЯ

Відповідальний

Іван Коваль

Створив

Admin Denys

Дата створення

04.06.2026 01:11

Дедлайн

05.06.2026 03:10

Розпочато

04.06.2026 01:12

Рис 3.12 Демонстрація створеної нової заявки

У ході безпосереднього усунення несправності технічний спеціаліст фіксує списання запчастин, необхідних для проведення ремонту (рис. 3.13), що дозволяє автоматично зменшувати залишки на складі та розраховувати матеріальні витрати.

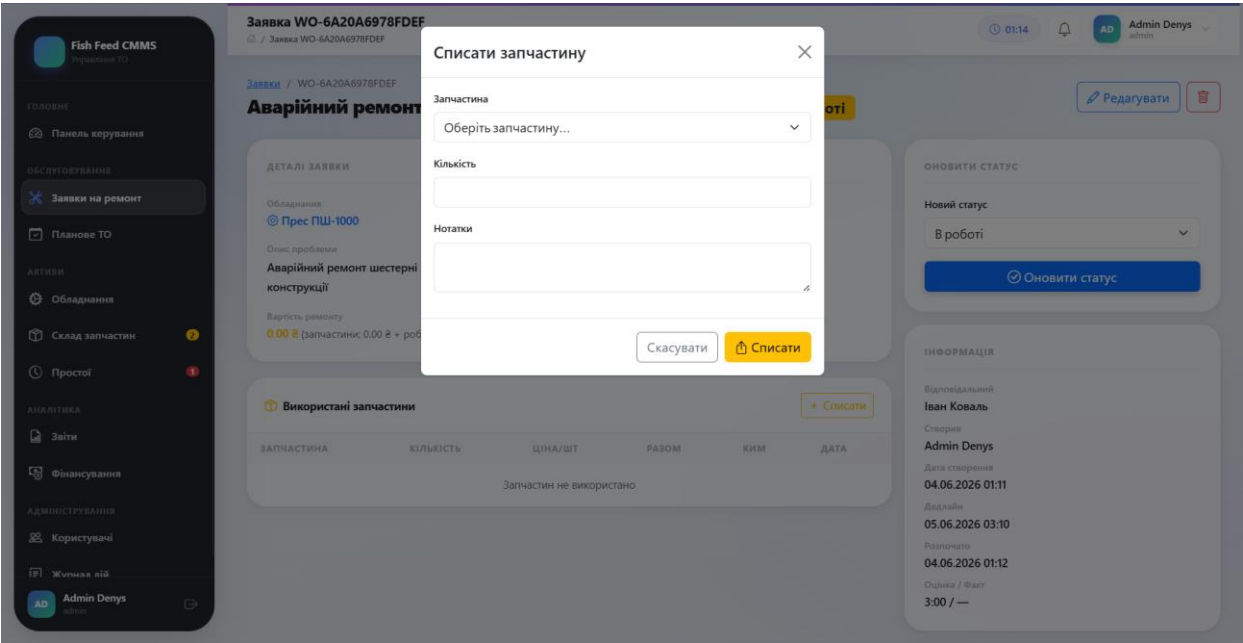


Рис 3.13 Списання запчастин які потрібні під час ремонту

Після завершення фізичних ремонтних робіт виконавець здійснює закриття наряду через форму завершення процесу ремонту (рис. 3.14).

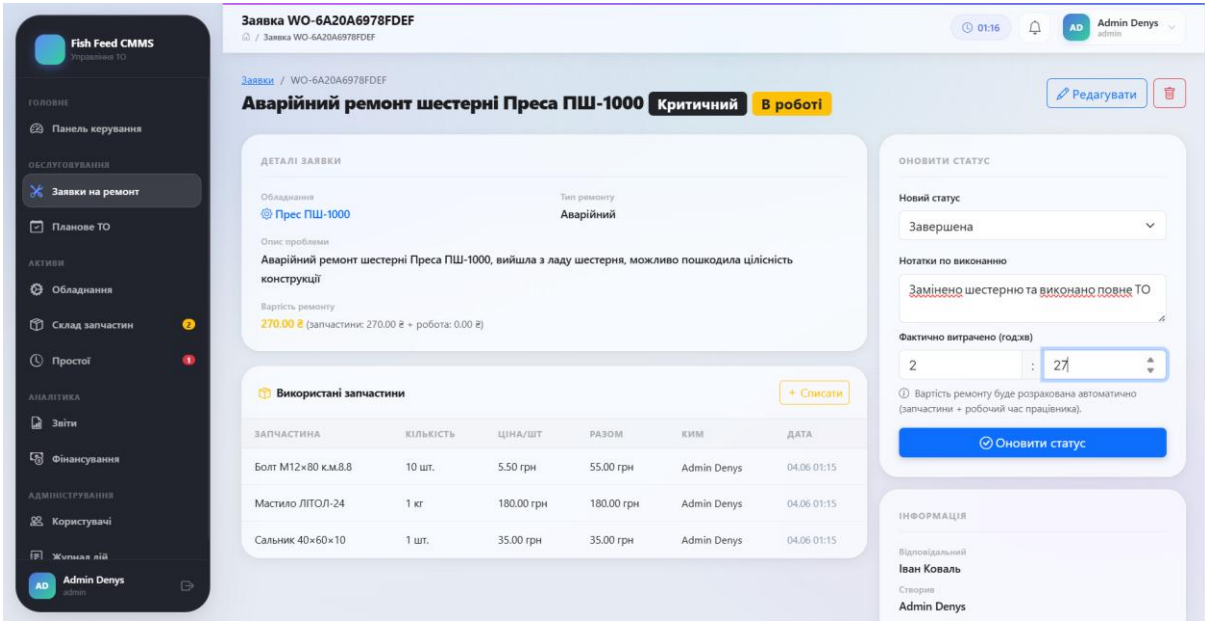


Рис 3.14 Завершення процесу ремонту

Якщо поломка призвела до повної зупинки виробничої лінії, у системі передбачено інтерфейс для фіксування простою обладнання під час ремонту (рис. 3.15) із зазначенням часу початку та завершення вимушеної зупинки.

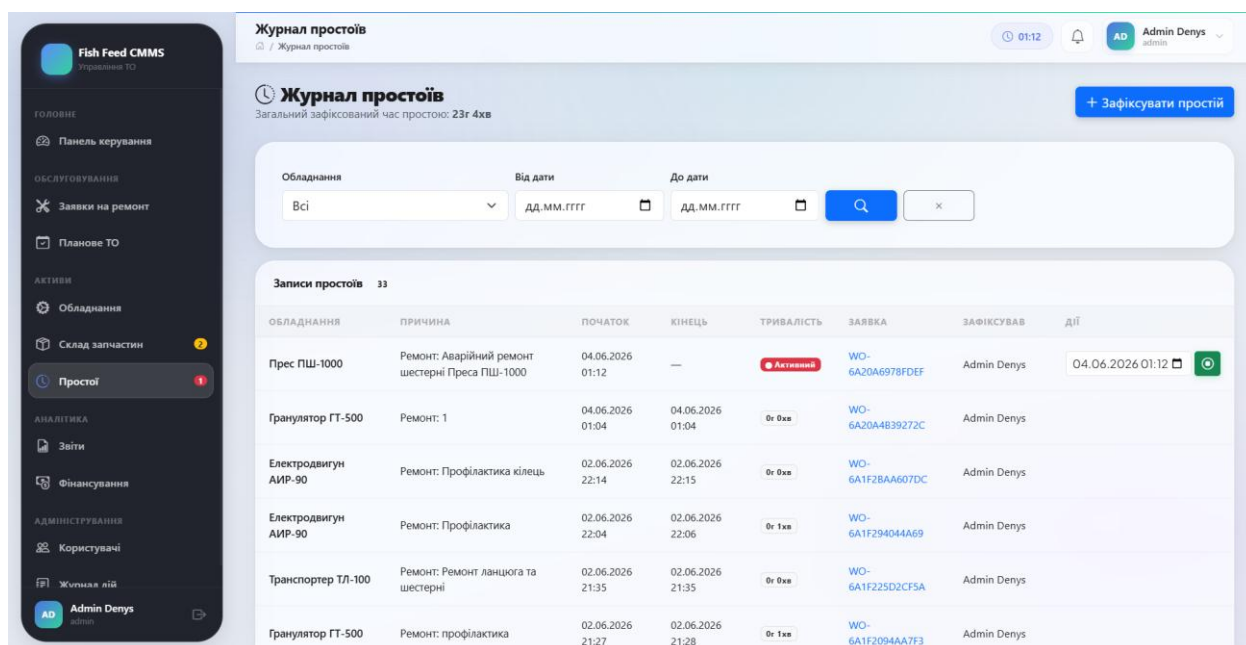


Рис 3.15 Фіксування простою обладнання під час ремонту

Для перегляду поточного стану виробничих потужностей призначено інтерфейс, що відображає обладнання, яке зараз перебуває на ремонті (рис. 3.16). Це дозволяє диспетчеру оцінювати ризики невиконання планів виробництва кормів.

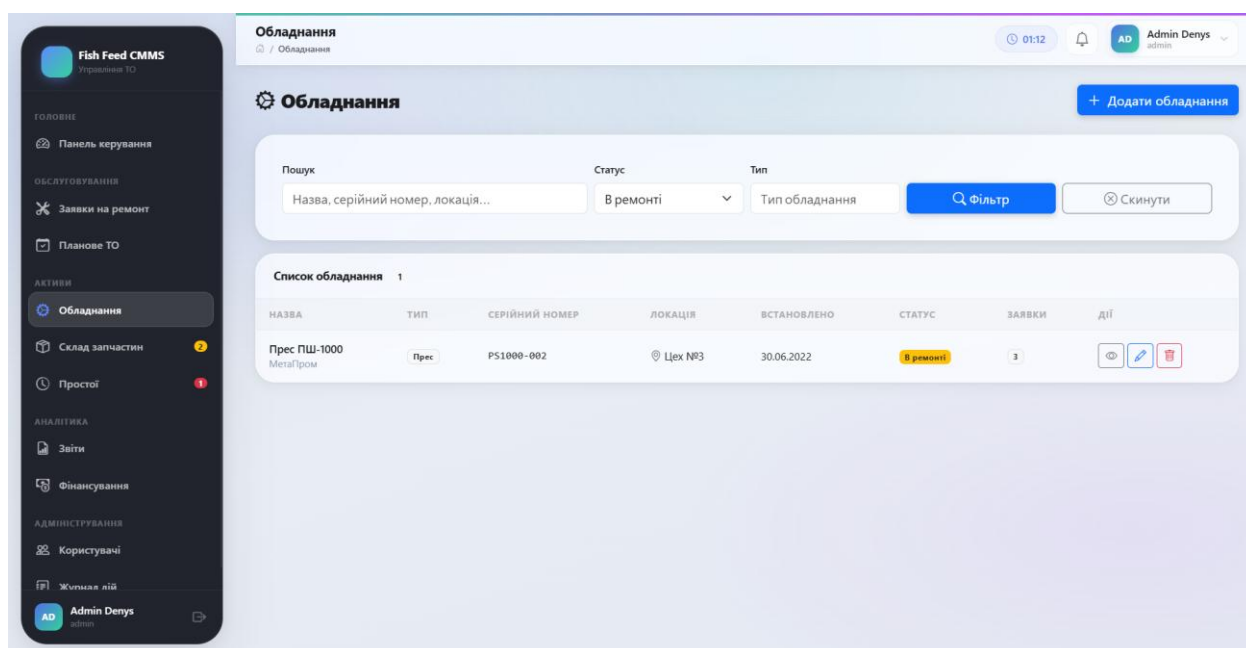


Рис 3.16 Обладнання яке зараз на ремонті

Детальні паспортні дані, історія ремонтів та поточний знос окремої машини відображаються на сторінці повної інформації про обладнання (рис. 3.17).

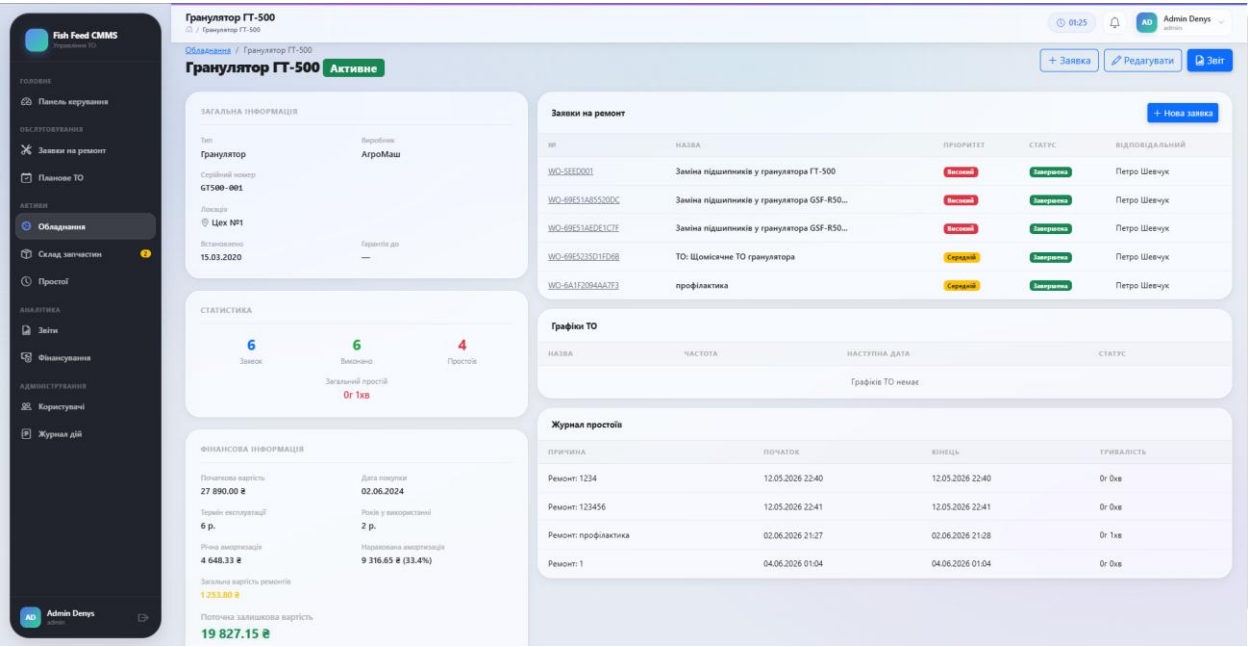


Рис 3.17 Сторінка повної інформації про обладнання

Після успішного завершення робіт система переводить обладнання в статус працездатного та виконує остаточний прорахунок вартості ремонту (рис. 3.18), який сумує вартість витрачених запчастин та вартість людино-годин роботи майстра.

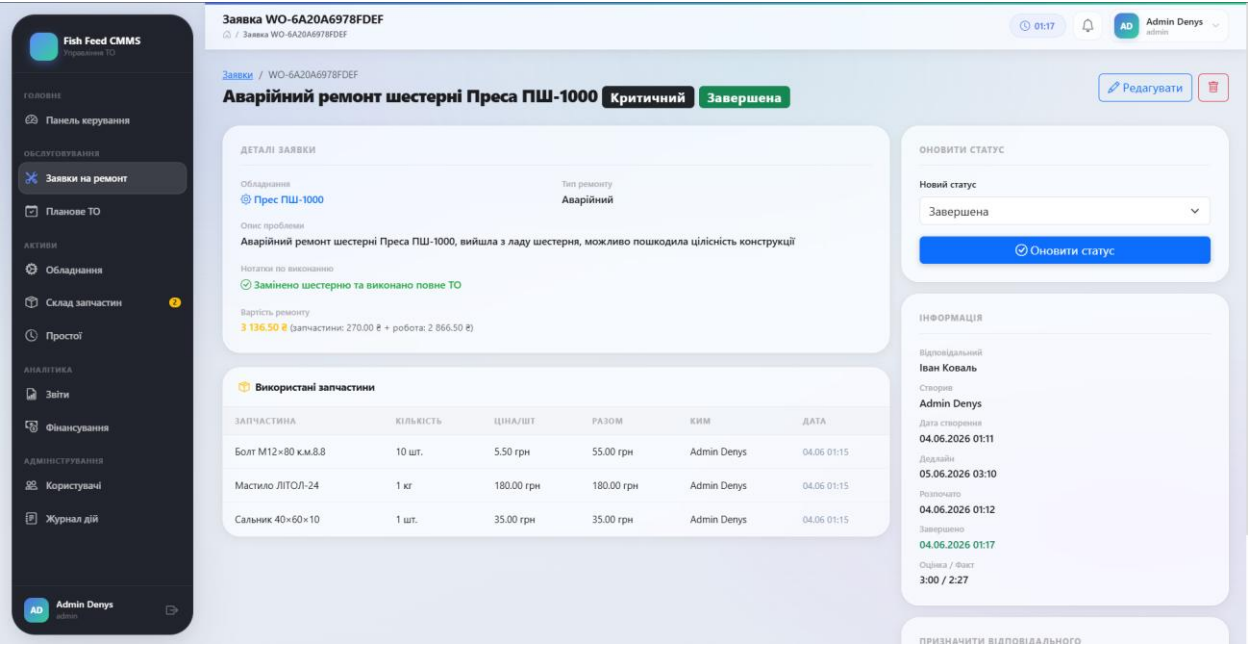


Рис 3.18 Завершення ремонту, прорахунок вартості ремонту

Для попередження аварійних ситуацій інженер може скористатися вікном створення планового ТО (рис. 3.19), задаючи періодичність та обсяг профілактичних робіт для кожної одиниці техніки.

Fish Feed CMMS

Панель керування

Обслуговування

Завдання на ремонт

Планове ТО

Активні

Обладнання

Склад запчастин

Простої

Аналітика

Звіт

Фінансування

Адміністрування

Користувачі

Журнал дій

Admin Denys

admin

Новий графік ТО

Новий графік ТО

01:20

Admin Denys

Планове ТО / Новий

Створити графік ТО

Назва \*

напр. Щомісячне ТО гранулятора

Обладнання \*

Відповідальний

Оберіть...

Не призначено

Частота \*

Оцінка часу (години) \*

Щоденно

2

:

0

Хвилини не можуть бути більше 59

Дата початку \*

04.06.2026

Опис / Чек-лист

Перелік робіт, чек-лист...

Створити

Скасувати

Перегляд графіка та керування всіма регламентними роботами здійснюється на сторінці всіх планових ТО (рис. 3.20).

Fish Feed CMMS

Планове ТО

ГОЛОВНІ

Панель керування

ОБСЛУГОВУВАННЯ

Завдання на ремонт

Планове ТО

АКТИВНІ

Обладнання

Склад запчастин

Простої

АНАЛІТИКА

Звіти

Фінансування

Адміністрування

Користувачі

Журнал дій

Admin Denys

Планове ТО

01:28

Admin Denys

Планове технічне обслуговування

+ Новий графік ТО

Пошук

Частота

Статус

Назва ТО...

Bci

Bci

🔍

✕

Графіки ТО 5

| НАЗВА                                               | ОБЛАДНАННЯ               | ЧАСТОТА        | НАСТУПНА ДАТА             | ВІДПОВІДАЛЬНИЙ | СТАТУС   | ДІЇ         |
|-----------------------------------------------------|--------------------------|----------------|---------------------------|----------------|----------|-------------|
| Сушарка Д5-300                                      | Сушарка барабанна ДВ-300 | Щоднічно       | 24.06.2024                | —              | Активний | 👁️ ⚙️ ✎️ 🗑️ |
| ТО Дозатора ДВ-200                                  | Дозатор ДВ-200           | Щотижнево      | 01.06.2024<br>Через 4 дні | Іван Коваль    | Активний | 👁️ ⚙️ ✎️ 🗑️ |
| ТО насоса НВ-50                                     | Насос відцентровий НВ-50 | Раз на 2 тижні | 16.06.2024                | Петро Шевчук   | Активний | 👁️ ⚙️ ✎️ 🗑️ |
| Піврічна перевірка компресора (відхилення атмосфер) | Прес ПШ-1000             | Раз на рік     | 01.12.2024                | Іван Коваль    | Активний | 👁️ ⚙️ ✎️ 🗑️ |
| Щотижнева перевірка компресора                      | Компресор КВ-75          | Щотижнево      | 01.06.2024<br>Через 4 дні | Петро Шевчук   | Активний | 👁️ ⚙️ ✎️ 🗑️ |

Для контролю за наявністю ремонтних комплектів на підприємстві реалізовано сторінку складу запчастин (рис. 3.21), де виводиться список деталей, їх поточна кількість на складі та вартість.

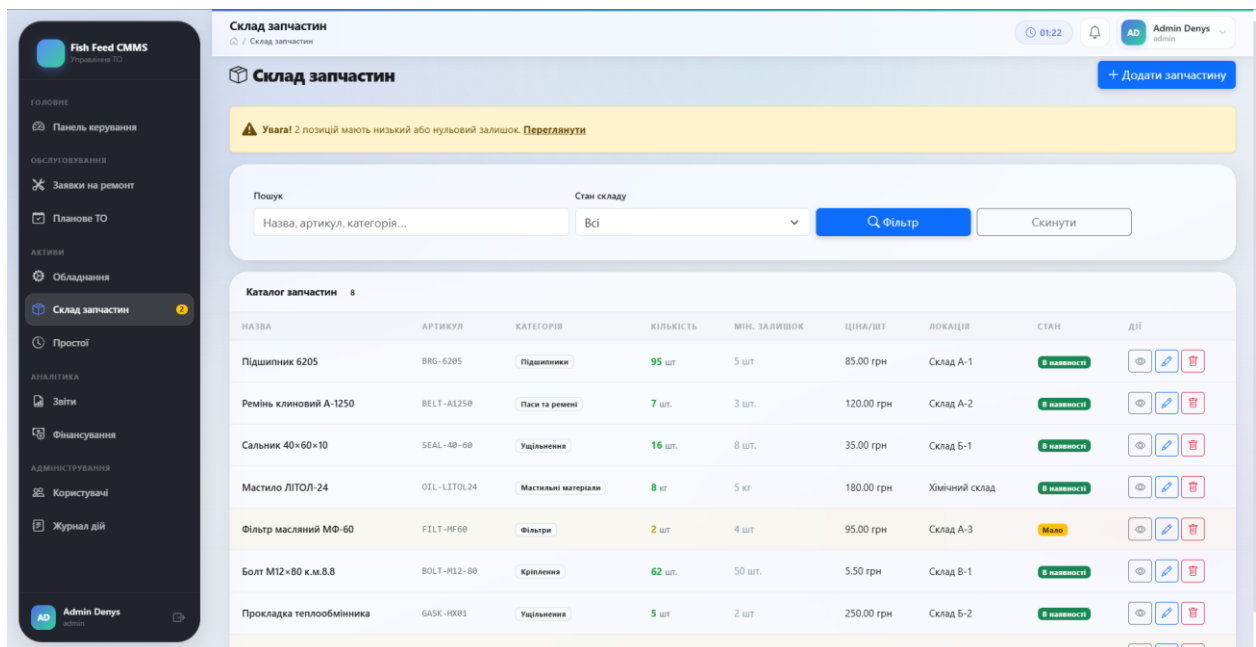


Рис 3.21 Склад запчастин

Додавання нових матеріалів або оприбуткування партії запчастин здійснюється за допомогою відповідної екранної форми (рис. 3.22).

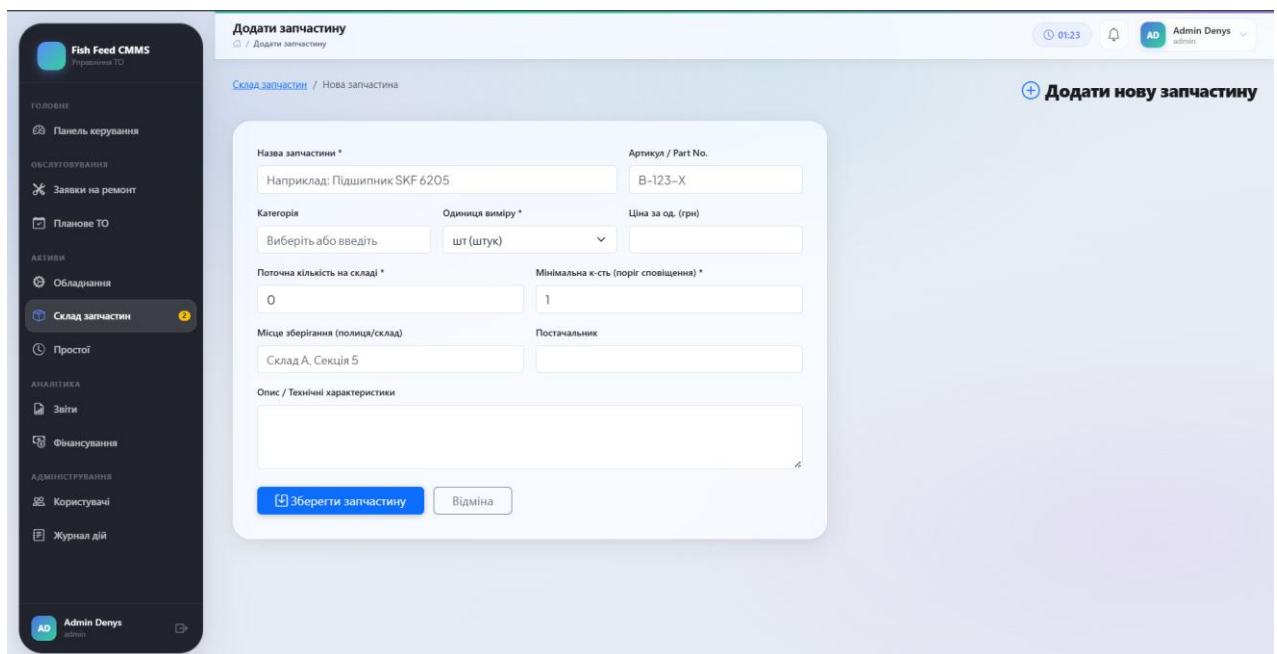


Рис 3.22 Додавання нової запчастини

Модуль звітності надає користувачам можливість формувати різноманітні аналітичні документи. Головна сторінка модуля звітів по заявках, обладнанню, простою та амортизації представлена на (рис. 3.23).

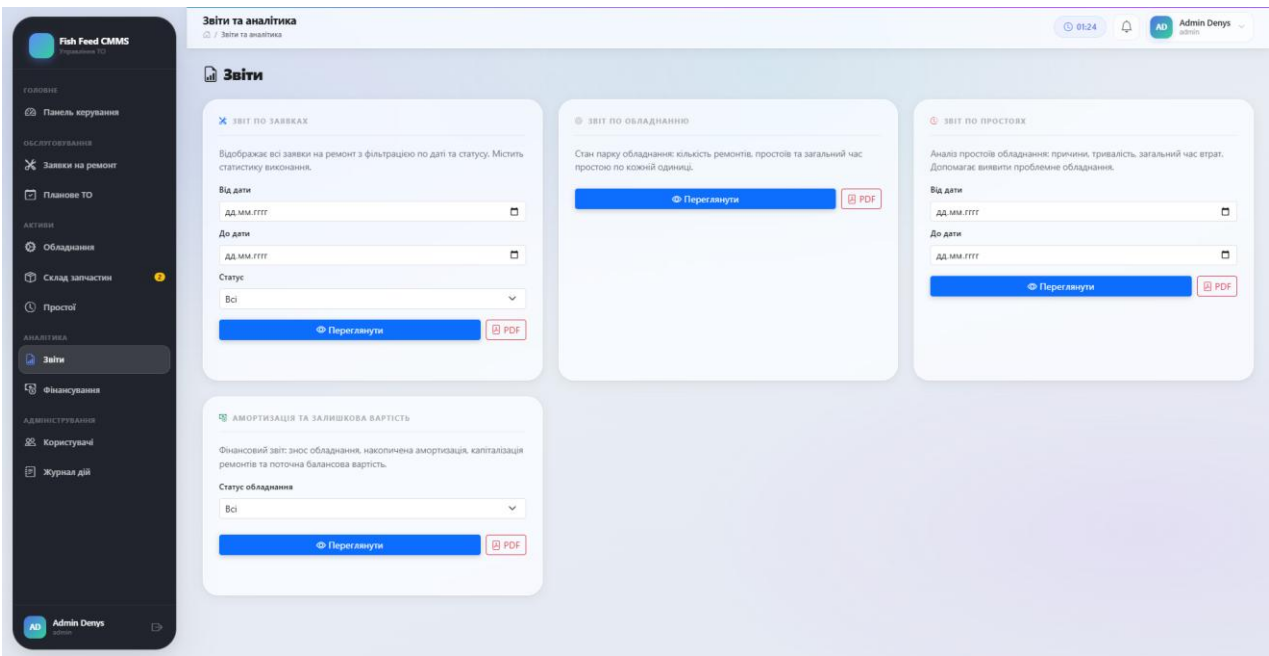


Рис 3.23 Звіти по заявках, обладнанню, простоях та амортизації

Для аналізу ефективності роботи ремонтної бригади призначена сторінка налаштування та створення звіту по заявках на ремонт (рис. 3.24).

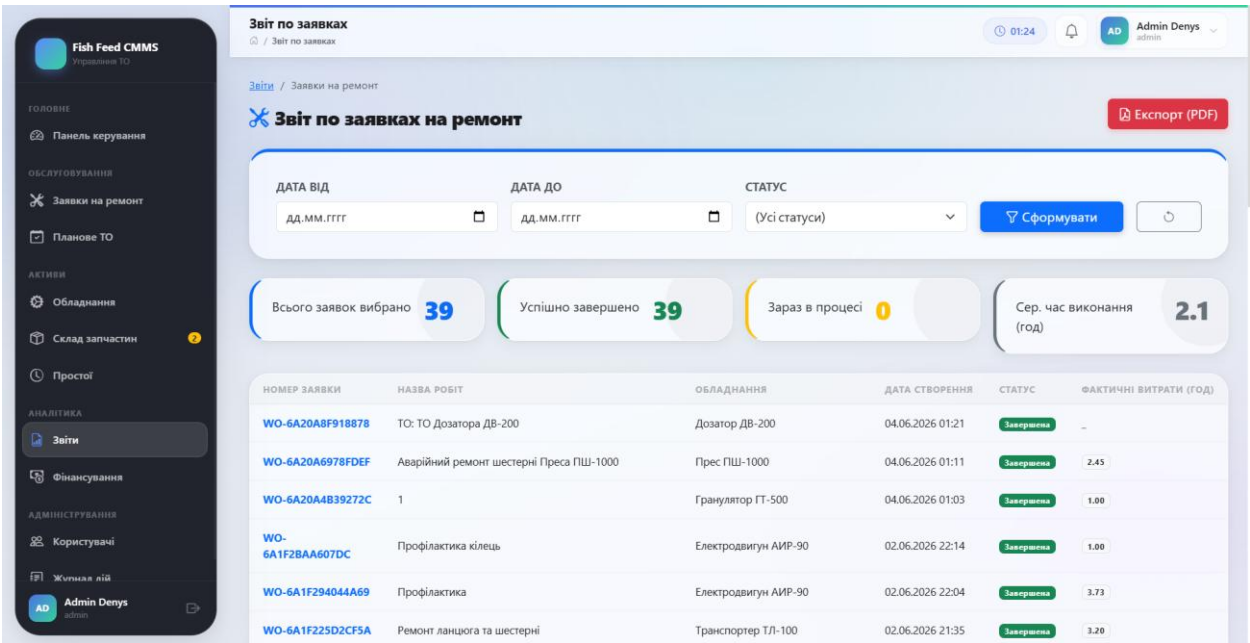


Рис 3.24 Створення звіту по заявках на ремонт

Сгенерований документ (рис. 3.25) містить детальну зведену таблицю щодо кількості закритих нарядів, витраченого часу та коштів за обраний період.

| Звіт по заявках на ремонт                           |                                         |                          |                |           |                   |            |            |
|-----------------------------------------------------|-----------------------------------------|--------------------------|----------------|-----------|-------------------|------------|------------|
| Fish Feed Plant CMMS   Сформовано: 04.06.2026 01:24 |                                         |                          |                |           |                   |            |            |
| 39                                                  | 39                                      | 0                        | 2.1            |           |                   |            |            |
| Всього                                              | Завершено                               | В роботі                 | Ср. час (год.) |           |                   |            |            |
| Номер                                               | Назва                                   | Обладнання               | Пріоритет      | Статус    | Відповідальний    | Факт. год. | Дата       |
| WO-6A20AB918878                                     | ТО: ТО Дозатора ДВ-200                  | Дозатор ДВ-200           | Середній       | Завершено | Іван Коваль       | —          | 04.06.2026 |
| WO-6A20A6978FDEF                                    | Аварійний ремонт шестерні Преса ПШ-1000 | Прес ПШ-1000             | Критичний      | Завершено | Іван Коваль       | 2.45       | 04.06.2026 |
| WO-6A20A4B39272C                                    | 1                                       | Гранулятор ГТ-500        | Середній       | Завершено | Григорій Павленко | 1.00       | 04.06.2026 |
| WO-6A1F2BA46070C                                    | Профілактика клієць                     | Електроагрегат АІР-90    | Середній       | Завершено | Григорій Павленко | 1.00       | 02.06.2026 |
| WO-6A1F294044A69                                    | Профілактика                            | Електроагрегат АІР-90    | Середній       | Завершено | Григорій Павленко | 3.73       | 02.06.2026 |
| WO-6A1F225D2CF3A                                    | Ремонт ланцюга та шестерні              | Транспортер ТЛ-100       | Високий        | Завершено | Григорій Павленко | 3.20       | 02.06.2026 |
| WO-6A1F2094A47F3                                    | профілактика                            | Гранулятор ГТ-500        | Середній       | Завершено | Петро Шевчук      | 0.57       | 02.06.2026 |
| WO-6A1F18EC2FCDE                                    | Аварійний ремонт                        | Компресор КВ-75          | Критичний      | Завершено | Іван Коваль       | 2.00       | 02.06.2026 |
| WO-6A1F14C89D4C                                     | Профілактика                            | Компресор КВ-75          | Низький        | Завершено | Григорій Павленко | 1.93       | 02.06.2026 |
| WO-6A1F1971424FB                                    | ТО: Щотижнева перевірка компресора      | Компресор КВ-75          | Середній       | Завершено | Петро Шевчук      | 2.00       | 02.06.2026 |
| WO-6A1F196883919                                    | ТО: тест 4                              | Дозатор ДВ-200           | Середній       | Завершено | Іван Коваль       | 1.00       | 02.06.2026 |
| WO-6A1F19575D30C                                    | ТО: тест 2                              | Насос відцентровий НВ-50 | Середній       | Завершено | Петро Шевчук      | 4.00       | 02.06.2026 |

Рис 3.25 Автоматично створений звіт по заявках на ремонт

Для отримання загальної картини зносу активів використовується інтерфейс створення звіту по всьому обладнанню (рис. 3.26) та його автоматично створений результат (рис. 3.27).

| Звіт по обладнанню                       |                     |             |         |               |                     |  |  |
|------------------------------------------|---------------------|-------------|---------|---------------|---------------------|--|--|
| Звіт по обладнанню                       |                     |             |         |               |                     |  |  |
| Звіти / Обладнання                       |                     |             |         |               |                     |  |  |
| Звіт по обладнанню                       |                     |             |         |               |                     |  |  |
| Експорт (PDF)                            |                     |             |         |               |                     |  |  |
| Оберіть обладнання для детального звіту: |                     |             |         |               |                     |  |  |
| -- Всі (загальний звіт) --               |                     |             |         |               |                     |  |  |
| Сформувати                               |                     |             |         |               |                     |  |  |
| Всього одиниць 10                        |                     |             |         |               |                     |  |  |
| Активні (В роботі) 9                     |                     |             |         |               |                     |  |  |
| В ремонті зараз 0                        |                     |             |         |               |                     |  |  |
| Списані 1                                |                     |             |         |               |                     |  |  |
| НАЗВА ОБЛАДНАННЯ                         | МОДЕЛЬ / СЕРІЙНИЙ № | ЛОКАЦІЯ     | СТАТУС  | ВСЬОГО ЗАЯВОК | ЧАС ПРОСТОїв (ХВ)   |  |  |
| Гранулятор ГТ-500                        | GT500-001           | Цех №1      | Активне | 6             | 1.65 хв             |  |  |
| Змішувач СГ-200                          | SG200-001           | Цех №1      | Списане | 0             | —                   |  |  |
| Транспортер ТЛ-100                       | TL100-003           | Цех №2      | Активне | 2             | 0.38333333333333 хв |  |  |
| Сушарка барабанна ДВ-300                 | DB300-001           | Цех №2      | Активне | 4             | 79.633333333333 хв  |  |  |
| Компресор КВ-75                          | KV75-002            | Компресорна | Активне | 7             | 13.066666666667 хв  |  |  |

Рис 3.26 Створення звіту по всьому обладнанню

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 76   |

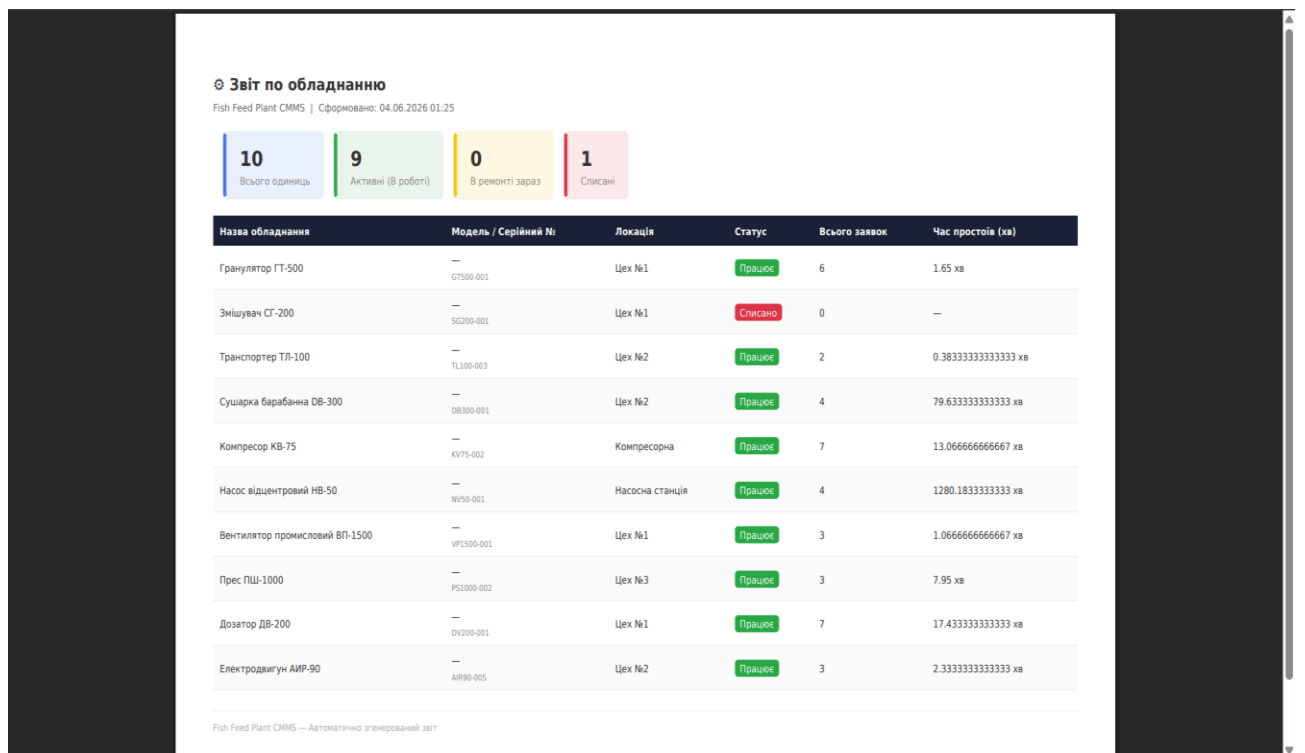


Рис 3.27 Автоматично створений звіт по всьому обладнанню

Якщо необхідно проаналізувати життєвий цикл конкретної дороговартісної машини (наприклад, головного екструдера), система дозволяє виконати створення звіту по одній одиниці обладнання (рис. 3.28).

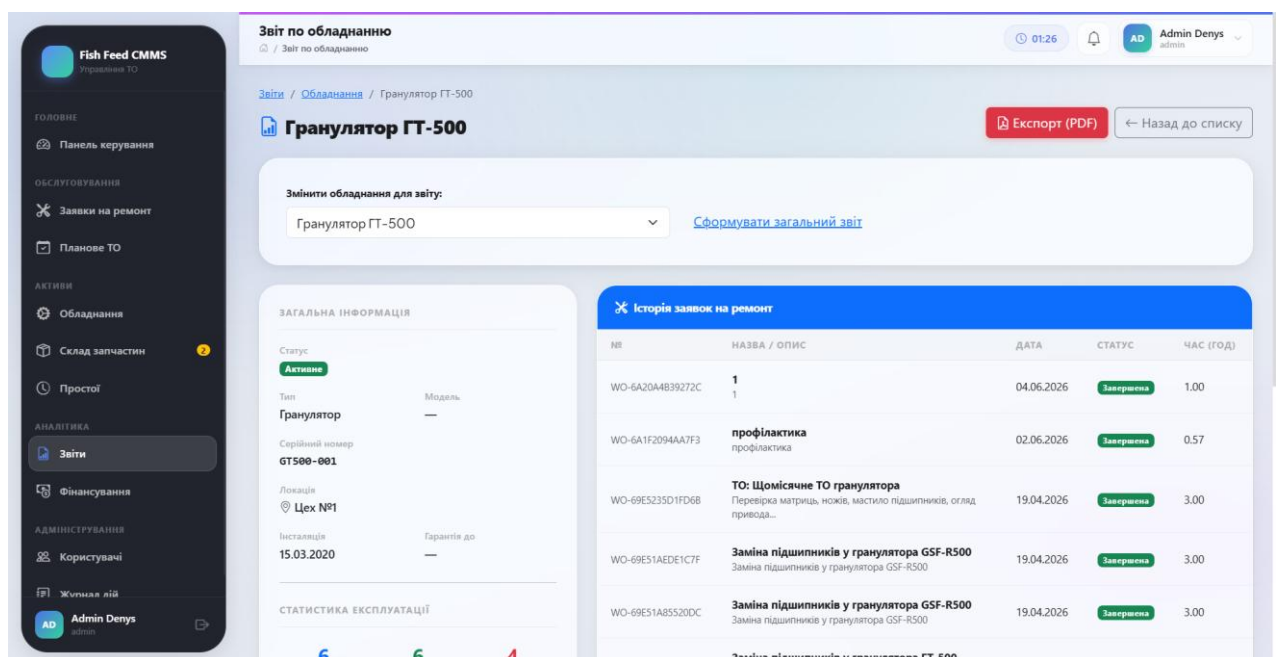


Рис 3.28 Створення звіту по одній одиниці обладнання

Результат генерації звіту по одній одиниці обладнання наведено на (рис. 3.29). Сформований звіт містить повну інформацію про вибране обладнання, зокрема його основні характеристики, технічні параметри, поточний стан, а також дані щодо проведених робіт з обслуговування та ремонту.

ТЕХНІЧНИЙ ЗВІТ ОБЛАДНАННЯ

Гранулятор GT-500

Сформовано: 04.06.2026 10:27

Загальна інформація

|             |            |             |           |
|-------------|------------|-------------|-----------|
| Тип:        | Гранулятор | Версія:     | АгроMax   |
| Модель:     | —          | Серійний №: | GT500-001 |
| Локація:    | Цех №1     | Статус:     | active    |
| Інсталяція: | 15.03.2020 | Гарантія:   | —         |

Статистика експлуатації

|                |   |                    |   |                     |   |                    |        |
|----------------|---|--------------------|---|---------------------|---|--------------------|--------|
| Всього заявок: | 6 | Виконано ремонтів: | 6 | Кількість простоїв: | 4 | Загальний простій: | 0: 1хв |
|----------------|---|--------------------|---|---------------------|---|--------------------|--------|

Історія заявок на ремонт

| №                | Назва / Опис                              | Дата       | Статус    | Відповідальний    |
|------------------|-------------------------------------------|------------|-----------|-------------------|
| №0-64204838072C  | 1                                         | 04.06.2026 | completed | Григорій Павленко |
| №0-641720944473  | профілактика                              | 02.06.2026 | completed | Петро Шенчук      |
| №0-686123200704B | ТО: Щоденне ТО гранулятора                | 19.04.2026 | completed | Петро Шенчук      |
| №0-686123200704B | Заміна підшипників у гранулятора GSF-8500 | 19.04.2026 | completed | Петро Шенчук      |
| №0-686123200704B | Заміна підшипників у гранулятора GSF-8500 | 19.04.2026 | completed | Петро Шенчук      |
| №0-686123200704B | Заміна підшипників у гранулятора GT-500   | 13.04.2026 | completed | Петро Шенчук      |

Графіки планового ТО

| Назва | Частота | Останнє ТО | Наступне ТО | Активний |
|-------|---------|------------|-------------|----------|
|-------|---------|------------|-------------|----------|

Журнал простоїв

| Примітка             | Початок          | Кінець           | Тривалість |
|----------------------|------------------|------------------|------------|
| Ремонт: 1            | 04.06.2026 01:04 | 04.06.2026 01:04 | 0: 0хв     |
| Ремонт: профілактика | 02.06.2026 22:27 | 02.06.2026 23:28 | 0: 1хв     |

Рис 3.29 Автоматично створений звіт по одній одиниці обладнання

Для детального аналізу надійності та ефективності використання виробничих потужностей у системі передбачено спеціалізовані інтерфейси створення аналітичного звіту по простоях обладнання (рис. 3.30).

| Звіт по простоях          |  | 01:28                                   |  | Admin Denys                              |  |
|---------------------------|--|-----------------------------------------|--|------------------------------------------|--|
| Звіт по простоях          |  | Звіт по простоях                        |  | Експорт (PDF)                            |  |
| Звіт / Простой обладнання |  | ДАТА ПОЧАТКУ (ВІД)                      |  | ДАТА ПОЧАТКУ (ДО)                        |  |
| Звіт по простоях          |  | ДД.ММ.ГГГГ                              |  | ДД.ММ.ГГГГ                               |  |
| Звіт по простоях          |  | Сформувати                              |  | Звіт по простоях                         |  |
| Звіт по простоях          |  | Загальна кількість зафіксованих зупинок |  | Сумарний час простоїв за вибраний період |  |
| Звіт по простоях          |  | 34                                      |  | 23 год. 9 хв.                            |  |
| Звіт по простоях          |  | ОБЛАДНАННЯ                              |  | ПОЧАТОК ПРОСТОЮ                          |  |
| Звіт по простоях          |  | КІНЕЦЬ ПРОСТОЮ                          |  | ТРИВАЛІСТЬ (ХВ)                          |  |
| Звіт по простоях          |  | ПРИЧИНА (ОПИС)                          |  | ЗАФІКСУВАВ                               |  |
| Звіт по простоях          |  | Дозатор ДВ-200                          |  | 04.06.2026 01:21                         |  |
| Звіт по простоях          |  | Прес ПШ-1000                            |  | 04.06.2026 01:12                         |  |
| Звіт по простоях          |  | Гранулятор ГТ-500                       |  | 04.06.2026 01:04                         |  |
| Звіт по простоях          |  | Електродвигун АИР-90                    |  | 02.06.2026 22:14                         |  |
| Звіт по простоях          |  | Електродвигун АИР-90                    |  | 02.06.2026 22:06                         |  |
| Звіт по простоях          |  | Транспортер ТЛ-100                      |  | 02.06.2026 21:35                         |  |

Рис 3.30 Створення звіту по простоям

Перегляд автоматично сформованого звіту по простоях (рис. 3.31). Сторінка формування звіту дозволяє інженеру задати часовий діапазон аналізу, вказавши дату початку та закінчення вибірки. На підставі введених параметрів контролер ReportController відбирає всі записи з таблиці downtime\_logs та обчислює сукупний час простою, виражаючи результат у годинах та хвилинах.

**Звіт по простоях обладнання**  
Fish Feed Plant CMMS | Сформовано: 04.06.2026 01:28

**34** **23 год. 9 хв.**  
Всього зафіксовано простой Сукупний час зупинки

| Обладнання                     | Початок простою  | Кінець простою   | Тривалість (хв) | Причина (опис)                                  | Зафіксував        |
|--------------------------------|------------------|------------------|-----------------|-------------------------------------------------|-------------------|
| Дозатор ДВ-200                 | 04.06.2026 01:21 | 04.06.2026 01:21 | —               | Ремонт: ТО: ДО Дозатора ДВ-200                  | Admin Denys       |
| Прес ПШ-1000                   | 04.06.2026 01:12 | 04.06.2026 01:17 | 5 хв            | Ремонт: Аварійний ремонт шестерні Преса ПШ-1000 | Admin Denys       |
| Гранулятор ГТ-500              | 04.06.2026 01:04 | 04.06.2026 01:04 | —               | Ремонт: 1                                       | Admin Denys       |
| Електродвигун АІР-90           | 02.06.2026 22:14 | 02.06.2026 22:15 | —               | Ремонт: Профілактика кілець                     | Admin Denys       |
| Електродвигун АІР-90           | 02.06.2026 22:04 | 02.06.2026 22:06 | 1 хв            | Ремонт: Профілактика                            | Admin Denys       |
| Транспортер ТЛ-100             | 02.06.2026 21:35 | 02.06.2026 21:35 | —               | Ремонт: Ремонт ланцюга та шестерні              | Admin Denys       |
| Гранулятор ГТ-500              | 02.06.2026 21:27 | 02.06.2026 21:28 | 1 хв            | Ремонт: профілактика                            | Admin Denys       |
| Компресор КВ-75                | 02.06.2026 21:09 | 02.06.2026 21:19 | 9 хв            | Ремонт: Аварійний ремонт                        | Admin Denys       |
| Компресор КВ-75                | 02.06.2026 21:07 | 02.06.2026 21:08 | —               | Ремонт: Аварійний ремонт                        | Admin Denys       |
| Компресор КВ-75                | 02.06.2026 21:02 | 02.06.2026 21:02 | —               | Ремонт: Профілактика                            | Admin Denys       |
| Компресор КВ-75                | 02.06.2026 20:57 | 02.06.2026 20:58 | 1 хв            | Ремонт: ТО: Щоденна перевірка компресора        | Admin Denys       |
| Дозатор ДВ-200                 | 02.06.2026 20:56 | 02.06.2026 21:01 | 4 хв            | Ремонт: ТО: тест 4                              | Admin Denys       |
| Насос відцентровий НВ-50       | 02.06.2026 20:56 | 02.06.2026 21:03 | 6 хв            | Ремонт: ТО: тест 2                              | Admin Denys       |
| Дозатор ДВ-200                 | 25.05.2026 22:56 | 25.05.2026 22:59 | 3 хв            | Ремонт: Звх                                     | Admin Denys       |
| Компресор КВ-75                | 25.05.2026 22:54 | 25.05.2026 22:54 | —               | Ремонт: вілпи                                   | Григорій Павленко |
| Сушарка барабанна ОБ-300       | 25.05.2026 21:16 | 25.05.2026 22:17 | 60 хв           | Ремонт: 12123123                                | Admin Denys       |
| Вентилятор промисловий ВП-1500 | 25.05.2026 21:06 | 25.05.2026 21:07 | —               | Ремонт: хмсин                                   | Admin Denys       |

Рис 3.31 Автоматично створений звіт простоям

**Звіт по амортизації**  
Звіт / Амортизація

Статус обладнання: -- Всі -- **Сформувати**

**Початкова вартість 210 690.00 ₴** **Залишкова вартість 148 044.62 ₴** **Витрати на ремонт 15 243.50 ₴** **Амортизація 77 948.70 ₴** **Експорт (PDF)**

| НАЗВА ОБЛАДНАННЯ     | ДАТА ПОКУПИ | РОКІВ у ВИКОР. | ПОЧАТКОВА ВАРТІСТЬ | ВАРТІСТЬ РЕМОНТІВ | АМОРТИЗАЦІЯ         | ЗАЛИШКОВА ВАРТІСТЬ |
|----------------------|-------------|----------------|--------------------|-------------------|---------------------|--------------------|
| Гранулятор ГТ-500    | 02.06.2024  | 2 р.           | 27 890.00 ₴        | 1 253.80 ₴        | 9 316.65 ₴ (33.4%)  | 19 827.15 ₴        |
| Транспортер ТЛ-100   | 01.03.2023  | 3.09 р.        | 157 000.00 ₴       | 3 015.00 ₴        | 48 574.23 ₴ (30.9%) | 111 440.77 ₴       |
| Компресор КВ-75      | 01.01.2024  | 2.42 р.        | 12 000.00 ₴        | 6 406.00 ₴        | 12 000.00 ₴ (100%)  | 6 406.00 ₴         |
| Електродвигун АІР-90 | 02.02.2024  | 2.34 р.        | 13 800.00 ₴        | 4 568.70 ₴        | 8 057.82 ₴ (58.4%)  | 10 310.88 ₴        |

Рис 3.32 Створення звіту по амортизації

Оцінка фінансового зносу активів та розрахунок залишкової вартості фондів виконується через форму створення звіту по амортизації (рис. 3.32). Готовий звіт про амортизаційні нарахування представлено на рис. 3.33.

|      |      |               |        |      |                           |      |
|------|------|---------------|--------|------|---------------------------|------|
|      |      | Козачок Д. П. |        |      | ПІЗ.КР.Б – 121 – 26 – ПІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                           |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                           | 79   |

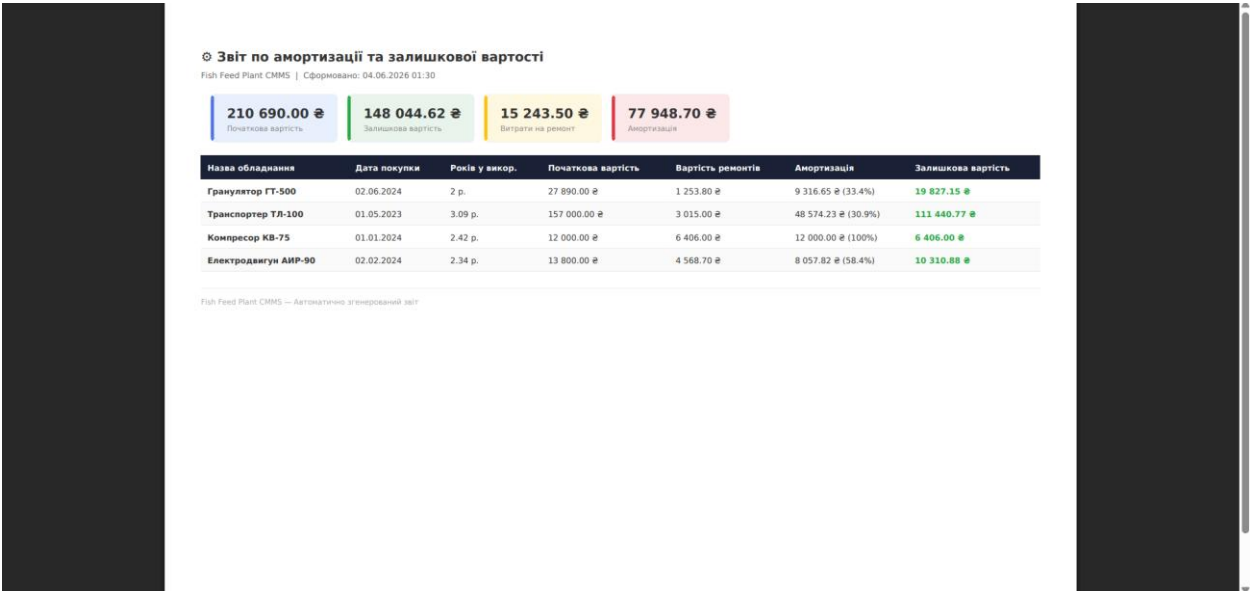


Рис 3.33 Автоматично створений звіт по амортизації

Фінансовий модуль системи відображає загальні бюджети на утримання заводу. Сторінка обліку фінансування (рис. 3.34) дозволяє бачити надходження та загальний ліміт витрат на ТО.

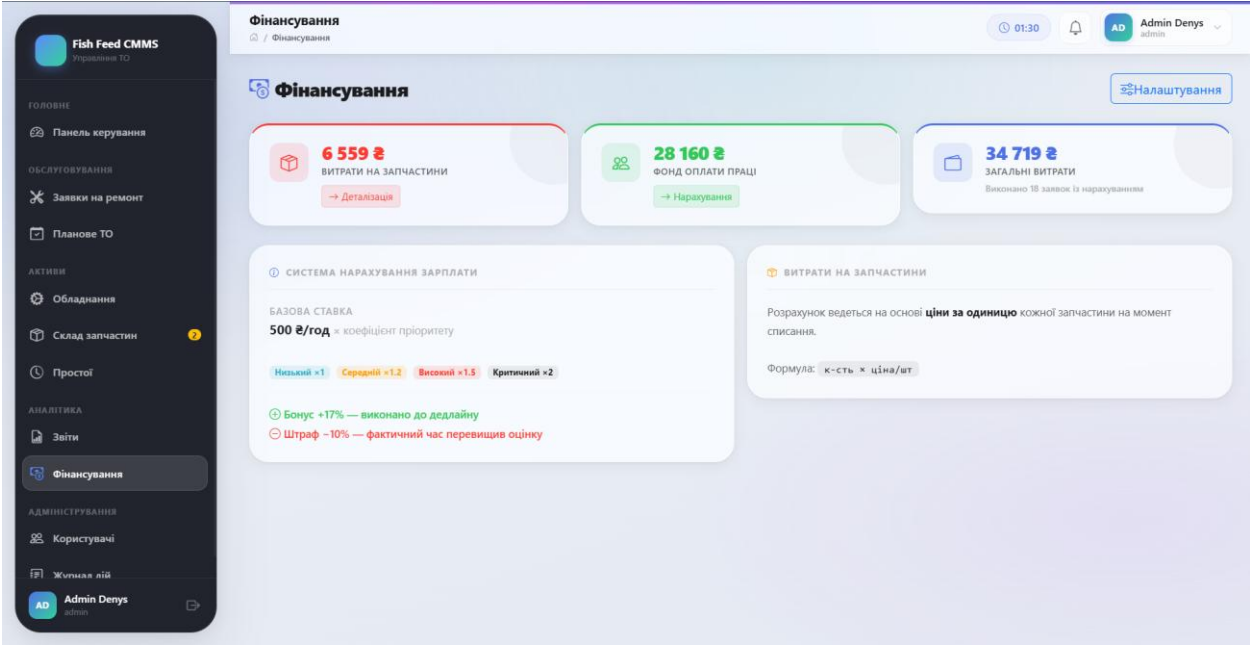


Рис 3.34 Облік фінансівання

Для гнучкого розрахунку вартості праці призначено інтерфейс налаштування коефіцієнтів нарахування вартості за ремонт (рис. 3.35), де можна вказати погодинну тарифну ставку або коефіцієнт складності робіт.

**Фінансування**

Фінансування / Налаштування

### Налаштування фінансування

**БАЗОВІ ПАРАМЕТРИ**

Базова годинна ставка (₴/год)  
500

Базова сума оплати за фактично відпрацьовані години до застосування коефіцієнтів та бонусів.

Бонус за дотримування термінів (%)  
17

Відсоток надбавки до базової оплати, якщо ремонт завершено до дедлайну.

Штраф за перевищення часу (%)  
10

Відсоток вирахування від базової оплати, якщо фактичний час перевищив оцінку.

**% КОЕФІЦІЄНТИ ПРІОРИТЕТУ**

| Коефіцієнт | Коефіцієнт |
|------------|------------|
| Низький    | 1          |
| Середній   | 1,2        |
| Високий    | 1,5        |
| Критичний  | 2          |

**ПОТОЧНІ НАЛАШТУВАННЯ**

**БАЗОВА СТАВКА**  
**500 ₴/год**

**БОНУС**  
**+17%**  
до дедлайну

**ШТРАФ**  
**-10%**  
перевищення часу

**КОЕФІЦІЄНТИ ПРІОРИТЕТУ**

|           |      |
|-----------|------|
| Низький   | x1   |
| Середній  | x1.2 |
| Високий   | x1.5 |
| Критичний | x2   |

Рис 3.35 налаштування коефіцієнтів нарахування вартості за ремонт

Сторінка обліку витрат на запчастини (рис. 3.36) акумулює фінансову звітність щодо придбання та списання деталей зі складу.

**Фінансування**

Фінансування / Витрати на запчастини

### Витрати на запчастини

Від дати: DD.MM.YYYY До дати: DD.MM.YYYY Обладнання: Bci Тип заявки: Bci

**Загальна сума витрат**  
**6 559.00 ₴**

Кількість записів: 24

**ТОП ЗАПЧАСТИН ЗА ВИТРАТАМИ**

|                          |        |            |
|--------------------------|--------|------------|
| Підшипник 6205           | 31 шт  | 2 635.00 ₴ |
| Прокладка теплообмінника | 8 шт   | 2 000.00 ₴ |
| Масило ЛІТОЛ-24          | 7 кг   | 1 260.00 ₴ |
| Болт М12×80 к.м.8.8      | 38 шт. | 209.00 ₴   |
| Запобіжник BWF25         | 3 кг   | 195.00 ₴   |

**Деталізація списань 24**

| ЗАПЧАСТИНА                  | К-СТЬ | ЦІНА/ШТ | СУМА    | ЗАЯВКА          | ОБЛАДНАННЯ   | КИМ СПИСАНО | ДАТА             |
|-----------------------------|-------|---------|---------|-----------------|--------------|-------------|------------------|
| Сальник 40×60×10 SEAL-40-60 | 1 шт. | 35.00 ₴ | 35.00 ₴ | №: 6A20M69Z8DEE | Прес ПШ-1000 | Admin Denys | 04.06.2026 01:15 |

Рис 3.36 Облік витрат на запчастини

Система також підтримує автоматичний прорахунок заробітної плати кожному працівнику (рис. 3.37) на основі кількості та складності виконаних ним заявок на ремонт за місяць.

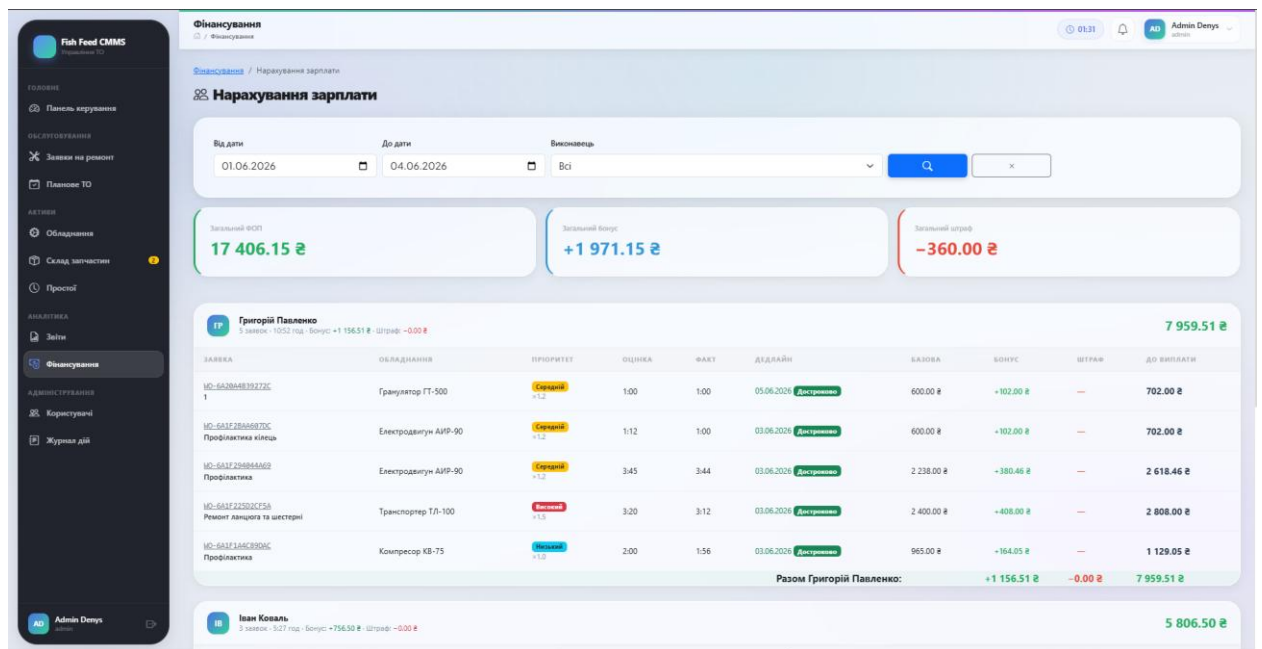


Рис 3.37 Прорахунок зарплати кожному працівнику

Для контролю доступу адміністратор використовує сторінку користувачів (рис. 3.38), де можна додавати, блокувати та редагувати облікові записи співробітників.

| ПІВ               | EMAIL                  | РОЛЬ       | ПОСАДА / ВІДДІЛ                           | ТЕЛЕФОН       | СТАТУС   | ДІЙ                     |
|-------------------|------------------------|------------|-------------------------------------------|---------------|----------|-------------------------|
| Григорій Павленко | technician2@cmms.local | technician | Технік<br>Технічний відділ                | 0679969186    | Активний | [Редагувати] [Видалити] |
| Admin Denys       | admin@cmms.local       | admin      | Системний адміністратор<br>ІТ відділ      | +380679969186 | Активний | [Редагувати]            |
| Іван Коваль       | engineer@cmms.local    | engineer   | Головний інженер<br>Технічний відділ      | +380672345678 | Активний | [Редагувати] [Видалити] |
| Петро Шевчук      | technician@cmms.local  | technician | Технік<br>Технічний відділ                | +380673456789 | Активний | [Редагувати] [Видалити] |
| Марія Бондар      | operator@cmms.local    | operator   | Оператор виробництва<br>Виробничий відділ | +380674567890 | Активний | [Редагувати] [Видалити] |

Рис 3.38 Користувачі

З метою аудиту безпеки та відстеження дій персоналу розроблено журнал дій користувачів (рис. 3.39), куди записуються всі важливі системні події із міткою часу та ID користувача.

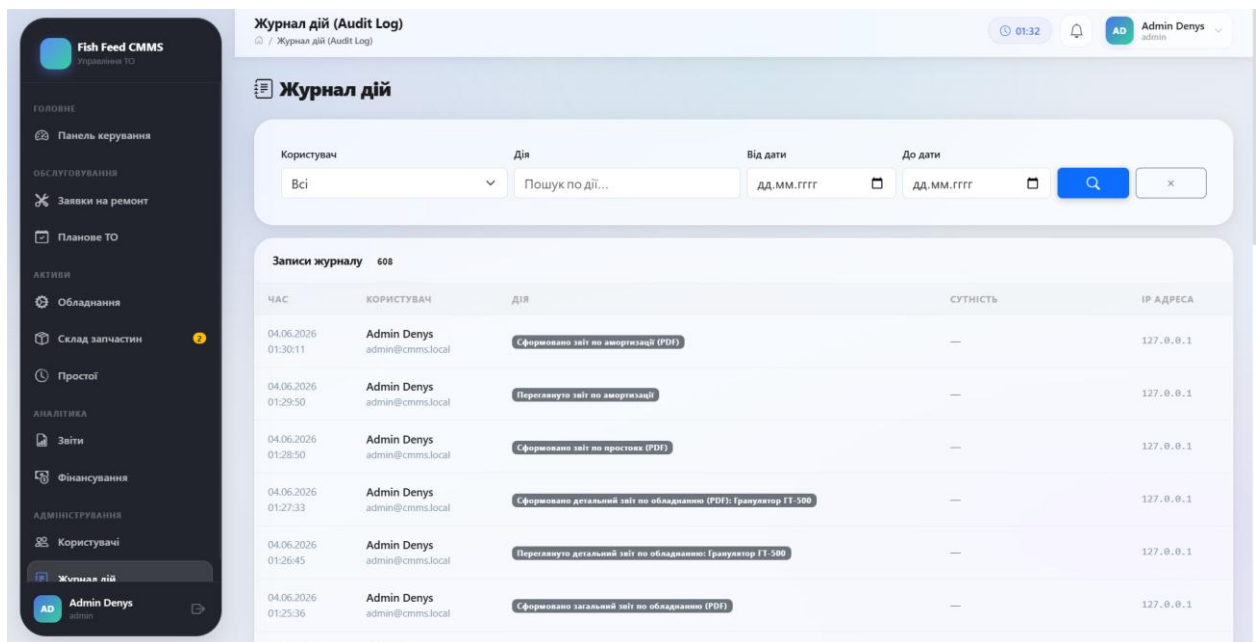


Рис 3.39 Журнал дій користувачів

На представленому інтерфейсі журналу дій користувачів (рис. 3.39) наочно відображено табличну структуру фіксації системних подій, яка містить інформацію про ім'я оператора, характер виконаної транзакції, тип ідентифікованої моделі, а також точні технічні параметри, включаючи ІР-адресу клієнта та системні дані пристрою.

### 3.3 Тестування системи

Для підтвердження працездатності розробленого програмного забезпечення та відповідності вимогам технічного завдання було проведено комплексне функціональне тестування. Найбільше часу було виділено на перевірку ядра системи - алгоритмів розрахунку вартості нарядів на ремонт, автоматичного списання запчастин та перерахунку залишкової амортизаційної вартості обладнання, оскільки від точності цих обчислень залежить коректність всієї аналітичної звітності підприємства.

Тестування алгоритмів обліку витрат та амортизації включало наступні кроки: перевірка реєстрації нового користувача та тестування форми створення облікового запису з валідацією полів (унікальність email, відповідність паролів).

**Реєстрація**  
Створити акаунт у CMMS системі

Повне ім'я  
Денис Козачок

Email адреса  
admin@cmms.local  
The email has already been taken.

Номер телефону  
admin@cmms.local

Пароль  
.....

Підтвердження пароля  
Повторіть пароль

[Зареєструватися](#)

Вже є акаунт? [Увійти](#)

Рис. 3.40 Введення email при реєстрації який вже використовується

**Реєстрація**  
Створити акаунт у CMMS системі

Повне ім'я  
Денис Козачок

Email адреса  
kdp@cmms.local

Номер телефону  
+380679969186

Пароль  
Мінімум 8 символів  
The password field must be at least 8 characters.

Підтвердження пароля  
Повторіть пароль

[Зареєструватися](#)

Вже є акаунт? [Увійти](#)

Рис. 3.41 Введення паролю по довжині менше за 8 символів

Тестування процесу автентифікації: перевірка входу з коректними та некоректними даними (неправильний пароль або email), а також перевірка роботи функції скидання пароля.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 84   |

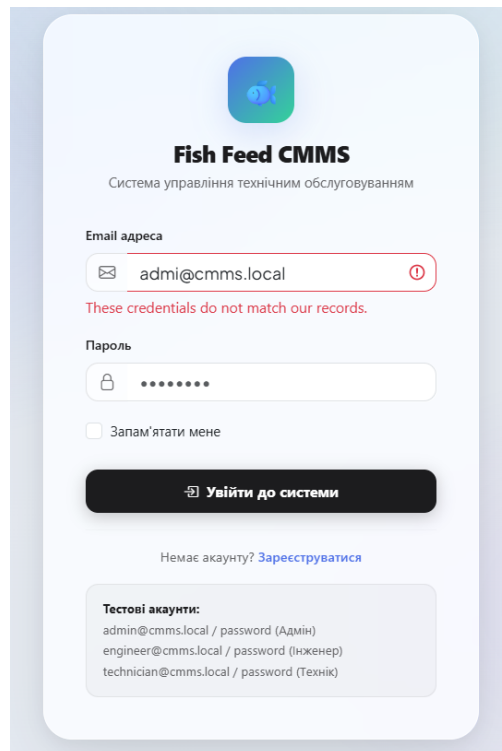


Рис. 3.42 форма авторизації та показ помилки при введенні  
неправильних даних

Перевірка розрахунку вартості оплати праці (labor\_cost) на основі фактично витраченого часу (actual\_hours), базової годинної ставки та коефіцієнтів пріоритету (low, medium, high, critical), що зберігаються в налаштуваннях системи.

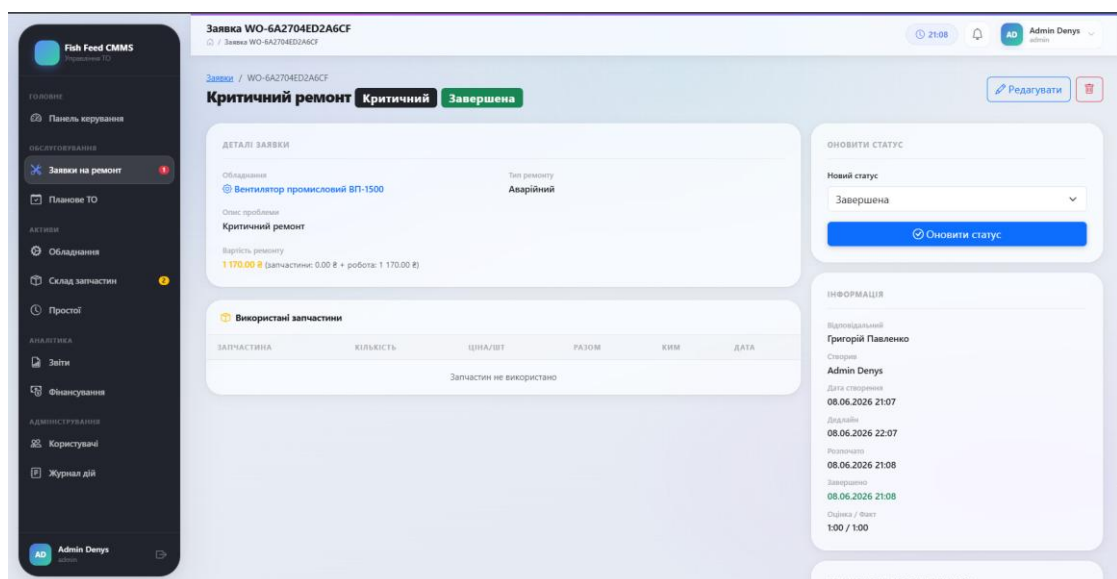


Рис. 3.43 Тестування розрахунку вартості ремонту для робіт різного  
пріоритету

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 85   |

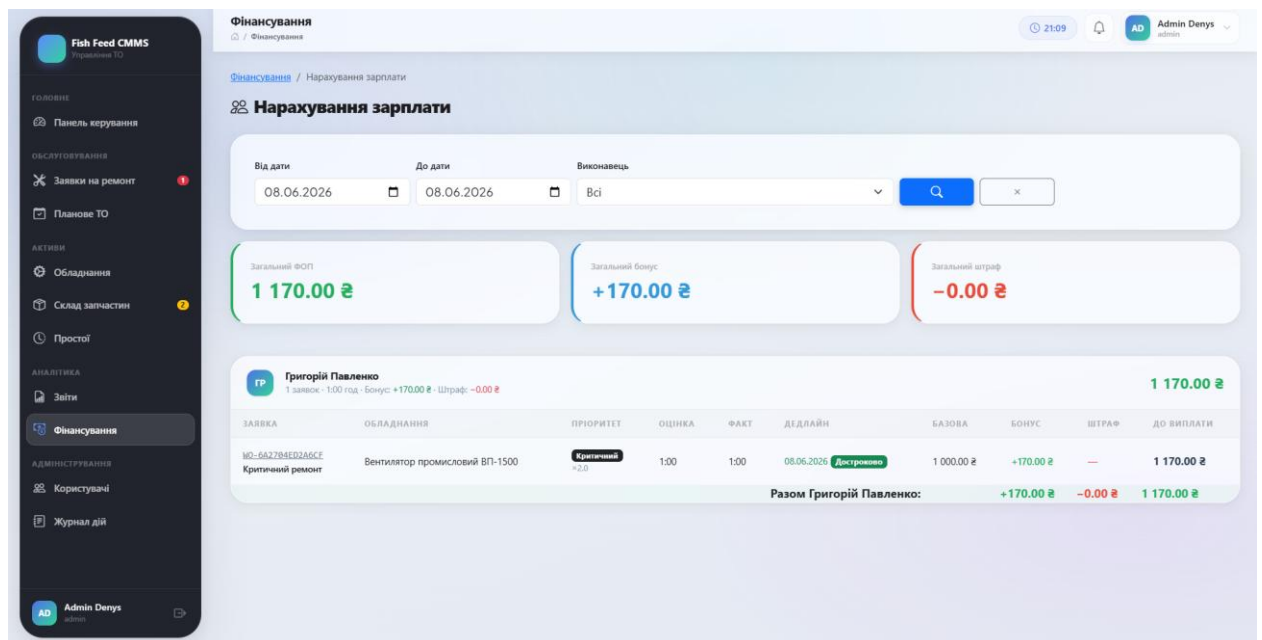


Рис. 3.44 Нарахування виплати за ремонт з врахуванням пріоритету

Тестування системи нарахування бонусів та штрафів: перевірка автоматичного додавання бонусу при достроковому виконанні наряду (completion\_notes до настання due\_date) та накладання штрафу у разі перевищення планового часу роботи (actual\_hours > estimated\_hours).

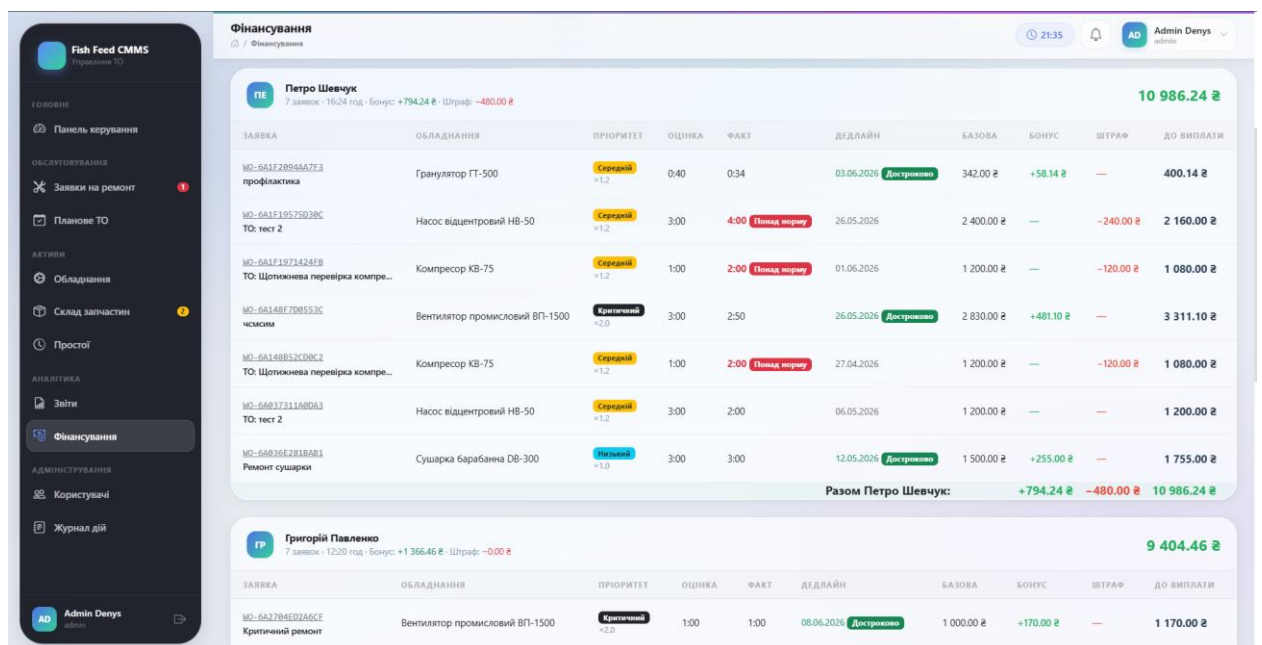


Рис. 3.45 Тестування нарахування штрафів та бонусів виконавцям

Перевірка підсумовування вартості витрачених матеріалів (materials\_cost) шляхом зв'язування таблиць використаних деталей (sparepartusage) та їхньої початкової ціни зі складу (sparepart).

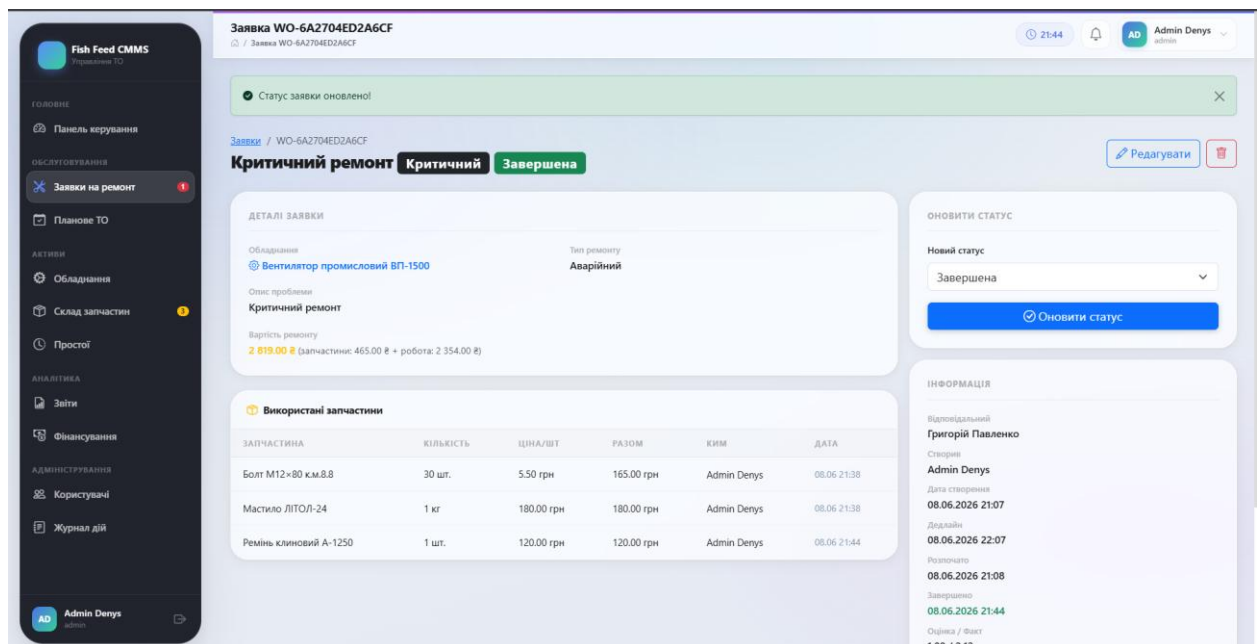


Рис. 3.46 перевірка обліку витрачених запчастин у наряді

Тестування інтеграції з сервісом амортизації (depreciationservice): перевірка, чи завершення ремонту (переведення наряду в статус completed) успішно тригерить перерахунок поточної вартості обладнання (current\_value) у базі даних (активи капіталізуються на суму проведеного ремонту).

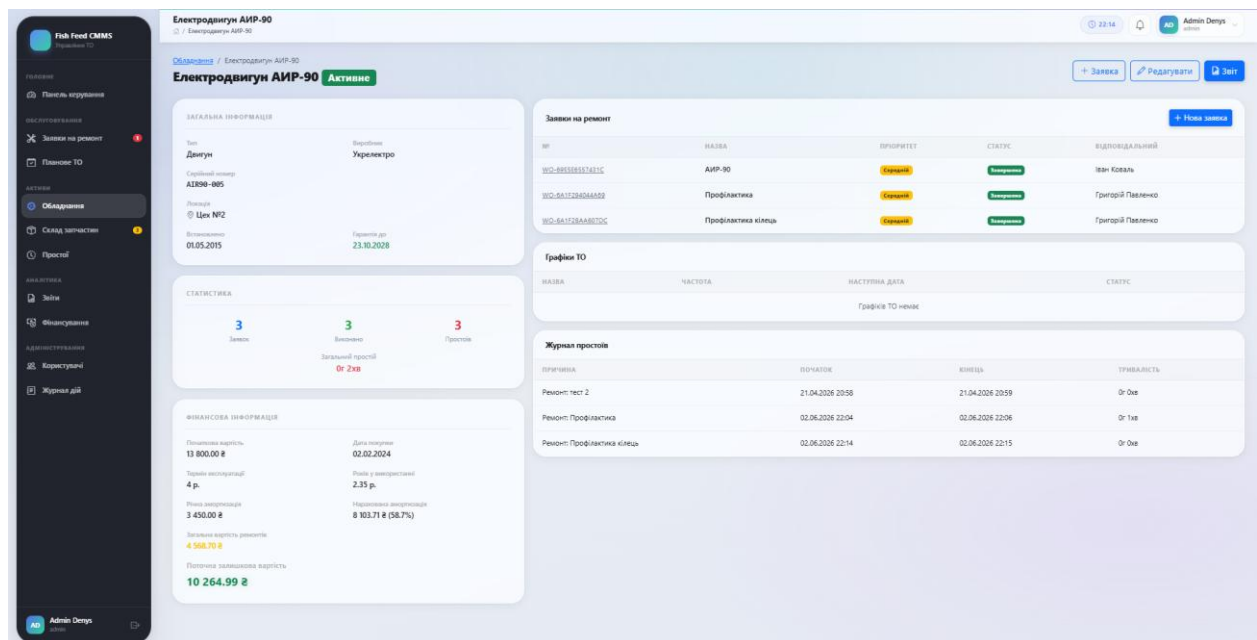


Рис. 3.47 відображення перерахованої вартості обладнання в системі

Тестування граничних значень та обробки виняткових ситуацій: перевірка поведінки коду при спробі списати запчастини у кількості, що перевищує наявну на складі.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          | 87   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

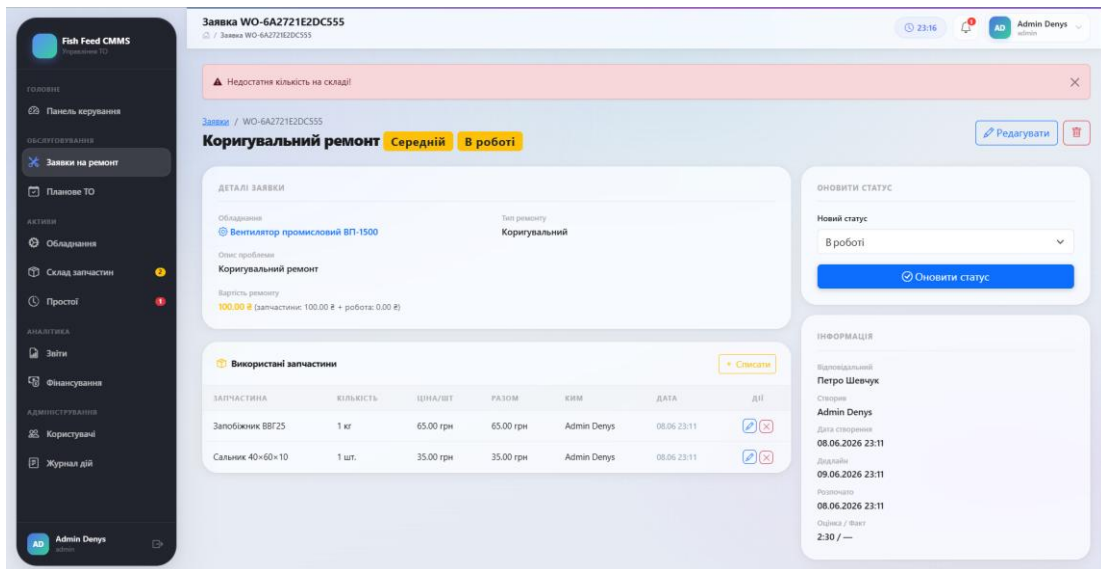


Рис. 3.48 повідомлення системи про недостатню кількість матеріалів на складі

Перед початком загального тестування програми було повністю підтверджено коректність розрахунків фінансового модуля. Переконавшись у точності математичної логіки бекенду, ми перейшли до тестування користувацького інтерфейсу та загальних сценаріїв взаємодії.

Тестування інтерфейсу та загальної функціональності включало:

Перевірку авторизації та розмежування прав доступу: перевірка коректності відображення елементів меню та блокування сторінок редагування графіків то для користувачів із роллю оператора.

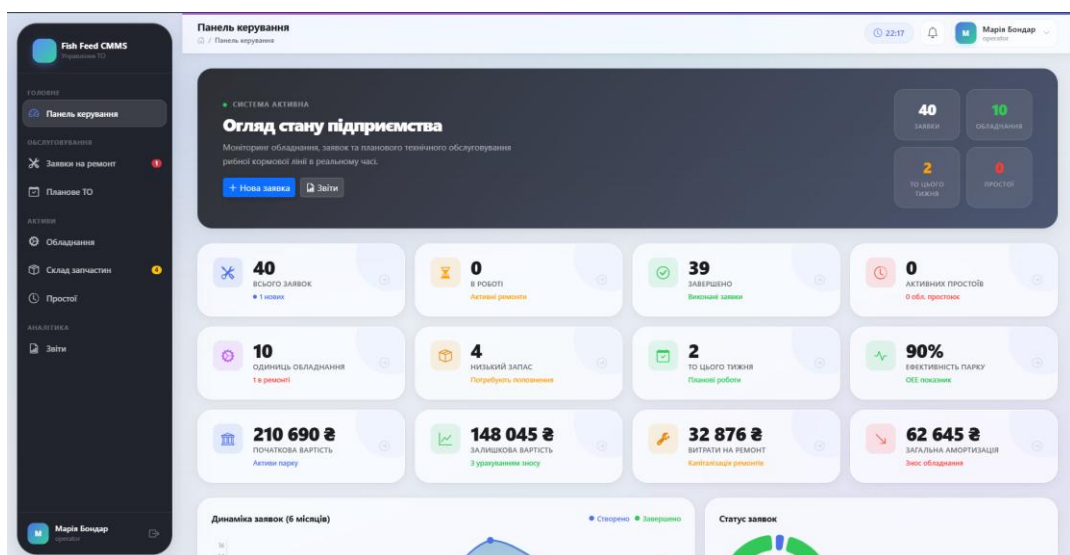


Рис. 3.49 відображення інтерфейсу системи для користувача з роллю оператора

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 88   |

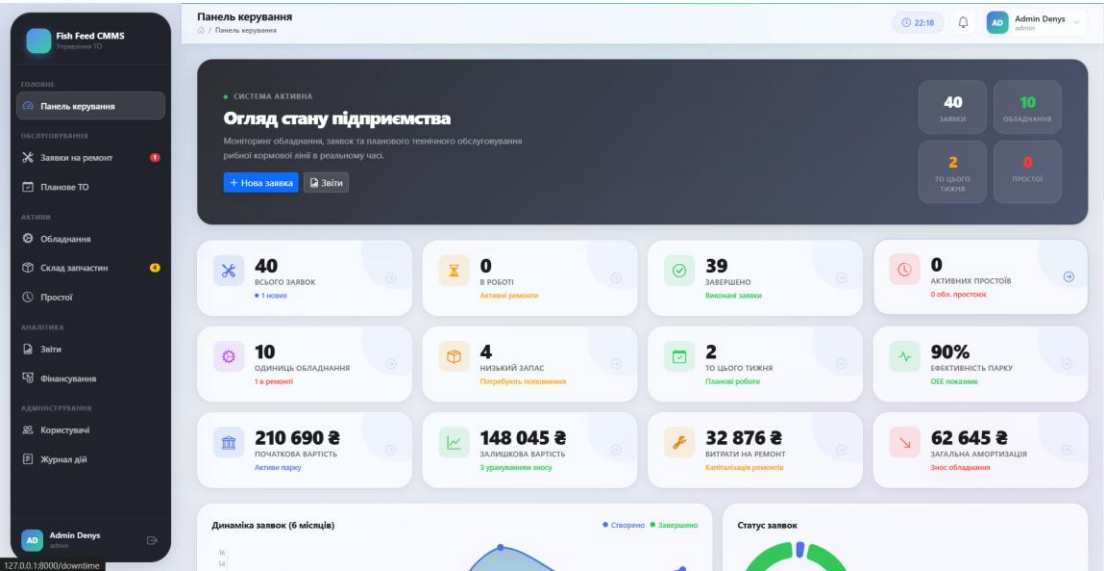


Рис. 3.50 відображення інтерфейсу системи для користувача з роллю адміністратор

Тестування адаптивності інтерфейсу: перевірка відображення таблиць обладнання, графіків то та черги заявок на екранах різного розширення.

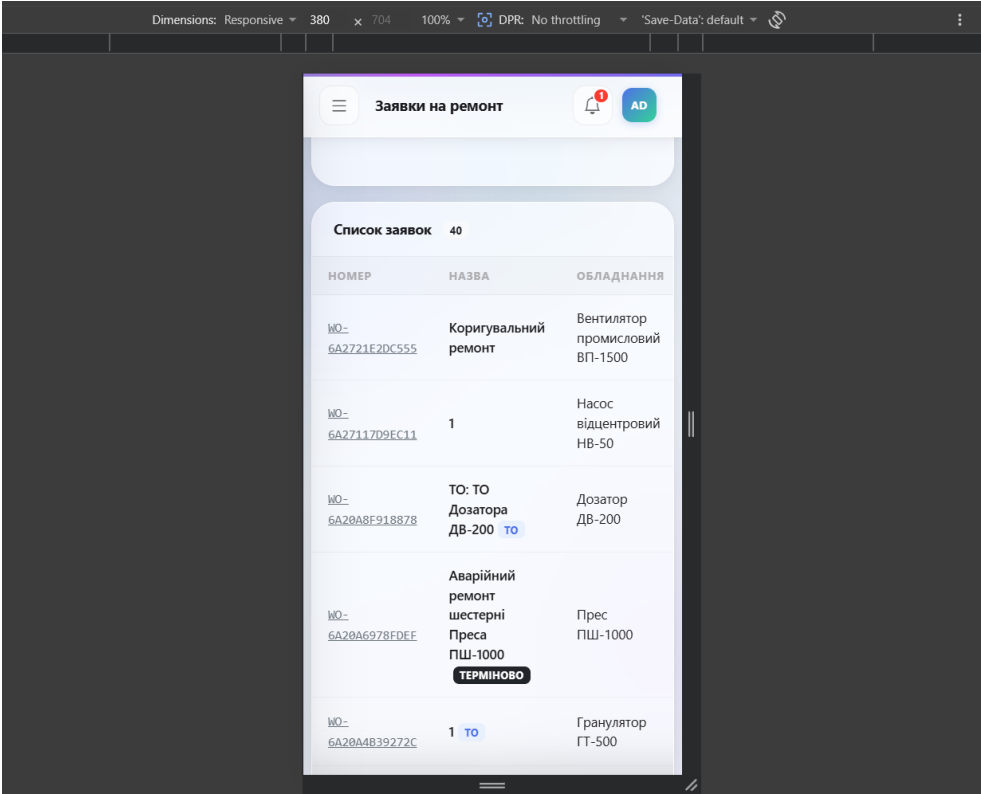


Рис. 3.51 Сторінка заявки на ремонт на екрані мобільного пристрою

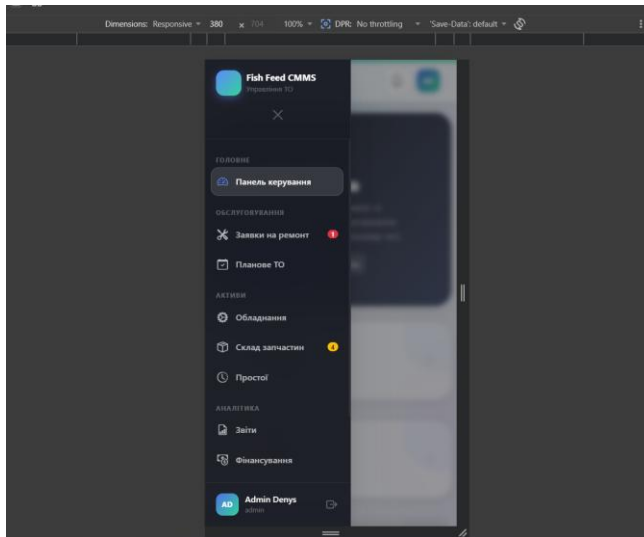


Рис. 3.52 Вигляд головної сторінки на екрані мобільного пристрою

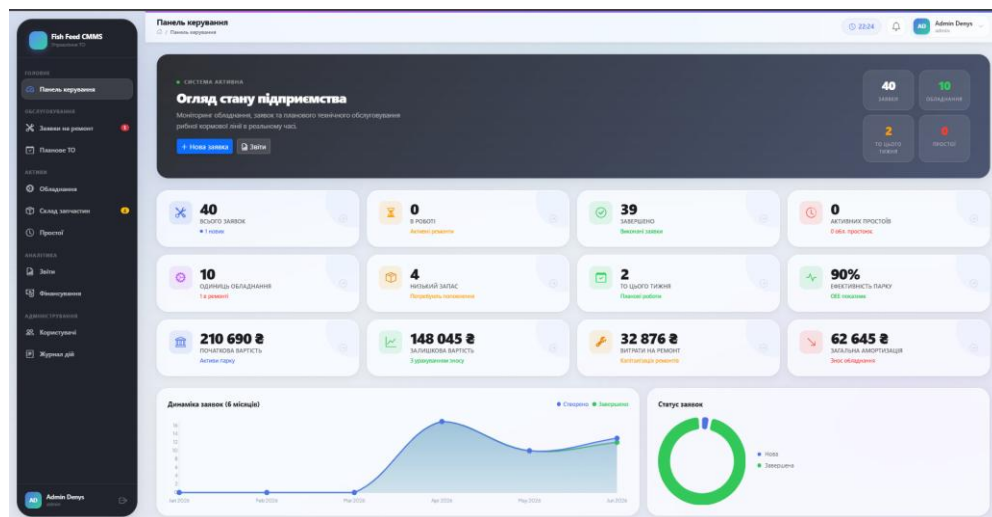


Рис. 3.53 Максимальний розмір головного вікна у повноекранному режимі

Перевірку системи сповіщень: контроль за тим, чи надходить email-повідомлення техніку при призначенні його на нове завдання.

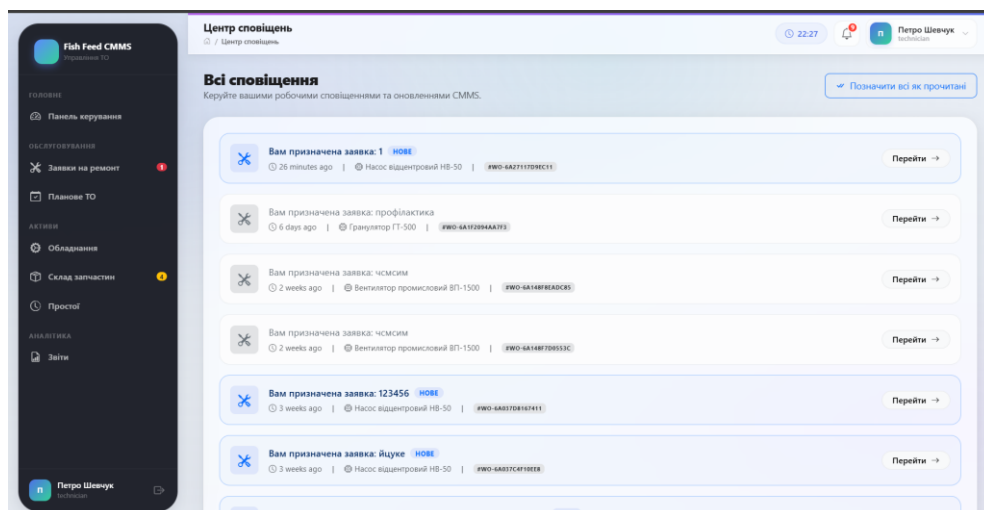


Рис. 3.54 Перегляд журналу сповіщень

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          | 90   |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

Проведені тести дозволили підтвердити, що система працює стабільно, коректно реагує на дії користувачів та забезпечує повну цілісність даних при виконанні бізнес-операцій.

### **Висновки до третього розділу**

У третьому розділі було детально розглянуто архітектуру реалізації, структуру користувацького інтерфейсу та особливості взаємодії користувачів із розробленою smtps-системою для контролю за технічним обслуговуванням обладнання.

Проаналізовано середовище розгортання та взаємодію між основними компонентами платформи: клієнтською частиною, сервером на laravel, базою даних mysql та docker-контейнерами. Описано ключові інтерфейсні модулі, зокрема інформаційну панель (dashboard) з інтерактивним календарем, картки обліку активів, інтерфейси управління нарядами на ремонт, обліку запчастин, фінансів та аналітичних звітів. Окрему увагу приділено алгоритмам автоматичного нарахування витрат на ремонти та інтегрованому сервісу depreciationservice для перерахунку амортизаційної вартості обладнання.

Описано процедуру автентифікації користувачів, розмежування прав на основі ролей (адміністратор, інженер, технік, оператор) та логування подій у журналі системного аудиту. Також у розділі наведено процес встановлення та конфігурування системи, виконано автоматизоване та ручне тестування ключових модулів. Результати тестування підтвердили стабільну роботу системи, коректне виконання бізнес-сценаріїв та готовність платформи до експлуатації.

Отже, реалізована веб-система повністю відповідає поставленим функціональним вимогам, має зручний адаптивний інтерфейс, поєднує технічний облік нарядів із фінансовим плануванням і аналітикою простоїв, забезпечуючи високу ефективність управління виробничими активами.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 91   |

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було успішно спроектовано, програмно реалізовано та протестовано веб-систему управління технічним обслуговуванням (CMMS) для підприємства з виробництва кормів для риб. Розроблене цифрове рішення повністю розв'язало завдання автоматизації процесів обліку активів, оптимізації роботи інженерних служб та зменшення аварійних простоїв конвеєрних ліній.

У ході проведення досліджень було детально вивчено специфіку експлуатації технологічного обладнання, виконано критичний порівняльний аналіз наявних методів ведення технічної документації та науково обґрунтовано доцільність впровадження спеціалізованого програмного продукту. На основі проведеного аналізу аналогів було сформовано вичерпний перелік функціональних та технічних вимог до системи.

На етапі проектування було розроблено детальну архітектурну модель застосунку, яка охоплює опис бізнес-процесів, реляційну структуру бази даних та схему взаємодії користувацьких інтерфейсів. Обраний технологічний стек на базі фреймворку Laravel 11 підтвердив свою відповідність вимогам масштабованості, безпеки даних та високої швидкодії в умовах промислового використання.

Створений програмний продукт успішно реалізує функції ведення детального реєстру обладнання з обліком його зносу, автоматичної генерації нарядів на роботу за графіками ППР, контролю виконання ремонтів у реальному часі та складського обліку деталей. Інтеграція аналітичного модуля для моніторингу часу простоїв та витрат надає керівництву підприємства дієвий інструмент для прийняття рішень щодо модернізації технічного парку.

Одержані результати роботи повністю підтверджують практичну ефективність та доцільність обраних архітектурних підходів і технологічних рішень. Створена система є готовим до впровадження інструментом цифровізації інженерної служби заводу та має потенціал для подальшої інтеграції в загальну екосистему автоматизації підприємства.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 92   |

## ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Документація Laravel 11 [Електронний ресурс] – Режим доступу до ресурсу: <https://laravel.com/docs/11.x>
2. Документація PHP 8.2 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/docs.php>
3. MySQL 8.0 Reference Manual [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.mysql.com/doc/refman/8.0/en/>
4. Документація Composer [Електронний ресурс] – Режим доступу до ресурсу: <https://getcomposer.org/doc/>
5. Spatie Laravel-Permission [Електронний ресурс] – Режим доступу до ресурсу: <https://spatie.be/docs/laravel-permission/v6/introduction>
6. Laravel Breeze [Електронний ресурс] – Режим доступу до ресурсу: <https://laravel.com/docs/11.x/starter-kits#laravel-breeze>
7. Документація Vite [Електронний ресурс] – Режим доступу до ресурсу: <https://vitejs.dev/guide/>
8. Документація Docker [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.docker.com/>
9. Документація Bootstrap 5 [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com/docs/5.3/>
10. Документація Chart.js [Електронний ресурс] – Режим доступу до ресурсу: <https://www.chartjs.org/docs/latest/>
11. FullCalendar - JavaScript Event Calendar [Електронний ресурс] – Режим доступу до ресурсу: <https://fullcalendar.io/docs>
12. Barryvdh Laravel DomPDF [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/barryvdh/laravel-dompdf>
13. Документація PHPUnit [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.phpunit.de/en/11.0/>
14. Fiix CMMS Platform [Електронний ресурс] – Режим доступу до ресурсу: <https://www.fiixsoftware.com>

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 93   |

15. UpKeep Maintenance Management [Електронний ресурс] – Режим доступу до ресурсу: <https://www.onupkeep.com>
16. eMaint CMMS by Fluke Reliability [Електронний ресурс] – Режим доступу до ресурсу: <https://www.emaaint.com>
17. What is CMMS? A Guide to Computerized Maintenance Management Systems [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ibm.com/topics/cmms>
18. Документація Eloquent ORM [Електронний ресурс] – Режим доступу до ресурсу: <https://laravel.com/docs/11.x/eloquent>
19. Laravel Queues [Електронний ресурс] – Режим доступу до ресурсу: <https://laravel.com/docs/11.x/queues>
20. Документація Nginx [Електронний ресурс] – Режим доступу до ресурсу: <https://nginx.org/en/docs/>
21. Документація Git [Електронний ресурс] – Режим доступу до ресурсу: <https://git-scm.com/doc>
22. Документація JavaScript ES6+ [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
23. Role-Based Access Control (RBAC) [Електронний ресурс] – Режим доступу до ресурсу: <https://www.redhat.com/en/topics/security/what-is-role-based-access-control>
24. Predictive Maintenance and the Smart Factory [Електронний ресурс] – Режим доступу до ресурсу: <https://www.deloitte.com/us/en/services/consulting/services/predictive-maintenance-and-the-smart-factory.html>
25. Computerized Maintenance Management System (CMMS) Market Size, Share & Growth Report [Електронний ресурс] – Режим доступу до ресурсу: <https://www.zionmarketresearch.com/report/computerized-maintenance-management-system-market>
26. OEE (Overall Equipment Effectiveness) - Industry Standard for Measuring Manufacturing Productivity [Електронний ресурс] – Режим доступу до ресурсу: <https://www.oee.com/>

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 94   |

27. The Top Trends in Tech 2024: Enterprise Technology Adoption [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mckinsey.com/capabilities/tech-and-ai/our-insights/the-top-trends-in-tech-2024>
28. Straight-Line Depreciation Method [Електронний ресурс] – Режим доступу до ресурсу: <https://corporatefinanceinstitute.com/resources/accounting/straight-line-depreciation/>
29. Документація Blade Templates [Електронний ресурс] – Режим доступу до ресурсу: <https://laravel.com/docs/11.x/blade>
30. Тест-кейс - один із ключових інструментів тестувальників ПЗ [Електронний ресурс] – Режим доступу до ресурсу: <https://foxminded.ua/test-keis/>

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 95   |

# ДОДАТКИ

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 96   |

**Лістининг модель WorkOrder**

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Relations\HasMany;
class WorkOrder extends Model
{
    use SoftDeletes;
    protected $fillable = [
        'order_number', 'title', 'description', 'equipment_id',
        'created_by', 'assigned_to', 'maintenance_schedule_id',
        'priority', 'status', 'type', 'rejection_reason',
        'completion_notes', 'started_at', 'completed_at',
        'due_date', 'estimated_hours', 'actual_hours', 'repair_cost',
    ];
    protected static function boot(): void
    {
        parent::boot();
        // Автогенерація унікального номера наряду
        static::creating(function ($model) {
            if (!$model->order_number) {
                $model->order_number = 'WO-' . strtoupper(uniqid());
            }
        });
        // Автоматичний перерахунок вартості при збереженні
        static::saved(function ($workOrder) {

```

|      |      |               |        |      |                           |      |
|------|------|---------------|--------|------|---------------------------|------|
|      |      | Козачок Д. П. |        |      | ІІЗ.КР.Б – 121 – 26 – ІІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                           |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                           | 97   |

```

        $workOrder->updateRepairCost();
    });
}
// Розрахунок вартості праці з урахуванням коефіцієнтів
public function getLaborCostAttribute(): float
{
    $baseRate          = Setting::get('finance_base_rate', 200);
    $earlyBonusPercent  = Setting::get('finance_early_bonus', 10) / 100;
    $overtimePenaltyPercent = Setting::get('finance_overtime_penalty', 20) / 100;
    $multipliers = [
        'low'    => Setting::get('finance_priority_low', 1.0),
        'medium' => Setting::get('finance_priority_medium', 1.2),
        'high'   => Setting::get('finance_priority_high', 1.5),
        'critical' => Setting::get('finance_priority_critical', 2.0),
    ];
    $multiplier = $multipliers[$this->priority] ?? 1.0;
    $basePay    = round(((float) $this->actual_hours) * $baseRate * $multiplier,
2);
    // Бонус за дострокове виконання
    $bonus = 0;
    if ($this->due_date && $this->completed_at && $this->completed_at <
$this->due_date) {
        $bonus = round($basePay * $earlyBonusPercent, 2);
    }
    // Штраф за перевищення планового часу
    $penalty = 0;
    if ($this->estimated_hours && $this->actual_hours > $this->estimated_hours)
    {
        $penalty = round($basePay * $overtimePenaltyPercent, 2);
    }
}

```

|      |      |               |        |      |                           |      |
|------|------|---------------|--------|------|---------------------------|------|
|      |      | Козачок Д. П. |        |      | ІІЗ.КР.Б – 121 – 26 – ІІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                           |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                           | 98   |

```

        return max(0.0, round($basePay + $bonus - $penalty, 2));
    }

    // Розрахунок вартості витрачених матеріалів
    public function getMaterialsCostAttribute(): float
    {
        return (float) $this->sparePartUsages()
            ->with('sparePart')
            ->get()
            ->sum(function ($usage) {
                return ($usage->sparePart->unit_price ?? 0) * $usage->quantity_used;
            });
    }

    // Збереження підсумкової вартості та перерахунок амортизації
    public function updateRepairCost(): void
    {
        $totalCost = round($this->labor_cost + $this->materials_cost, 2);
        $this->updateQuietly(['repair_cost' => $totalCost]);
        if ($this->equipment) {
            app(\App\Services\DepreciationService::class)
                ->recalculateCurrentValue($this->equipment);
        }
    }
}

```

## Лістининг контролера **WorkOrderController**

```

php

public function updateStatus(Request $request, WorkOrder $workOrder):
    RedirectResponse
    {

```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 99   |

```

$user = auth()->user();
// Перевірка прав: лише адмін, інженер або призначений виконавець
if (!$user->hasRole(['admin', 'engineer'])) {
    if ($workOrder->assigned_to !== $user->id) {
        abort(403, 'Тільки призначений відповідальний може змінювати
статус.');
```

```

    }
}

$validated = $request->validate([
    'status'      => 'required|in:new,in_progress,completed,rejected,on_hold',
    'rejection_reason' => 'required_if:status,rejected|nullable|string',
    'completion_notes' => 'nullable|string',
    'actual_hours'  => 'nullable|numeric|min:0.01',
]);

$old = ['status' => $workOrder->status, 'repair_cost' => $workOrder-
>repair_cost];

// Фіксація часу початку роботи
if ($validated['status'] === 'in_progress' && !$workOrder->started_at) {
    $validated['started_at'] = now();
}

if (in_array($validated['status'], ['completed', 'rejected'])) {
    if ($validated['status'] === 'completed') {
        $validated['completed_at'] = now();
    }
}

// Закриття активних простоїв
DowntimeLog::where('work_order_id', $workOrder->id)
    ->whereNull('ended_at')
    ->update(['ended_at' => now()]);

// Прив'язка до графіку ТО та оновлення наступної дати
if (!$workOrder->maintenance_schedule_id) {
```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 100  |

```

$linkedSchedule = MaintenanceSchedule::where('equipment_id',
$workOrder->equipment_id)
    ->where('is_active', true)
    ->orderBy('next_due_date', 'asc')
    ->first();
if ($linkedSchedule) {
    $validated['maintenance_schedule_id'] = $linkedSchedule->id;
    $workOrder->setRelation('maintenanceSchedule', $linkedSchedule);
}
}

// Оновлення дат у графіку ТО
if ($workOrder->maintenance_schedule_id) {
    $schedule = $workOrder->maintenanceSchedule;
    if ($schedule) {
        $schedule->update([
            'last_completed_date' => now(),
            'next_due_date'      => now()->addDays(
                match ($schedule->frequency) {
                    'daily'      => 1, 'weekly'      => 7,
                    'biweekly'   => 14, 'monthly'     => 30,
                    'quarterly'  => 90, 'semi_annual' => 180,
                    'annual'     => 365, default      => 30,
                }
            ),
        ]);
    }
}
}

$workOrder->update($validated);

```

|      |      |               |        |      |                         |      |
|------|------|---------------|--------|------|-------------------------|------|
|      |      | Козачок Д. П. |        |      | ПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                         |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                         | 101  |

```

        AuditLog::record('Змінено статус заявки: ' . $validated['status'], $workOrder,
        $old, $validated);
        return back()->with('success', 'Статус заявки оновлено!');
    }

```

### Лістинг спостерігача **WorkOrderObserver**

```

php
<?php
namespace App\Observers;
use App\Models\WorkOrder;
use App\Models\Equipment;
use App\Models\DowntimeLog;
class WorkOrderObserver
{
    public function created(WorkOrder $workOrder): void
    {
        $this->syncEquipmentStatus($workOrder->equipment);
    }
    public function updated(WorkOrder $workOrder): void
    {
        $this->syncEquipmentStatus($workOrder->equipment);
        if ($workOrder->isDirty('status')) {
            $this->handleDowntime($workOrder);
        }
    }
}
// Синхронізація статусу обладнання на основі останнього наряду
protected function syncEquipmentStatus(?Equipment $equipment): void
{
    if (!$equipment || $equipment->status === 'decommissioned') return;

```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 102  |

```

$latestWO = $equipment->workOrders()
    ->latest('created_at')->orderBy('id', 'desc')->first();
if (!$latestWO) {
    $equipment->update(['status' => 'active']);
    return;
}
$newStatus = match ($latestWO->status) {
    'new', 'in_progress', 'on_hold' => 'in_repair',
    'completed', 'rejected'      => 'active',
    default                      => 'active',
};
if ($equipment->status !== $newStatus) {
    $equipment->update(['status' => $newStatus]);
}
}

// Автоматичне управління записами простоїв
protected function handleDowntime(WorkOrder $workOrder): void
{
    if ($workOrder->status === 'in_progress') {
        $exists = DowntimeLog::where('work_order_id', $workOrder->id)
            ->whereNull('ended_at')->exists();
        if (!$exists) {
            DowntimeLog::create([
                'equipment_id' => $workOrder->equipment_id,
                'work_order_id' => $workOrder->id,
                'reported_by' => auth()->id() ?? $workOrder->assigned_to,
                'reason'      => 'Ремонт: ' . $workOrder->title,
                'started_at'  => now(),
            ]);
        }
    }
}

```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ПІЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 103  |

```

    } elseif (in_array($workOrder->status, ['completed', 'rejected'])) {
        DowntimeLog::where('work_order_id', $workOrder->id)
            ->whereNull('ended_at')
            ->update(['ended_at' => now()]);
    }
}
}
}

```

### Лістинг сервісу DepreciationService

```

php
<?php
namespace App\Services;
use App\Models\Equipment;
class DepreciationService
{
    public function calcDepreciation(Equipment $equipment): array
    {
        if (!$equipment->purchase_date || !$equipment->purchase_price
            || !$equipment->useful_life_years || $equipment->useful_life_years <= 0) {
            return ['has_data' => false, ...];
        }
        $purchasePrice = (float) $equipment->purchase_price;
        $usefulLife     = (int) $equipment->useful_life_years;
        $yearsInUse     = round($equipment->purchase_date->diffInDays(now()) /
365.25, 4);
        // Річна амортизація (прямолінійний метод)
        $annualDepreciation = $purchasePrice / $usefulLife;
        $accumulatedDepreciation = min($purchasePrice, round($annualDepreciation
* $yearsInUse, 2));
    }
}

```

|      |      |               |        |      |                           |      |
|------|------|---------------|--------|------|---------------------------|------|
|      |      | Козачок Д. П. |        |      | ІІЗ.КР.Б – 121 – 26 – ІІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                           |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                           | 104  |

```

        $baseCurrentValue      = max(0, $purchasePrice -
$accumulatedDepreciation);

        // Сума витрат на ремонти з завершених нарядів
        $totalRepairCost = (float) $equipment->workOrders()
            ->where('status', 'completed')
            ->whereNotNull('repair_cost')
            ->sum('repair_cost');

        // Поточна вартість = залишкова + ремонти (капіталізація витрат)
        $currentValue = max(0, round($baseCurrentValue + $totalRepairCost, 2));

        return [
            'has_data'          => true,
            'annual_depreciation' => round($annualDepreciation, 2),
            'accumulated_depreciation' => $accumulatedDepreciation,
            'total_repair_cost'   => round($totalRepairCost, 2),
            'current_value'      => $currentValue,
            'depreciation_percent' => $purchasePrice > 0
                ? round(($accumulatedDepreciation / $purchasePrice) * 100, 1) : 0,
        ];
    }

    public function recalculateCurrentValue(Equipment $equipment): void
    {
        $info = $this->calcDepreciation($equipment);
        $equipment->updateQuietly(['current_value' => $info['current_value']]);
    }
}

```

## Лістининг контролера SparePartController

php

```

public function writeOff(Request $request): RedirectResponse

```

|      |      |               |        |      |                           |      |
|------|------|---------------|--------|------|---------------------------|------|
|      |      | Козачок Д. П. |        |      | ІІЗ.КР.Б – 121 – 26 – ІІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                           |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                           | 105  |

```

{
    $validated = $request->validate([
        'work_order_id' => 'required|exists:work_orders,id',
        'spare_part_id' => 'required|exists:spare_parts,id',
        'quantity_used' => 'required|integer|min:1',
        'notes'          => 'nullable|string',
    ]);
    $part = SparePart::findOrFail($validated['spare_part_id']);
    // Перевірка складського залишку
    if ($part->quantity < $validated['quantity_used']) {
        return back()->with('error', 'Недостатня кількість на складі!');
    }
    // Зменшення залишку та реєстрація факту списання
    $part->decrement('quantity', $validated['quantity_used']);
    SparePartUsage::create([
        ...$validated,
        'used_by' => auth()->id(),
    ]);
    AuditLog::record(
        'Списано запчастину: ' . $part->name .
        ' (' . $validated['quantity_used'] . ' ' . $part->unit . ')'
    );
    return back()->with('success', 'Запчастину списано!');
}

```

## Лістининг **GenerateMaintenanceTasks**

```

php
<?php
namespace App\Console\Commands;

```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 106  |

```

use App\Models\MaintenanceSchedule;
use App\Models\WorkOrder;
use App\Models\AuditLog;
use Illuminate\Console\Command;

class GenerateMaintenanceTasks extends Command
{
    protected $signature = 'cmms:generate-maintenance';
    protected $description = 'Auto-generate work orders for due maintenance
schedules';

    public function handle(): void
    {
        $systemUser = \App\Models\User::role('admin')->first();
        $systemUserId = $systemUser?->id ?? 1;
        // Відбір активних графіків, термін яких настав
        $due = MaintenanceSchedule::where('is_active', true)
            ->where('next_due_date', '<=', now()->addDay())
            ->with(['equipment', 'assignedTo'])
            ->get();
        $count = 0;
        foreach ($due as $schedule) {
            // Уникнення дублювання активних нарядів
            $exists = WorkOrder::where('maintenance_schedule_id', $schedule->id)
                ->whereNotIn('status', ['completed', 'rejected'])
                ->where('type', 'preventive')
                ->exists();
            if ($exists) continue;
            $workOrder = WorkOrder::create([
                'title'          => 'ТО: ' . $schedule->title,
                'description'     => $schedule->description
                ?? 'Автоматично згенероване планове ТО.',
            ]);
        }
    }
}

```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 107  |

```

        'equipment_id'      => $schedule->equipment_id,
        'created_by'        => $systemUserId,
        'assigned_to'       => $schedule->assigned_to,
        'maintenance_schedule_id' => $schedule->id,
        'priority'          => 'medium',
        'status'            => 'new',
        'type'              => 'preventive',
        'estimated_hours'    => $schedule->estimated_hours,
        'due_date'          => $schedule->next_due_date,
    ];

    AuditLog::record('Системна генерація заявки по ТО', $workOrder, [],
    $workOrder->toArray());

    if ($schedule->assignedTo) {
        $schedule->assignedTo->notify(
            new \App\Notifications\MaintenanceDue($schedule)
        );
    }

    $count++;
}

$this->info("Завершено. Згенеровано {$count} заявок.");
}
}

```

## Лістининг моделі AuditLog

```

php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

use Illuminate\Database\Eloquent\Relations\BelongsTo;

```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 108  |

```

class AuditLog extends Model
{
    protected $fillable = [
        'user_id', 'action', 'model_type', 'model_id',
        'old_values', 'new_values', 'ip_address', 'user_agent',
    ];
    protected $casts = [
        'old_values' => 'array',
        'new_values' => 'array',
    ];
    public function user(): BelongsTo
    {
        return $this->belongsTo(User::class);
    }
    // Статичний метод для запису будь-якої події в журнал аудиту
    public static function record(
        string $action,
        $model = null,
        array $old = [],
        array $new = []
    ): void {
        static::create([
            'user_id' => auth()->id(),
            'action' => $action,
            'model_type' => $model ? get_class($model) : null,
            'model_id' => $model?->id,
            'old_values' => $old ?: null,
            'new_values' => $new ?: null,
            'ip_address' => request()->ip(),
            'user_agent' => request()->userAgent(),

```

|      |      |               |        |      |                           |      |
|------|------|---------------|--------|------|---------------------------|------|
|      |      | Козачок Д. П. |        |      | ІІЗ.КР.Б – 121 – 26 – ІІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                           |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                           | 109  |

```

        });
    }
}

```

## Лістининг RoleAndPermissionSeeder

```

php
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

use Spatie\Permission\Models\Role;

use Spatie\Permission\Models\Permission;

class RoleAndPermissionSeeder extends Seeder
{
    public function run(): void
    {
        // Скидання кешу дозволів

        app()[\Spatie\Permission\PermissionRegistrar::class]-
        >forgetCachedPermissions();

        // Створення дозволів

        $permissions = [

            'manage users',

            'manage equipment',

            'create work orders',

            'assign work orders',

            'complete work orders',

```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 110  |

```

        'view reports',

        'manage inventory',

        'write off spare parts'

    ];

    foreach ($permissions as $permission) {

        Permission::findOrCreate($permission);

    }

    // Створення ролей та призначення дозволів

    Role::findOrCreate('admin')

        ->givePermissionTo(Permission::all());

    Role::findOrCreate('engineer')

        ->givePermissionTo([

            'manage equipment',

            'create work orders',

            'assign work orders',

            'view reports',

            'manage inventory'

        ]);

    Role::findOrCreate('technician')

        ->givePermissionTo([

            'complete work orders',

            'write off spare parts'

        ]);

    }

}

```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 111  |

## Лістининг web

php

<?php

```
use Illuminate\Support\Facades\Route;
```

```
use App\Http\Controllers\DashboardController;
```

```
use App\Http\Controllers\EquipmentController;
```

```
use App\Http\Controllers\WorkOrderController;
```

```
use App\Http\Controllers\SparePartController;
```

```
use App\Http\Controllers\ReportController;
```

```
Route::get('/', function () {
```

```
    return redirect()->route('login');
```

```
});
```

```
Route::middleware(['auth', 'verified'])->group(function () {
```

```
    Route::get('/dashboard', [DashboardController::class, 'index'])-  
>name('dashboard');
```

```
    // Доступ для всіх авторизованих користувачів
```

```
    Route::resource('work-orders', WorkOrderController::class);
```

```
    // Доступ лише для Адміністраторів та Інженерів
```

```
    Route::middleware(['role:admin|engineer'])->group(function () {
```

```
        Route::resource('equipment', EquipmentController::class);
```

```
        Route::get('/reports/downtime', [ReportController::class, 'downtimeReport'])-  
>name('reports.downtime');
```

```
        Route::get('/reports/downtime/pdf', [ReportController::class,  
'exportDowntimePdf'])->name('reports.downtime.pdf');
```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 112  |

```

});

// Списання запчастин (доступно і для техніків під час ремонту)

Route::post('/spare-parts/{sparePart}/write-off', [SparePartController::class,
'writeOff'])

->name('spare-parts.write-off')

->middleware('permission:write off spare parts');

});

require __DIR__.'/auth.php';

```

## Лістинг CreateWorkOrdersTable

```

php

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    public function up(): void
    {
        Schema::create('work_orders', function (Blueprint $table) {
            $table->id();

            $table->foreignId('equipment_id')->constrained()->onDelete('cascade');

            $table->foreignId('creator_id')->constrained('users');

            $table->foreignId('assigned_to')->nullable()->constrained('users');

```

|      |      |               |        |      |                           |      |
|------|------|---------------|--------|------|---------------------------|------|
|      |      | Козачок Д. П. |        |      | ІІЗ.КР.Б – 121 – 26 – ІІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                           |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                           | 113  |

```

$table->string('subject');

$table->text('description');


$table->enum('priority', ['low', 'medium', 'high', 'critical'])-
>default('medium');

$table->enum('status', ['open', 'in_progress', 'completed', 'cancelled'])-
>default('open');


$table->timestamp('started_at')->nullable();

$table->timestamp('completed_at')->nullable();


$table->decimal('labor_cost', 10, 2)->default(0.00);

$table->decimal('material_cost', 10, 2)->default(0.00);

$table->decimal('total_cost', 10, 2)->storedAs('labor_cost + material_cost');


$table->timestamps();

$table->softDeletes();

});

}

public function down(): void
{
    Schema::dropIfExists('work_orders');
}

};

```

**Лістинг GenerateMaintenanceTasks**

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          | 114  |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          |      |

```

php
<?php

namespace App\Console\Commands;

use Illuminate\Console\Command;

use App\Models\MaintenanceSchedule;

use App\Models\WorkOrder;

use Carbon\Carbon;

class GenerateMaintenanceTasks extends Command
{
    protected $signature = 'maintenance:generate-tasks';

    protected $description = 'Generate preventive maintenance work orders based on
schedules';

    public function handle(): int
    {
        $today = Carbon::today();

        $schedules = MaintenanceSchedule::where('next_run_date', '<=', $today)
            ->where('is_active', true)
            ->get();

        foreach ($schedules as $schedule) {
            // Створення заявки
            WorkOrder::create([
                'equipment_id' => $schedule->equipment_id,
                'creator_id' => 1, // Системний користувач / Admin
                'subject' => "Регламентне ТО: " . $schedule->title,
                'description' => $schedule->description,
            ]);
        }
    }
}

```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 115  |

```

        'priority' => $schedule->priority,
        'status' => 'open',
    ]);
    // Розрахунок наступної дати запуску
    $schedule->update([
        'last_run_date' => $today,
        'next_run_date' => $schedule->calculateNextRunDate($today),
    ]);
}

$this->info('Preventive maintenance work orders generated successfully.');
```

return Command::SUCCESS;

```

}

}

```

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 116  |

## Глосарій

**Веб-система CMMS** – програмний комплекс для автоматизації технічного обслуговування заводу, що забезпечує облік обладнання, планування ремонтів та управління складськими запасами запчастин.

**Виробниче обладнання (Equipment)** – агрегати та лінії з виробництва кормів, що є основними активами системи та потребують регулярного моніторингу і сервісу.

**Заявка на ремонт (Work Order)** – основний документ системи, що містить опис технічного завдання, призначеного виконавця, перелік потрібних запчастин та терміни виконання робіт.

**ПЗР (Планово-запобіжні роботи)** – комплекс регламентних технічних заходів, що проводяться за заздалегідь визначеним графіком для попередження поломок та виходу обладнання з ладу.

**Графік ТО (Maintenance Schedule)** – системний календарний план, що автоматично розраховує дати майбутніх сервісних робіт на основі заданої періодичності (щодня, щомісяця тощо).

**Час простою (Downtime)** – період часу, протягом якого обладнання не функціонує через поломку або планове обслуговування. Фіксується в системі для подальшого аналізу ефективності.

**Запасні частини (Spare Parts)** – вузли, деталі та витратні матеріали, необхідні для проведення ремонтних робіт, облік яких ведеться у модулі складу.

**Склад запчастин (Inventory)** – розділ системи для контролю залишків, списання матеріалів під конкретні заявки та моніторингу критичного рівня запасів.

**MTBF (Mean Time Between Failures)** – середній час між відмовами обладнання; показник надійності.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 117  |

**MTTR (Mean Time To Repair)** – середній час на виконання одного ремонту.

**Оператор (Operator)** – роль користувача, що має право повідомляти про поломки.

**Технік (Technician)** – виконавець ремонтних робіт.

**Інженер (Engineer)** – роль із правами створення графіків ТО та аналізу технічних звітів.

**Адміністратор (Admin)** – користувач із повним доступом до налаштувань системи.

**Журнал аудиту (Audit Log)** – системна таблиця, що автоматично фіксує всі критичні зміни.

**Критичність обладнання** – категорія пріоритетності агрегату для виробничого процесу.

**Тип обслуговування** – класифікація робіт у системі (аварійне, планове, модернізація).

**Діагностика** – процес виявлення причин несправності обладнання.

**Звітність (Reporting)** – програмний інструмент для генерації аналітичних документів у форматі PDF.

**Посередник (Middleware)** – шар програмного коду для перевірки прав доступу.

**Список перегляду (Watchlist)** - персоналізований перелік медіаконтенту, сформований користувачем із фільмів та серіалів, які він планує переглянути в майбутньому. Цей функціонал дозволяє швидко отримувати доступ до обраних творів.

**Жанр (Genre)** – атрибут класифікації кінематографічних творів, що об'єднує їх за спільними художніми, сюжетними або стилістичними ознаками (наприклад, драма, комедія, наукова фантастика), який використовується для фільтрації контенту в каталозі.

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ПЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 118  |

**Фільтрація** - процес відбору медіаконтенту з загальної бази даних за заданими критеріями, такими як рік випуску, жанр, країна виробництва або рейтинг, для звуження результатів пошуку.

**Рекомендаційна система** - програмний модуль, який на основі аналізу попередніх оцінок, переглядів та списків користувача автоматично формує пропозиції схожого контенту, який може його зацікавити.

**Фільмографія** - повний хронологічний список кінофільмів та серіалів, у створенні яких брала участь конкретна персона (актор, режисер, сценарист), що відображається на сторінці детальної інформації про цю особу.

**Особистий кабінет (User Profile)** - захищена частина веб-додатку, доступна лише авторизованому користувачу, де він може керувати своїми налаштуваннями, змінювати персональні дані, переглядати історію активності та власні колекції.

**Зовнішній API (External API)** - програмний інтерфейс сторонніх сервісів (наприклад, TMDb або IMDb), який система Kovix використовує для автоматичного отримання, оновлення та синхронізації метаданих про фільми, постерів та інформації про акторів.

**Статус перегляду (Watch Status)** - мітка, яку користувач присвоює конкретному фільму чи серіалу для відстеження прогресу (наприклад, «Планую подивитися», «Дивлюся», «Переглянуто», «Покинуто»).

|      |      |               |        |      |                          |      |
|------|------|---------------|--------|------|--------------------------|------|
|      |      | Козачок Д. П. |        |      | ІПЗ.КР.Б – 121 – 26 – ІЗ | Арк. |
|      |      | Вакалюк Т.А   |        |      |                          |      |
| Змн. | Арк. | № докум.      | Підпис | Дата |                          | 119  |