

SCADA System

Посібник користувача

Версія 1.0 — Квітень 2026

Зміст

Передмова

Швидкий старт — 15 хвилин

1. Авторизація та реєстрація

2. Головна панель (Dashboard)

3. Мнемосхеми

4. Редактор мнемосхем (Schema Builder)

5. Перегляд мнемосхем (Schema Viewer)

6. Пристрої

7. ПЛК

8. Послідовності

9. Рецепти та змінні

10. Події та аварії

11. Опції

12. Імпульсні модулі

13. Керування партіями

14. Виробничі лінії

15. Тренди

16. Аварійні сигнали

17. Математичні функції

18. Точки вводу-виводу

19. Провізіонування I/O

20. Графічні статуси

21. Конфігуратор обладнання

22. Керування користувачами

23. Керування ролями

24. Журнал активності

25. Налаштування

26. AI-асистент

27. Експорт PLC-проєкту

28. Експорт та імпорт проєкту

29. Танки

Навігація додатка (Sidebar)

Глобальні елементи інтерфейсу

Гарячі клавіші

Практичні приклади

Усунення несправностей і FAQ

Безпека та найкращі практики

Глосарій

Додаток А: Довідник WebSocket-каналів

Додаток В: Довідник API (зведена таблиця)

Додаток С: Коды помилок і аварій

Версія документа: 1.0

Дата: Квітень 2026

Мови інтерфейсу: Багатомовний (EN, RU, UK, DE, FR, PL, AR та інші)

Передмова

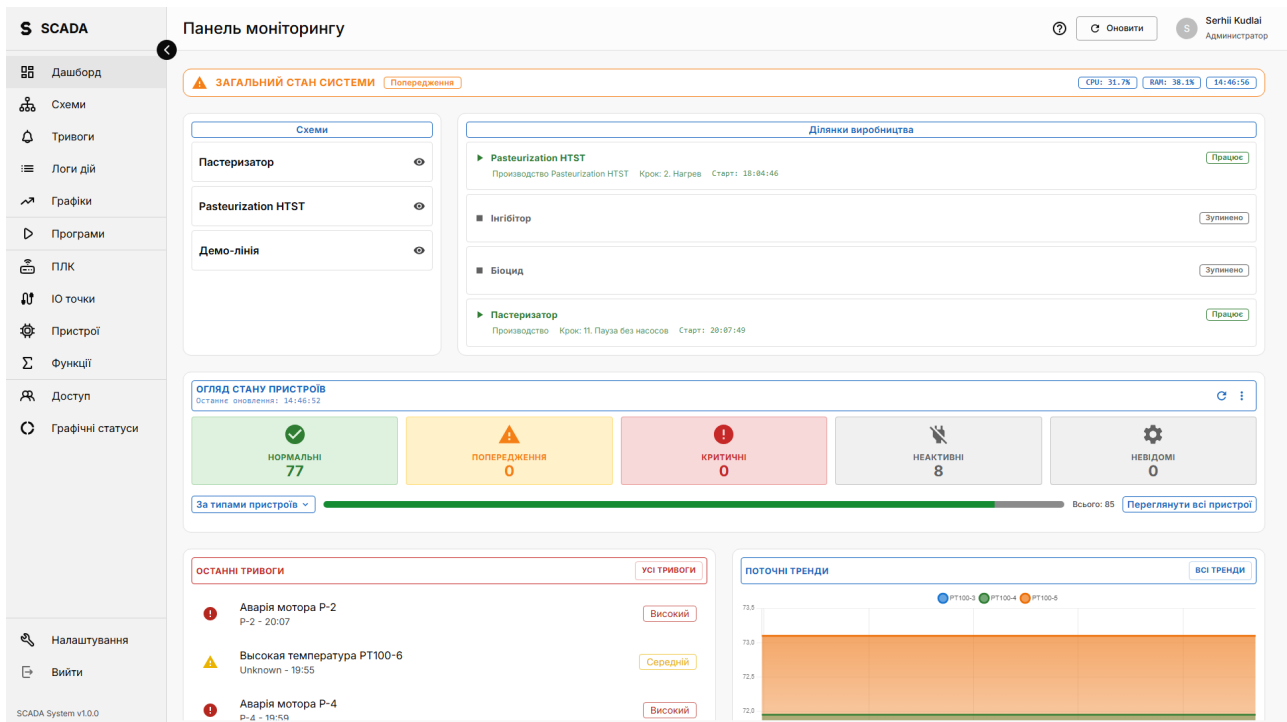
Про систему

SCADA System — це сучасна платформа диспетчерського керування та збору даних (Supervisory Control and Data Acquisition), розроблена з нуля для потреб промислової автоматизації нового покоління. Система призначена для повного циклу керування технологічними процесами — від конфігурування обладнання та створення мнемосхем до оперативного керування виробничими лініями в реальному часі.

На відміну від традиційних SCADA-систем, що народилися в епоху настільних застосунків та пропріетарних протоколів, ця платформа побудована на сучасному веб-стеку та принципах мікросервісної архітектури. Це означає, що оператор може працювати з системою з будь-якого браузера — на промисловому ПК у цеху, на планшеті в лабораторії або на ноутбучі з офісу — без встановлення додаткового програмного забезпечення.

Система охоплює весь життєвий цикл автоматизованого виробництва:

- Проектування — візуальний конструктор мнемосхем, конфігурування пристроїв, створення алгоритмів керування (послідовностей) та рецептів
- Розгортання — автоматична генерація PLC-коду (Siemens SCL/XML), провізійонування точок вводу-виводу, деплой через TIA Portal Agent
- Експлуатація — моніторинг процесу в реальному часі, керування обладнанням, трендовий аналіз, керування партіями
- Обслуговування — журнал активності, аварійні сигнали, діагностика, автоматичні оновлення, резервне копіювання



Колаж із 4 основних екранів системи: Dashboard (верхній лівий), Schema Viewer з працюючою схемою (верхній правий), Trends з графіками (нижній лівий), Sequence Editor (нижній правий). Демонструє широту функціональності.

Цільова аудиторія

Роль	Як використовує систему
Інженер-проектувальник АСУ ТП	Створює мнемосхеми, налаштовує пристрої та послідовності, конфігурує PLC, генерує код для TIA Portal
Інженер-технолог	Розробляє виробничі рецепти, налаштовує параметри процесу, визначає аварійні пороги
Оператор виробничої лінії	Спостерігає за мнемосхемою в реальному часі, керує обладнанням (пуск/стоп, уставки), працює з партіями та рецептами
Начальник зміни / Майстер	Контролює стан усіх ліній через Dashboard, аналізує тренди, працює з журналом аварій
Системний	Керує користувачами та ролями, оновлює систему, налаштовує підключення до PLC та мережі
Керівництво підприємства	Переглядає звіти, тренди ефективності, статуси виробничих ліній з будь-якого місця через браузер

Ключові переваги

1. Повністю веб-орієнтована платформа

Система працює в будь-якому сучасному браузері без встановлення клієнтського ПЗ. Це принципово відрізняє її від класичних SCADA-систем (WinCC, Citect, InTouch), які потребують встановлення товстого клієнта на кожній робочій станції, прив'язані до конкретної ОС (як правило, Windows) та потребують купівлі окремих ліцензій для кожного робочого місця.

- Доступ з будь-якої точки мережі (інтранет або через VPN)
- Робота на будь-якому пристрої: ПК, планшет, промисловий термінал
- Немає прив'язки до операційної системи
- Миттєве оновлення інтерфейсу для всіх користувачів одночасно

2. Оновлення даних у реальному часі

Дворівнева архітектура WebSocket забезпечує миттєву доставку змін усім підключеним клієнтам:

- Rust WebSocket сервер (порт 8001) — високопродуктивна доставка масових оновлень пристроїв (до тисяч параметрів на секунду). Написаний на Rust для мінімальної затримки та максимальної пропускну здатності.
- Django Channels WebSocket — двонаправлена взаємодія для команд керування (пуск/стоп мотора, відкриття/закриття клапана, зміна уставки PID).

Дельта-оновлення передають лише змінені поля, а пакетне надсилання (batching кожні 250 мс) мінімізує мережеве навантаження при оновленні сотень параметрів одночасно.

3. Інтегрована генерація PLC-коду

Унікальна функція, що не має аналогів у більшості SCADA-систем: з конфігурації пристроїв, послідовностей, рецептів та аварій система автоматично генерує повноцінний проєкт для Siemens TIA Portal:

- 11 функціональних блоків (SCL) для кожного типу пристрою (Motor FB, Valve FB, PID FB, AI/AO/DI/DO FB, COS FB, Counter FB, Timer FB, Tank FB)
- Sequence FB — автомат станів для кожної послідовності
- Event / Accident FBs — обробники подій та аварій
- Data Blocks — екземпляри БД пристроїв та рецептів
- OB1 Main — головна програма з викликом усіх FB у правильному порядку
- XML Tag Tables — таблиці тегів у форматі SimaticML для прямого імпорту в TIA Portal

Це скорочує час розгортання нової виробничої лінії з тижнів до годин: інженер конфігурує процес у SCADA, натискає «Генерувати», завантажує ZIP у TIA Portal — і PLC готовий до роботи.

4. Візуальний конструктор мнемосхем

Потужний редактор мнемосхем на базі Konva.js (HTML5 Canvas) замінює необхідність використання окремих SCADA-графічних редакторів:

- Drag-and-drop розміщення пристроїв, трубопроводів, тексту, зображень
- Автоматична візуалізація стану пристроїв (копір, анімація потоку)
- Підтримка фонових зображень (підкладка P&ID креслення)
- Конструктор танків з вибором форми кришок та автоматичним розміщенням сенсорів
- Багатомовні підписи та переклади
- Необмежене масштабування та панорамування
- Множинне виділення та групові операції

5. Гнучка система рецептів та послідовностей

Система рецептів реалізує стандарт ISA-88 (Batch Control), дозволяючи розділяти:

- Послідовність (Sequence) — алгоритм: порядок кроків та умови переходів
- Рецепт кроків (Recipe) — конкретні значення пристроїв для кожного кроку
- Виробничий рецепт (Production Recipe) — набір параметрів для конкретного продукту

Одна послідовність може використовуватись із різними рецептами для виробництва різних продуктів на одному обладнанні. Змінні рецепта дозволяють параметризувати процес без зміни алгоритму.

6. Керування партіями (Batch Management)

Повноцінна система керування виробничими партіями:

- Створення партій з прив'язкою до рецепта та виробничої лінії
- Відстеження проходження партії за етапами лінії
- Черга партій з пріоритетами
- Автоматичний трансфер між етапами
- Історія проходження кожної партії

7. AI-асистент

Вбудований AI-помічник на базі LLM (через OpenRouter) значно прискорює роботу з системою:

- Керування через природну мову: «Увімкни мотор MOT_001», «Які пристрої в аварії?»
- Автоматичний аналіз P&ID креслень: завантаження зображення → розпізнавання пристроїв та зв'язків → пропозиція розміщення на мнемосхемі
- Контекстна допомога та пояснення
- Агентний цикл з інструментами (до 15 ітерацій) для складних завдань

8. Багатомовність

Повна підтримка 10+ мов інтерфейсу з можливістю перекладу не лише UI, але й користувацького контенту:

- Інтерфейс: EN, UK, RU, DE, FR, PL, AR (з RTL) та інші
- Імена пристроїв, послідовностей, кроків, рецептів, опцій — з полем name_translations
- Тексти на мнемосхемах — з полем translations
- Назви мнемосхем — з полем name_translations
- Перемикання мови «на льоту» без перезавантаження

9. Масштабована рольова модель доступу

Гранулярна система дозволів дозволяє точно налаштувати, хто й що може робити:

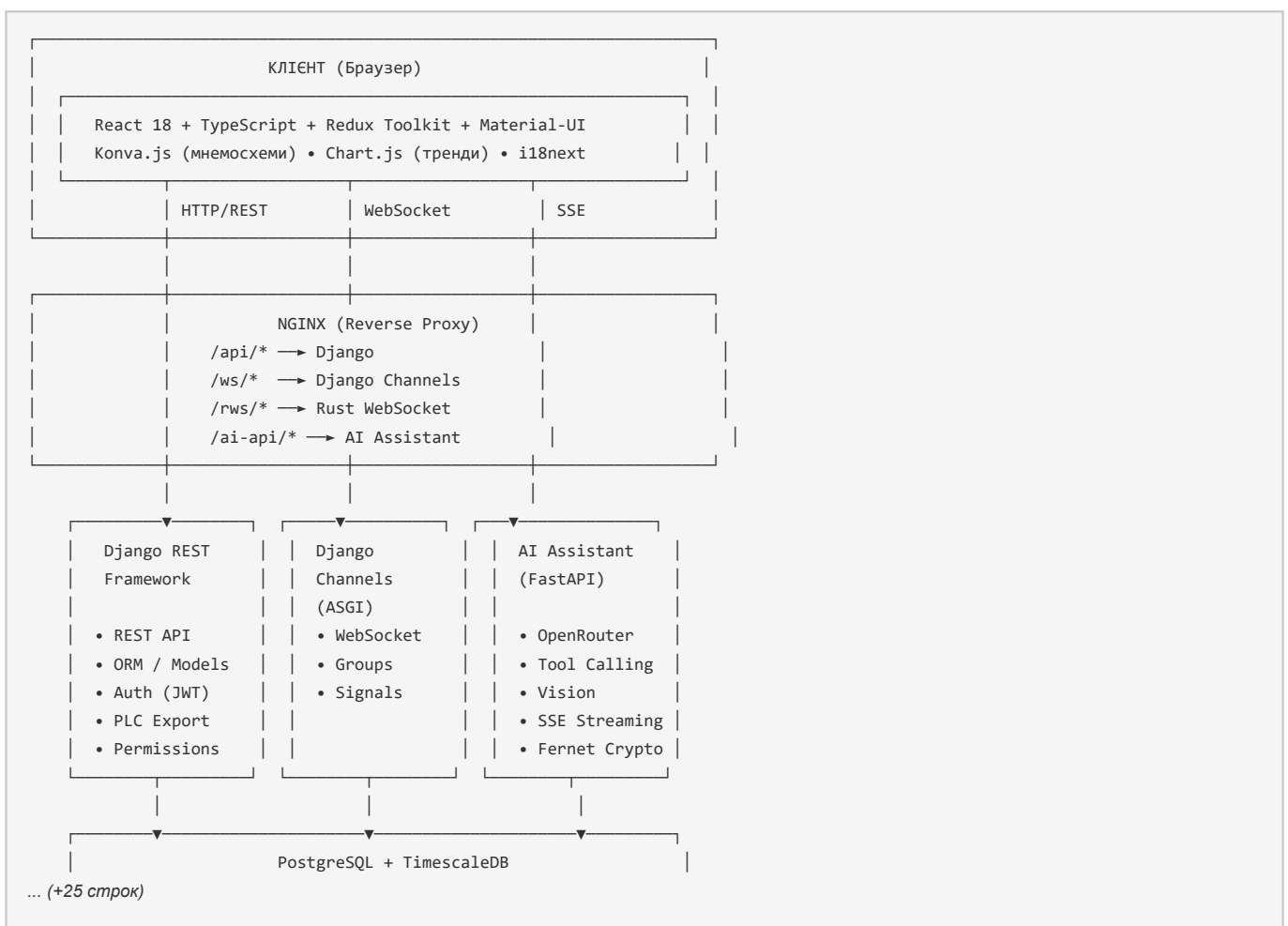
- Ролі з матрицею дозволів за 12 категоріями та безліччю дій
- Керування доступом до окремих мнемосхем (per-schema access)
- Журнал усіх дій користувачів (аудит)
- IP-фільтрація доступу

10. Промислова надійність

- Автоматичне перепідключення WebSocket з експоненційною затримкою
- Dual WebSocket: Rust для швидкості + Django для керування (fallback)
- Повне резервне копіювання та відновлення БД
- Експорт/імпорт проєктів для перенесення між установками
- Система оновлень з rollback та dry-run
- Smoke-тести після оновлень

Архітектура системи

Система побудована на мікросервісній архітектурі з чітким розподілом відповідальності:



Опис компонентів

Frontend (React + TypeScript)

Клієнтський застосунок, що працює в браузері користувача. Побудований на React 18 з використанням TypeScript для типобезпеки, Redux Toolkit для керування станом, Material-UI для компонентів інтерфейсу. Мнемосхеми відмальовуються через Konva.js (HTML5 Canvas), тренди — через Chart.js. Багатомовність

реалізована через i18next з підтримкою 10+ мов, включаючи RTL (арабська). Взаємодіє з бекендом через REST API (axios), WebSocket (нативний API) та SSE (для AI-стрімінгу).

NGINX (Reverse Proxy)

Точка входу для всіх запитів. Маршрутизує трафік між мікросервісами:

- /api/* → Django REST API
- /ws/* → Django Channels (WebSocket)
- /rws/* → Rust WebSocket сервер (високопродуктивний read-only)
- /ai-api/* → AI Assistant (FastAPI)
- /media/* → Статичні файли (іконки, зображення)

Django REST Framework (Backend API)

Ядро бізнес-логіки:

- Усі CRUD-операції для 40+ типів ресурсів
- Автентифікація та авторизація (JWT + RBAC)
- Керування користувачами, ролями та дозволами
- Генерація PLC-коду (11 типів функціональних блоків, автомат станів, Data Blocks, Tag Tables)
- Експорт/імпорт проєктів
- Системне обслуговування (оновлення, бекапи, діагностика)

Django Channels (ASGI WebSocket)

Двонаправлена взаємодія в реальному часі:

- 17+ типів WebSocket-каналів (пристрої, схеми, послідовності, партії, аварії, мат. функції, активність, система)
- Redis Channel Layer для group broadcasting
- Обробка команд керування (пуск/стоп пристроїв, керування партіями)
- Сигнали оновлення при зміні моделей Django

AI Assistant (FastAPI)

Автономний мікросервіс AI-помічника:

- FastAPI на Python 3.11 (порт 8080)
- Інтеграція з OpenRouter API (LLM з підтримкою tool calling)
- Агентний цикл-оркестратор (до 15 ітерацій) з 4 категоріями інструментів
- Аналіз P&ID зображень через vision-модель
- SSE-стрімінг відповідей (token, toolcall, toolresult, done)
- Шифрування API-ключів (Fernet/AES)
- JWT-автентифікація (проксується через Django /api/users/me/)
- Власні таблиці у спільній PostgreSQL БД (префікс ai_assistant_)
- Alembic міграції

PostgreSQL + TimescaleDB

Єдина база даних для всіх сервісів:

- Основні таблиці (core_*) — пристрої, схеми, послідовності, рецепти, ролі, користувачі
- Гіпертаблиці TimescaleDB — зберігання історичних даних (тренди) з автоматичним партиціонуванням за часом та стисненням
- Таблиці AI-асистента (ai_assistant_*) — сесії, повідомлення, налаштування

TimescaleDB забезпечує продуктивність запитів до часових рядів на порядки вищу за стандартний PostgreSQL при великих обсягах даних.

Redis

Комунікаційна шина:

- Channel Layer для Django Channels (групи WebSocket)
- Pub/Sub для сповіщень між Django та Rust Worker
- Кешування частих запитів

Rust Worker (scada-worker)

Високопродуктивний сервіс обробки в реальному часі, написаний на Rust:

- Sequence Engine (engine.rs) — виконання послідовностей: відстеження поточного кроку, перевірка умов переходу, активація/деактивація пристроїв, робота з рецептами
- Events Processor — обробка подій послідовностей
- Accident Processor — виявлення аварійних ситуацій, створення записів аварій
- Tank Volume Calculator — періодичний розрахунок об'ємів танків (кожні 1000 мс) за даними сенсорів рівня та тиску
- Math Function Calculator — перерахунок математичних функцій
- DB Batchер — пакетний запис змін у PostgreSQL (оптимізація I/O)
- WebSocket Batchер — пакетна відправка оновлень через WebSocket (інтервал 250 мс)
- Rust WebSocket Server (порт 8001) — виділений сервер для read-only стрімінгу статусів пристроїв. Значно швидший за Django Channels для масових оновлень.

Вибір Rust для цього компонента зумовлений вимогами до мінімальної затримки, передбачуваного використання пам'яті та здатності обробляти тисячі оновлень на секунду без garbage collection пауз.

Переваги над традиційними SCADA-системами

Критерій	Традиційні SCADA	Ця система
Клієнт	Товстий клієнт (встановлення на кожен ПК)	Веб-браузер (нічого не потрібно встановлювати)
ОС	Тільки Windows	Будь-яка (Windows, Linux, macOS, Android, iOS)
Ліцензування	Per-seat (за кожне робоче місце)	Серверна ліцензія (необмежені клієнти)
Оновлення	Ручне на кожному ПК	Автоматичне, централізоване (усі клієнти одразу)
Мобільний доступ	Немає або окремий модуль	Адаптивний інтерфейс з коробки
Генерація PLC-коду	Немає (ручне програмування)	Автоматична з конфігурації SCADA
AI-помічник	Немає	Вбудований (аналіз P&ID, керування, допомога)
Багатомовність	Обмежена (інтерфейс)	Повна (інтерфейс + контент + RTL)
Реальний час	Пропріетарні протоколи	WebSocket + Rust (sub-second latency)
Рецепти	Базова підтримка	ISA-88 (Sequence + Recipe + Production Recipe)
Batch Management	Окремий модуль (дод. ліцензія)	Вбудований
Трендовий аналіз	Вбудований, обмежений	TimescaleDB (петабайти даних, автостиснення)
Аудит	Базовий	Повний журнал з WebSocket-оновленнями
Бекап/Відновлення	Зовнішні інструменти	Вбудований (бекап, експорт, імпорт, rollback)
Вартість розгортання	Висока (ліцензії + установка + налаштування)	Низька (Docker, автоматизоване розгортання)
Розширюваність	Закрита екосистема	Відкритий REST API + WebSocket + SSE

Системні вимоги

Серверна частина

Компонент	Мінімальні	Рекомендовані
Процесор	4 ядра	8+ ядер
Оперативна пам'ять	8 ГБ	16+ ГБ
Дисковий простір	20 ГБ SSD	100+ ГБ SSD (для трендів)
ОС	Linux (Ubuntu 22.04+) або Windows Server 2019+	Linux (Ubuntu 22.04 LTS)
Docker	Docker Engine 24+	Docker Engine 24+ з Docker Compose v2
PostgreSQL	15+ з TimescaleDB	16+ з TimescaleDB 2.x
Redis	7+	7+

Клієнтська частина (браузер)

Компонент	Вимога
Браузер	Chrome 90+, Firefox 90+, Edge 90+, Safari 15+
Роздільна здатність екрана	Мінімум 1280×720, рекомендовано 1920×1080+
Мережа	Стабільне підключення до сервера (LAN/VPN)

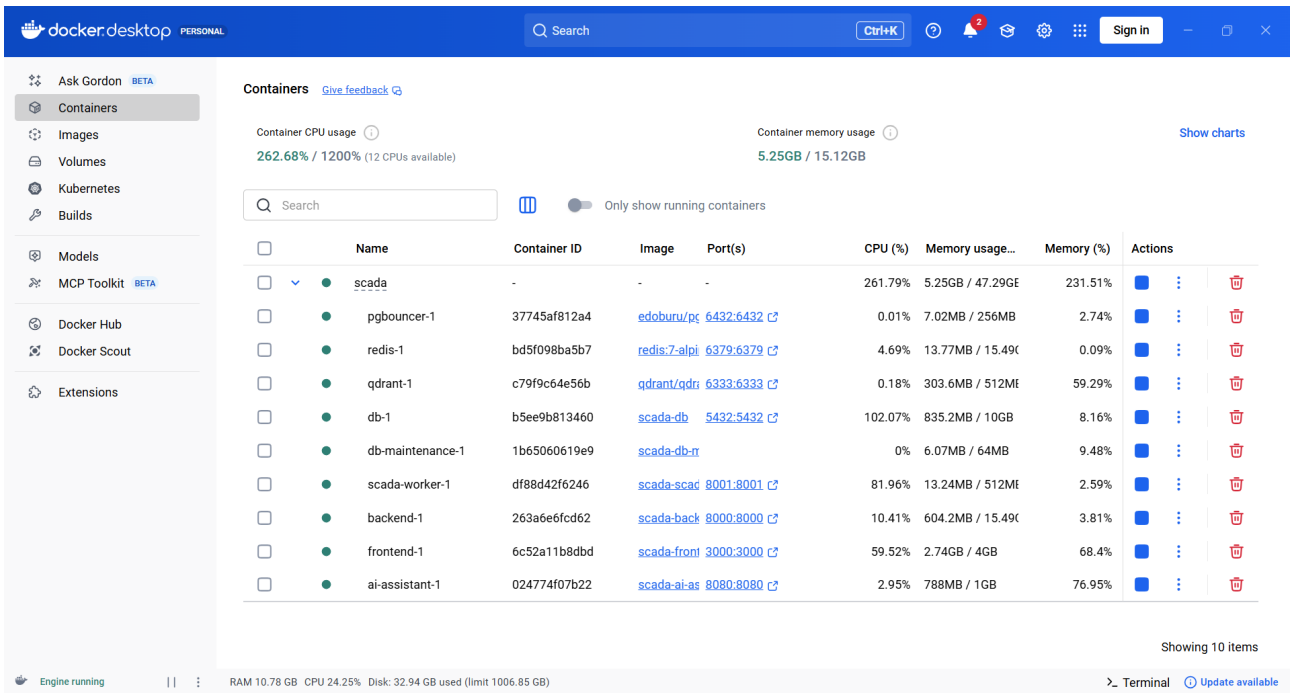
Розгортання

Система постачається у вигляді Docker-контейнерів та розгортається однією командою:

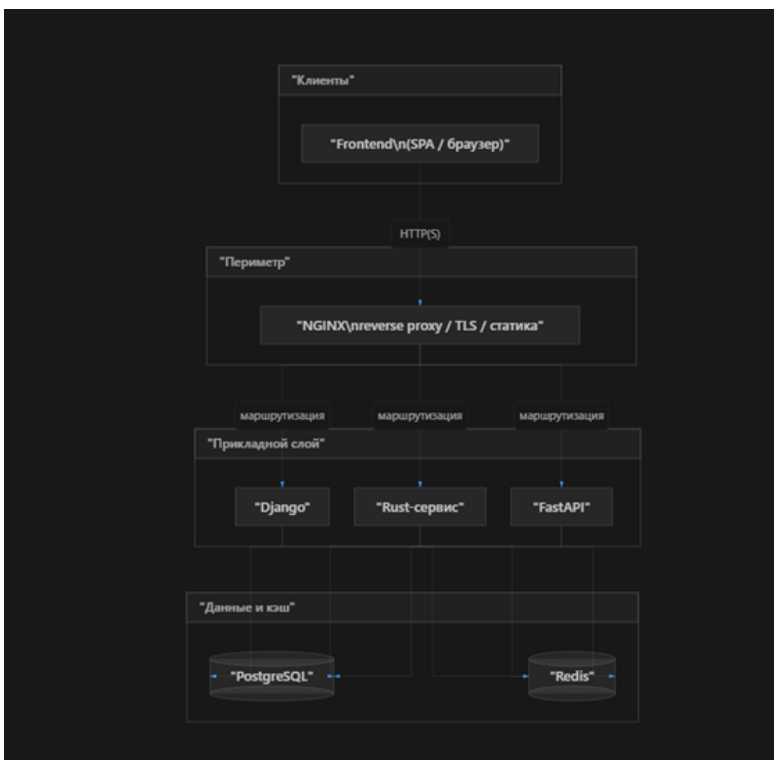
```
docker-compose up -d
```

Склад контейнерів:

- backend — Django REST API + Django Channels
- scada-worker — Rust Worker
- ai-assistant — FastAPI AI сервіс
- postgres — PostgreSQL + TimescaleDB
- redis — Redis
- nginx — Reverse Proxy



Термінал з виведенням `docker-compose ps`: усі 6 контейнерів у статусі «Ур», порти, час роботи.



Архітектурна діаграма системи: блоки компонентів з лініями зв'язку (REST, WebSocket, Redis Pub/Sub, SQL). Кольорове кодування: синій — Frontend, зелений — Django, помаранчевий — Rust, фіолетовий — AI, сірий — інфраструктура.

Швидкий старт — 15 хвилин

Цей розділ проведе вас від чистої інсталяції до робочої мнемосхеми з одним двигуном та одним клапаном. Якщо ви вперше запускаєте SCADA — почніть звідси.

Крок 1. Перший вхід (2 хв)

- Відкрийте у браузері адресу інсталяції (наприклад, <http://localhost> або <http://scada.local>).
- На екрані входу введіть облікові дані адміністратора, створені інсталятором (за замовчуванням admin / пароль, заданий під час установлення).
- Під час першого входу система може запросити активацію ліцензії — введіть ключ і натисніть Активувати. Без активної ліцензії доступ до більшості функцій обмежено.
- Після успішного входу ви потрапите на Dashboard.

Підказка. Оберіть мову інтерфейсу в лівому нижньому куті форми входу — її буде збережено для вашого облікового запису.

Крок 2. Створення першого ПЛК (2 хв)

Навіть якщо ви поки що працюєте лише в симуляції, система вимагає наявності хоча б одного ПЛК, до якого прив'язуються пристрої.

- У боковому меню відкрийте PLCs.
- Натисніть «+» у правому верхньому куті.
- Заповніть:
 - Name: PLC_DEMO
 - IP Address: 192.168.0.100 (адресу можна змінити пізніше)
 - Type: Siemens S7-1200 (або будь-який доступний у списку)
 - Rack / Slot: 0 / 1
- Натисніть Зберегти. Статус ПЛК буде «Offline» — це нормально для демо.

Крок 3. Створення перших пристроїв (3 хв)

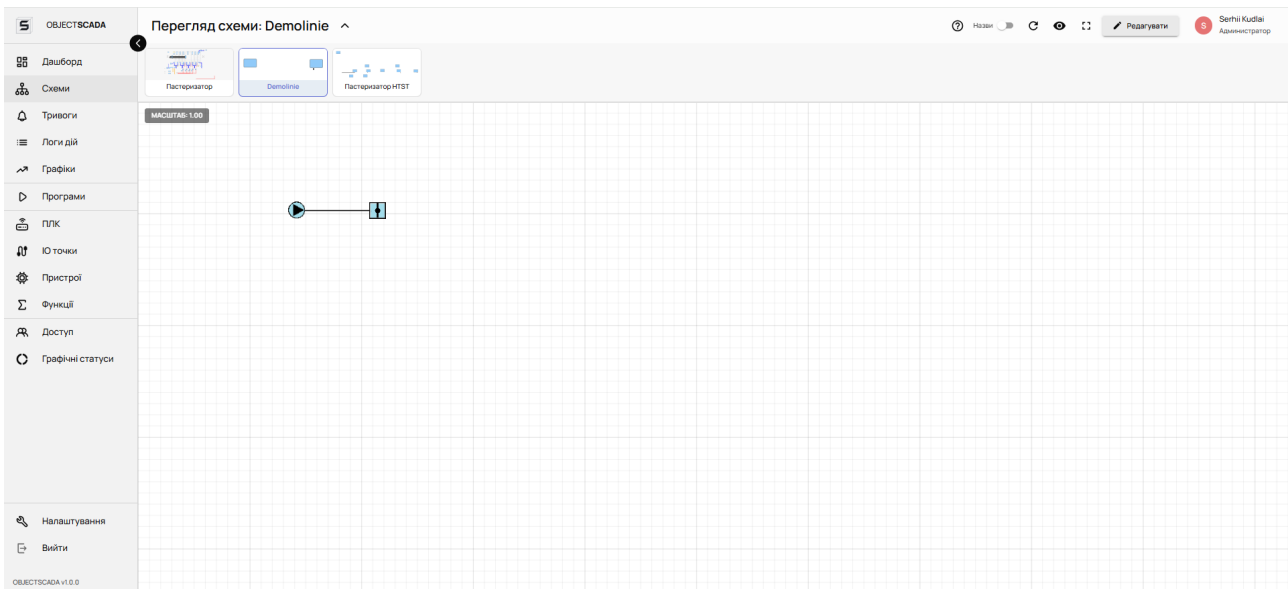
Додамо один двигун і один клапан.

- Перейдіть до Devices → «+» → Motor.
- Заповніть:
 - ID: mot_001
 - Name: Насос подачі
 - PLC: PLC_DEMO
 - Description: «Демонстраційний двигун»
- Натисніть Зберегти.
- Аналогічно створіть клапан («+» → Valve):
 - ID: val_001
 - Name: Клапан лінії
 - PLC: PLC_DEMO

Важливо. ID пристрою — це текстовий ідентифікатор (mot_001, а не число). Він використовується в усіх WebSocket-каналах, PLC-експорті та в умовах переходів. Дотримуйтесь єдиної схеми іменування від початку.

Крок 4. Створення першої мнемосхеми (3 хв)

- Відкрийте Schemas → «+».
- Вкажіть Name: «Демо-лінія» і натисніть Створити.
- У списку схем натисніть олівець (Edit) на щойно створеній схемі — відкриється Schema Builder.
- У лівій панелі натисніть Пристрій () і у відкритому діалозі оберіть mot_001 → Додати. Двигун з'явиться на полотні — перетягніть його мишею.
- Знову Пристрій → val_001 → Додати — розмістіть клапан поруч.
- Оберіть Труба (), клацніть на двигун → потім на клапан — між ними з'явиться лінія з'єднання.
- Натисніть Ctrl+S або кнопку Зберегти у верхній панелі.



Проста демо-схема: один двигун (зелений кружок) з'єднаний трубою з клапаном (жовтий трикутник), на задньому плані — сітка полотна.

Крок 5. Запуск перегляду та керування (3 хв)

- У верхній панелі редактора натисніть Перегляд — ви перейдете до Schema Viewer.
- Клацніть на іконці двигуна — відкриється роруп керування. Перемкніть режим на Manual і натисніть Start. Індикатор стане зеленим — двигун «працює».
- Клацніть на клапан → Open. Якщо у труби увімкнено моніторинг процесу, з'явиться анімація потоку.
- Натисніть Stop на двигуні — анімація зупиниться.

Вітаємо — у вас є перша робоча мнемосхема з живим керуванням.

Крок 6. Що робити далі (2 хв)

Тепер у вас є базовий каркас. Наступні кроки залежать від вашого завдання:

Завдання	Куди йти
Автоматизувати процес (запуск/зупинка за умовами)	Розділ 8 «Послідовності» та практичний приклад «Пастеризатор молока»

Завдання	Куди йти
Налаштувати рецепти для різних продуктів	Розділ 9 «Рецепти та змінні»
Прив'язати реальні входи/виходи ПЛК	Розділи 18 «I/O Points» та 19 «Провізіонування I/O»
Налаштувати аварійні пороги	Розділ 10 «Події та аварії»
Згенерувати проект для TIA Portal	Розділ 27 «Експорт PLC-проекту»
Аналіз трендів	Розділ 15 «Тренди»

Підказка. На будь-якій сторінці у шапці є кнопка «?» — вона запускає інтерактивний навчальний тур з підсвічуванням елементів.

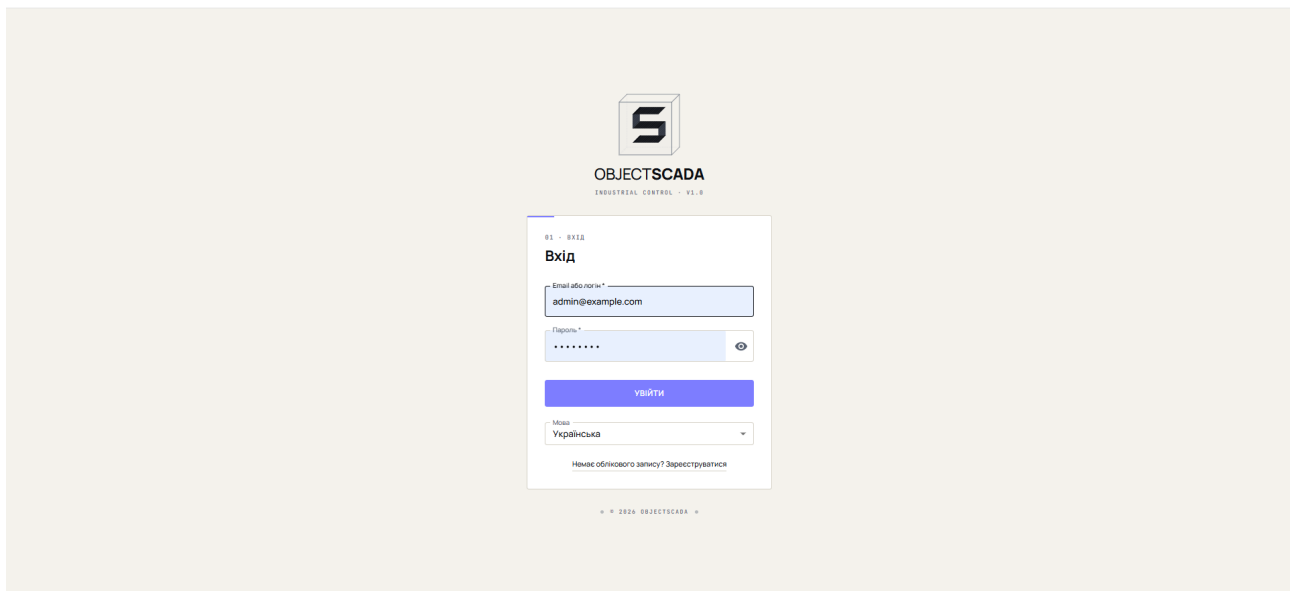
1. Авторизація та реєстрація

Для користувача

Екран входу — перше, що ви бачите під час відкриття системи.

Елементи екрана:

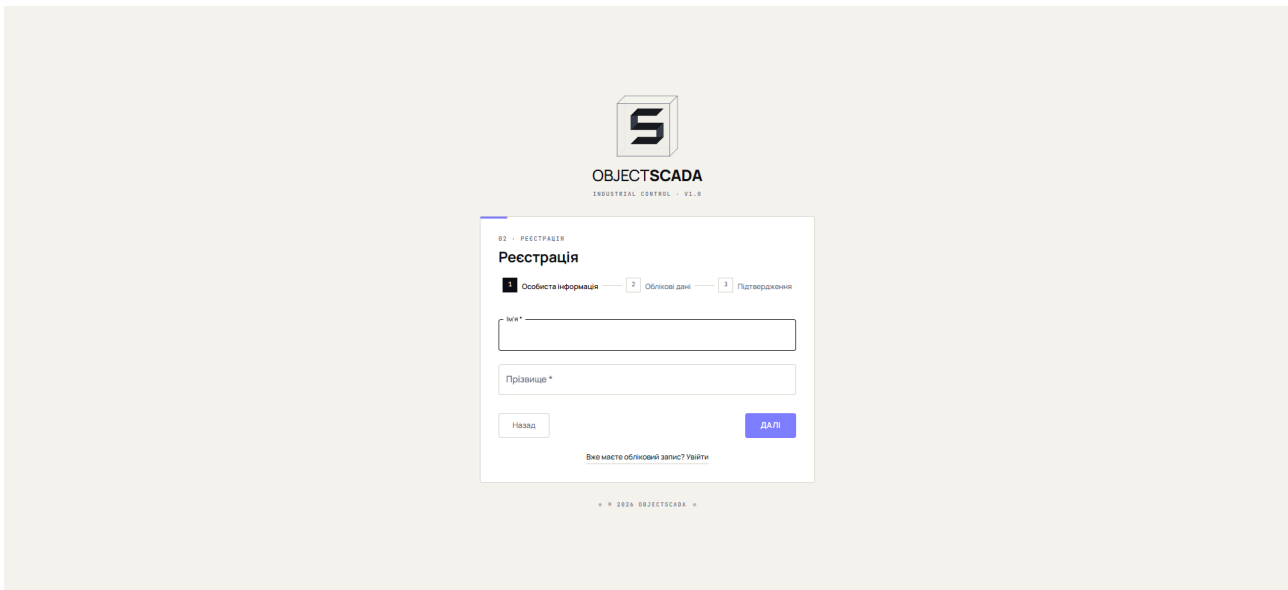
Елемент	Опис
Поле «Ім'я користувача / Email»	Введіть ваш логін або електронну пошту
Поле «Пароль»	Введіть пароль. Іконка праворуч перемикає видимість пароля
Вибір мови	Випадний список для вибору мови інтерфейсу (зберігається після входу)
Кнопка «Увійти»	Виконує вхід у систему
Посилання «Забули пароль?»	Перехід до відновлення пароля
Посилання «Реєстрація»	Перехід до форми реєстрації нового облікового запису



Сторінка входу: поля «Username/Email» і «Password» (з іконкою ока), випадний список мови, кнопка «Увійти», посилання «Забули пароль?» і «Реєстрація». Темна/світла тема залежно від налаштувань.

Екран реєстрації:

Елемент	Опис
Поле «Ім'я користувача»	Унікальне ім'я для входу
Поле «Email»	Електронна пошта
Поле «Пароль»	Пароль з індикатором надійності
Поле «Підтвердіть пароль»	Повторне введення пароля
Поле «Повне ім'я»	Ім'я та прізвище
Кнопка «Зареєструватися»	Створення облікового запису



Форма реєстрації: Username, Email, Password (з індикатором надійності), Confirm Password, Full Name, кнопка «Зареєструватися».

Технічна документація

API автентифікації:

Метод	Endpoint	Опис	Тіло запиту
POST	/api/auth/login	Вхід у систему	{"username": "...", "password": "..."}
POST	/api/auth/register	Реєстрація	{"username": "...", "email": "...", "password": "...", "first_name": "...", "last_name": "..."}
GET	/api/auth/me	Поточний користувач	—
POST	/api/auth/token/refresh/	Оновлення JWT токена	{"refresh": "..."}

Відповідь при вході:

```
{
  "access": "eyJhbGciOi...",
  "refresh": "eyJhbGciOi...",
  "user": {
    "id": 1,
    "username": "admin",
    "email": "admin@example.com",
    "roles": [...]
  }
}
```

Авторизація: Усі захищені endpoints потребують заголовок Authorization: Bearer .

2. Головна панель (Dashboard)

Для користувача

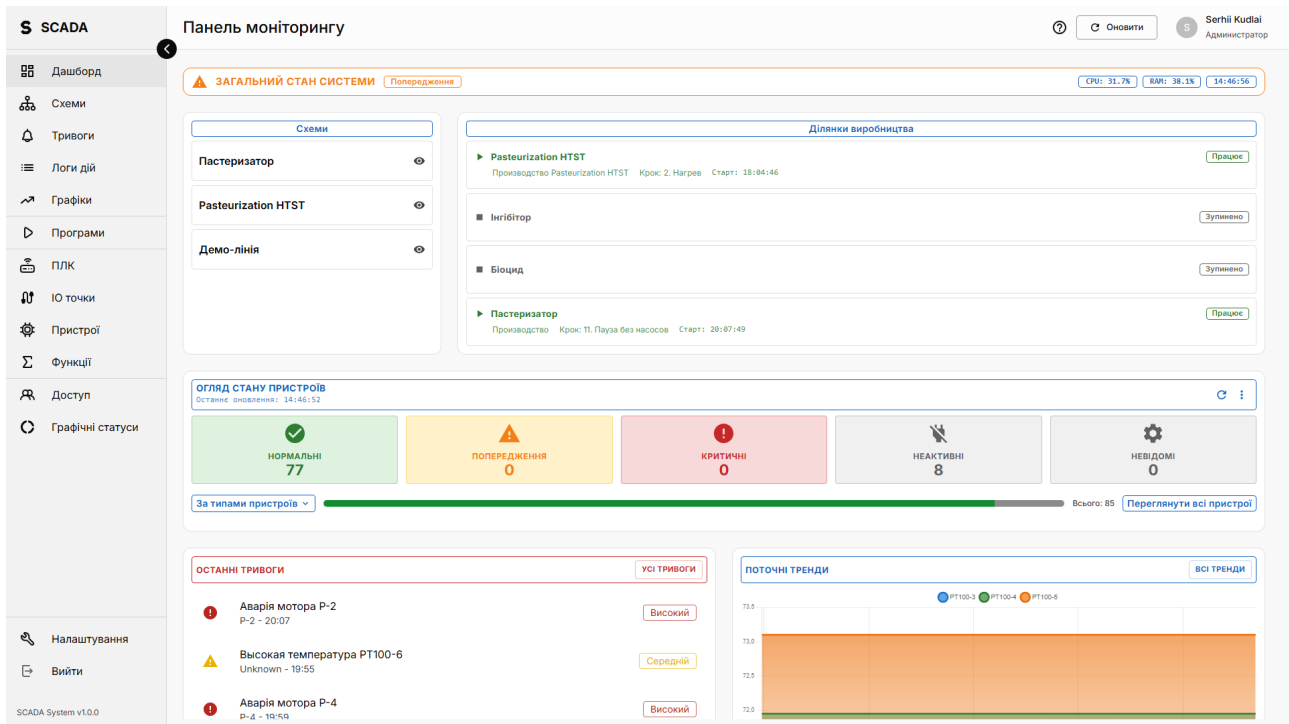
Dashboard — центральна сторінка системи, що відображає загальний стан.

Елементи екрана:

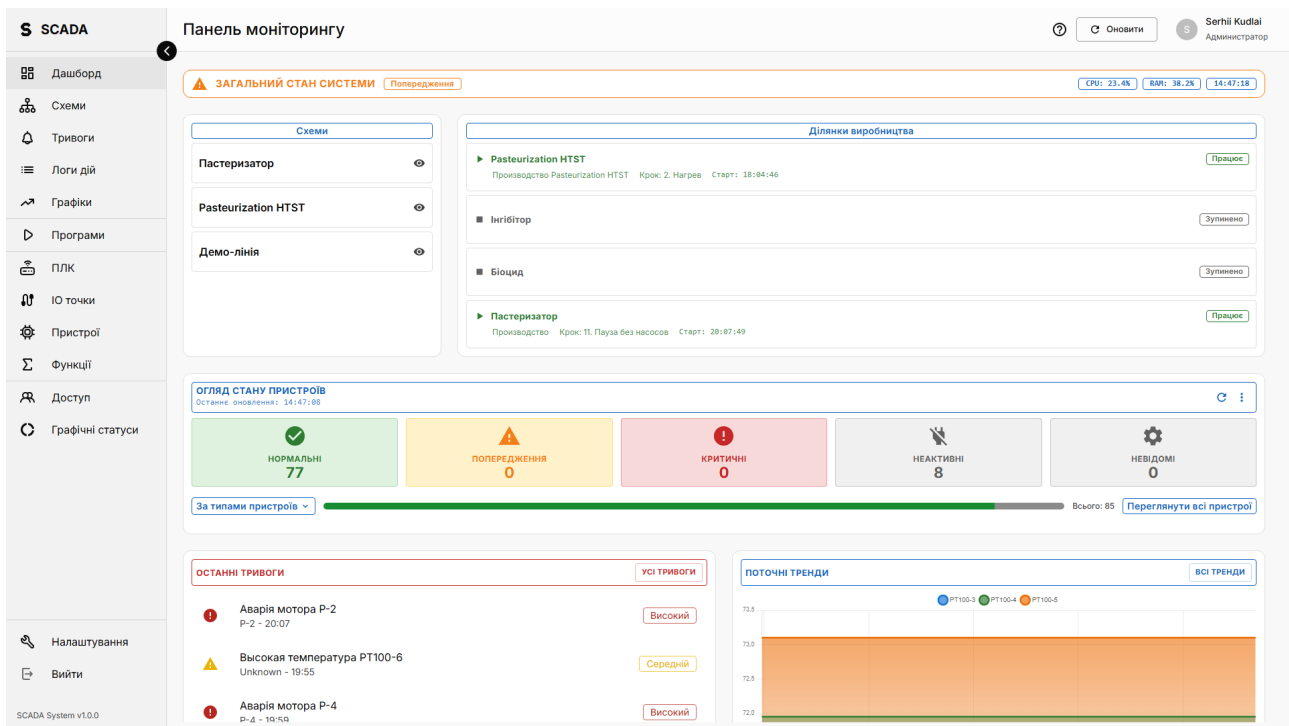
Елемент	Опис
Картки статистики	Стисла зведена інформація: кількість пристроїв, активних схем, аварій, послідовностей
Огляд стану	Візуальні індикатори стану пристроїв та системи
Список аварій	Останні аварійні сигнали з кольоровою індикацією серйозності
Міні-тренди	Мініатюрні графіки ключових параметрів
Статуси послідовностей	Поточний крок і прогрес кожної запущеної послідовності
Системні метрики	CPU, RAM, Uptime, температура, дисковий простір
Кнопки швидкого доступу	Швидкий перехід до основних функцій

Кнопки в шапці:

Кнопка	Опис
Оновити	Оновити дані на Dashboard
Тур	Запуск інтерактивного навчального туру елементами Dashboard
Експорт проекту	Відкриває діалог експорту всього проекту
Імпорт проекту	Відкриває діалог імпорту проекту з файлу



Повний екран Dashboard: картки статистики зверху, блок аварій ліворуч, міні-тренди праворуч, статуси послідовностей унизу. Бокова панель навігації ліворуч.



Блок системних метрик: CPU usage (%), RAM usage (%), Disk usage, Uptime, Network stats. Кожен параметр з кольоровим індикатором (зелений/жовтий/червоний).

Технічна документація

API системних метрик:

Метод	Endpoint	Опис
GET	/api/system/info/	Інформація про систему
GET	/api/system/health-status/	Стан здоров'я системи

WebSocket: Системні метрики (стрімінг кожні 2 с)

Підключення: ws://ws/system

```
{
  "type": "system_metrics",
  "cpu": 45.2,
  "memory": 67.8,
  "disk": 34.1,
  "network": {"rx": 1024, "tx": 512},
  "uptime": 86400,
  "temperature": 52.0,
  "lastUpdate": "2026-04-07T12:00:00Z"
}
```

WebSocket: Статус ПЛК (стрімінг кожні 2 с)

```
{
  "type": "plc_status",
  "is_active": true,
  "status": "running",
  "service_running": true,
  "timestamp": "2026-04-07T12:00:00Z"
}
```

3. Мнемосхеми

Для користувача

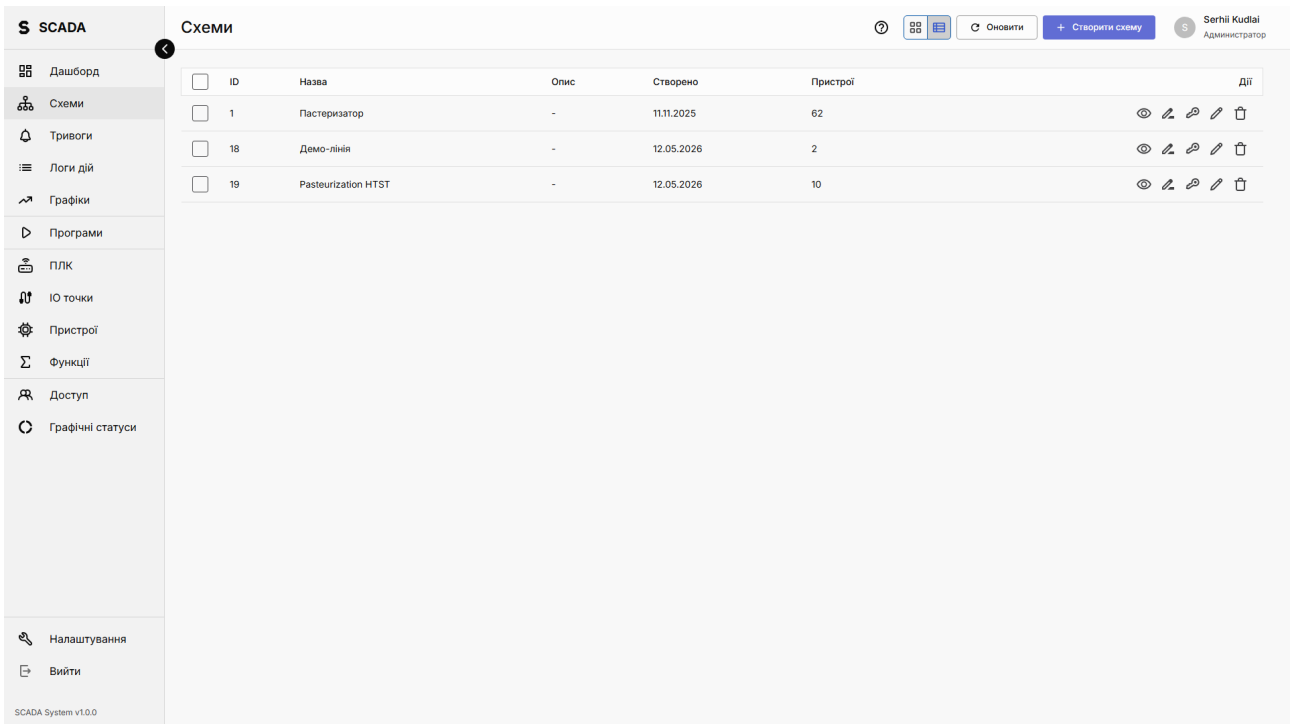
Мнемосхеми — візуальні діаграми вашого виробничого процесу. Це головний інструмент для моніторингу та керування.

Елементи екрана списку:

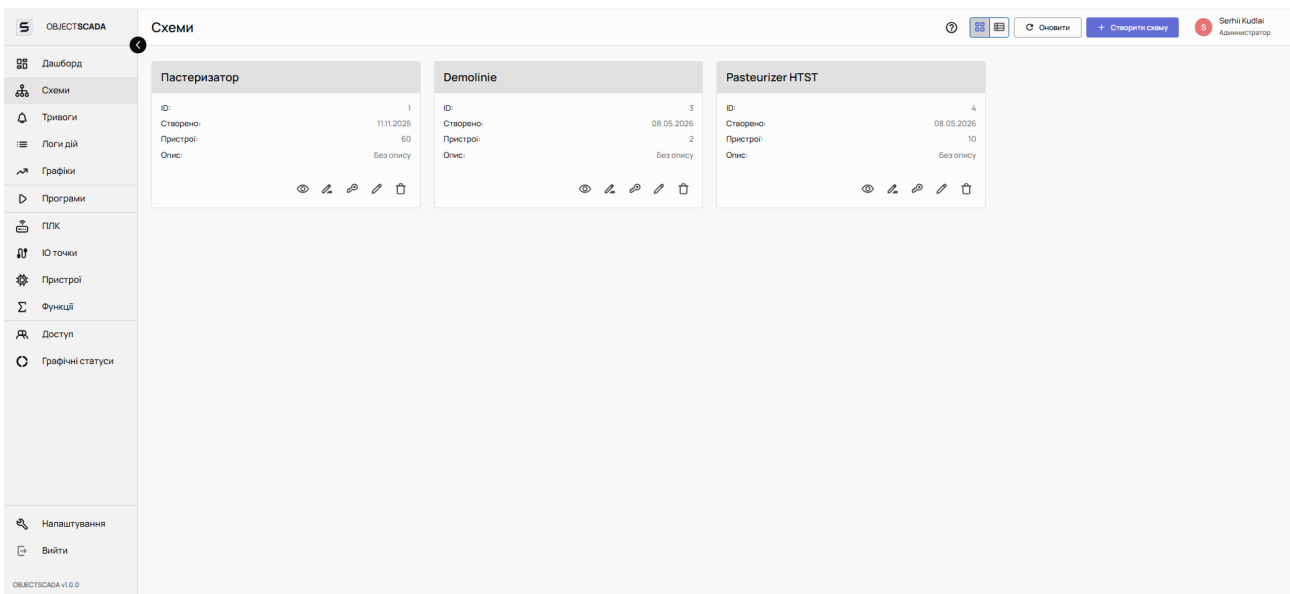
Елемент	Опис
Список/сітка схем	Картки або рядки таблиці з мнемосхемами
Перемикач вигляду	Сітка / Список (іконки праворуч зверху)
Кнопка «+»	Створення нової мнемосхеми
Пошук	Поле пошуку за назвою
Кнопка оновлення	Перезавантаження списку

Дії для кожної схеми:

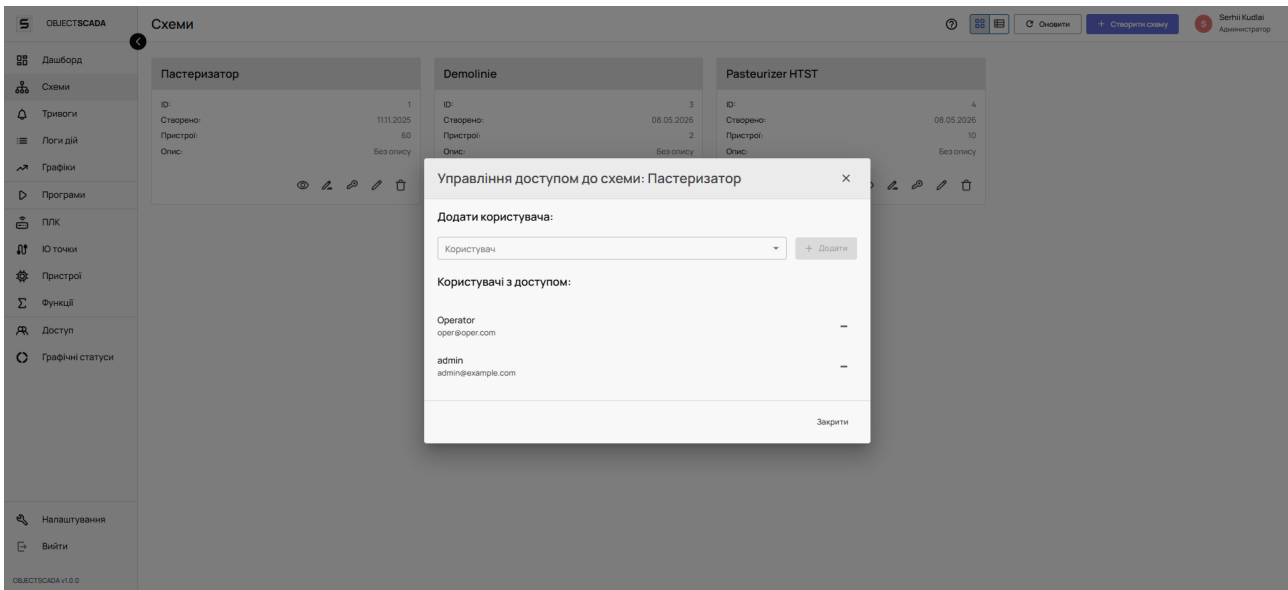
Кнопка	Іконка	Опис
Перегляд		Відкриває схему в режимі перегляду (SchemaViewer)
Редагувати		Відкриває схему в редакторі (SchemaBuilder)
Видалити		Видалення схеми (з підтвердженням)
Керування доступом		Налаштування доступу користувачів до схеми
Перейменувати		Зміна назви та перекладів назви
Налаштування		Налаштування параметрів схеми (опис, переклади)



Сторінка списку мнемосхем у вигляді сітки карток. Кожна картка показує прев'ю схеми, назву, дату зміни, кнопки дій (перегляд, редагування, видалення). Кнопка «+» у правому верхньому куті.



Та ж сторінка, але в табличному вигляді: колонки «Назва», «Опис», «Дата створення», «Останнє змінення», «Дії».



Діалог керування доступом: список користувачів з чекбоксами, кнопки «Додати» та «Зберегти».

Технічна документація

API мнемосхем:

Метод	Endpoint	Опис	Параметри
GET	/api/schemas/	Список усіх схем	?search=...&ordering=...
POST	/api/schemas/	Створення схеми	{"name": "...", "description": "...", "name_translations": {...}}
GET	/api/schemas/{id}/	Отримання схеми	—
PUT	/api/schemas/{id}/	Оновлення схеми	Повний об'єкт схеми
PATCH	/api/schemas/{id}/	Часткове оновлення	{"name": "...", "description": "...", "name_translations": {...}}
DELETE	/api/schemas/{id}/	Видалення схеми	—
GET	/api/schemas/{id}/user_access/	Список користувачів з доступом	—
POST	/api/schemas/{id}/user_access/	Додати доступ	{"user_id": 1}
DELETE	/api/schemas/{id}/user_access/	Прибрати доступ	{"user_id": 1}

4. Редактор мнемосхем (Schema Builder)

Для користувача

Редактор мнемосхем — потужний візуальний інструмент для створення та редагування технологічних діаграм. Працює на полотні (canvas) з функціями перетягування, масштабування та обертання елементів.

Ліва панель інструментів

Кнопка	Іконка	Опис
Виділення		Режим вибору та переміщення елементів
Пристрій		Додати пристрій з каталогу на схему
Труба/Сигнал		Намалювати трубопровід (з'єднання між пристроями)
Текст	T	Додати текстову анотацію на схему
Зображення		Додати фонове зображення (підкладку P&ID)
Послідовність		Прив'язати послідовність до схеми
Опція		Додати кнопку опції послідовності
Змінна рецепта		Додати відображення змінної рецепта
Виробнича лінія		Додати виробничу лінію
Мат. функція	f	Додати математичну функцію
Танк		Відкрити конструктор танка (окрема кнопка)
Створити пристрій		Швидке створення нового пристрою

Верхня панель інструментів

Кнопка	Опис
Зберегти	Зберігає поточний стан схеми
Скасувати (Ctrl+Z)	Скасування останньої дії
Повторити (Ctrl+Y)	Повтор скасованої дії
Перегляд	Перемикання в режим SchemaViewer
Перемикач «Імена»	Показати/приховати назви пристроїв на схемі
Назва схеми	Відображення поточного імені схеми

Керування полотном

Кнопка	Опис
Збільшити (+)	Збільшення масштабу
Зменшити (-)	Зменшення масштабу
Сітка	Увімкнути/вимкнути сітку на полотні
Колір фону	Змінити колір фону схеми
Прозорість фону	Повзунок прозорості фону
Згорнути	Згорнути/розгорнути ліву панель

Права панель властивостей

Панель відображається при виборі елемента на схемі та залежить від типу вибраного елемента:

Для пристрою (DevicePropertyPanel):

Властивість	Опис
Позиція X, Y	Координати на полотні
Кут повороту	Поворот елемента (0-360°)
Розмір W, H	Ширина та висота
Колір	Колір елемента (палітра)
Дзеркальне відображення	Відобразити по горизонталі/вертикалі
Показувати ім'я	Увімк/вимк підпис пристрою
Графічний статус	Прив'язка до шаблону статусів
Конфігурація	Пристрій-специфічні параметри

Для трубопроводу (PipePropertyPanel):

Властивість	Опис
Назва	Ім'я трубопроводу
Колір	Колір лінії
Товщина	Товщина лінії (px)
Видимість імені	Показувати/приховувати назву
Моніторинг процесу	Увімкнути анімацію потоку

Для тексту (TextPropertyPanel):

Властивість	Опис
Текст	Вміст
Розмір шрифту	Розмір тексту
Колір	Колір тексту
Жирний / Курсив	Стилізація
Переклади	Багатомовні переклади тексту

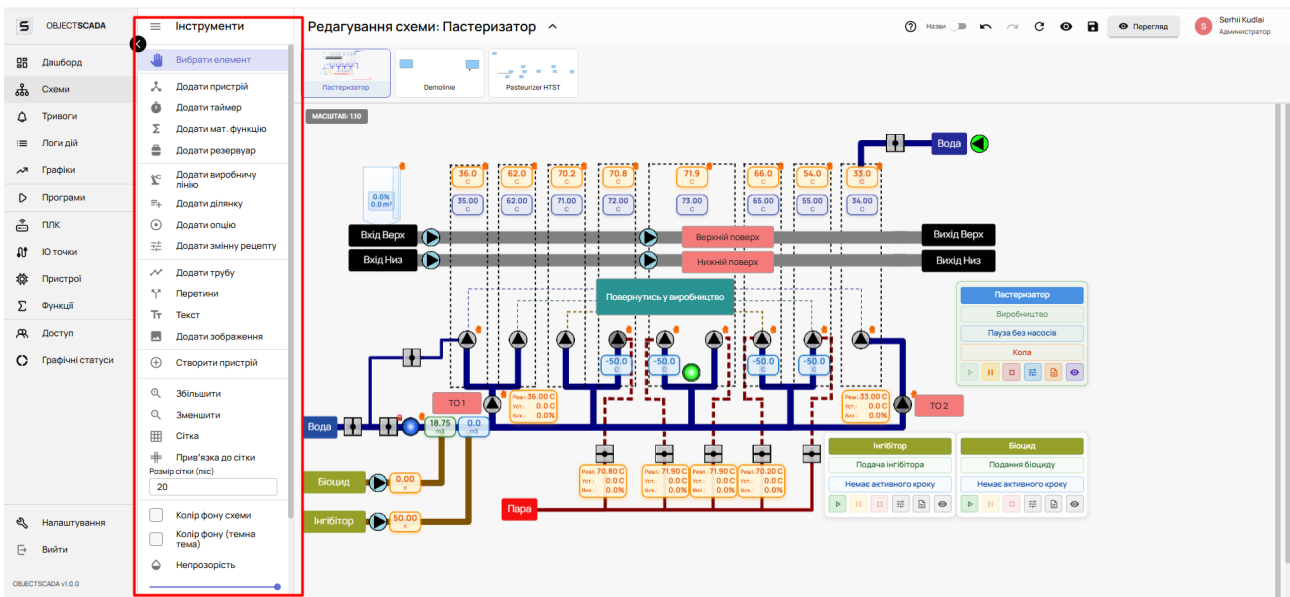
Для послідовності (SequencePropertyPanel):

Властивість	Опис
Позиція	Координати
Розмір	Ширина × Висота
Відображення	Налаштування відображення (ім'я, статус, кроки)


















Діалоги редактора

Діалог	Опис
DeviceDialog	Вибір пристрою з каталогу для розміщення на схемі. Фільтрація за типом.
PipeDialog	Налаштування імені, типу та видимості трубопроводу
AddSequenceDialog	Вибір послідовності для прив'язки до схеми

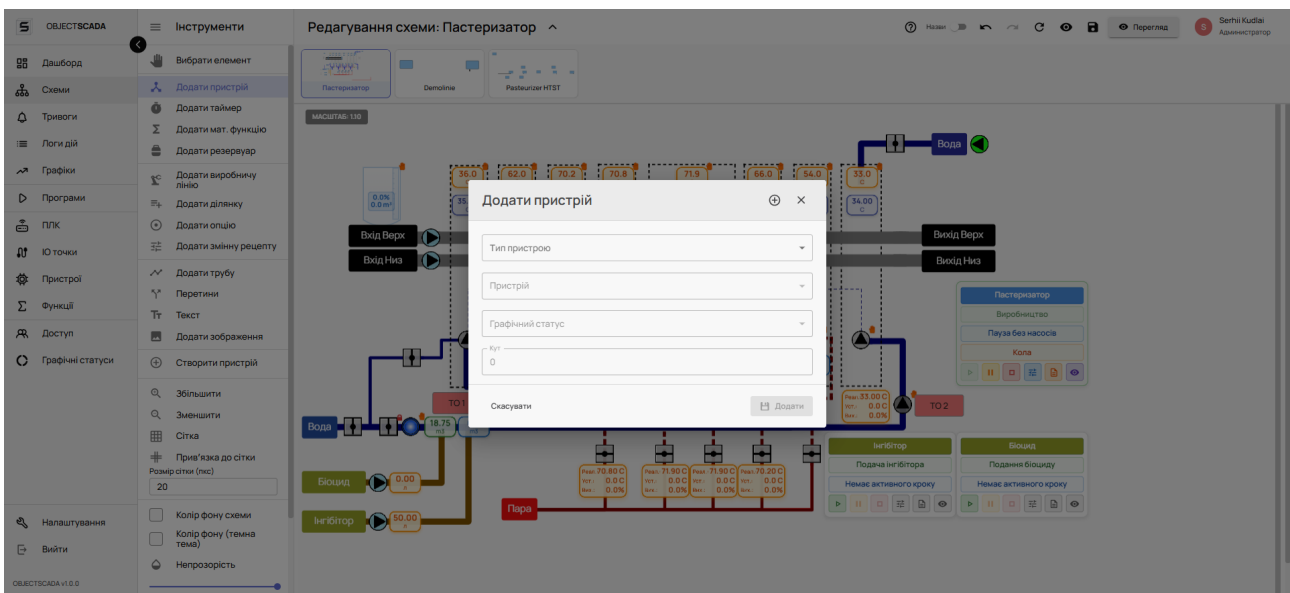
Діалог	Опис
AddOptionDialog	Вибір опції послідовності
AddProductionLineDialog	Вибір виробничої лінії
AddRecipeVariableDialog	Вибір змінної рецепта
MathFunctionDialog	Вибір математичної функції
TankConstructorDialog	Конструктор танка: вибір пристрою, типи кришок (FLAT/DOME/DISH/CONE), розміри, прев'ю SVG, авто-розміщення сенсорів
ImageDialog	Завантаження фонового зображення
TextDialog	Додавання/редагування тексту



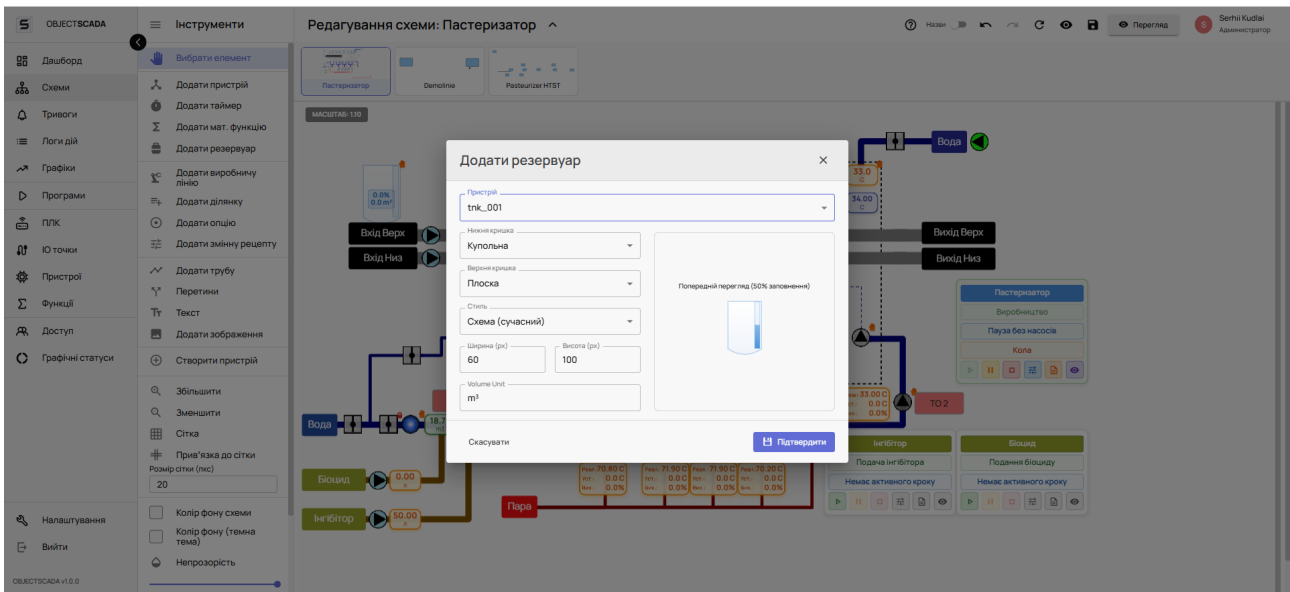
Повний екран редактора: ліва панель інструментів з іконками, верхня панель з кнопками «Зберегти», «Скасувати», «Повторити», центральне полотно зі схемою (пристрої, труби, тексти), права панель властивостей вибраного елемента.

-  Додати таймер
-  Додати мат. функці
-  Додати резервуар
-  Додати виробничу лінію
-  Додати ділянку
-  Додати опцію
-  Додати змінну рецп
-  Додати трубу
-  Перетини
-  Текст
-  Додати зображенн
-  Створити пристрій
-  Збільшити
-  Зменшити
-  Сітка
-  Прив'язка до сітки
- Розмір сітки (пкс)
- Колір фону схеми
- Непрозорість
-
-  Рамка схеми
- Готовий формат

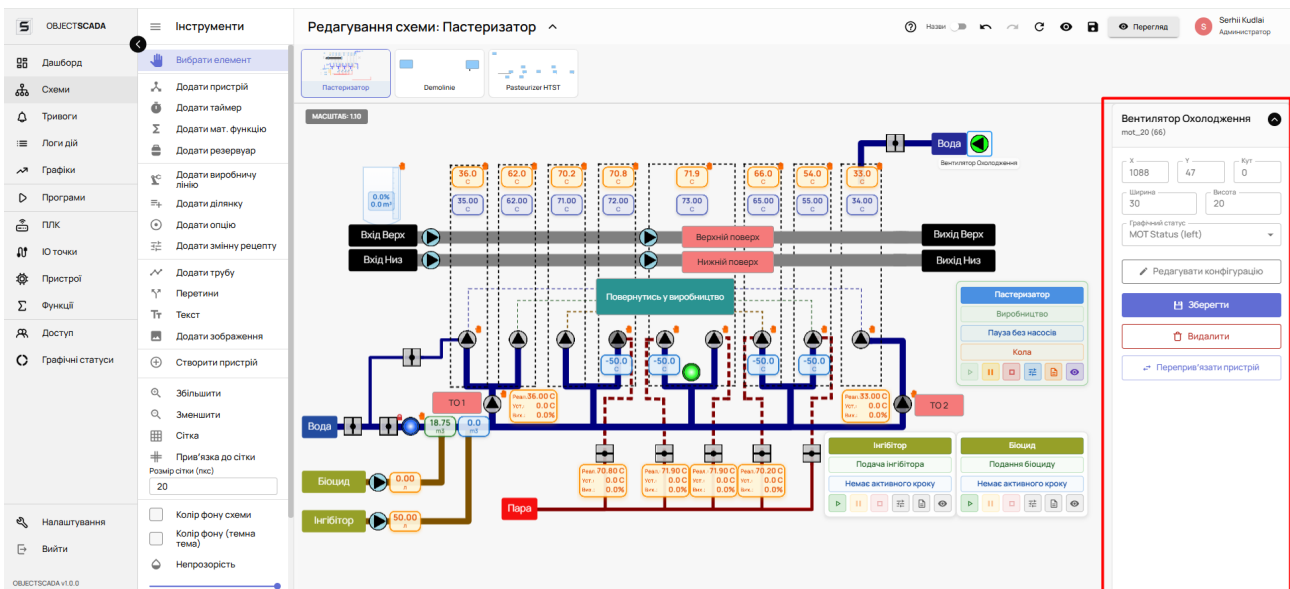
Великий план лівої панелі інструментів: кожна кнопка підписана (Виділення, Пристрій, Труба, Текст, Зображення, Послідовність, Опція, Змінна рецепта, Виробнична лінія, Мат. функція, Танк).



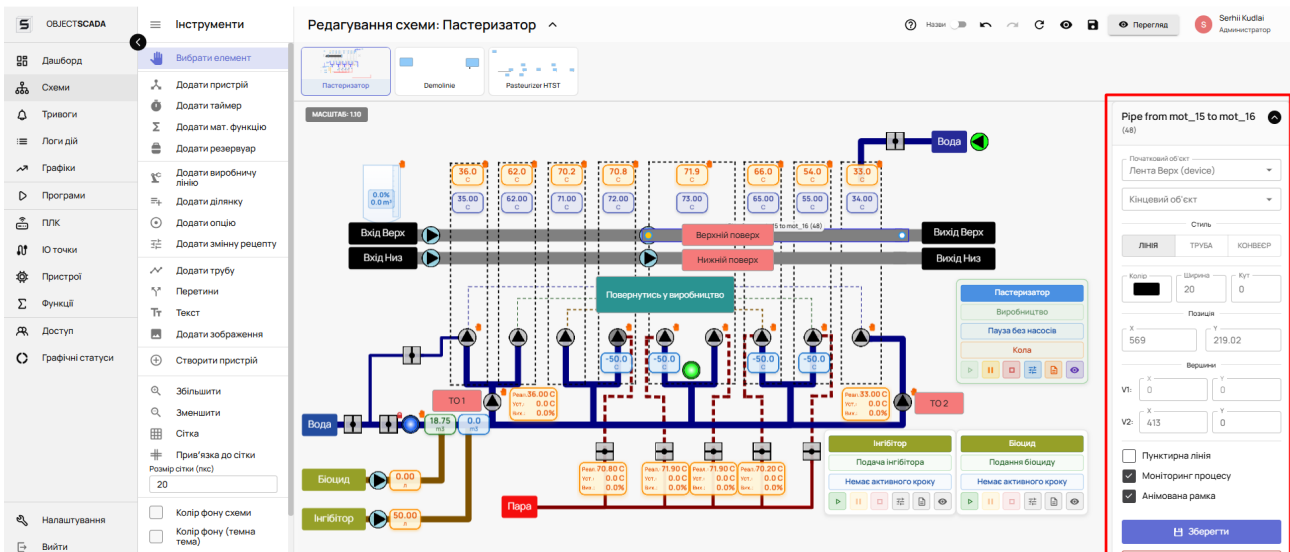
Діалог вибору пристрою: список пристроїв з фільтрацією за типом (Motor, Valve, AI, AO, DI, DO, PID, Counter, Timer, Tank, COS), поле пошуку, кнопка «Додати».



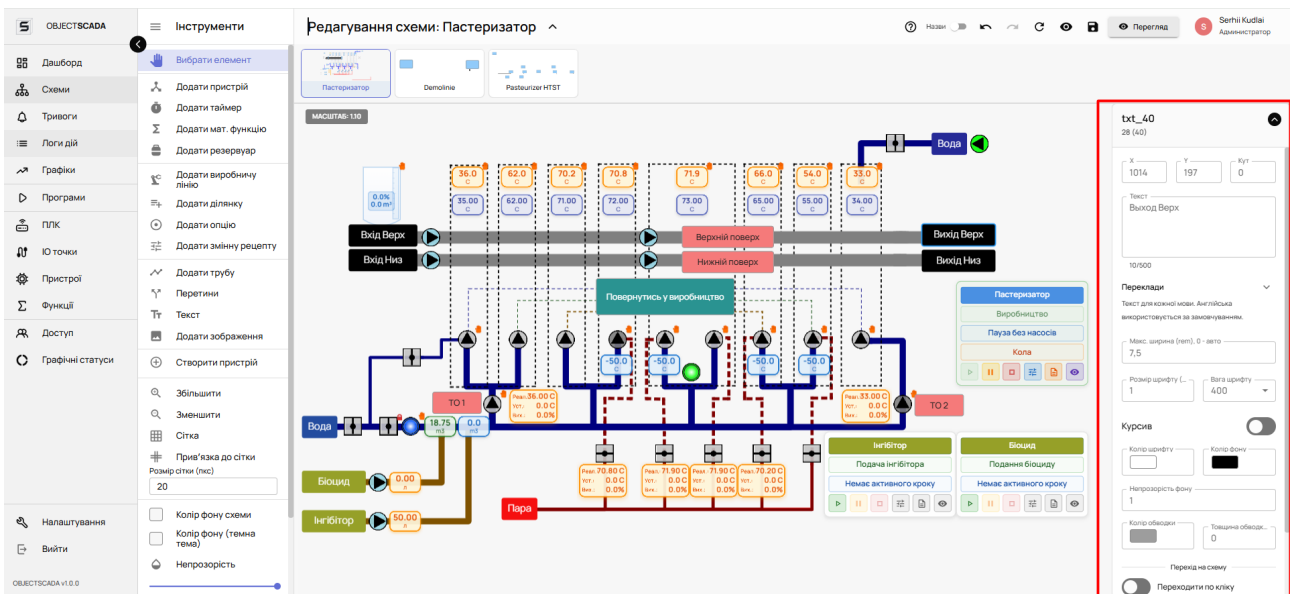
Конструктор танка: випадні списки типу верхньої та нижньої кришки (FLAT, DOME, DISH, CONE), поля ширини/висоти, SVG-прев'ю форми танка, вибір пристрою танка.



Права панель властивостей при виборі пристрою: поля Position X/Y, Angle, Width/Height, Color picker, перемикачі Mirror/Show Name.



Права панель властивостей при виборі труби: поле Name, Color picker, Thickness slider, Show Name toggle, Process Monitoring toggle.



Права панель властивостей при виборі тексту: поле Text, Font Size, Color, Bold/Italic toggles, блок Translations з полями для кожної мови.

Технічна документація

API елементів схеми:

Метод	Endpoint	Опис
POST	/api/schemas/{id}/add_motor/	Додати мотор
POST	/api/schemas/{id}/add_valve/	Додати клапан
POST	/api/schemas/{id}/add_analog_input/	Додати AI
POST	/api/schemas/{id}/add_analog_output/	Додати АО
POST	/api/schemas/{id}/add_digital_input/	Додати DI
POST	/api/schemas/{id}/add_digital_output/	Додати DO
POST	/api/schemas/{id}/add_pid_controller/	Додати PID
POST	/api/schemas/{id}/add_counter/	Додати лічильник

Метод	Endpoint	Опис
POST	/api/schemas/{id}/add_variable_speed_drive/	Додати ЧП
POST	/api/schemas/{id}/add_sequence/	Додати послідовність
POST	/api/schemas/{id}/add_option/	Додати опцію
POST	/api/schemas/{id}/add_production_line/	Додати лінію
POST	/api/schemas/{id}/add_recipe_variable/	Додати змінну
POST	/api/schemas/{id}/add-math-function/	Додати мат. функцію
POST	/api/schemas/{id}/bulk_add_schema_devices/	Масове додавання
PATCH	/api/schemas/{id}/update_positions/	Масове оновлення позицій
PATCH	/api/schemas/{id}/devices/{deviceId}/	Оновити елемент
DELET	/api/schemas/{id}/remove_device/	Видалити елемент

E

Тіло запиту додавання пристрою:

```
{
  "device_id": "mot_001",
  "content_type": "motor",
  "position_x": 100,
  "position_y": 200,
  "angle": 0,
  "configuration": {
    "color": "#FF0000",
    "mirror_x": false,
    "mirror_y": false,
    "show_name": true,
    "tank_visual": {
      "bottom_type": "DOME",
      "top_type": "FLAT",
      "width": 60,
      "height": 100
    }
  }
}
```

API трубопроводів:

Метод	Endpoint	Опис
GET	/api/pipes/?schema_id={id}	Усі труби схеми
POST	/api/pipes/	Створити трубу
PATCH	/api/pipes/{id}/	Оновити трубу
DELETE	/api/pipes/{id}/	Видалити трубу

5. Перегляд мнемосхем (Schema Viewer)

Для користувача

Режим перегляду надає живу візуалізацію процесу з оновленням даних у реальному часі.

Можливості:

Функція	Опис
Живий статус пристроїв	Кольори та анімації оновлюються в реальному часі
Натискання на пристрій	Відкриває рорир з керуванням та інформацією
Масштабування	Колесо миші або кнопки «+» / «-»
Панорамування	Затискання правої кнопки миші + переміщення
Смуга аварій	Зверху/знизу екрана — поточні активні аварії
Перемикач імен	Показати/приховати підписи пристроїв

Рорир керування пристроєм

При натисканні на пристрій відкривається спливне вікно з вкладками:

Вкладка «Керування» (Control):

Для кожного типу пристрою:

Тип	Елементи керування
Мотор (Motor)	Кнопки Пуск/Стоп, перемикач Авто/Ручний, керування частотою
Клапан (Valve)	Кнопки Відкрити/Закрити, Авто/Ручний, %, витрата
Аналоговий вхід (AI)	Авто/Ручний, введення значення, індикатор шкали
Аналоговий вихід (AO)	Авто/Ручний, введення значення, Min/Max
Дискретний вхід (DI)	Авто/Ручний, індикатор ON/OFF, блокування
Дискретний вихід (DO)	Авто/Ручний, кнопки ON/OFF, блокування
PID-регулятор	Уставка, Авто/Ручний, P/I/D параметри, похибка та вихід
Таймер	Пуск/Стоп/Скидання, встановлення тривалості
Лічильник	Скидання, поточне значення, встановлення значення
Танк (Tank)	Рівень заповнення, об'єм, дані сенсорів

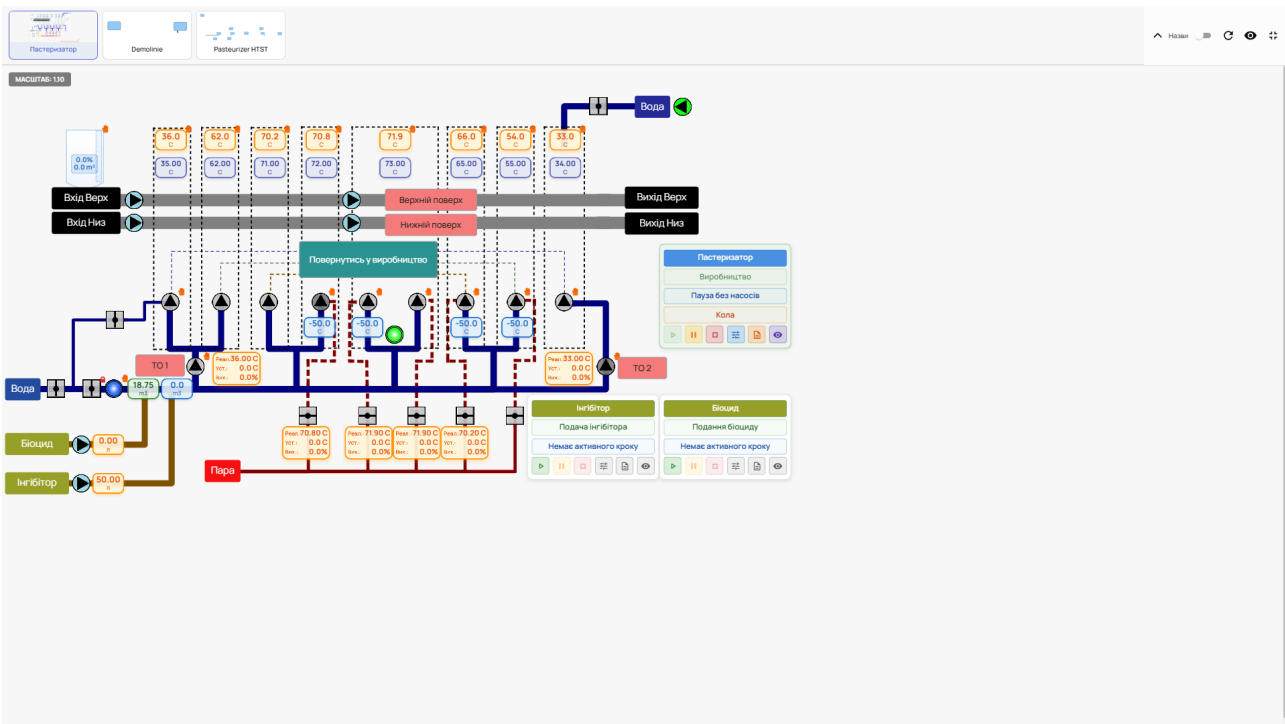
Вкладка «Статус» (Status): Поточний стан у реальному часі.

Вкладка «Таблиця» (Table): Табличне представлення параметрів.

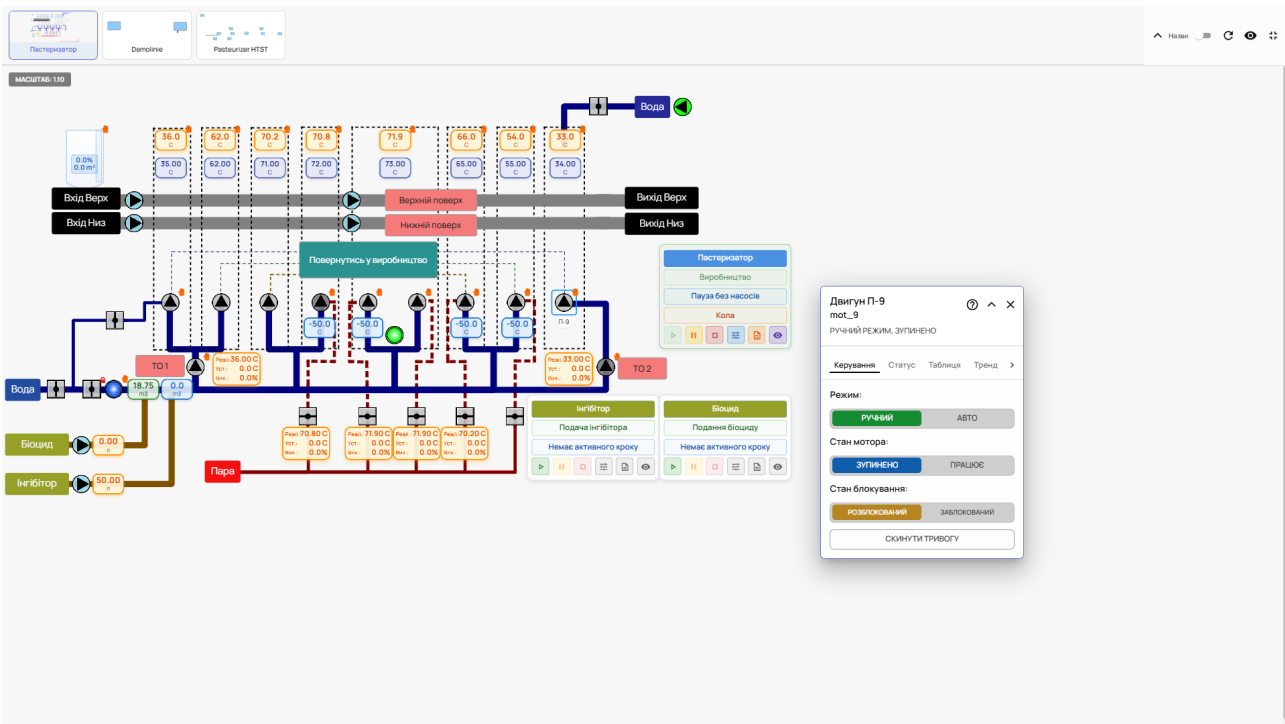
Вкладка «Тренд» (Trend): Мініатюрний графік історії значень.

Вкладка «Логи» (Logs): Журнал подій пристрою.

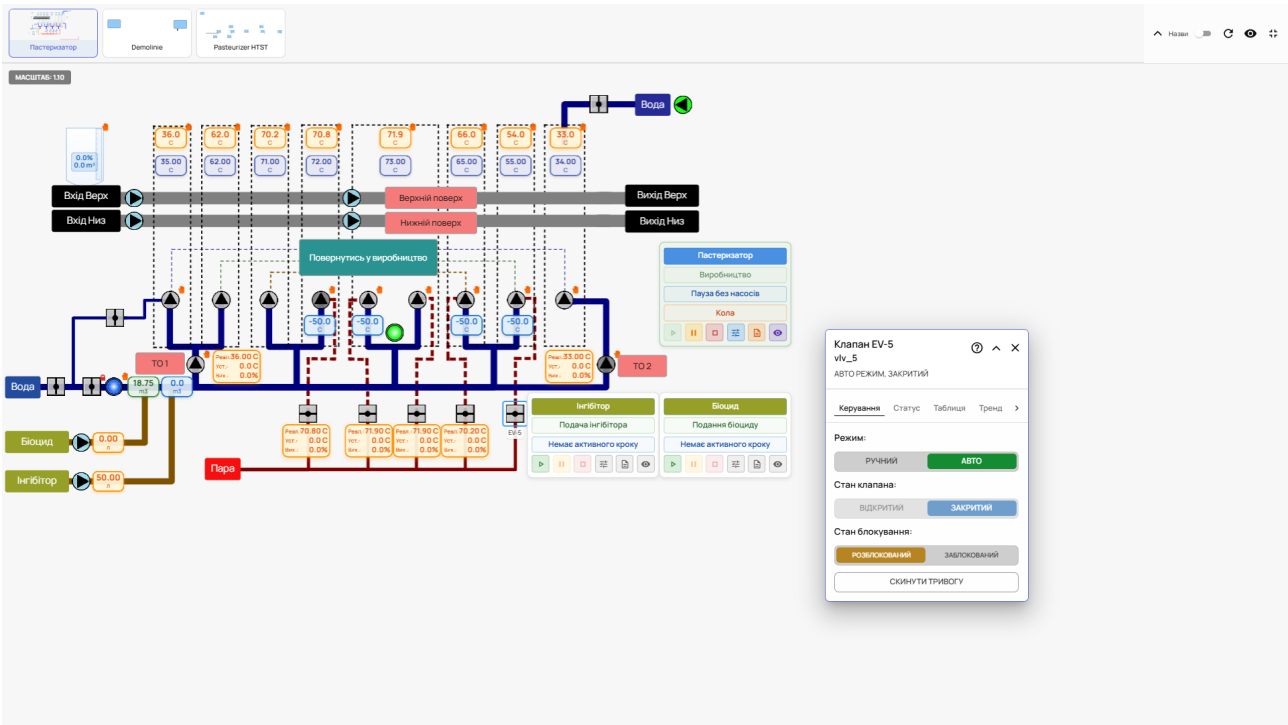
Вкладка «Умови переходу» (Transition Conditions): Активні умови послідовності (якщо пристрій бере участь).



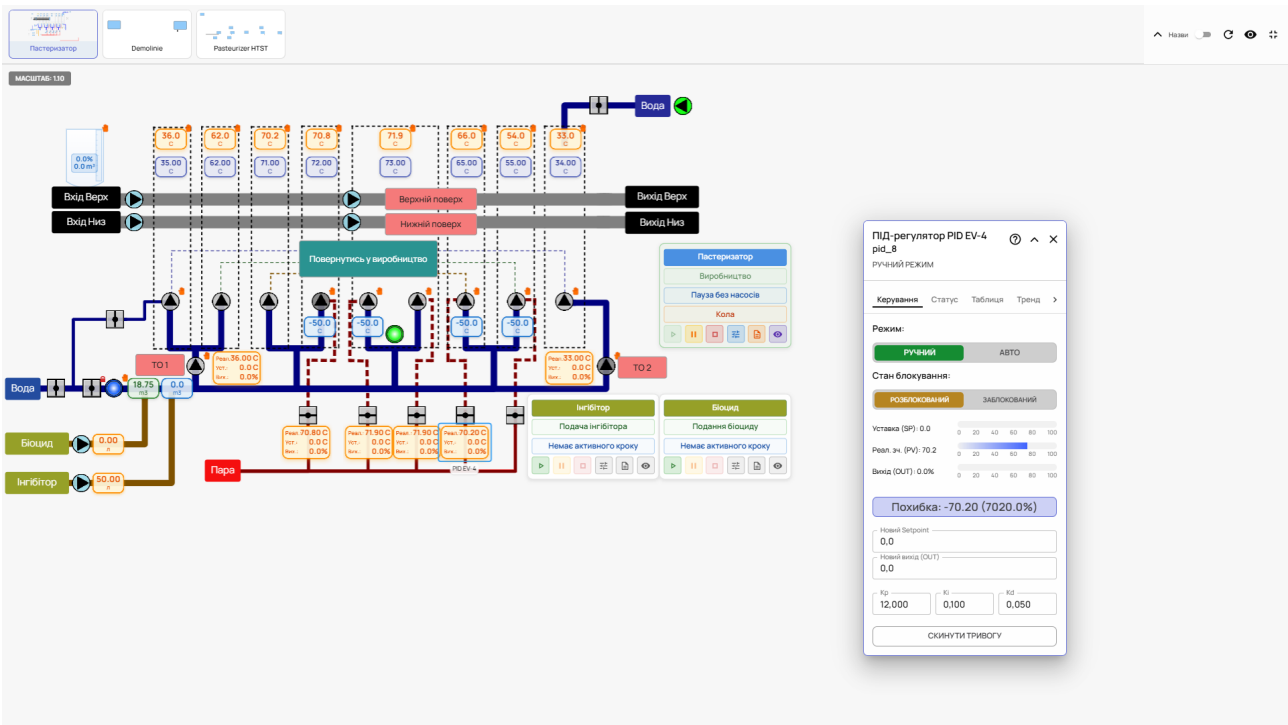
Повний екран перегляду схеми: кольорові пристрої з анімацією потоку по трубах, блок послідовності з поточним кроком, смуга аварій зверху.



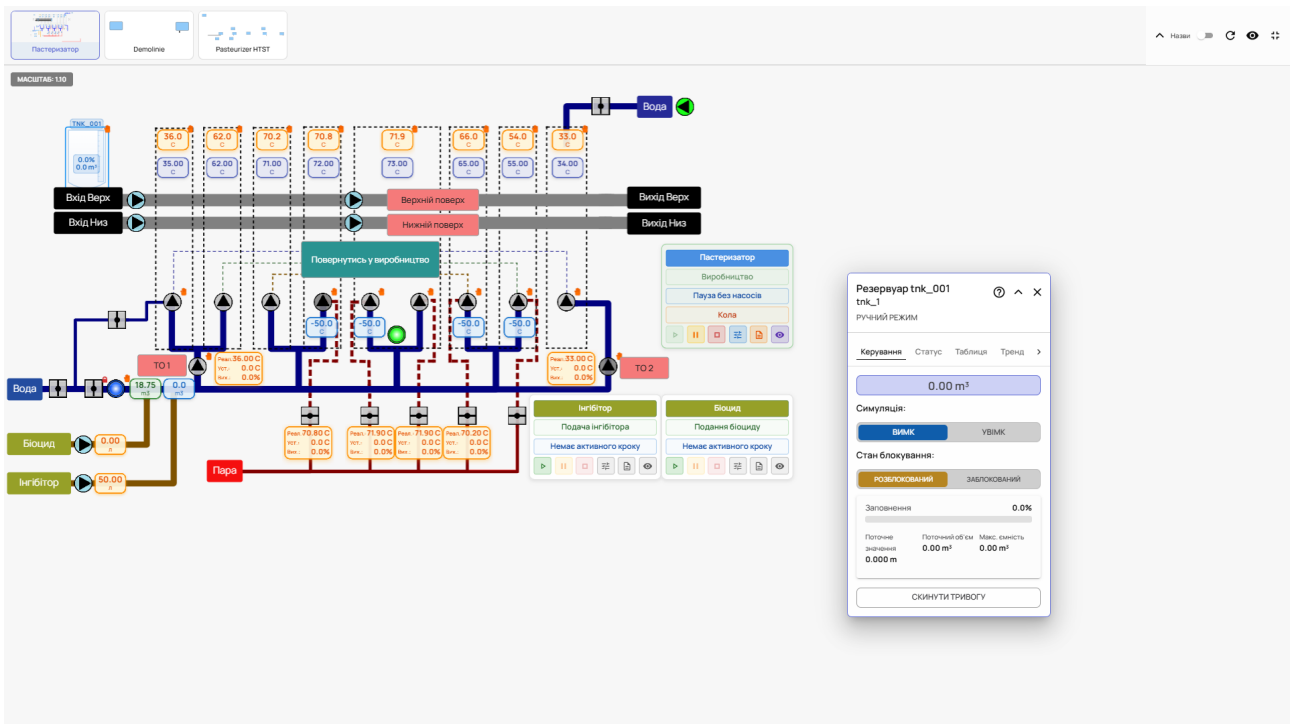
Рорир мотора: вкладка Control з кнопками Start/Stop, перемикач Auto/Manual, частота. Зелений індикатор — мотор працює.



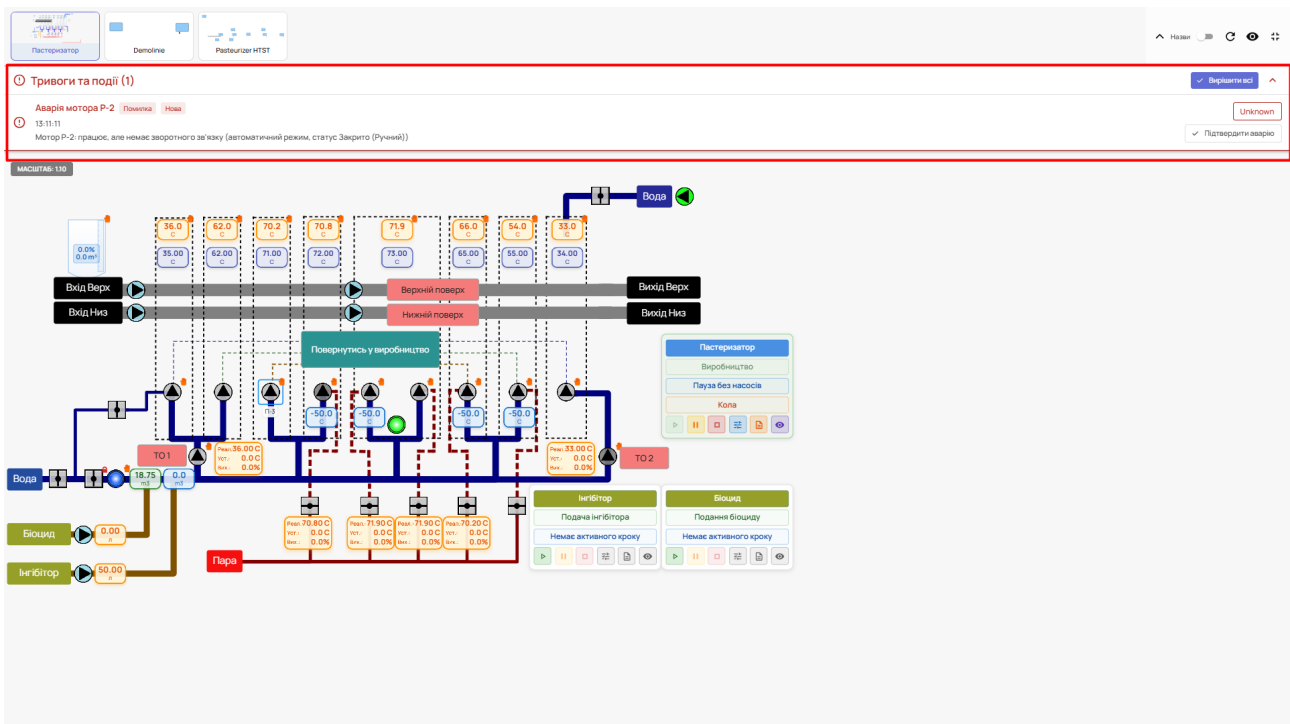
Рорир клапана: кнопки Open/Close, слайдер відсотка відкриття, режим Auto/Manual.



Рорир PID: поля Setpoint, P, I, D, індикатор Error і Output, перемикач Auto/Manual.



Рорир танка: візуалізація рівня заповнення, таблиця сенсорів (рівень, тиск), об'єм.



Смуга аварій зверху екрана: червоні/жовті повідомлення з timestamp, серйозністю та ім'ям пристрою.

Технічна документація

WebSocket підключення для перегляду схеми:

WebSocket шлях	Опис
ws/schema/{schemaid}/devices	Усі пристрої на схемі (реалтайм)
ws/schema/{schemaid}/pipes	Усі трубопроводи (анімація потоку)

WebSocket шлях	Опис
ws/schema/{schemald}/sequences	Усі послідовності на схемі
ws/schema/{schemald}/batches	Партії та виробничі лінії
ws/schema/{schemald}/accidents	Аварії для даної схеми

Формат події оновлення пристроїв:

```
{
  "type": "schema_devices_status",
  "devices": {
    "mot_001": {
      "device": "mot_001",
      "current_status": {
        "value": 1,
        "is_running": true,
        "mode": "auto",
        "frequency": 50.0,
        "interlock": false
      },
      "status_icon": "/media/icons/motor_running.svg",
      "position_x": 100,
      "position_y": 200,
      "angle": 0,
      "updated_at": "2026-04-07T12:00:00Z"
    }
  }
}
```

Дельта-оновлення (лише змінені поля):

```
{
  "type": "device_status_update",
  "device": "mot_001",
  "delta": true,
  "changes": {
    "frequency": 45.0
  }
}
```

API керування пристроєм:

Метод	Endpoint	Опис	Тіло
POST	/api/{device-type}/{id}/set_mode/	Авто/Ручний	{"mode": "manual"}
POST	/api/{device-type}/{id}/set_manual_value/	Встановити значення	{"value": 50.0}
POST	/api/{device-type}/{id}/set_simulation/	Симуляція	{"simulation": true}
POST	/api/{device-type}/{id}/set_interlock/	Блокування	{"interlock": true}
POST	/api/{device-type}/{id}/update_status/	Оновити статус	{"current_status": {...}}

WebSocket команди (через фронтенд websocketService):

```
{
  "type": "device_command",
  "device": "mot_001",
  "command": "start",
  "params": {},
  "options": {}
}
```

Відповідь:

```
{
  "type": "command_response",
  "device": "mot_001",
  "command": "start",
  "status": "success",
  "data": {...},
  "timestamp": "2026-04-07T12:00:00Z"
}
```

6. Пристрої

Для користувача

Сторінка керування пристроями дозволяє створювати, редагувати та видаляти всі типи пристроїв.

Типи пристроїв:

Тип	Код	Опис
Клапан	VLV	Запірна/регульовальна арматура
Мотор	MOT	Електродвигуни
Аналоговий вхід	AI	Аналогові датчики (температура, тиск, витрата)
Аналоговий вихід	AO	Аналогові виходи (4-20 mA, 0-10V)
Дискретний вхід	DI	Сигнали увімк/вимк (кінцеві, готовності)
Дискретний вихід	DO	Дискретні команди
Координаційний сигнал	COS	Міжсистемні координаційні сигнали
PID-регулятор	PID	Конттури автоматичного регулювання
Лічильник	CNT	Лічильники імпульсів
Танк	TNK	Ємності з розрахунком об'єму
Таймер	TMR	Програмні таймери

Елементи екрана:

Елемент	Опис
Перемикач вигляду	Сітка / Таблиця
Кнопка «+»	Створення нового пристрою
Фільтр за типом	Випадний список типів пристроїв
Пошук	Пошук за назвою
Оновити	Перезавантаження списку

Форма пристрою (створення/редагування):

Поле	Опис
ID пристрою	Унікальний ідентифікатор (наприклад, mot_001)
Назва	Людиночитана назва
Переклади назв	Багатомовні назви
Тип	Вибір типу пристрою
Опис	Текстовий опис
Значення за замовчуванням	Початкове значення
Одиниця вимірювання	Інженерна одиниця
Min / Max	Діапазон допустимих значень
Масштабування	Параметри перетворення сигналу
Кнопка «Зберегти»	Збереження
Кнопка «Скасувати»	Повернення без збереження

Поле	Опис
Кнопка «Видалити»	Видалення пристрою (у режимі редагування)

SCADA Пристрої

Тип пристрою: ПЛК

Назва: Пошук за назвою

ID	Назва	Тип	IP	Статус	Місце активації	Дії
viv_1	EV-1	Клапан	Не вказано	Норма	Пастеризатор → Увімкнення клапана EV-1 [Відкриття]	✎ ✕
viv_10	EV-10	Клапан	Не вказано	Норма	Тест → Етап 1 [Відкриття]	✎ ✕
viv_11	EV-11	Клапан	Не вказано	Норма	Пастеризатор → Набір води [Відкриття]	✎ ✕
viv_12	EV-12	Клапан	Не вказано	Норма	—	✎ ✕
viv_2	EV-2	Клапан	Не вказано	Норма	Пастеризатор → Увімкнення клапана EV-2 [Відкриття]	✎ ✕
viv_3	EV-3	Клапан	Не вказано	Норма	Пастеризатор → Увімкнення клапана EV-3 [Відкриття]	✎ ✕
viv_4	EV-4	Клапан	Не вказано	Норма	Пастеризатор → Увімкнення клапана EV-4 [Відкриття]	✎ ✕
viv_5	EV-5	Клапан	Не вказано	Норма	Пастеризатор → Увімкнення клапана EV-5 [Відкриття]	✎ ✕
viv_6	EV-6	Клапан	Не вказано	Норма	Пастеризатор → Набір води [Відкриття] Пастеризатор → Набір води [Відкриття]	✎ ✕
viv_7	EV-7	Клапан	Не вказано	Норма	—	✎ ✕
viv_8	EV-8	Клапан	Не вказано	Норма	—	✎ ✕
viv_9	EV-9	Клапан	Не вказано	Норма	—	✎ ✕
mot1	П-1	Двигун	Не вказано	Норма	Пастеризатор → Старт насосів 1, 9 [Старт] Пастеризатор → Старт насосів 2, 8 [Старт]	✎ ✕
mot1R	П-1 П.П.	Двигун	Не вказано	Норма	Пастеризатор → Старт насосів 2, 8 [Старт]	✎ ✕

Рядків на сторінці: 25

Назад 1 2 3 4 Далі 1-25 з 85

Сторінка пристроїв у вигляді сітки: картки з іконками пристроїв (мотор, клапан тощо), назвою, типом, значенням. Фільтр за типом зверху.

ОБ'ЄКТSCADA Пристрої

Тип пристрою: ПЛК

Назва: Пошук за назвою

ID	Назва	Тип	Значення за замовчуванням	Одиниця	Дії
EV-1	EV-1	Клапан	Не вказано	Норма	✎ ✕
EV-10	EV-10	Клапан	Не вказано	Норма	✎ ✕
EV-11	EV-11	Клапан	Не вказано	Норма	✎ ✕
EV-12	EV-12	Клапан	Не вказано	Норма	✎ ✕
EV-2	EV-2	Клапан	Не вказано	Норма	✎ ✕
EV-3	EV-3	Клапан	Не вказано	Норма	✎ ✕
EV-4	EV-4	Клапан	Не вказано	Норма	✎ ✕
EV-5	EV-5	Клапан	Не вказано	Норма	✎ ✕
EV-6	EV-6	Клапан	Не вказано	Норма	✎ ✕
EV-7	EV-7	Клапан	Не вказано	Норма	✎ ✕

Рядків на сторінці: 10

Назад 1 2 3 4 5 ... 9 Далі 1-10 з 85

Табличний вигляд: колонки ID, Назва, Тип, Значення за замовчуванням, Одиниця, Дії.

S SCADA Пристрої Serhii Kudlai
Адміністратор

Створити новий пристрій

1 Основна інформація — 2 Конфігурація пристрою

Назва пристрою *

Переклади назви
Назва кожною мовою. Англійська використовується як резервна.

Тип пристрою *

ПЛК

Опис

Скасувати Далі >

SCADA System v1.0.0

Форма створення мотора: основні поля — ID, Name, PLC, Description, тип пристрою.

OBJECTSCADA Пристрої Serhii Kudlai
Адміністратор

Редагувати пристрій RT100-1 (ai_1)

1 Основна інформація — 2 Конфігурація пристрою

Точки вводу/виводу
Точка вводу/виводу
RT100-1_Ai_Value

Логування
 Логування

Масштабування

Тип даних

Мінімальне значення

Максимальне значення
150

Нижня межа RAW

Верхня межа RAW
28500

Скорість викиру
С

Коефіцієнт масштабування

Зсув

Тривоги

Загальна тривога

Тривога: значення нижче нижньої межі

Тривога: значення вище верхньої межі

Тривога: сигнал поза діапазоном

Контроль лімітів (запит ПЛК)

Контроль лімітів (конфігурація)

Попередження: високе значення

Попередження: низьке значення

Попередження: надто низьке значення

HN threshold (high-high)

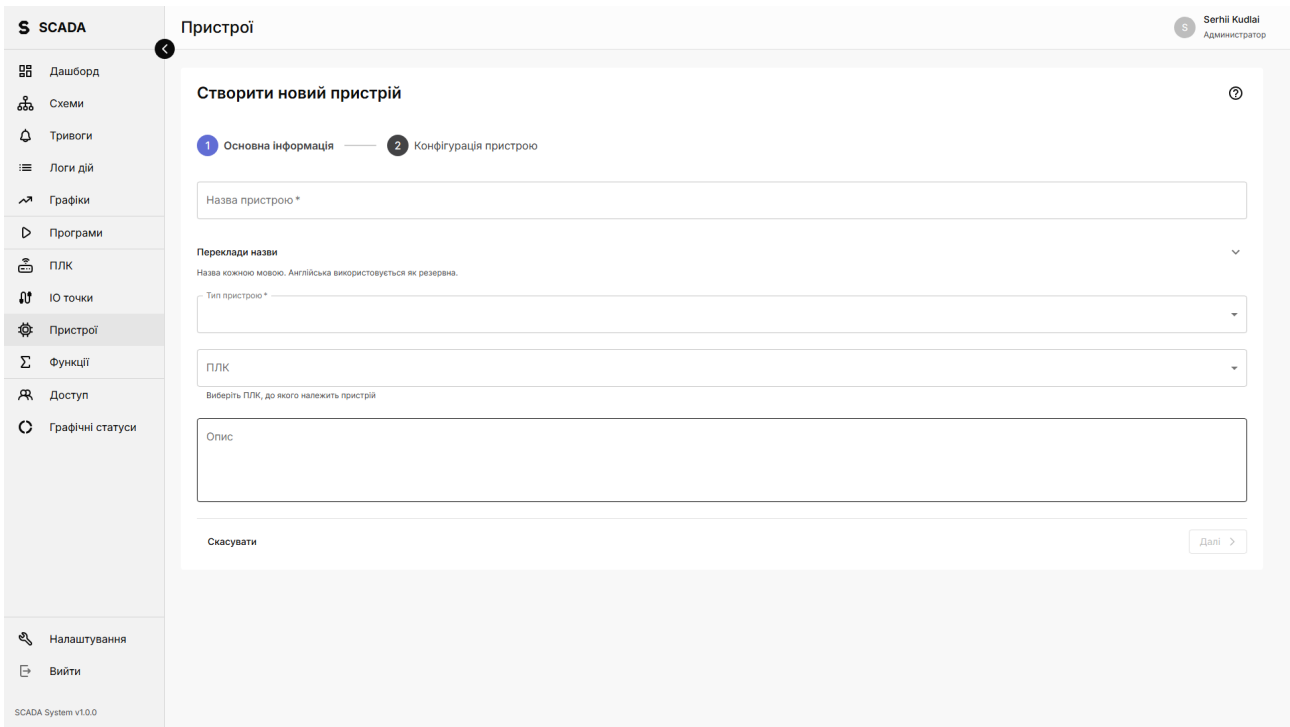
H threshold (high)

L threshold (low)

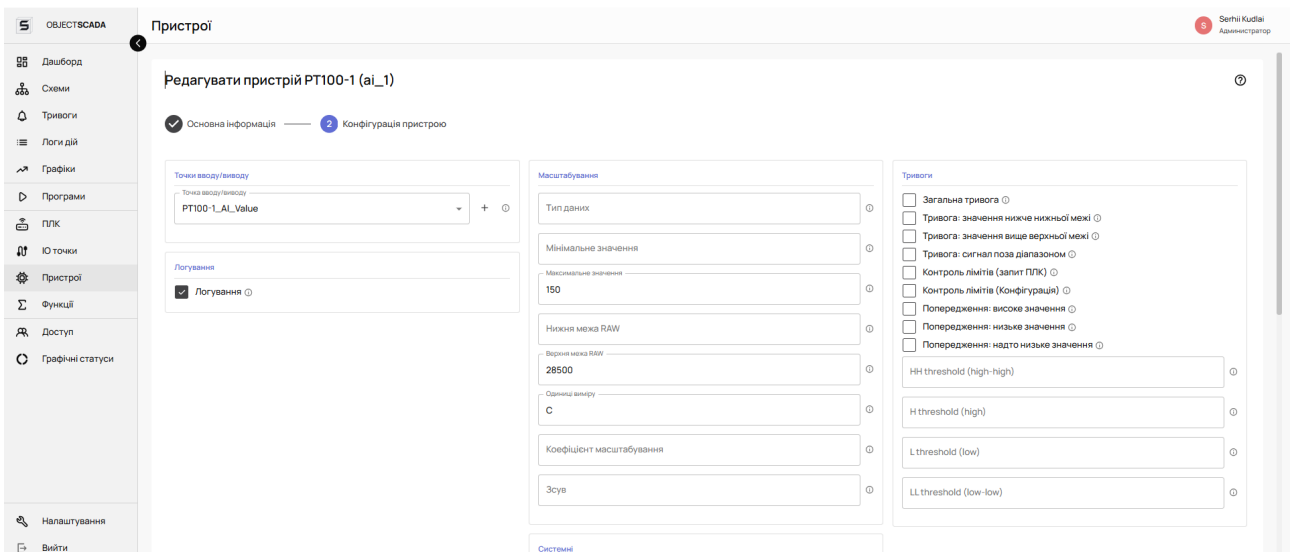
LL threshold (low-low)

Системні

Форма мотора: min/max speed, затримки, generate_alarm, кнопки збереження.



Форма аналогового входу: Min/Max, Units, Scaling, Raw Min/Max.



Форма AI: пороги попереджень, значення за замовчуванням та інші поля.

Технічна документація

API пристроїв (спільний для всіх типів):

Метод	Endpoint	Опис
GET	/api/devices/all_devices/	Усі пристрої всіх типів
GET	/api/{device-type}/	Список пристроїв типу
POST	/api/{device-type}/	Створення пристрою
GET	/api/{device-type}/{id}/	Отримання пристрою
PUT	/api/{device-type}/{id}/	Оновлення

Метод	Endpoint	Опис
PATCH	/api/{device-type}/{id}/	Часткове оновлення
DELETE	/api/{device-type}/{id}/	Видалення

Типи endpoint'ів:

- /api/analog-inputs/
- /api/analog-outputs/
- /api/digital-inputs/
- /api/digital-outputs/
- /api/motors/
- /api/valves/
- /api/pid-controllers/
- /api/counters/
- /api/timers/
- /api/coordination-signals/
- /api/variable-speed-drives/

Додаткові дії (для кожного типу):

Метод	Endpoint	Опис
POST	/{id}/set_mode/	Перемкнути Авто/Ручний
POST	/{id}/set_simulation/	Увімк/вимк симуляцію
POST	/{id}/set_interlock/	Встановити/зняти блокування
POST	/{id}/update_status/	Оновити поточний статус
POST	/{id}/set_manual_value/	Встановити ручне значення
POST	/{id}/trigger_websocket_update/	Примусово оновити WS (AllowAny)

WebSocket: Оновлення пристрою

Підключення: ws://ws/device/{device_id}

```
{
  "type": "device_status_update",
  "device": "ai_001",
  "data": {
    "current_status": {
      "value": 23.5,
      "mode": "auto",
      "simulation": false,
      "interlock": false,
      "quality": "good"
    },
    "status_icon": null,
    "updated_at": "2026-04-07T12:00:00Z"
  }
}
```

7. ПЛК

Для користувача

Сторінка керування програмованими логічними контролерами.

Елементи екрана:

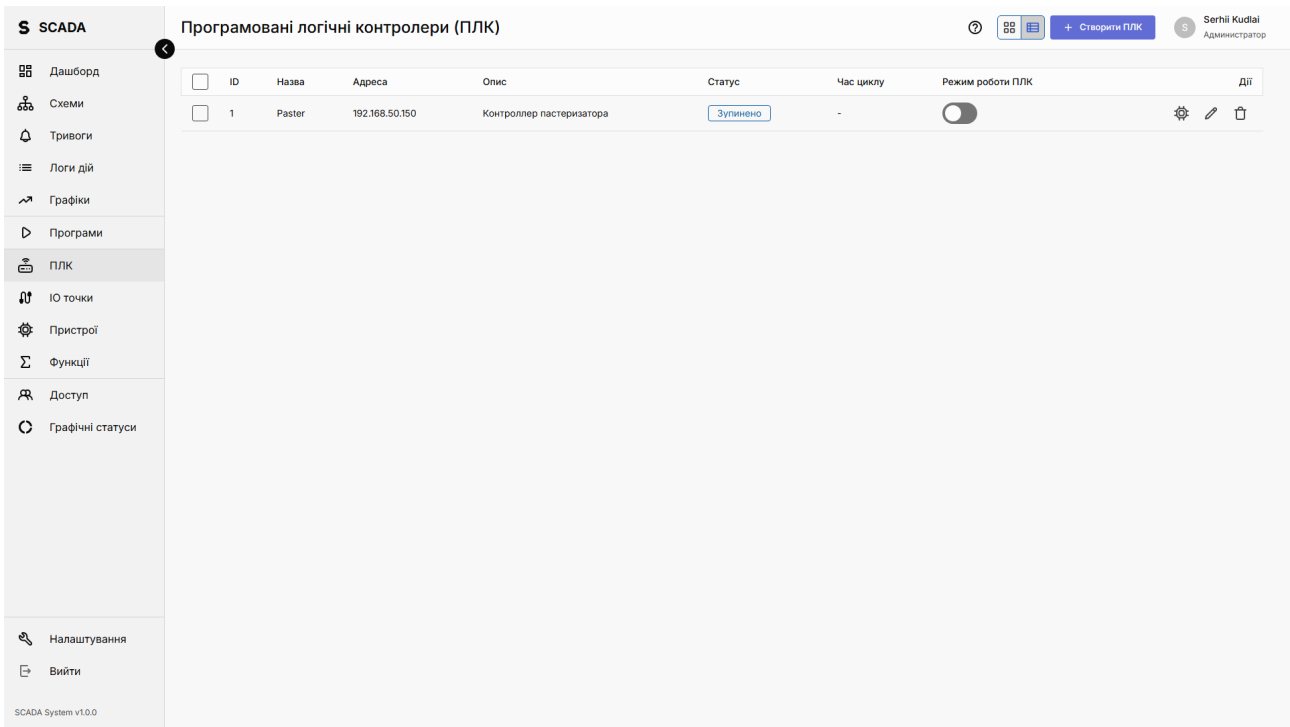
Елемент	Опис
Список ПЛК	Картки/таблиця з ПЛК
Індикатор статусу	Зелений — онлайн, сірий — офлайн, червоний — помилка
Кнопка «+»	Додати новий ПЛК
Кнопка «Hardware Config»	Перейти в конфігуратор обладнання
Перемикач Run Mode	Увімк/вимк режим виконання ПЛК

Інформація на картці ПЛК:

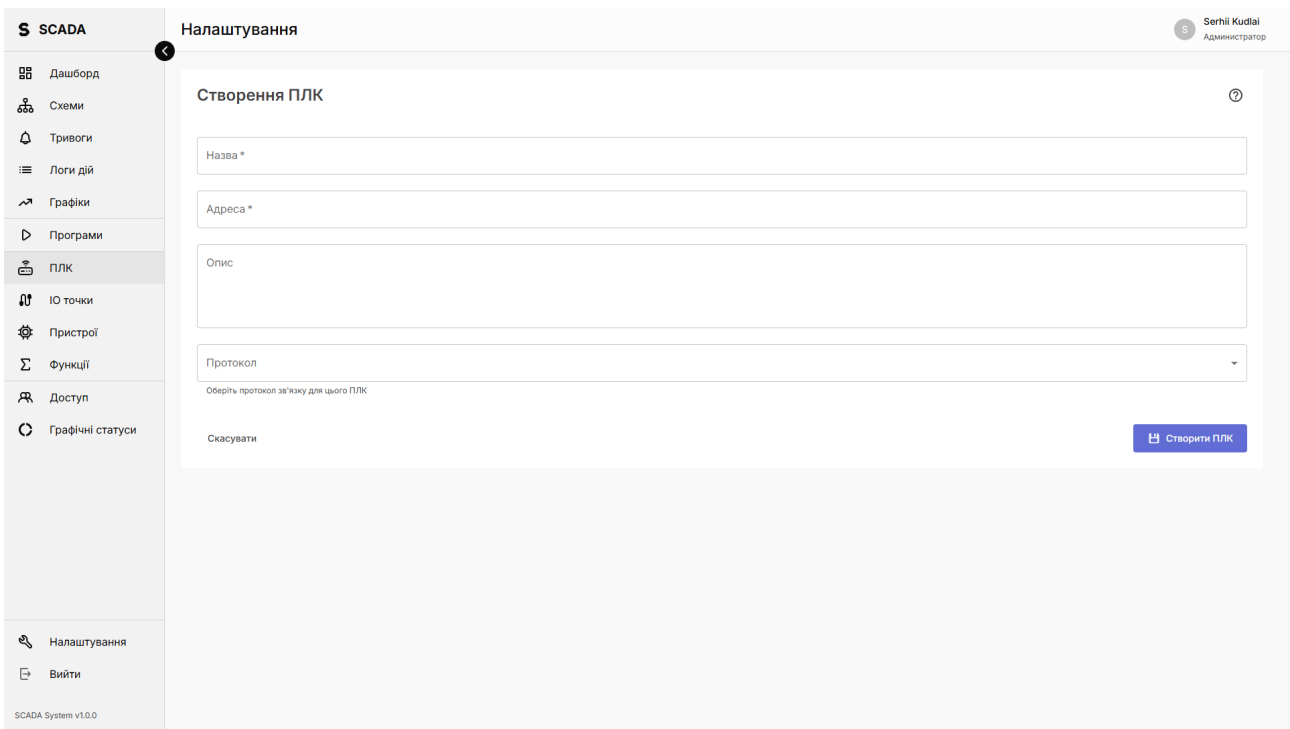
Поле	Опис
Назва	Ім'я ПЛК
Протокол	Тип зв'язку (Siemens S7, Modbus, OPC UA)
Статус	Online / Offline
Час циклу	Швидкість опитування
IP-адреса	Мережева адреса
Кількість пристроїв	К-сть прив'язаних пристроїв

Форма ПЛК:

Поле	Опис
Назва	Ім'я ПЛК
Опис	Текстовий опис
Протокол	Вибір протоколу зв'язку
IP / Порт	Мережеві налаштування
Rack / Slot	Для Siemens S7
Кнопка «Тест з'єднання»	Перевірка зв'язку з ПЛК
Кнопка «Зберегти»	Збереження конфігурації



Список ПЛК: картки з індикатором статусу (зелений/сірий), назвою, протоколом, IP-адресою, перемикачем Run Mode, кнопкою Hardware Config.



Форма налаштування ПЛК: поля Name, Protocol dropdown, IP Address, Port, Rack, Slot, кнопка Test Connection.

Технічна документація

API ПЛК:

Метод	Endpoint	Опис
GET	/api/plcs/	Список ПЛК

Метод	Endpoint	Опис
POST	/api/plcs/	Створення ПЛК
GET	/api/plcs/{id}/	Отримання ПЛК
PUT	/api/plcs/{id}/	Оновлення
DELETE	/api/plcs/{id}/	Видалення
GET	/api/plcs/{id}/status/	Статус ПЛК
POST	/api/plcs/{id}/set-run-mode/	Переключити Run Mode
GET	/api/plcs/status/	Статус усіх ПЛК

Симуляція ПЛК:

Метод	Endpoint	Опис
POST	/api/plc/simulation/run/	Запустити симуляцію
GET	/api/plc/simulation/status/	Статус симуляції
POST	/api/plc/service/start/	Запустити сервіс ПЛК
POST	/api/plc/service/stop/	Зупинити сервіс ПЛК

8. Послідовності

Для користувача

Послідовності визначають порядок роботи виробничого обладнання — кроки, переходи між ними та умови перемикання.

Елементи екрана:

Елемент	Опис
Список послідовностей	Картки/таблиця з фільтрацією
Кнопка «+»	Створити нову послідовність
Індикатор статусу	Зелений — працює, сірий — зупинено, жовтий — пауза
Фільтр за статусом	Усі / Працюючі / Зупинені
Пошук	Пошук за ім'ям

Редактор послідовності (вкладки):

Вкладка	Опис
Властивості	Загальні налаштування (ім'я, опис, прив'язка пристроїв)
Кроки	Список кроків з порядком виконання
Умови переходу	Налаштування умов переходу між кроками
Події	Обробка подій (Events)
Аварії	Обробка аварій (Accidents)
Опції	Налаштування опцій
Змінні рецепта	Визначення змінних
Виробничі рецепти	Рецепти для даної послідовності
Імпульсні модулі	Керування імпульсними модулями
Партії	Керування партіями (Batches)

Налаштування кроку:

Поле	Опис
Порядковий номер	Номер кроку в послідовності
Назва	Ім'я кроку (+ переклади)
Опис	Текстовий опис
Пристрої	Прив'язані пристрої та їх значення на цьому кроці
Наступні кроки	JSON-визначення переходів: {"target_step_order": [condition_id, ...]}
Дія з партією	NONE / TRANSFER / COMPLETE

Умова переходу (Transition Condition):

Поле	Опис
Тип	Device (за значенням пристрою), Timer (за часом), Option (за опцією)
Пристрій	Який пристрій відстежувати (для Device)

Поле	Опис
Оператор	==, !=, >, =, <=
Значення	Цільове значення для порівняння
Тривалість	Час (для Timer)
Симуляція	Увімк/вимк симуляцію умови (для тестування)

Програми

Програми (4)

ID ↓	Назва	Ділянка	Створено	Опис	Дії
4	Production Pasteurization HTST	Pasteurization HTST	12.05.2026, 17:56:24	-	[Edit] [Copy] [Delete]
3	Подання біоциду	Біоцид	16.12.2025	-	[Edit] [Copy] [Delete]
2	Подача інгібітора	Інгібітор	16.12.2025	-	[Edit] [Copy] [Delete]
1	Виробництво	Пастеризатор	19.11.2025	-	[Edit] [Copy] [Delete]

Список послідовностей: картки з назвою, індикатором статусу (зелений/сірий), поточним кроком, кнопками Edit/Delete.

Програми

Редагування програми Виробництво (1)

Кроки програми

1. Перед стартом
2. Набір води 1 Подай 1 Аварія
3. Старт насосів 1, 9 1 Подай
4. Старт насосів 2, 8 1 Подай
5. Старт насосів 3, 7 1 Подай
6. Старт насоса 4 1 Подай
7. Старт насосів 5, 6 1 Подай
8. Нагрів 8 Подай 5 Аварія
9. Виробництво 10 Подай 10 Аварія
10. Пауза із насосами 8 Подай
11. Пауза без насосів
12. Підготовка до закінчення 1 Подай
13. Кінець

Основна інформація

Назва кроку

Переклади назви

Опис кроку

Стани пристроїв в роботі

Клапани (12)

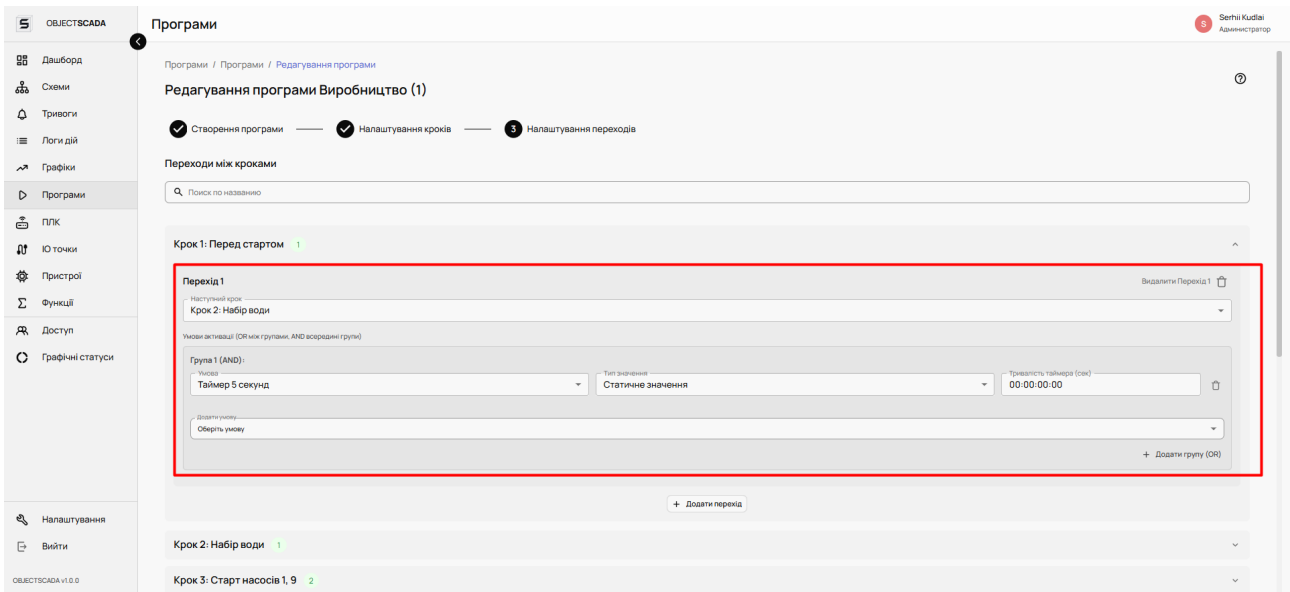
Моторнінасоси (19)

ПІД-регулятори (7)

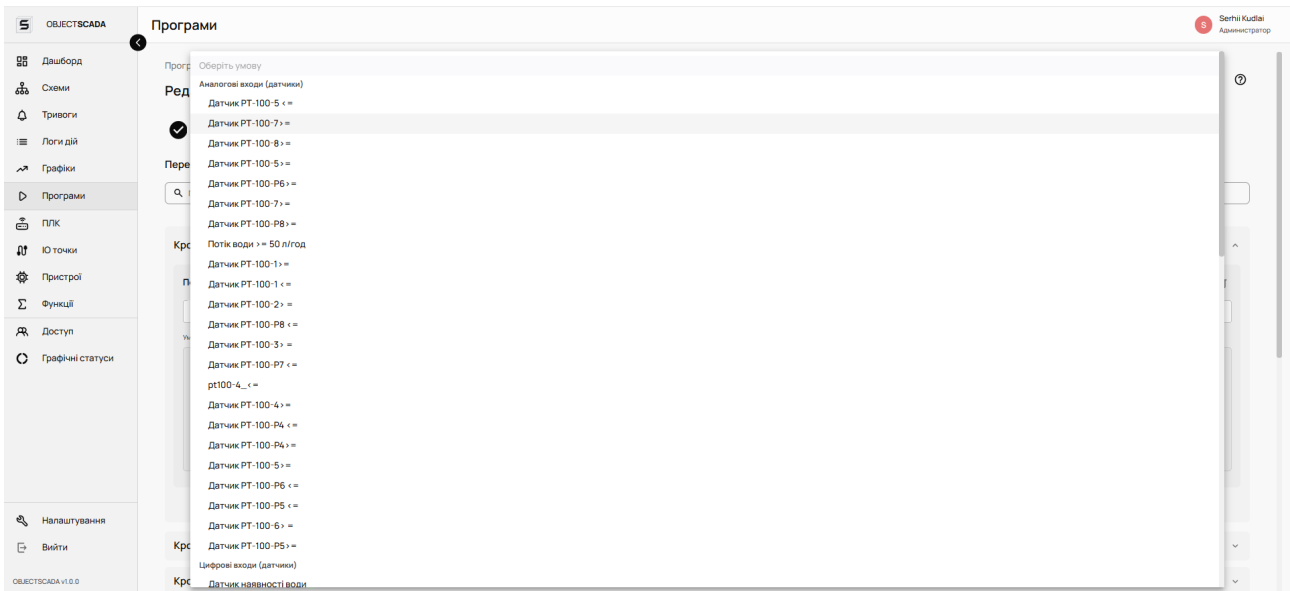
Ущільнювачі (1)

Координаційні сигнали (2)

Редактор кроків: нумерований список кроків, для кожного — назва, прив'язані пристрої, стрілки переходів, кнопки Add/Delete Step.



Налаштування умов переходу: таблиця з колонками Type, Device, Operator, Value, Duration, Simulation toggle.



Діалог створення умови переходу: вибір типу (Device/Timer/Option), вибір пристрою з dropdown, оператор порівняння, поле значення.

Технічна документація

API послідовностей:

Метод	Endpoint	Опис
GET	/api/sequences/	Список послідовностей
POST	/api/sequences/	Створення
GET	/api/sequences/{id}/	Отримання
PUT	/api/sequences/{id}/	Оновлення
DELET	/api/sequences/{id}/	Видалення
GET	/api/sequences/{id}/steps/	Список кроків
POST	/api/sequences/{id}/add_step/	Додати крок
POST	/api/sequences/{id}/set_steps_order/	Змінити порядок кроків

Метод	Endpoint	Опис
POST	/api/sequences/{id}/add_transition_conditions/	Додати умови

API кроків:

Метод	Endpoint	Опис
GET	/api/sequence-steps/	Усі кроки
POST	/api/sequence-steps/	Створити крок
PUT	/api/sequence-steps/{id}/	Оновити
DELETE	/api/sequence-steps/{id}/	Видалити

API умов переходу:

Метод	Endpoint	Опис
GET	/api/transition-conditions/	Усі умови
POST	/api/transition-conditions/	Створити
PUT	/api/transition-conditions/{id}/	Оновити
DELET	/api/transition-conditions/{id}/	Видалити
POST	/api/transition-conditions/{id}/simulate/	Симуляція умови
POST	/api/transition-conditions/clear_simulation/	Скидання симуляції кроку

WebSocket: Статус послідовності

Підключення: ws://ws/sequence/{sequence_id}

```
{
  "type": "sequence_status",
  "id": 1,
  "name": "Production A",
  "is_running": true,
  "is_pause": false,
  "is_stop": false,
  "active_step": 3,
  "active_step_data": {
    "id": 3,
    "order": 3,
    "name": "Heating",
    "name_translations": {"uk": "Нагрів"},
    "description": "..."
  },
  "previous_step": 2,
  "current_step_start_time": "2026-04-07T11:30:00Z",
  "monitored_devices": {
    "ai_001": {"value": 72.3, "mode": "auto"}
  },
  "activated_devices": {
    "mot_001": {"value": 1, "is_running": true}
  },
  "conditions": [
    {
      "id": 10,
      "type": "device",
      "device_id": "ai_001",
      "operator": ">=",
      "value": 80.0,
    }
  ]
}
```

... (+9 строк)

9. Рецепти та змінні

Для користувача

Рецепти визначають конкретні значення параметрів пристроїв на кожному кроці послідовності. Змінні рецепта дозволяють параметризувати ці значення.

Типи рецептів:

Тип	Опис
Recipe (Рецепт кроків)	Визначає значення пристроїв для кожного кроку послідовності
Production Recipe (Виробничий рецепт)	Набір змінних для конкретного виробничого завдання

Форма рецепта кроків:

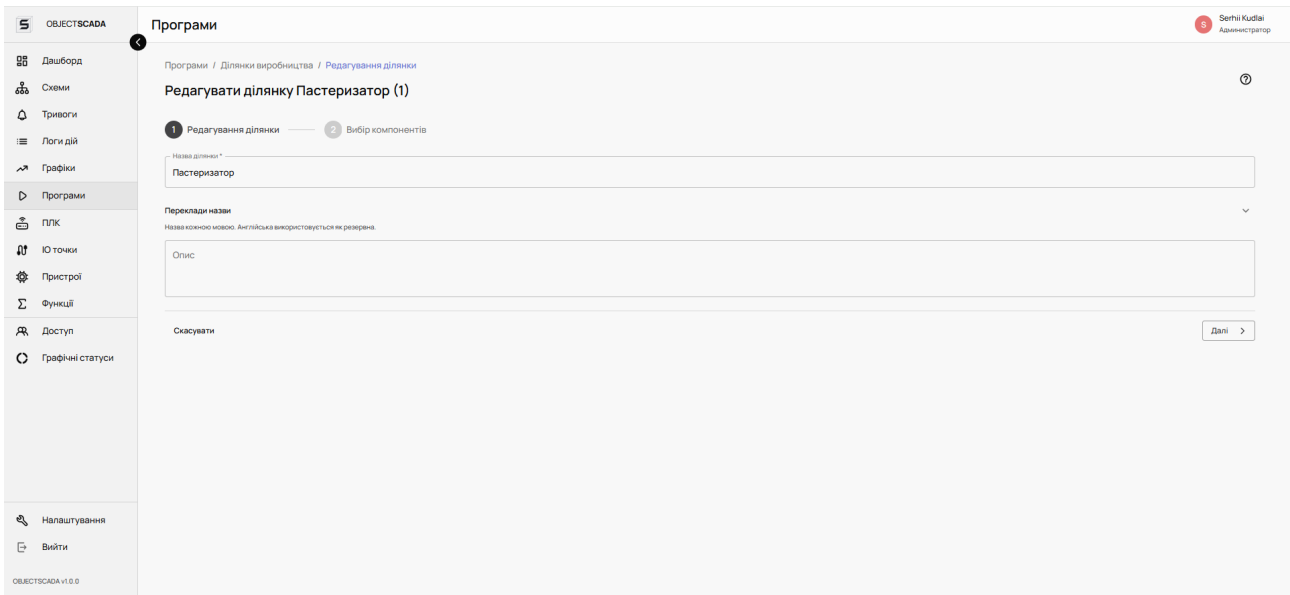
Поле	Опис
Назва	Ім'я рецепта (+ переклади)
Послідовність	Прив'язка до послідовності
Кроки	Для кожного кроку — набір «пристрій → значення»

Форма виробничого рецепта:

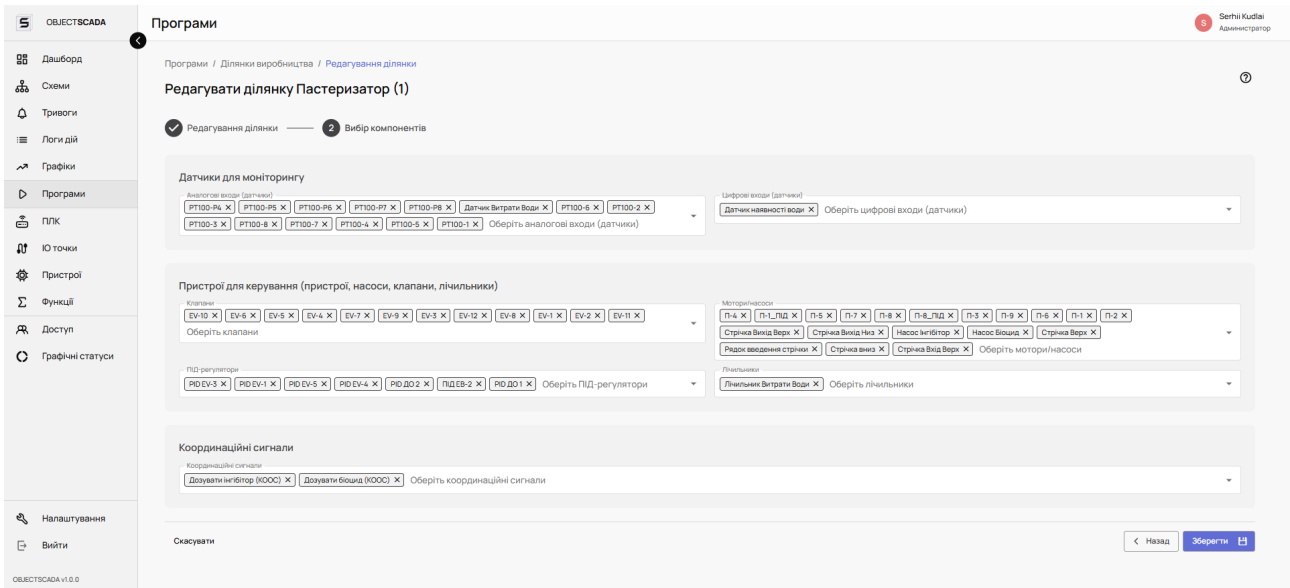
Поле	Опис
Назва	Ім'я рецепта
Прив'язка	До послідовності АБО до виробничої лінії (XOR)
Змінні	Список змінних зі значеннями

Змінні рецепта:

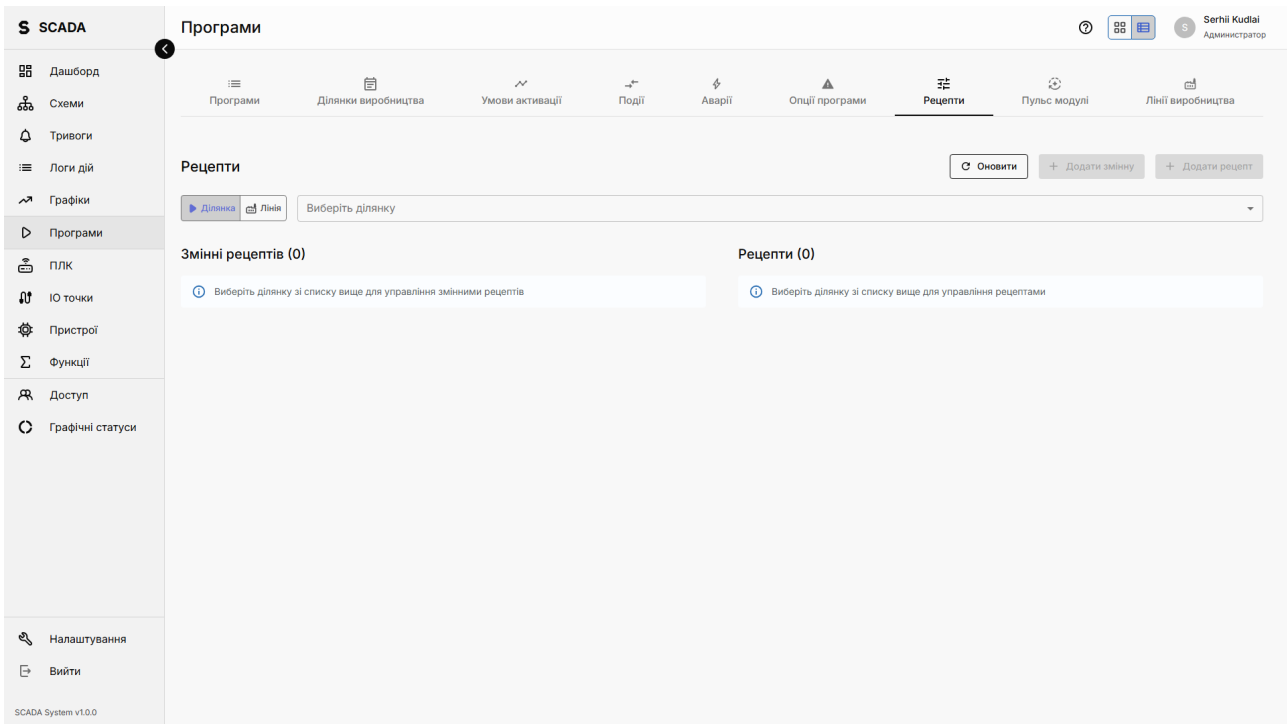
Поле	Опис
Назва	Ім'я змінної (+ переклади)
Прив'язка	До послідовності АБО до виробничої лінії (XOR)
Тип	Числовий, текстовий
Значення за замовчуванням	Початкове значення
Min / Max	Обмеження



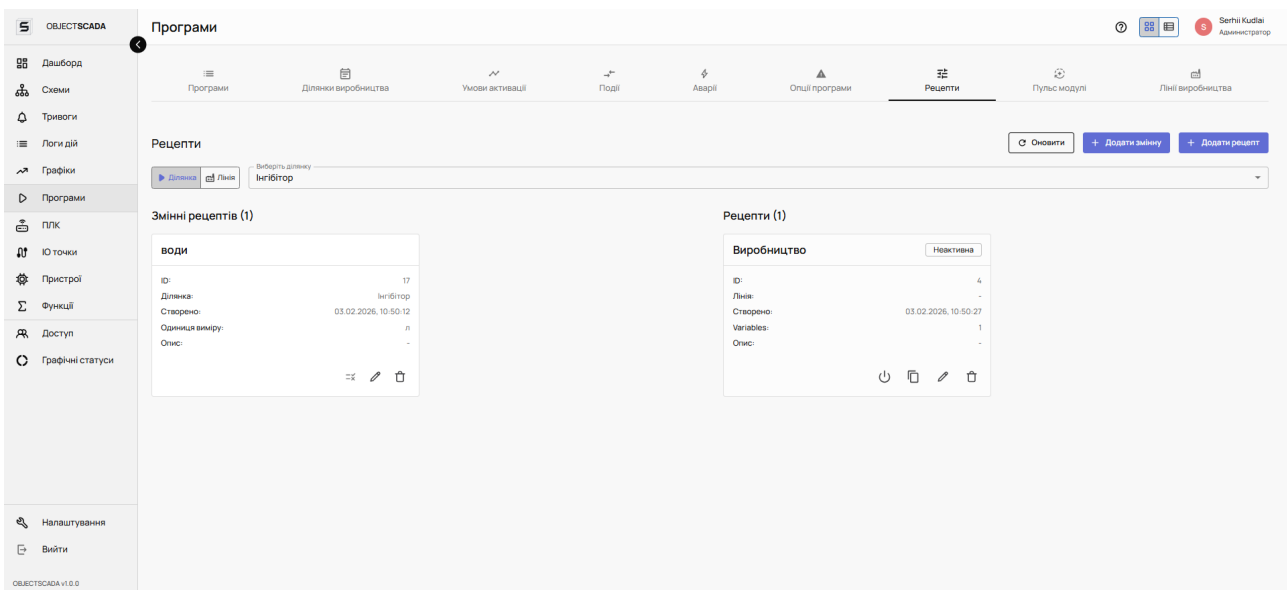
Майстер створення рецепта — крок 1: назва та прив'язка до послідовності.



Майстер створення рецепта — крок 2: таблиця «крок → пристрій → значення».



Керування виробничими рецептами: таблиця з рецептами, кнопки Create/Edit/Delete/Duplicate, перемикач Sequence/Production Line.



Керування змінними: список змінних з ім'ям, типом, значенням за замовчуванням, Min/Max. Кнопки додавання/видалення.

Технічна документація

API рецептів:

Метод	Endpoint	Опис
GET	/api/recipes/	Список рецептів кроків
POST	/api/recipes/	Створити рецепт
PUT	/api/recipes/{id}/	Оновити
DELETE	/api/recipes/{id}/	Видалити

API виробничих рецептів:

Метод	Endpoint	Опис	Параметри
GET	/api/production-recipes/	Усі рецепти	?sequence=ID або ?production_line=ID
POST	/api/production-recipes/	Створити	—
PUT	/api/production-recipes/{id}/	Оновити	—
DELETE	/api/production-recipes/{id}/	Видалити	—
POST	/api/production-recipes/{id}/select-production-recipe/	Обрати активний	—

API змінних:

Метод	Endpoint	Опис	Параметри
GET	/api/production-recipe-variables/	Усі змінні	?sequence=ID або ?production_line=ID
POST	/api/production-recipe-variables/	Створити	—
PUT	/api/production-recipe-variables/{id}/	Оновити	—
DELETE	/api/production-recipe-variables/{id}/	Видалити	—

E

10. Події та аварії

Для користувача

Події (Events) — умови, що виконують дії під час роботи послідовності (наприклад, перемикання при досягненні значення).

Аварії (Accidents) — критичні умови, що зупиняють процес при виникненні небезпечної ситуації.

Форма події:

Поле	Опис
Назва	Ім'я події (+ переклади)
Тип	Тип спрацювання
Пристрій	Відстежуваний пристрій
Умова	Оператор порівняння та значення
Дія	Що виконати при спрацюванні

Форма аварії:

Поле	Опис
Назва	Ім'я аварії (+ переклади)
Серйозність	INFO / WARNING / ERROR / CRITICAL
Пристрій	Відстежуваний пристрій
Умова	Оператор та значення
Блокування	Які пристрої заблокувати

Скріншот інтерфейсу програми "Програми" у розділі "Події". Таблиця показує 10 подій з такими колонками: ID, Назва, Тип дії, Ділянка, Умови активації, Статус, Дії.

ID	Назва	Тип дії	Ділянка	Умови активації	Статус	Дії
1	Набір води	Активувати пристрій	Пастеризатор	Датчик наявності води	Активна	⏻ ⏸ ⏹
2	Увімнення верхнього поверку	Активувати пристрій	Пастеризатор	Селектор Пастеризатора "Верхній поверк"	Активна	⏻ ⏸ ⏹
3	Увімнення нижнього поверку	Активувати пристрій	Пастеризатор	Селектор Пастеризатора "Нижній поверк"	Активна	⏻ ⏸ ⏹
4	Увімнення клапана EV-1	Активувати пристрій	Пастеризатор	Датчик RT-100-P6 <=	Активна	⏻ ⏸ ⏹
5	Увімнення клапана EV-3	Активувати пристрій	Пастеризатор	Датчик RT-100-P6 <=	Активна	⏻ ⏸ ⏹
6	Увімнення клапана EV-2	Активувати пристрій	Пастеризатор	Датчик RT-100-P5 <=	Активна	⏻ ⏸ ⏹
7	Увімнення клапана EV-4	Активувати пристрій	Пастеризатор	Датчик RT-100-P7 <=	Активна	⏻ ⏸ ⏹
8	Увімнення клапана EV-5	Активувати пристрій	Пастеризатор	Датчик RT-100-P8 <=	Активна	⏻ ⏸ ⏹
9	Активізація вентиляторів	Активувати пристрій	Пастеризатор	Датчик води (30 м)	Активна	⏻ ⏸ ⏹
10	Активізація TO 2	Активувати пристрій	Пастеризатор	Селектор Теплообмінник Зона 2	Активна	⏻ ⏸ ⏹

Керування подіями: таблиця з подіями, колонки Name, Device, Condition, Action, кнопки Add/Edit/Delete.

Програми

Події аварій

ID ↑	Назва	Шаблон заголовку	Шаблон опису	Пріоритет	Схема	Ділення	Умов	Затримка	Одноразова	Опис	Активна	Дії
1	Аварія нагріву RT100-1	Высокая температура RT100-1	-	50	-	-	1	0	Ні	-	Активна	⏻ ⏹ ⏸ ⏶
2	Аварія нагріву RT100-2	Высокая температура RT100-2	-	50	-	-	1	0	Ні	-	Активна	⏻ ⏹ ⏸ ⏶
3	Аварія нагріву RT100-3	Высокая температура RT100-3	-	50	-	-	1	0	Ні	-	Активна	⏻ ⏹ ⏸ ⏶
4	Аварія нагріву RT100-4	Высокая температура RT100-4	-	50	-	-	1	0	Ні	-	Активна	⏻ ⏹ ⏸ ⏶
5	Аварія нагріву RT100-5	Высокая температура RT100-5	-	50	-	-	1	0	Ні	-	Активна	⏻ ⏹ ⏸ ⏶
6	Аварія нагріву RT100-6	Высокая температура RT100-6	-	50	-	-	1	0	Ні	-	Активна	⏻ ⏹ ⏸ ⏶
7	Аварія нагріву RT100-7	Высокая температура RT100-7	-	50	-	-	1	0	Ні	-	Активна	⏻ ⏹ ⏸ ⏶
8	Аварія нагріву RT100-8	Высокая температура RT100-8	-	50	-	-	1	0	Ні	-	Активна	⏻ ⏹ ⏸ ⏶
9	Аварія температури пари RT100-P4	Авария температуры пара датчика RT100-P4 для мотора P4	-	50	-	-	1	0	Ні	-	Неактивна	⏻ ⏹ ⏸ ⏶
10	Аварія температури пари RT100-P5	Авария температуры пара датчика RT100-P5 для мотора P5	-	50	-	-	1	0	Ні	-	Неактивна	⏻ ⏹ ⏸ ⏶

Рядків на сторінці: 10 | 1 | 2 | Дані > 1-10 з 15

Керування аваріями: таблиця з кольоровою індикацією серйозності (INFO-синій, WARNING-жовтий, ERROR-червоний, CRITICAL-темно-червоний).

Технічна документація

API подій:

Метод	Endpoint	Опис
GET	/api/events/	Список подій
POST	/api/events/	Створити
PUT	/api/events/{id}/	Оновити
DELETE	/api/events/{id}/	Видалити
POST	/api/events/{id}/set_active/	Активувати/деактивувати

API аварій:

Метод	Endpoint	Опис
GET	/api/accident-events/	Список визначень аварій
POST	/api/accident-events/	Створити визначення
GET	/api/accidents/	Список подій, що сталися
POST	/api/accidents/{id}/acknowledge/	Підтвердити аварію
POST	/api/accidents/{id}/resolve/	Розв'язати аварію
POST	/api/accidents/resolve-all/	Розв'язати всі
GET	/api/accidents/active_events/	Поточні активні аварії

WebSocket: Аварії

Підключення: ws:///ws/accidents

```
{
  "type": "accident_created",
  "accident_id": 42,
  "title": "Висока температура",
  "description": "AI_001 перевищив поріг 95°C",
  "status": "NEW",
  "severity_level": "CRITICAL",
  "device_id": "ai_001",
  "device_name": "Temperature Sensor 1",
  "schema_id": 1,
  "occurred_at": "2026-04-07T12:00:00Z"
}
```

11. Опції

Для користувача

Опції — кнопки/перемикачі, доступні оператору під час роботи послідовності. Дозволяють обирати альтернативні шляхи виконання.

Форма опції:

Поле	Опис
Назва	Ім'я опції (+ переклади)
Послідовність	Прив'язка до послідовності
Опис	Текстовий опис

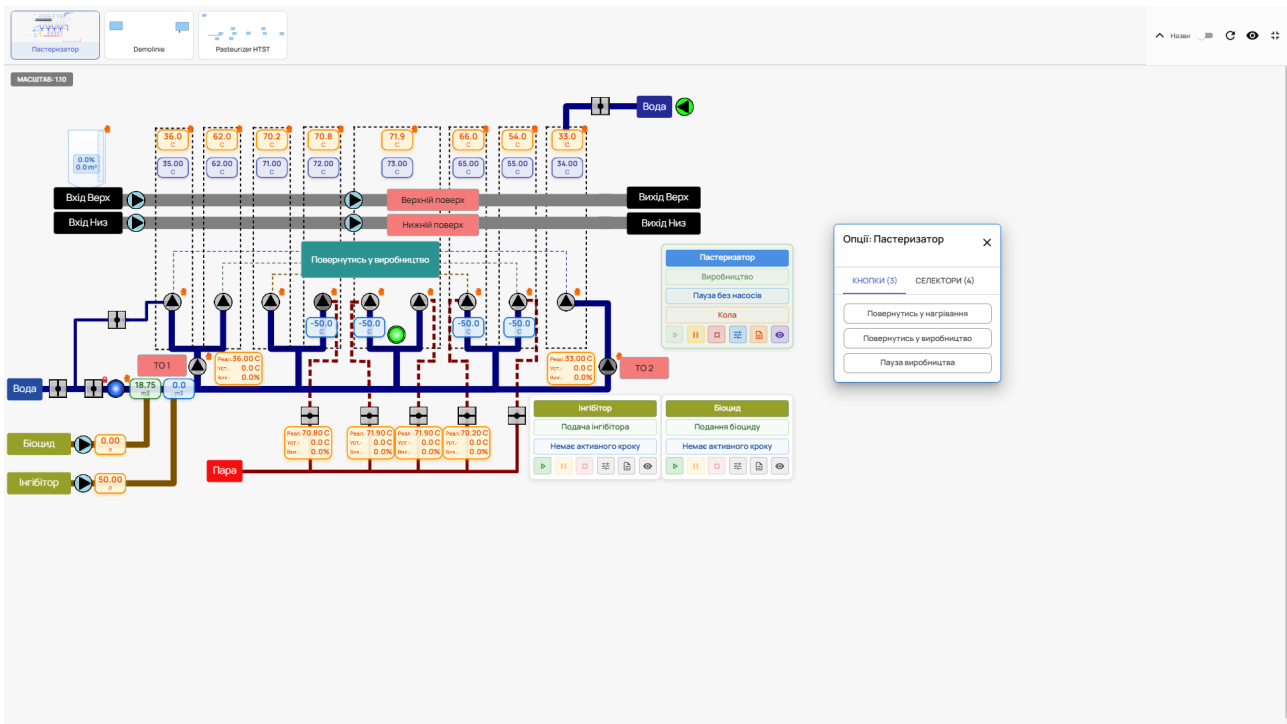
На схемі (Schema Viewer):

- Опція відображається як кнопка
- Натискання перемикає опцію (увімк/вимк)
- Колір кнопки змінюється залежно від стану

Скріншот інтерфейсу системи SCADA, що демонструє список опцій програм. Інтерфейс складається з лівого меню навігації та головного робочого простору. У меню вибрано пункт 'Програми'. У верхній частині робочого простору є панель інструментів з вкладками: Програми, Діяльність виробництва, Умови активації, Події, Аварії, Опції програм (активна), Рецепти, Пульс модулі, Лінії виробництва. Під панеллю інструментів знаходиться заголовок 'Опції програм (7)' та кнопки 'Оновити' та 'Додати опцію'. Нижче розташований пошуковий рядок з текстом 'Введіть назву опції'. Основна частина екрана зайнята таблицею з наступними стовпцями: ID, Назва, Опис, Тип, Ділення, Activated, Conditions, Доступність, Статус, Дії. У таблиці наведено 7 записів, кожен з яких має чекбокс та набор іконок для управління (вимкнення, оновлення, редагування, видалення).

ID	Назва	Опис	Тип	Ділення	Activated	Conditions	Доступність	Статус	Дії
1	Пауза виробництва	-	Кнопка (тригер)	Пастеризатор	31.03.2026, 13:43:31	1	Доступна для взаємодії	Неактивна	🔌 🔄 ✎ 🗑️
2	Верхній поверх	-	Селектор	Пастеризатор	-	1	Доступна для взаємодії	Неактивна	🔌 🔄 ✎ 🗑️
3	Нижній поверх	-	Селектор	Пастеризатор	-	1	Доступна для взаємодії	Неактивна	🔌 🔄 ✎ 🗑️
4	Повернутись у виробництво	-	Кнопка (тригер)	Пастеризатор	09.04.2026, 14:01:30	1	Доступна для взаємодії	Неактивна	🔌 🔄 ✎ 🗑️
5	Повернутись у нагрівання	-	Кнопка (тригер)	Пастеризатор	14.05.2026, 13:11:00	1	Доступна для взаємодії	Неактивна	🔌 🔄 ✎ 🗑️
6	ТО1	-	Селектор	Пастеризатор	-	1	Доступна для взаємодії	Неактивна	🔌 🔄 ✎ 🗑️
7	ТО2	-	Селектор	Пастеризатор	-	1	Доступна для взаємодії	Неактивна	🔌 🔄 ✎ 🗑️

Керування опціями: список опцій з назвою, послідовністю, описом, кнопками Edit/Delete.



Кнопка опції на схемі: прямокутна кнопка з назвою, зелена — активна, сіра — неактивна.

Технічна документація

API опцій:

Метод	Endpoint	Опис
GET	/api/sequence-options/	Список опцій
POST	/api/sequence-options/	Створити
PUT	/api/sequence-options/{id}/	Оновити
DELETE	/api/sequence-options/{id}/	Видалити

12. Імпульсні модулі

Для користувача

Імпульсні модулі генерують циклічні послідовності увімкнень/вимкнень пристроїв (наприклад, циклічна подача води).

Структура:

Рівень	Опис
Модуль	Контейнер для кроків
Крок	Один етап циклу
Фаза	Часовий інтервал усередині кроку (увімк/вимк)
Пристрій фази	Який пристрій вмикається/вимикається

The screenshot shows the 'Програми' (Programs) section of the SCADA System. The 'Пульт модулі' (Pulse Modules) sub-section is active. A table lists the following module:

ID	Назва	Опис	Циклія	Етапів	Статус	Дії
1	ТЕст	-	5	2	Зупинено	[Power Off] [Refresh] [Edit] [Delete]

Керування імпульсними модулями: ієрархічний список — модуль → кроки → фази → пристрої. Кнопки Add/Edit/Delete на кожному рівні.

Технічна документація

API імпульсних модулів:

Метод	Endpoint	Опис
GET	/api/pulse-modules/	Список модулів
POST	/api/pulse-modules/	Створити
POST	/api/pulse-modules/{id}/start/	Запустити
POST	/api/pulse-modules/{id}/stop/	Зупинити
GET	/api/pulse-module-steps/	Кроки модуля

Метод	Endpoint	Опис
GET	/api/pulse-module-step-phases/	Фази кроку
GET	/api/pulse-module-devices/	Пристрої фази

13. Керування партіями

Для користувача

Система керування партіями відстежує виробничі порції (batches) протягом усього процесу.

Елементи панелі партій:

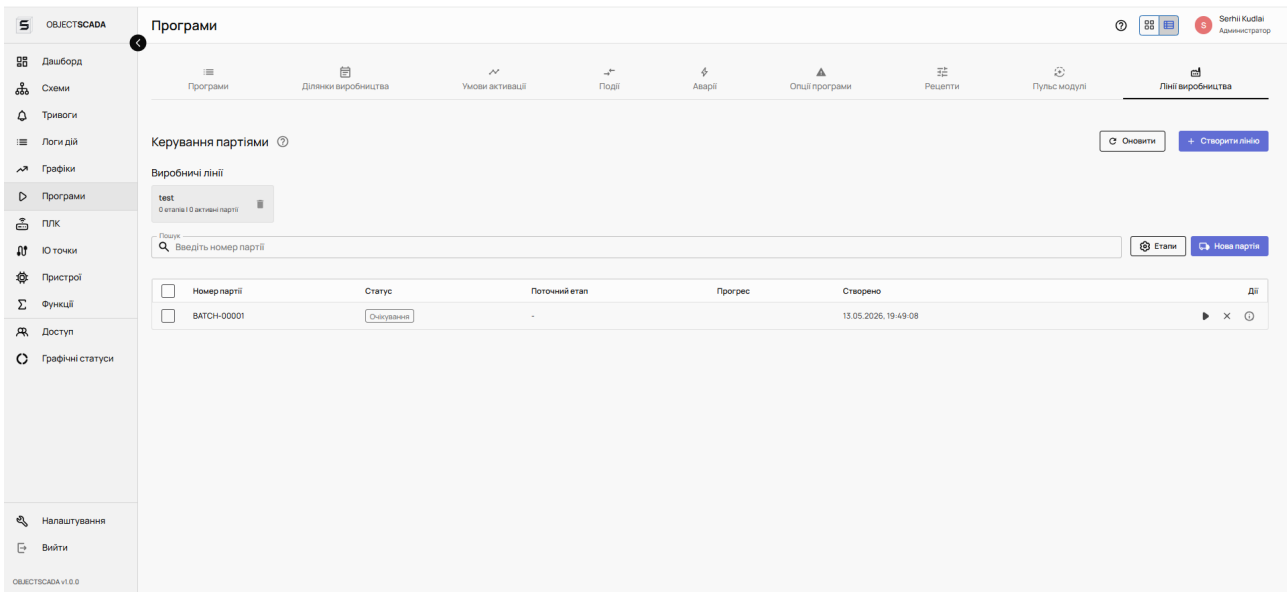
Елемент	Опис
Створити партію	Кнопка створення нової партії
Таблиця партій	Список з номером, рецептом, статусом, прогресом
Статус партії	Pending / Queued / In Progress / Paused / Completed / Failed / Cancelled
Черга	Порядок обробки партій

Дії з партією:

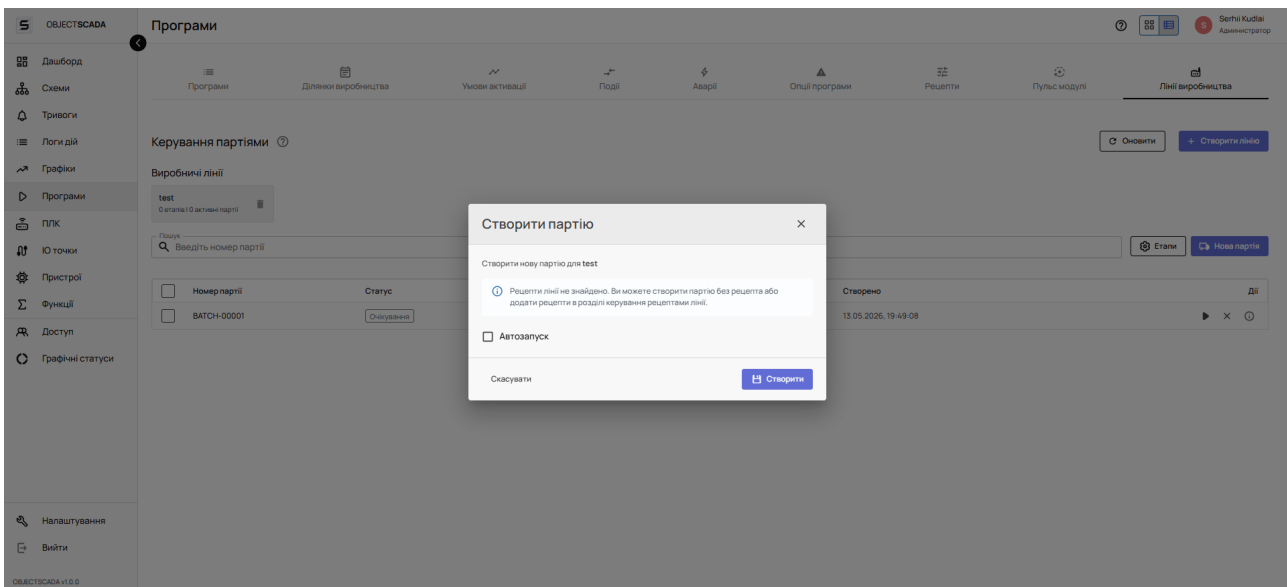
Кнопка	Опис
Play	Запуск/відновлення партії
Pause	Призупинення
Stop	Повна зупинка
Skip	Пропуск поточного кроку
Cancel	Скасування партії
Delete	Видалення зі списку

Діалог створення партії:

Поле	Опис
Виробнича лінія	Вибір лінії
Рецепт	Вибір рецепта
Номер партії	Унікальний номер (BAT-001)
Авто-старт	Запустити одразу після створення



Панель керування партіями: таблиця з партіями, кольорова індикація статусу (зелений — In Progress, сірий — Pending, синій — Completed, червоний — Failed), кнопки керування.



Діалог створення: вибір лінії, рецепта, введення номера, чекбокс «Авто-старт», кнопки «Створити» та «Скасувати».

Технічна документація

API партій:

Метод	Endpoint	Опис
GET	/api/batches/	Усі партії
POST	/api/production-lines/{id}/create_batch/	Створити партію
POST	/api/batches/{id}/start/	Запустити
POST	/api/batches/{id}/pause/	Пауза
POST	/api/batches/{id}/resume/	Продовжити
POST	/api/batches/{id}/cancel/	Скасувати
POST	/api/batches/{id}/transfer/	Перевести на наступний етап

Метод	Endpoint	Опис
GET	/api/batches/{id}/history/	Історія етапів
GET	/api/batch-queue/	Черга партій
POST	/api/batch-queue/reorder/	Змінити порядок черги

WebSocket: Статус партій

Підключення: ws://ws/schema/{schemaid}/batches

```
{
  "type": "batch_status",
  "batch_id": "uuid-...",
  "batch_number": "BAT-001",
  "status": "IN_PROGRESS",
  "current_stage": "Mixing",
  "progress": 65,
  "production_line_id": "uuid-...",
  "recipe_name": "Recipe A",
  "updated_at": "2026-04-07T12:00:00Z"
}
```

WebSocket команди (Client → Server):

```
{
  "type": "batch_command",
  "command": "start",
  "batch_id": "uuid-...",
  "production_line_id": "uuid-..."
}
```

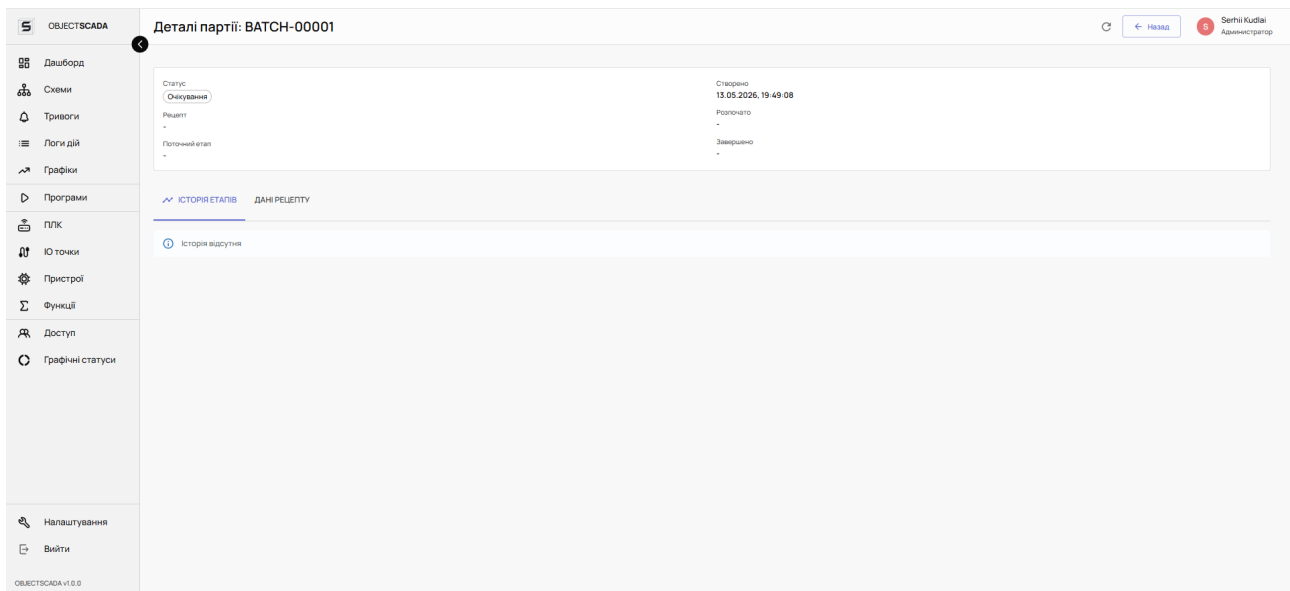
14. Виробничі лінії

Для користувача

Виробнича лінія об'єднує декілька послідовностей у єдиний виробничий маршрут з етапами.

Елементи:

Поле	Опис
Назва	Ім'я лінії (+ переклади)
Етапи	Список етапів (Production Line Stages), кожен прив'язаний до послідовності
Порядок етапів	Визначає маршрут проходження партії
Статус	Активна / Неактивна



Керування лініями: таблиця з назвою, кількістю етапів, статусом, кнопками Edit/Delete. Детальний вигляд — список етапів з перетягуванням для зміни порядку.

Технічна документація

API виробничих ліній:

Метод	Endpoint	Опис
GET	/api/production-lines/	Список ліній
POST	/api/production-lines/	Створити
PATCH	/api/production-lines/{id}/	Оновити
DELETE	/api/production-lines/{id}/	Видалити
POST	/api/production-lines/{id}/activate/	Активувати
POST	/api/production-lines/{id}/deactivate/	Деактивувати
GET	/api/production-line-stages/	Усі етапи
POST	/api/production-line-stages/	Створити етап

Метод	Endpoint	Опис
PATCH	/api/production-line-stages/{id}/	Оновити
DELETE	/api/production-line-stages/{id}/	Видалити

15. Тренди

Для користувача

Тренди — візуалізація історичних даних з пристроїв у вигляді графіків.

Елементи екрана:

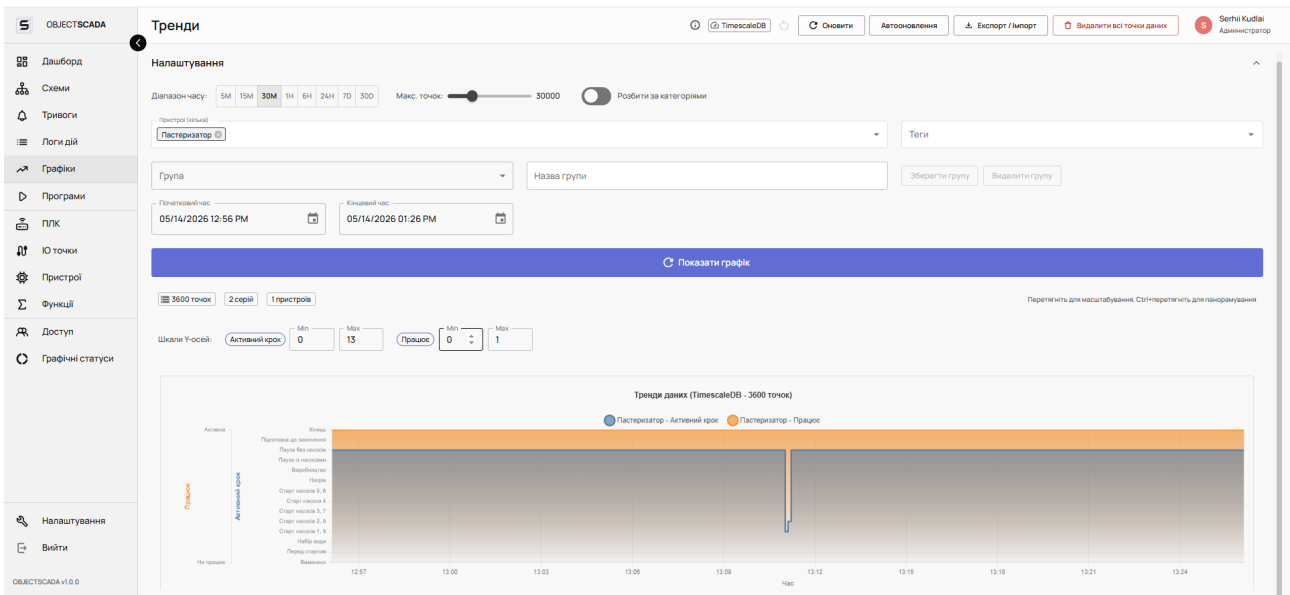
Елемент	Опис
Вибір пристроїв	Мультиселект пристроїв/параметрів для графіка
Часовий діапазон	Пресети: 5 хв, 15 хв, 30 хв, 1 година, 6 годин, 24 години, 7 днів, 30 днів
Користувачький період	Вибір довільного діапазону дат/часу
Графік	Лінійний графік Chart.js з масштабуванням
Групи пристроїв	Збереження/завантаження наборів пристроїв
Сховище	Інформація про використання дискового простору

Взаємодія з графіком:

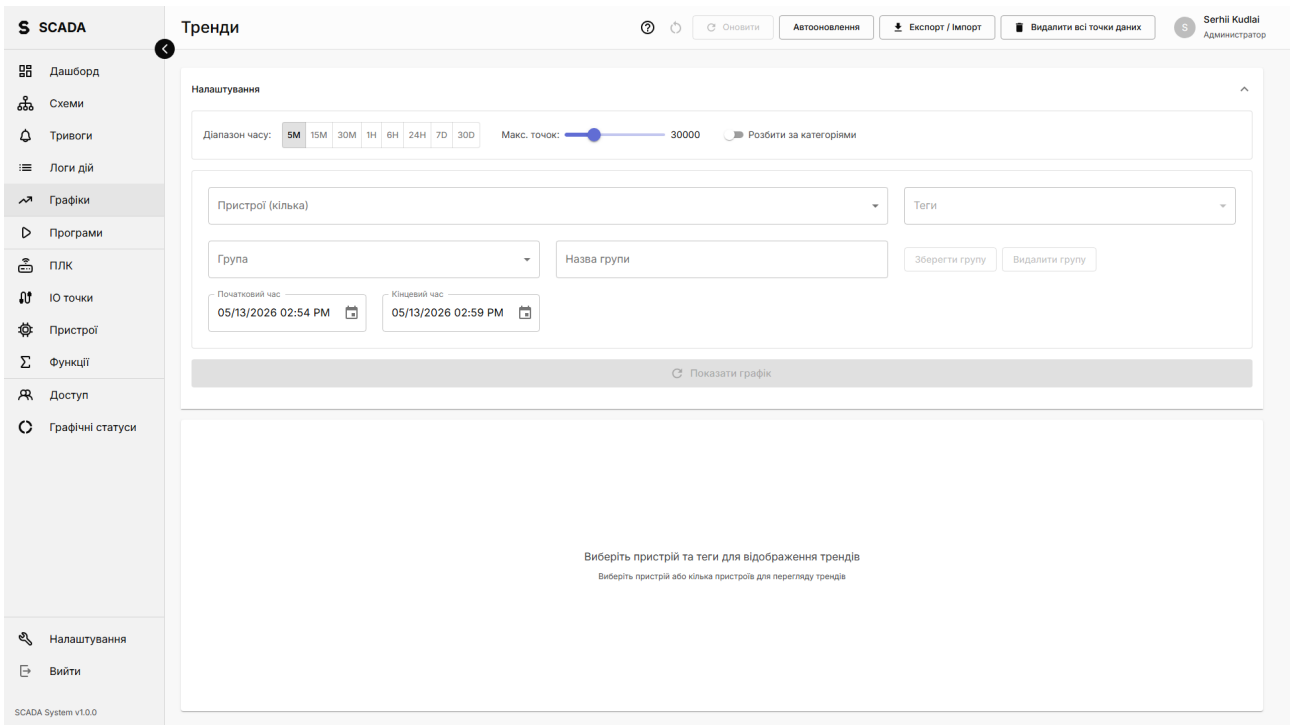
Дія	Опис
Drag (перетягування)	Виділення області для збільшення
Scroll (колесо миші)	Масштабування
ПКМ + перетягування	Панорамування
Наведення	Спливна підказка зі значеннями
Reset zoom	Скидання масштабу до початкового

Кнопки:

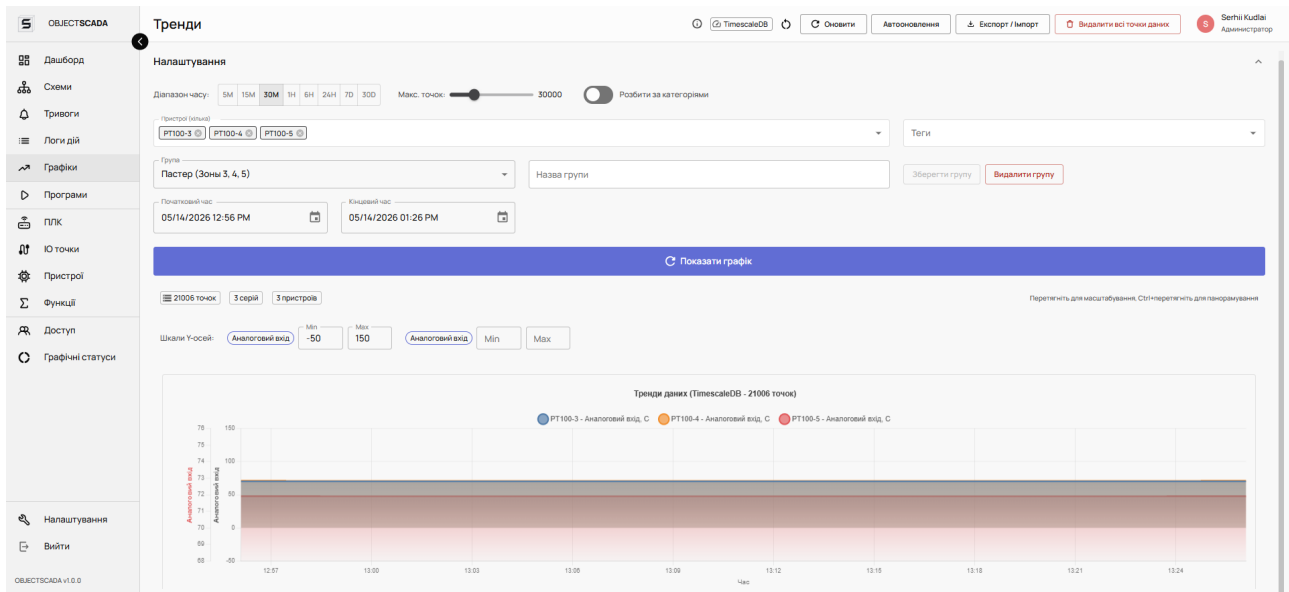
Кнопка	Опис
Додати пристрої	Відкриває мультиселект пристроїв
Зберегти групу	Зберігає поточний набір пристроїв
Завантажити групу	Вибір збереженої групи
Видалити групу	Видалення збереженої групи
Завантажити	Експорт даних (CSV/JSON)
Завантажити графік	Експорт зображення графіка
Видалити дані	Видалення точок даних (суперкористувач)
Оновити	Примусове оновлення даних



Повний екран трендів: зверху — вибір пристроїв (multi-select chips), вибір часового діапазону (кнопки 5m/15m/1h/6h/24h/7d/30d та DateTimePicker). По центру — лінійний графік з декількома лініями різних кольорів. Праворуч — легенда.



Графік після масштабування: виділена область збільшена, видно Reset zoom кнопку.



Dropdown збережених груп пристроїв: «Група 1 — Температури», «Група 2 — Тиски», кнопки Save/Load/Delete.

Технічна документація

API трендів (TimescaleDB):

Метод	Endpoint	Опис	Параметри
GET	/api/graphs/graph/	Дані графіка	?start_time=ISO&end_time=ISO&device_ids=id1,id2
GET	/api/graphs/available-parameters/	Доступні параметри	—
GET	/api/graphs/timescale-status/	Статус	—
DELET	/api/graphs/delete-all/	Видалити всі дані	—
GET	/api/data-points/	Точки даних	?limit=100&offset=0

Дані експорту:

Метод	Endpoint	Опис
POST	/api/project/export-graph-data/	Експорт графіків
POST	/api/project/import-graph-data/	Імпорт графіків
GET	/api/project/graph-data-stats/	Статистика даних

16. Аварійні сигнали

Для користувача

Централізований журнал усіх аварійних подій у системі.

Елементи таблиці:

Колонка	Опис
Час	Дата і час виникнення
Серйозність	Error (червоний) / Warning (жовтий)
Повідомлення	Опис аварії
Пристрій	Джерело
Статус	Active / Acknowledged / Resolved
Дії	Кнопки керування

Фільтри:

Фільтр	Опис
За серйозністю	Warning / Error / All
За статусом	Active / Acknowledged / Resolved / All
За пристроєм	Вибір пристрою
Пошук	Текстовий пошук

Кнопки:

Кнопка	Опис
Acknowledge	Підтвердити (прийняти до відома)
Resolve	Розв'язати (проблему усунено)
Resolve All	Розв'язати всі аварії
Delete	Видалити запис
Refresh	Оновити список

S SCADA

- Дашборд
- Схеми
- Тривоги**
- Логи дій
- Графіки
- Програми
- ПЛК
- ІО точки
- Пристрої
- Функції
- Доступ
- Графічні статуси
- Налаштування
- Вийти

SCADA System v1.0.0

Тривоги та події

Пріоритет: Vcl

Статус: Vcl

Тег пристрою: Vcl

Пошук:

	Час виникнення	Пристрій	Тип аварії	Опис	Важливість	Статус	Дії
<input type="checkbox"/>	12.05.2026, 20:07:49	P-2	Аварія мотора P-2	Мотор P-2: працює, але немає зворотного зв'я...	Висока	Вирішена	
<input type="checkbox"/>	12.05.2026, 19:55:24	Невідомий пристрій	Висока температура РТ100-6		Середня	Вирішена	
<input type="checkbox"/>	08.05.2026, 19:59:26	P-4	Аварія мотора P-4	Мотор P-4: працює, але немає зворотного зв'я...	Висока	Вирішена	
<input type="checkbox"/>	08.05.2026, 19:59:16	P-4	Аварія мотора P-4	Мотор P-4: працює, але немає зворотного зв'я...	Висока	Вирішена	
<input type="checkbox"/>	28.04.2026, 17:24:52	Невідомий пристрій	Висока температура РТ100-4		Середня	Вирішена	
<input type="checkbox"/>	06.04.2026, 11:41:30	P-1_PID	Аварія мотора P-1_PID	Мотор P-1_PID: працює, але немає зворотного ...	Висока	Вирішена	
<input type="checkbox"/>	06.04.2026, 11:40:40	P-1_PID	Аварія мотора P-1_PID	Мотор P-1_PID: працює, але немає зворотного ...	Висока	Вирішена	
<input type="checkbox"/>	06.04.2026, 11:40:09	P-1_PID	Аварія мотора P-1_PID	Мотор P-1_PID: працює, але немає зворотного ...	Висока	Вирішена	
<input type="checkbox"/>	06.04.2026, 11:39:39	P-1_PID	Аварія мотора P-1_PID	Мотор P-1_PID: працює, але немає зворотного ...	Висока	Вирішена	
<input type="checkbox"/>	06.04.2026, 01:13:12	P-1_PID	Аварія мотора P-1_PID	Мотор P-1_PID: працює, але немає зворотного ...	Висока	Вирішена	

Рядків на сторінці: 10 < Назад 1 2 3 4 Далі > 1-10 з 36

Сторінка аварій: таблиця з кольоровою індикацією серйозності (червона смуга ліворуч — Error, жовта — Warning). Фільтри зверху. Кнопки Acknowledge/Resolve для кожного рядка.

Технічна документація

API аварій: Див. розділ 10.

WebSocket: Підключення ws/accidents — отримання нових аварій та оновлень статусу в реальному часі.

17. Математичні функції

Для користувача

Математичні функції дозволяють створювати обчислювані значення на основі даних з пристроїв та змінних рецепта.

Елементи екрана:

Елемент	Опис
Список функцій	Таблиця/картки функцій
Кнопка «+»	Створити функцію
Пошук	За назвою

Редактор виразів (блочний):

Вираз будується з блоків, що додаються послідовно:

Тип блоку	Опис
Число	Константне числове значення
Пристрій	Поточне значення пристрою
Змінна рецепта	Значення змінної
Оператор	+, -, ×, ÷, %, ^
Дужки	() для групування
Функція	sqrt, log10, round, floor, ceil

Кнопки:

Кнопка	Опис
Додати блок	Додати новий блок у вираз
Видалити блок	Видалити блок з виразу
Перевірити	Валідація синтаксису
Тест	Обчислити з поточними значеннями
Зберегти	Збереження функції

ID	Назва	Вираз	Результат	Одиниці	Статус	Останній розрахунок	Дії
func_014	Alarm PT-100-5 Cold	Пастеризатор · PT100_5 Пастеризатор · Cold_diff	72.0000	°C	Активна	06.05.2026, 16:13:43	[Edit] [Delete] [Test]
func_016	RECIPE_70.PT100,1	Пастеризатор · PT100,1	35.0000	°C	Активна	06.05.2026, 23:02:03	[Edit] [Delete] [Test]
func_020	RECIPE_70.PT100,2	Пастеризатор · PT100,2	62.0000	°C	Активна	09.05.2026, 15:38:09	[Edit] [Delete] [Test]
func_019	RECIPE_70.PT100,3	Пастеризатор · PT100,3	71.0000	°C	Активна	06.05.2026, 16:13:43	[Edit] [Delete] [Test]
func_018	RECIPE_70.PT100,4	Пастеризатор · PT100,4	72.0000	°C	Активна	06.05.2026, 16:13:43	[Edit] [Delete] [Test]
func_017	RECIPE_70.PT100,5	Пастеризатор · PT100,5	73.0000	°C	Активна	06.05.2026, 16:13:43	[Edit] [Delete] [Test]
func_021	RECIPE_70.PT100,6	Пастеризатор · PT100,6	65.0000	°C	Активна	06.05.2026, 16:13:43	[Edit] [Delete] [Test]
func_022	RECIPE_70.PT100,7	Пастеризатор · PT100,7	55.0000	°C	Активна	06.05.2026, 16:13:43	[Edit] [Delete] [Test]
func_023	RECIPE_70.PT100,8	Пастеризатор · PT100,8	34.0000	°C	Активна	06.05.2026, 16:13:43	[Edit] [Delete] [Test]
func_015	pt100-4_cold	Пастеризатор · PT100_4 Пастеризатор · Cold_diff	71.0000	°C	Активна	06.05.2026, 16:13:43	[Edit] [Delete] [Test]
func_001	Аварія RT100-1	Пастеризатор · RT100,1 Пастеризатор · Alarm_diff	90.0000	°C	Активна	08.05.2026, 17:39:43	[Edit] [Delete] [Test]
func_002	Аварія RT100-2	Пастеризатор · RT100,2 Пастеризатор · Alarm_diff	117.0000	°C	Активна	09.05.2026, 15:38:09	[Edit] [Delete] [Test]
func_003	Аварія RT100-3	Пастеризатор · RT100,3 Пастеризатор · Alarm_diff	126.0000	°C	Активна	08.05.2026, 17:39:43	[Edit] [Delete] [Test]
func_004	Аварія RT100-4	Пастеризатор · RT100,4 Пастеризатор · Alarm_diff	127.0000	°C	Активна	08.05.2026, 17:39:43	[Edit] [Delete] [Test]

Список функцій: таблиця з колонками Name, Expression, Result, Unit, кнопки Edit/Delete/Test.

Редактор виразів: ряд блоків — [AI_001] [*] [2.5] [+] [sqrt] [(] [AI_002] [)]. Кнопки додавання блоків знизу. Результат тесту праворуч.

Технічна документація

API математичних функцій:

Метод	Endpoint	Опис
GET	/api/math-functions/	Список функцій
POST	/api/math-functions/	Створити

Метод	Endpoint	Опис
PUT	/api/math-functions/{id}/	Оновити
DELETE	/api/math-functions/{id}/	Видалити
POST	/api/math-functions/{id}/calculate/	Обчислити
POST	/api/math-functions/{id}/validate_expression/	Валідація
POST	/api/math-functions/{id}/test_expression/	Тест
POST	/api/math-functions/validate_all/	Валідація всіх

WebSocket: Оновлення мат. функцій

Підключення: ws://ws/math-functions

```
{
  "type": "math_function_update",
  "function_id": 5,
  "name": "Total Flow",
  "expression": "ai_001 * 2.5 + sqrt(ai_002)",
  "result": 157.3,
  "units": "L/h",
  "last_calculated": "2026-04-07T12:00:00Z",
  "calculation_error": null,
  "is_active": true
}
```

18. Точки вводу-виводу

Для користувача

I/O Points визначають зв'язок між фізичними сигналами ПЛК та логічними пристроями SCADA.

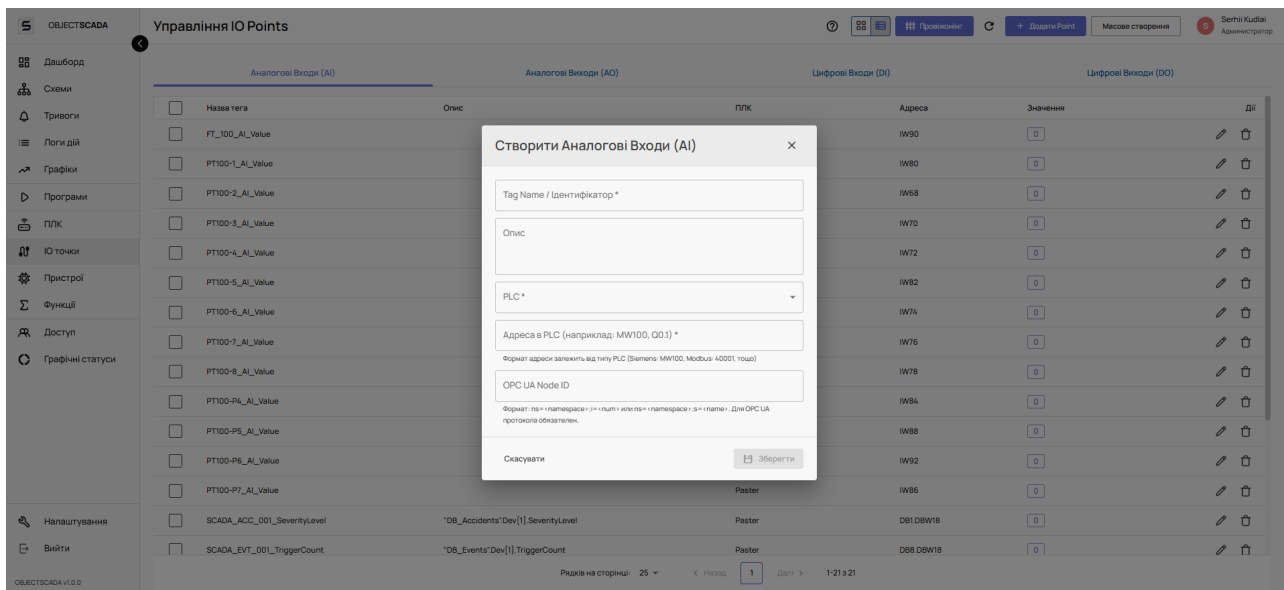
Вкладки:

Вкладка	Опис
Analog Inputs	Аналогові входи (4-20 mA, 0-10V)
Analog Outputs	Аналогові виходи
Digital Inputs	Дискретні входи
Digital Outputs	Дискретні виходи

Форма I/O точки (IoPointDialog):

Поле	Опис
ПЛК	Вибір контролера
Номер/Адреса	Адреса в ПЛК (IW0, QW0, IO.0, QO.0)
Пристрій	Прив'язка до логічного пристрою
Масштабування	Min/Max (для аналогових)
Тип даних	INT, REAL, BOOL
Одиниця	Інженерна одиниця

Сторінка I/O Points: 4 вкладки (AI, AO, DI, DO), таблиця з колонками PLC, Address, Device, Scale, Unit, кнопки Add/Edit/Delete.



Діалог створення I/O точки: dropdown ПЛК, поле адреси, dropdown пристрою, поля Min/Max Scale, Unit.

Технічна документація

API I/O точок:

Метод	Endpoint	Опис
GET	/api/points/analog-inputs/	AI точки
POST	/api/points/analog-inputs/	Створити AI
GET	/api/points/analog-outputs/	АО точки
POST	/api/points/analog-outputs/	Створити АО
GET	/api/points/digital-inputs/	DI точки
POST	/api/points/digital-inputs/	Створити DI
GET	/api/points/digital-outputs/	DO точки
POST	/api/points/digital-outputs/	Створити DO

Для всіх типів: стандартні CRUD операції (GET, POST, PUT, PATCH, DELETE) + /validate/ та /test/.

19. Провізіонування I/O

Для користувача

Масове створення I/O точок на основі конфігурації пристроїв.

Елементи екрана:

Елемент	Опис
Вибір ПЛК	Цільовий контролер
Тип ПЛК	S7-1200 або S7-1500
Зберегти адреси	Чекбокс — не перезаписувати існуючі адреси
Список сутностей	Чекбоксы для кожного типу (мотори, клапани, PID, AI, AO, DI, DO, COS, лічильники, таймери, танки, послідовності, події, аварії)
Прев'ю	Попередній перегляд результату
Провізіонувати	Запуск створення точок

Сторінка провізіонування: dropdown ПЛК, radio S7-1200/S7-1500, список сутностей з чекбоксами, кнопки Preview та Provision.

Технічна документація

API провізіонування:

Метод	Endpoint	Опис
POST	/api/plc-export/provision-io-points/preview/	Прев'ю
POST	/api/plc-export/provision-io-points/	Виконати

Метод	Endpoint	Опис
GET	/api/plc-export/provision-io-points/entities/	Доступні сутності

20. Графічні статуси

Для користувача

Графічні статуси — шаблони іконок для відображення стану пристроїв на мнемосхемах.

Елементи:

Елемент	Опис
Список шаблонів	Картки/таблиця
Кнопка «+»	Створити шаблон

Майстер створення:

- Крок 1: Ім'я, опис, тип пристрою
- Крок 2: Статуси та іконки
- Код статусу → Іконка (URL або завантаження)
- Галерея доступних іконок
- Завантаження своїх іконок

Кнопка	Опис
Додати статус	Новий рядок «код → іконка»
Видалити статус	Видалити рядок
Завантажити іконку	Upload файлу іконки
Галерея	Вибір з раніше завантажених
Попередній/Наступний	Навігація майстром
Зберегти	Збереження шаблону

ID	Назва	Опис	Кількість статусів	Створено	Оновлено	Дії
18	DI Green/Red	Немає опису	10	12.02.26 09:55:21	29.03.26 23:35:43	👁️ ✎️ 🗑️
17	VLV BF Status (NC vertical)	Немає опису	10	08.12.25 12:16:54	30.03.26 00:44:45	👁️ ✎️ 🗑️
14	DI Red/Green	Немає опису	10	27.10.25 19:21:49	30.03.26 10:48:20	👁️ ✎️ 🗑️
11	VLV RV Status (vertical)	Немає опису	10	27.10.25 19:07:54	29.03.26 23:21:34	👁️ ✎️ 🗑️
10	Valve RV Status (horizontal)	Немає опису	10	27.10.25 19:03:27	29.03.26 23:22:15	👁️ ✎️ 🗑️
9	VLV DS Status	Немає опису	12	27.10.25 18:48:51	29.03.26 23:21:11	👁️ ✎️ 🗑️
8	VLV Status BF (NO horizontal)	Немає опису	10	27.10.25 18:37:14	30.03.26 00:11:00	👁️ ✎️ 🗑️
6	MOT Status (up)	Немає опису	11	27.10.25 18:22:51	29.03.26 23:17:55	👁️ ✎️ 🗑️
5	MOT Status (right)	Немає опису	11	27.10.25 18:18:30	29.03.26 23:21:18	👁️ ✎️ 🗑️
4	MOT Status (left)	Немає опису	11	27.10.25 18:14:42	29.03.26 23:18:45	👁️ ✎️ 🗑️
3	MOT Status (down)	Немає опису	11	27.10.25 18:09:25	29.03.26 23:18:53	👁️ ✎️ 🗑️
2	Agitator Status	Немає опису	11	27.10.25 15:31:36	08.12.25 12:16:54	👁️ ✎️ 🗑️

Список графічних статусів: картки з прев'ю іконок, назва шаблону, тип пристрою.

Майстер створення: ліворуч — список статусів (код + прев'ю іконки), праворуч — галерея іконок для вибору.

Технічна документація

API графічних статусів:

Метод	Endpoint	Опис
GET	/api/graphical-statuses/	Список шаблонів
POST	/api/graphical-statuses/	Створити
PUT	/api/graphical-statuses/{id}/	Оновити
DELETE	/api/graphical-statuses/{id}/	Видалити
POST	/api/graphical-statuses/upload-icon/	Завантажити іконку (multipart)

Метод	Endpoint	Опис
GET	/api/graphical-statuses/list-icons/	Список іконок

21. Конфігуратор обладнання

Для користувача

Візуальний редактор апаратної конфігурації ПЛК для розгортання в TIA Portal.

Вкладки:

Вкладка	Опис
Rack View	Візуальний редактор стійки з модулями
PROFINET Topology	Мережева топологія пристроїв
Address Table	Таблиця адрес I/O

Редактор стійки (Rack Editor):

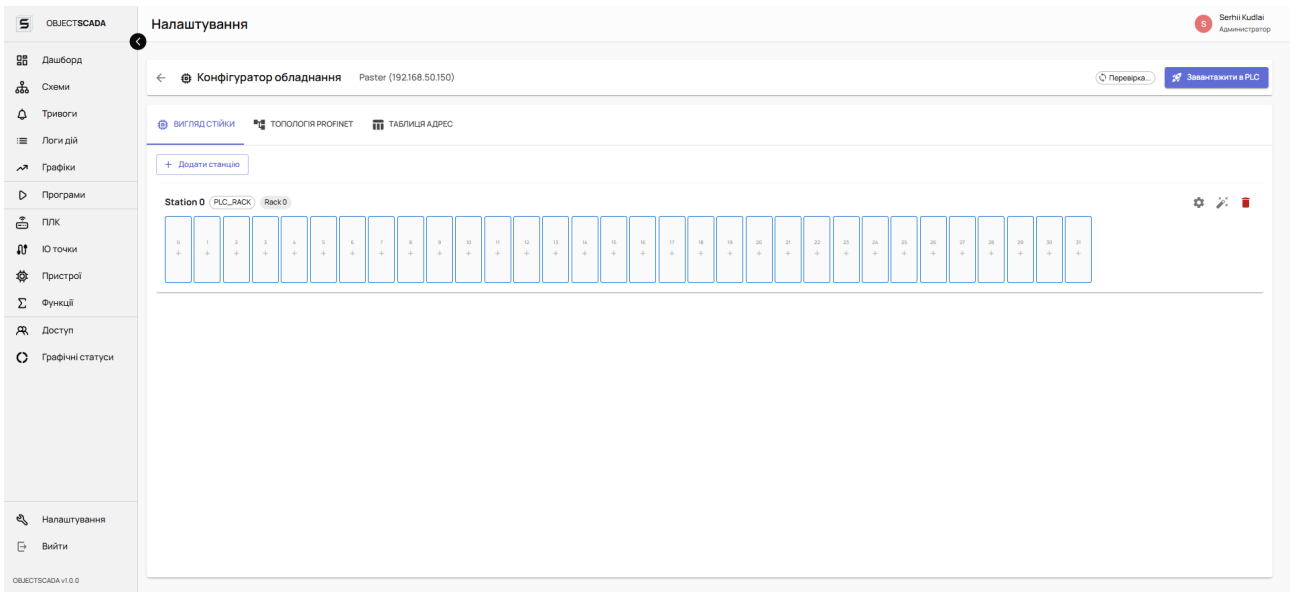
Елемент	Опис
Слоти стійки	Візуальне представлення модулів у стійці
Каталог модулів	Бокова панель з доступними модулями (drag-drop)
Панель властивостей	Конфігурація вибраного модуля
Додати модуль	Перетягування з каталогу
Видалити модуль	Видалення зі слота

Кнопки панелі інструментів:

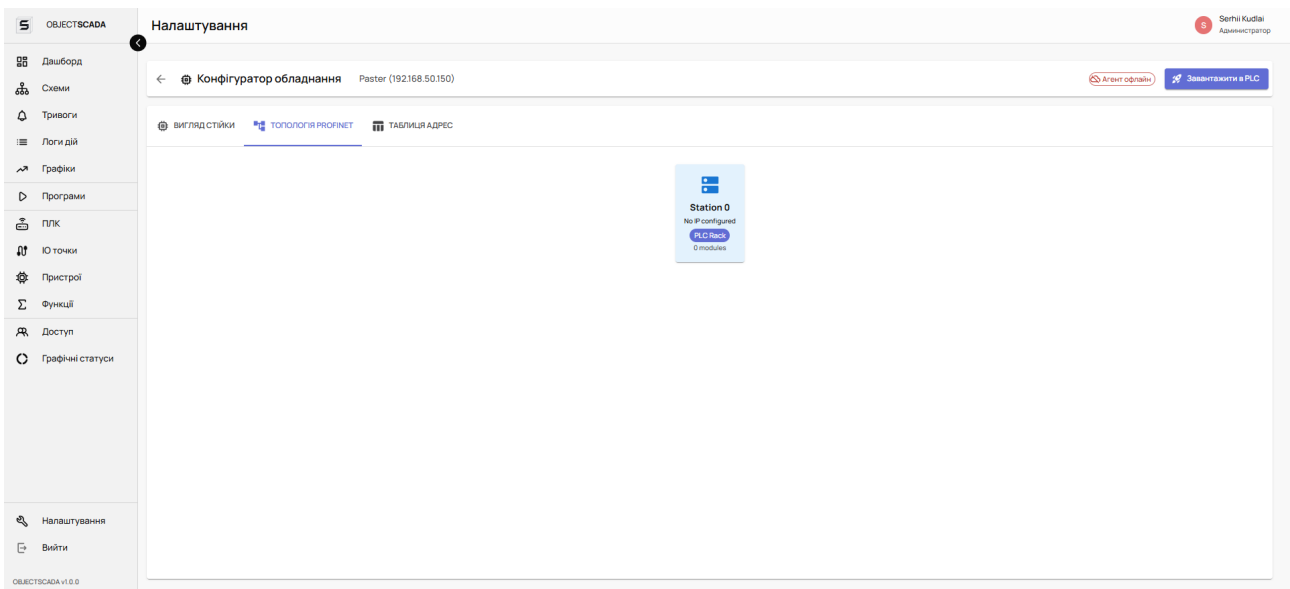
Кнопка	Опис
Deploy	Розгорнути конфігурацію в TIA Portal
Download	Завантажити конфігурацію
Import GSD	Імпорт файлу опису пристрою (GSD/GSDML)
TIA Agent Status	Індикатор підключення до TIA Agent
Back	Повернення до списку ПЛК

Діалоги:

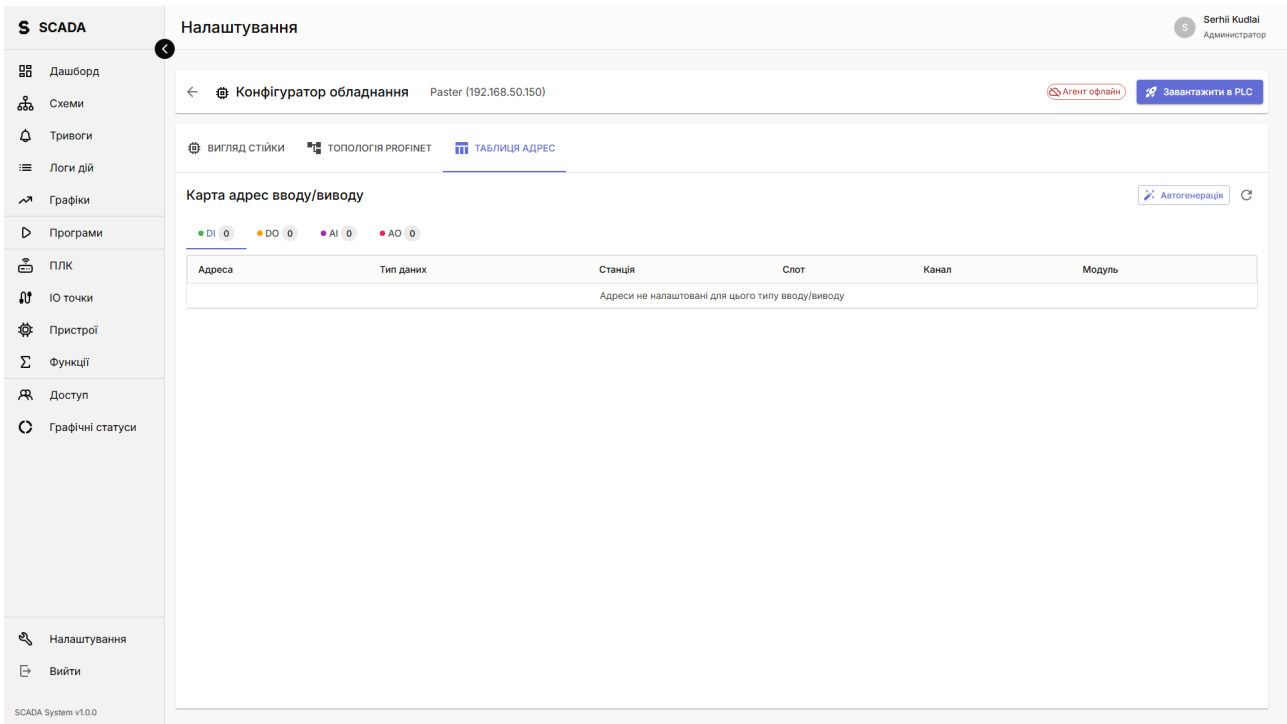
Діалог	Опис
TiaDeployDialog	Розгортання: вибір цільової системи, прогрес, статус
GsdImportDialog	Імпорт GSD: завантаження файлу, прев'ю модулів, підтвердження
NetworkConfigDialog	Налаштування мережі: IP, маска, шлюз, PROFINET параметри



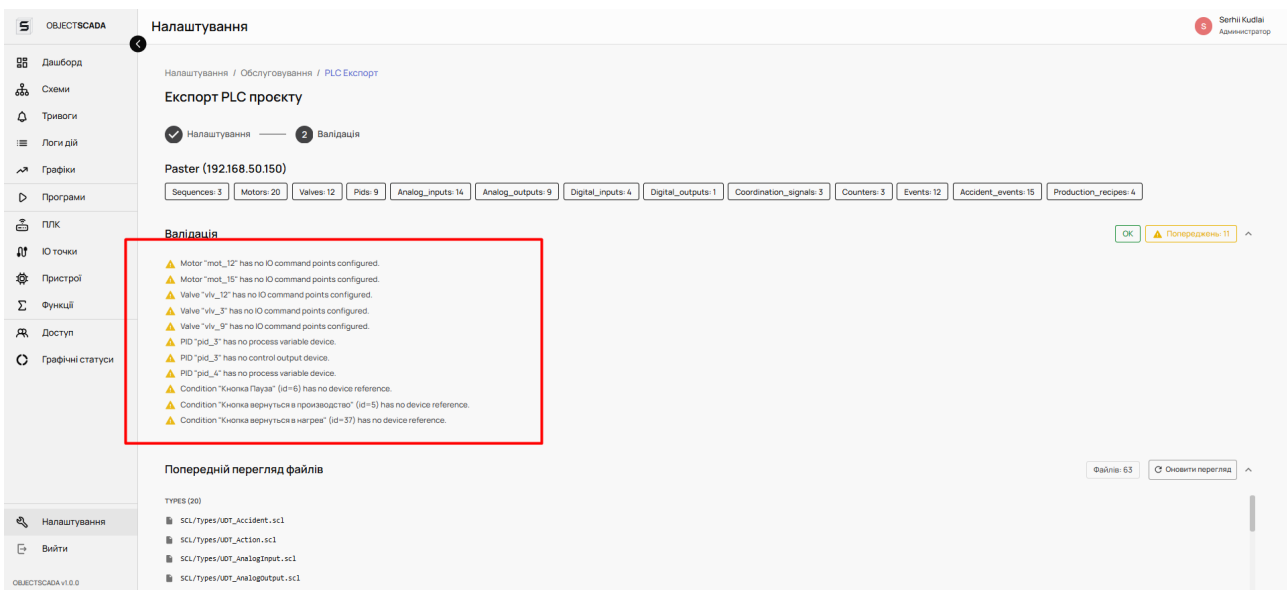
Редактор стійки: візуальна стійка з модулями у слотах, каталог модулів ліворуч (висувна панель), панель властивостей праворуч, кнопка Deploy у тулбарі.



Топологія PROFINET: мережева діаграма з вузлами пристроїв та зв'язками між ними.



Таблиця адрес: колонки Module, Slot, Channel, Address, Type (I/O), Device.



Діалог розгортання: вибір TIA Agent, прогрес-бар, лог операцій, кнопки Deploy/Cancel.

Технічна документація

API конфігуратора:

Метод	Endpoint	Опис
GET	/api/hw-catalog/	Каталог модулів
POST	/api/hw-catalog/import-gsd/	Імпорт GSD (multipart)
GET	/api/hw-stations/	Станції
POST	/api/hw-stations/	Створити станцію
POST	/api/hw-stations/{id}/auto-assign-addresses/	Авто-призначення адрес

Метод	Endpoint	Опис
GET	/api/hw-slots/	Слоти
POST	/api/hw-slots/	Створити слот
POST	/api/hw-slots/bulk-update/	Масове оновлення
GET	/api/hw-network/	Мережева конфігурація
POST	/api/tia/deploy/	Розгорнути в TIA Portal
GET	/api/tia-deploy-jobs/{id}/progress/	Прогрес розгортання
GET	/api/tia-agent-config/	Конфігурації TIA Agent
GET	/api/tia-agent-config/{id}/check-connection/	Тест з'єднання

22. Керування користувачами

Для користувача

Створення, редагування та видалення облікових записів користувачів.

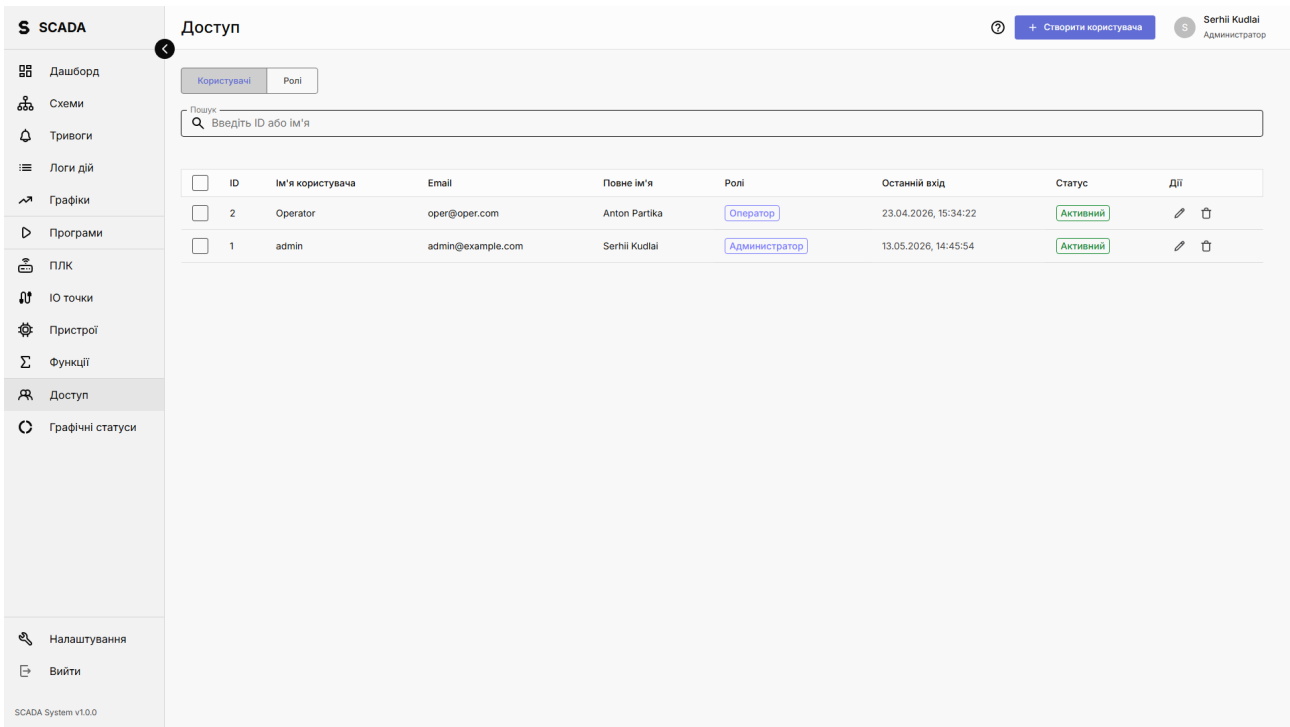
Таблиця користувачів:

Колонка	Опис
Ім'я користувача	Login
Повне ім'я	Ім'я та прізвище
Email	Електронна пошта
Ролі	Призначені ролі
Останній вхід	Дата та час
Статус	Active / Inactive
Дії	Edit / Delete

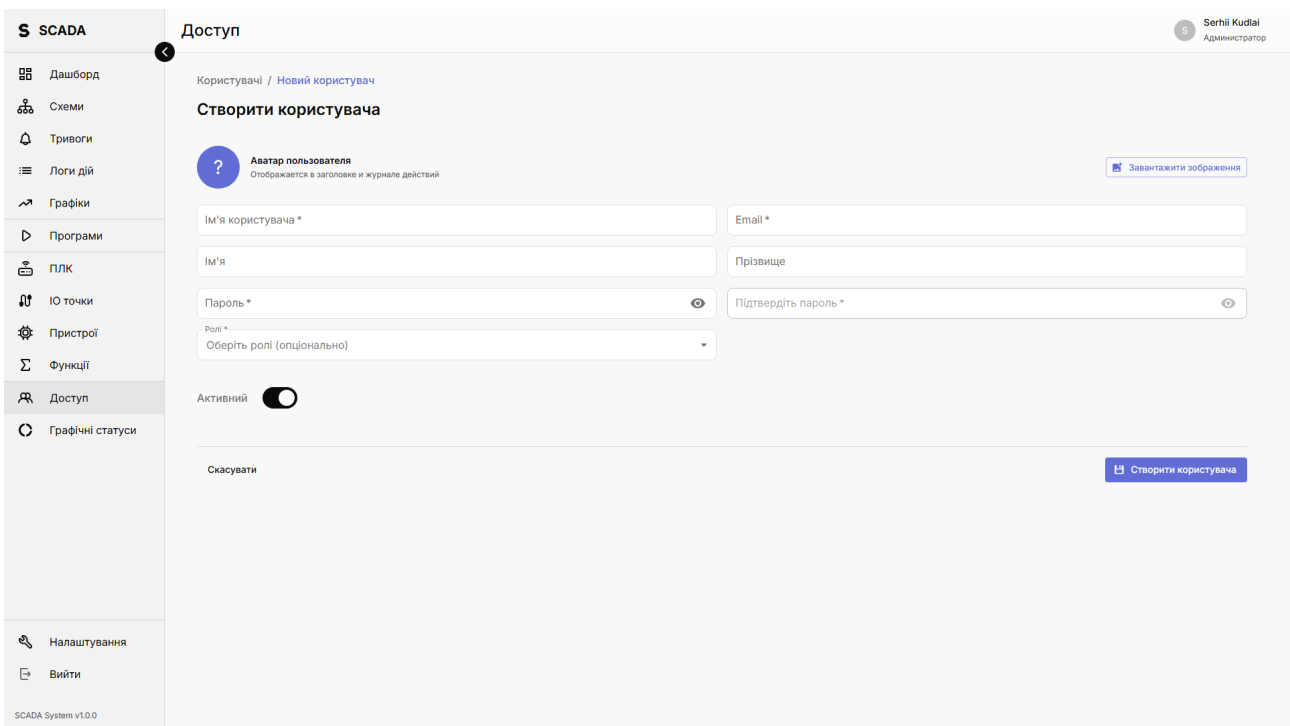
Форма користувача:

Поле	Опис
Username	Унікальний логін
Email	Електронна пошта
Пароль	Пароль (при створенні)
Підтвердження	Повторне введення пароля
Ім'я / Прізвище	Реальне ім'я
Аватар	Завантаження фото профілю
Ролі	Мультиселект ролей
Статус	Перемикач Active/Inactive

Кнопка	Опис
Зберегти	Збереження змін
Скасувати	Повернення без збереження
Видалити	Видалення облікового запису (тільки в редагуванні)
Змінити пароль	Змінити пароль користувача



Таблиця користувачів з колонками, аватарами, кольоровими бейджми ролей, індикатором статусу.



Форма створення/редагування: поля username, email, password, role selector, avatar upload, status toggle.

Технічна документація

API користувачів:

Метод	Endpoint	Опис
GET	/api/users/	Список користувачів
POST	/api/users/	Створити

Метод	Endpoint	Опис
GET	/api/users/{id}/	Отримати
PUT	/api/users/{id}/	Оновити
DELETE	/api/users/{id}/	Видалити
GET	/api/users/me/	Поточний користувач
POST	/api/users/{id}/set_password/	Змінити пароль
POST	/api/users/{id}/upload-avatar/	Завантажити аватар (multipart)
GET	/api/users/{id}/roles/	Ролі користувача
POST	/api/users/{id}/roles/	Призначити роль
DELETE	/api/users/{id}/roles/	Зняти роль

23. Керування ролями

Для користувача

Рольова модель керування доступом. Кожна роль визначає набір дозволів.

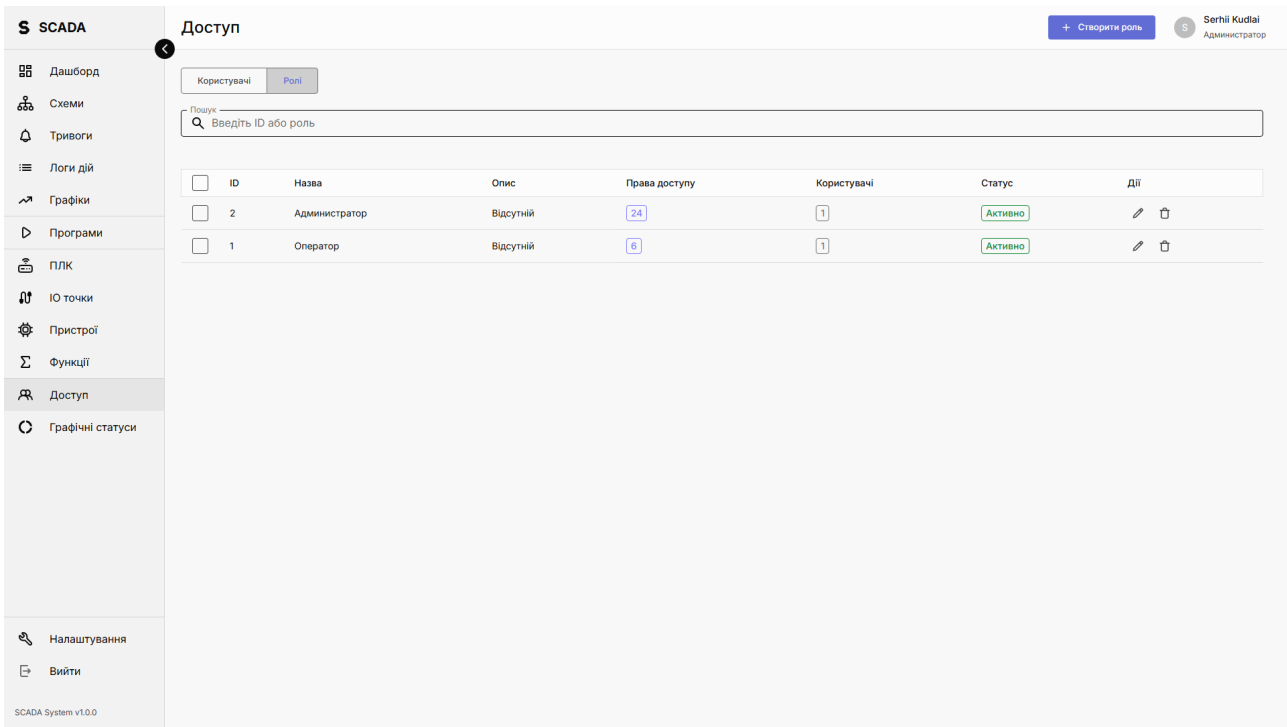
Таблиця ролей:

Колонка	Опис
Назва	Ім'я ролі
Опис	Текстовий опис
Користувачів	Кількість користувачів з цією роллю
Дії	Edit / Delete

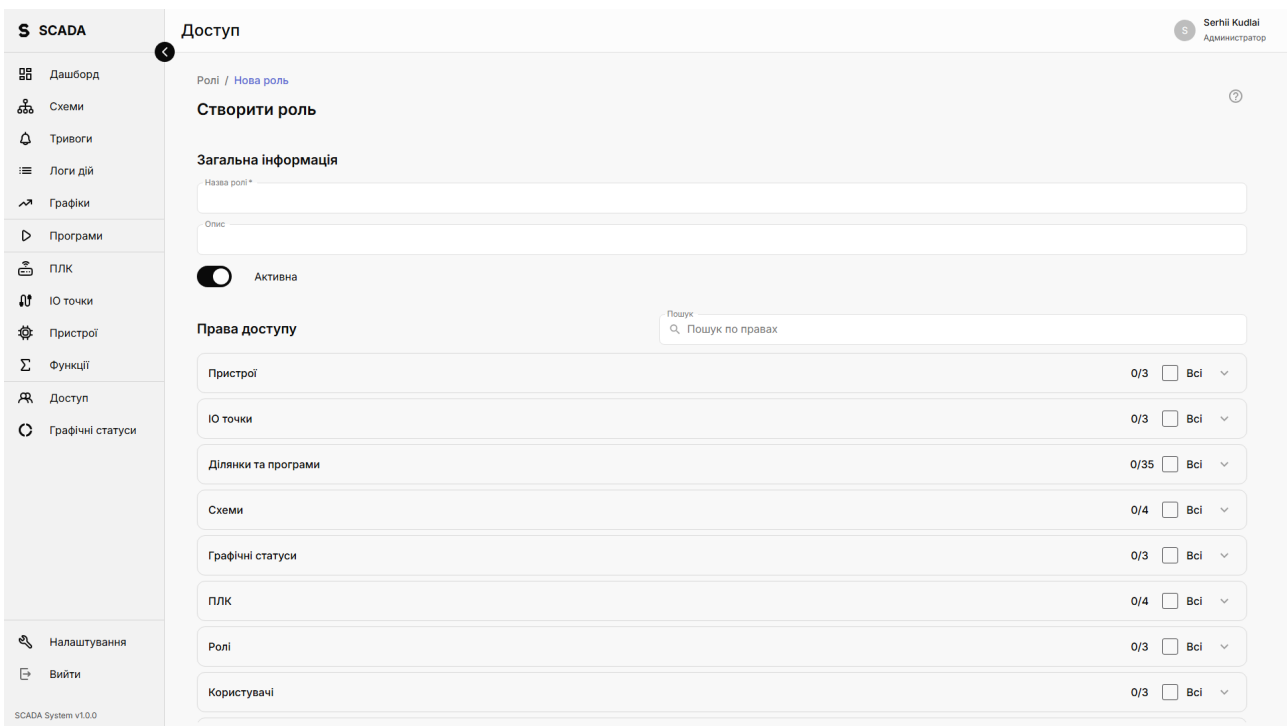
Матриця дозволів (у формі ролі):

Категорія	Дозволи
Пристрої	Створення, Перегляд, Редагування, Видалення
ПЛК	Створення, Перегляд, Редагування, Видалення, Hardware Config
Послідовності	Створення, Перегляд, Редагування, Видалення, Дублювання
Мнемосхеми	Створення, Перегляд, Редагування, Видалення
I/O Points	Створення, Перегляд, Редагування, Видалення
Користувачі	Створення, Перегляд, Редагування, Видалення
Ролі	Створення, Перегляд, Редагування, Видалення
Аварії	Підтвердження, Розв'язання, Видалення
Тренди	Перегляд, Експорт, Видалення даних
Графічні статуси	Створення, Перегляд, Редагування, Видалення
Керування	Підключення, Ліцензія, Оновлення

Кнопка	Опис
Select All	Вибрати всі дозволи
Deselect All	Зняти всі дозволи
Зберегти	Зберегти роль



Список ролей: таблиця з назвою, описом, кількістю користувачів, кнопками Edit/Delete.



Форма ролі: ім'я, опис, матриця чекбоксів за категоріями (Devices, PLCs, Sequences, Schemas...). Кнопки Select All/Deselect All.

Технічна документація

API ролей:

Метод	Endpoint	Опис
GET	/api/roles/	Список ролей
POST	/api/roles/	Створити

Метод	Endpoint	Опис
GET	/api/roles/{id}/	Отримати
PATCH	/api/roles/{id}/	Оновити
DELETE	/api/roles/{id}/	Видалити
GET	/api/roles/{id}/permissions/	Дозволи ролі
GET	/api/permissions/categories/	Категорії дозволів
GET	/api/permissions/actions/	Дії дозволів

24. Журнал активності

Для користувача

Аудит усіх дій користувачів у системі.

Елементи екрана:

Колонка	Опис
Час	Дата і час дії
Користувач	Хто виконав
Дія	Тип операції (create, update, delete, login...)
Опис	Деталі дії
IP-адреса	З якої адреси

Фільтри:

Фільтр	Опис
За користувачем	Вибір користувача
За дією	Тип операції
За датою	Діапазон дат
Пошук	Текстовий пошук

Статистика:

- Загальна кількість дій
- Зведення за користувачами
- Зведення за типами дій

SCADA
Логи дій користувачів
Очистити всі логи
Serhii Kudlai
Адміністратор

- Дашборд
- Схеми
- Тривоги
- Логи дій
- Графіки
- Програми
- ПЛК
- ІО точки
- Пристрої
- Функції
- Доступ
- Графічні статуси
- Налаштування
- Вийти

Всього дій (7 днів) **580**

Найчастіші дії

Створення пристрою: 11
Пристрій оновлено: 2
Видалення пристрою: 2
Пристрій активовано: 76
Пристрій деактивовано: 64

Користувачі
Всі

Тип дії
Всі

Період
13.05.2025 - 13.05.2026

Дата/Час	Користувач	Тип дії	Опис	Об'єкт	IP адреса
12.05.2026 20:49:40	admin	Видалення пристрою	Видалено пристрій asdasd	-	172.18.0.1
12.05.2026 20:48:59	admin	Створення пристрою	Створено пристрій asdasd (тип: Valve)	-	172.18.0.1
12.05.2026 20:21:34	admin	Оновлення ділянки	Оновлено ділянку Пастеризатор	Ділянка: Пастеризатор	172.18.0.1
12.05.2026 20:17:06	admin	Оновлення програми	Оновлено програму Виробництво	Програма: Виробництво	172.18.0.1
12.05.2026 20:07:43	admin	Зміна стану пристрою	Змінено стан П-2 на Авто	Двигун: П-2	-
12.05.2026 20:07:43	Анонімний	Пристрій активовано	Активовано пристрій П-2 (стан: Працює)	Двигун: П-2	-
12.05.2026 20:07:15	admin	Запуск ділянки	Запущено ділянку Пастеризатор	Ділянка: Пастеризатор	-
12.05.2026 20:05:53	admin	Оновлення схеми	Оновлено схему Пастеризатор	Схема: Пастеризатор	172.18.0.1
12.05.2026 19:56:58	admin	Зупинка ділянки	Зупинено ділянку Біоцид	Ділянка: Біоцид	-
12.05.2026 19:56:51	admin	Зупинка ділянки	Зупинено ділянку Інгібітор	Ділянка: Інгібітор	-
12.05.2026 19:56:49	admin	Зупинка ділянки	Зупинено ділянку Пастеризатор	Ділянка: Пастеризатор	-
12.05.2026 19:56:12	admin	Видалення пристрою	Видалено пристрій тест резервуар	-	172.18.0.1
12.05.2026 18:06:48	admin	Встановлено ручне значення пристрою	Встановлено ручне значення 33 для ai_L003	Аналоговий вхід: ai_003	172.18.0.1
12.05.2026 18:06:46	admin	Встановлено ручне значення пристрою	Встановлено ручне значення 33 для ai_L003	Аналоговий вхід: ai_003	172.18.0.1
12.05.2026 18:06:40	admin	Зміна стану пристрою	Змінено стан ai_003 на Симуляція	Аналоговий вхід: ai_003	-
12.05.2026 18:06:37	admin	Встановлено ручне значення пристрою	Встановлено ручне значення 20 для ai_L002	Аналоговий вхід: ai_002	172.18.0.1
12.05.2026 18:06:35	admin	Встановлено ручне значення пристрою	Встановлено ручне значення 20 для ai_L002	Аналоговий вхід: ai_002	172.18.0.1

Рядків на сторінці: 25 < Назад 1 2 3 4 5 ... 60 Далі > 1-25 з 1479

Журнал активності: таблиця з фільтрами зверху, статистика праворуч (графіки за типами дій та за користувачами). Пагінація знизу.

Технічна документація

API журналу активності:

Метод	Endpoint	Опис	Параметри
GET	/api/user-activity-logs/	Список логів	?user_id=X&action_type=X&limit=100&offset=0
GET	/api/user-activity-logs/recent/	За останні 24 год	—
GET	/api/user-activity-logs/statistics/	Статистика	—
DELET	/api/user-activity-logs/clear/	Очистити (admin)	—

WebSocket: Журнал у реальному часі

Підключення: ws:///ws/activity-log

```
{
  "type": "activity_log_entry",
  "id": 1234,
  "action_type": "device_update",
  "description": "Updated motor MOT_001 mode to manual",
  "user": {"id": 1, "username": "admin"},
  "content_object": {"type": "motor", "id": "mot_001"},
  "metadata": {},
  "created_at": "2026-04-07T12:00:00Z"
}
```

25. Налаштування

Для користувача

Сторінка налаштувань містить декілька вкладок:

Вкладка «Загальні»

Параметр	Опис
Тема	Вибір кольорової схеми (Blue, Green, Orange, Pink, Purple, Red)
Темний режим	Перемикач Light/Dark
Мова	Вибір мови інтерфейсу (EN, UK, RU, DE, FR, PL, AR...)
Версія дизайну	V1 / V2
AI-асистент	Увімкнути/вимкнути AI-помічника

Вкладка «Сповіщення»

Параметр	Опис
Email-сповіщення	Увімк/вимк
Push-сповіщення	Увімк/вимк
SMS-сповіщення	Увімк/вимк
Фільтр серйозності	Мінімальний рівень для сповіщень

Вкладка «Підключення»

Елемент	Опис
Список TIA Agent	Конфігурації підключень до TIA Portal
Додати	Новий TIA Agent (host, port)
Тест	Перевірка з'єднання
Індикатор статусу	Зелений/червоний

Вкладка «Ліцензія»

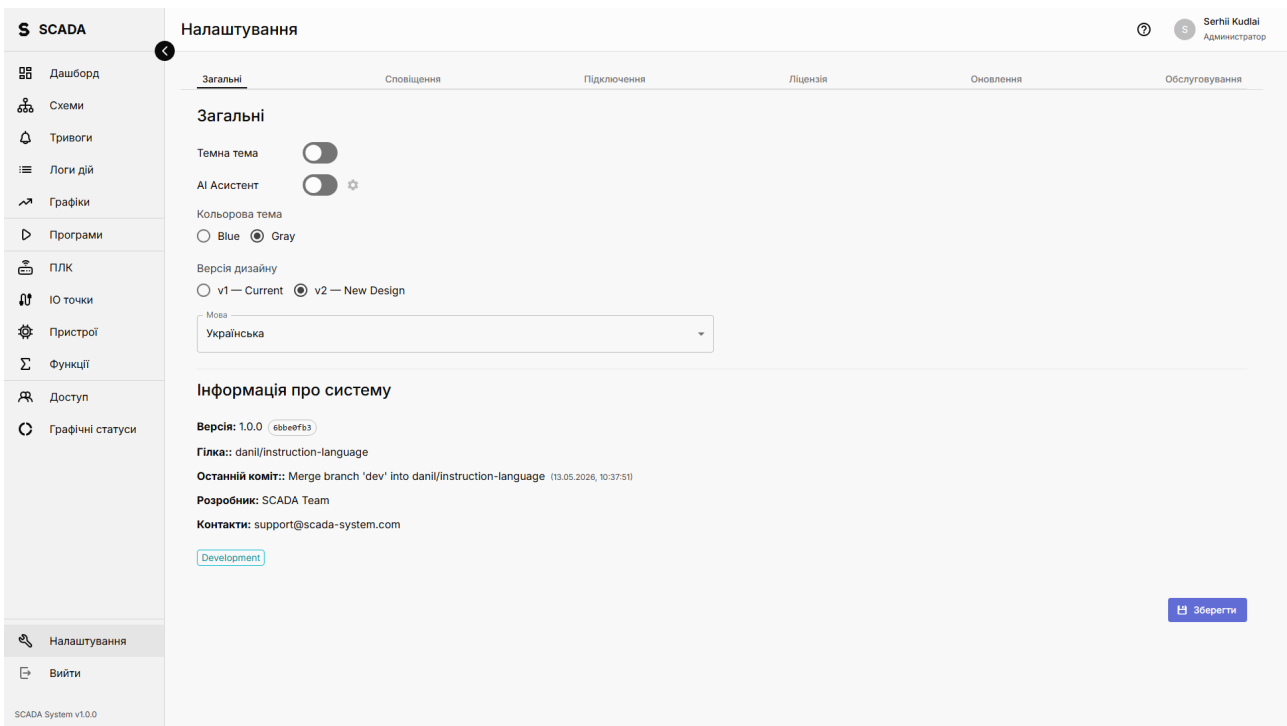
Поле	Опис
Статус	Valid / Invalid / Trial
Тип	Lifetime / Trial / Subscription
Днів залишилось	Зворотний відлік
Компанія	Назва компанії
Дата закінчення	Дата закінчення ліцензії
ID інсталяції	Унікальний ідентифікатор

Вкладка «Оновлення»

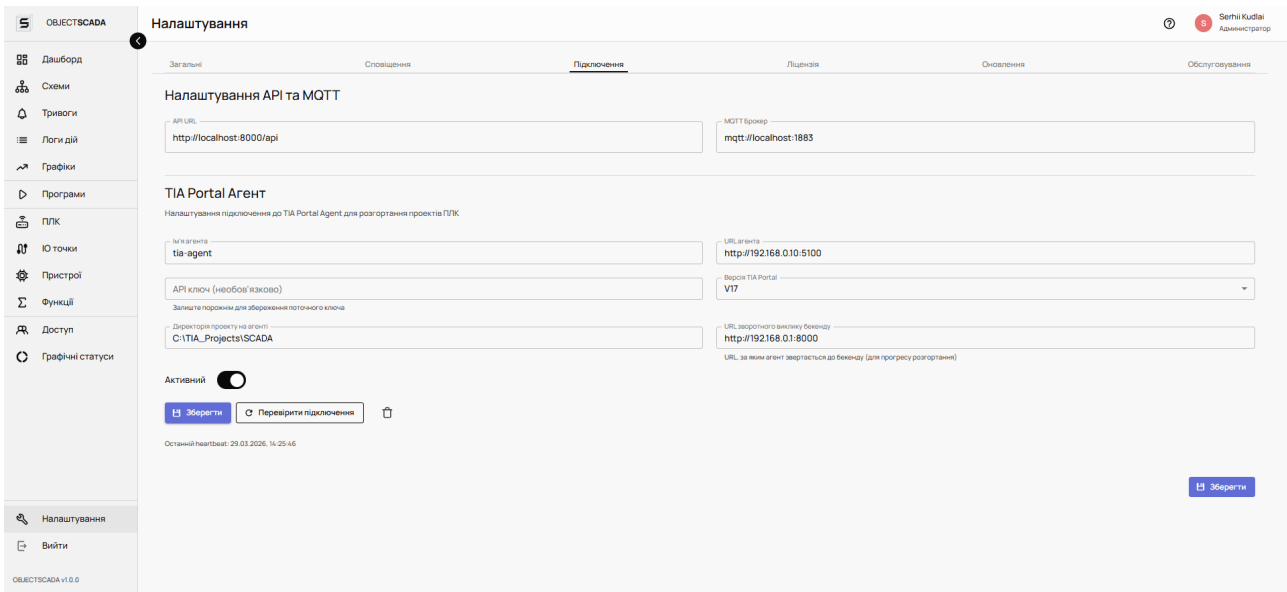
Елемент	Опис
Поточна версія	Номер поточної версії
Перевірити оновлення	Кнопка перевірки
Завантажити оновлення	Кнопка завантаження
Прогрес	Індикатор прогресу оновлення
Авто-оновлення	Перемикач

Вкладка «Обслуговування»

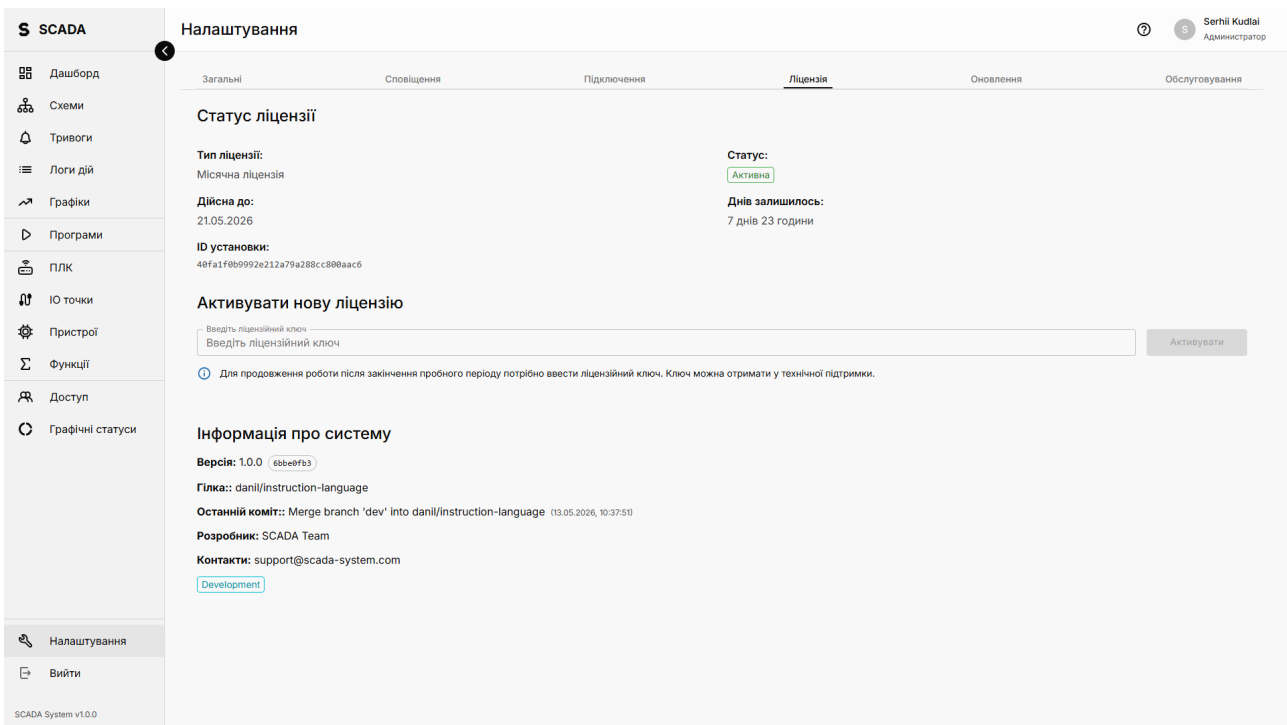
Елемент	Опис
База даних	Операції обслуговування БД
Кеш	Очищення кешу
Логи	Керування логами
Здоров'я системи	Перевірки стану
Бекап/Відновлення	Резервне копіювання та відновлення
Очищення диска	Видалення тимчасових файлів
Smoke-тест	Перевірка працездатності



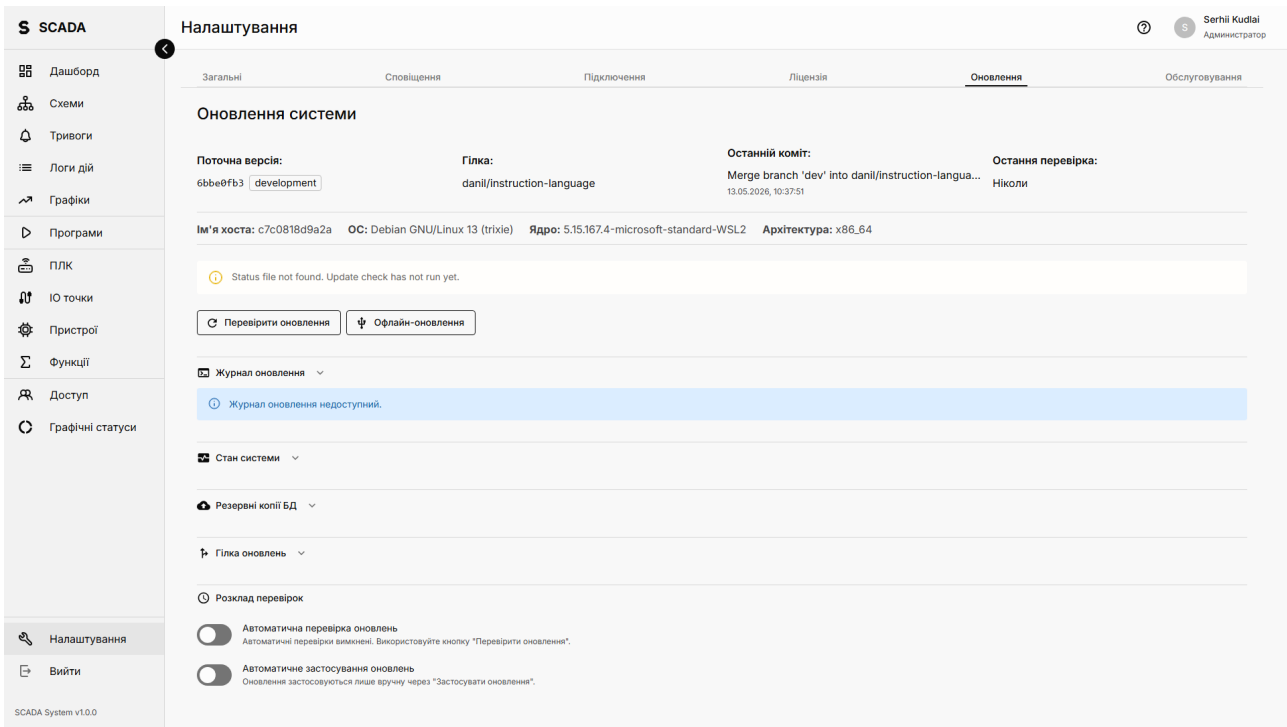
Вкладка General: перемикач Dark Mode, Color Picker (6 кольорів), dropdown мови, перемикач Design V1/V2, toggle AI Assistant.



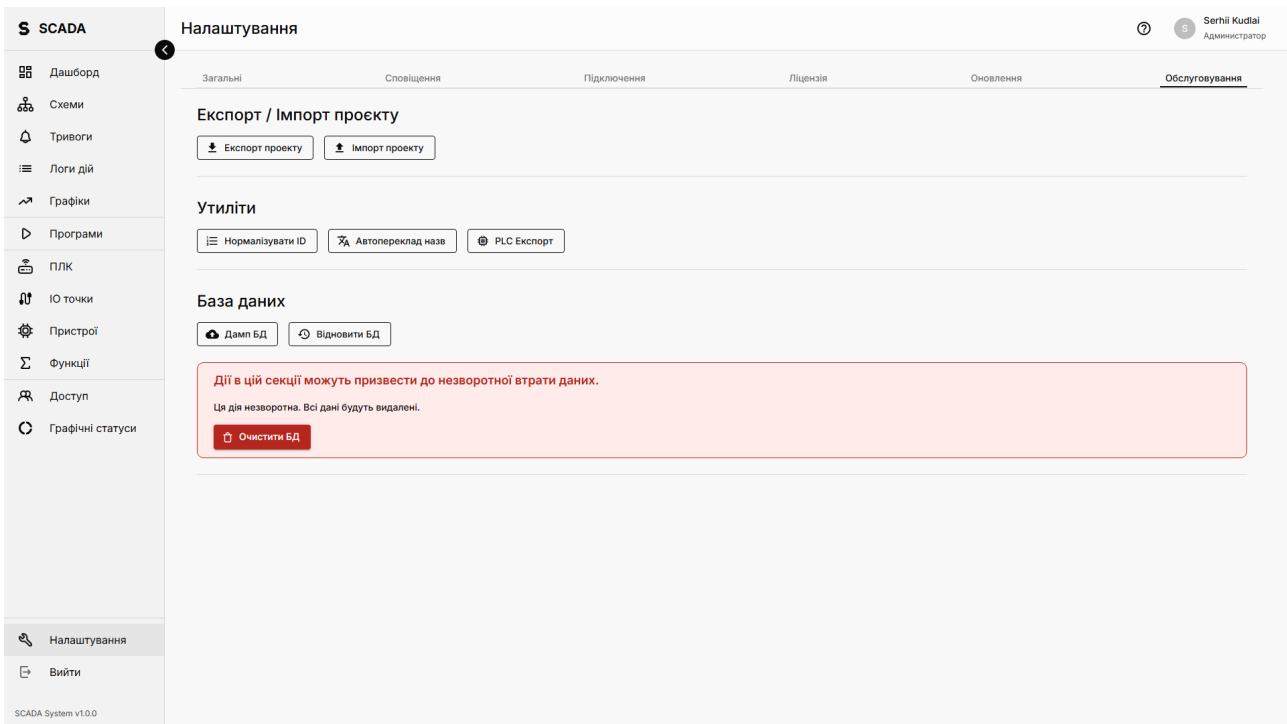
Вкладка Connections: список TIA Agent з host/port, кнопки Add/Edit/Delete/Test, індикатори статусу.



Вкладка License: статус ліцензії (зелений бейдж), тип, днів залишилось, компанія, дата закінчення.



Вкладка Updates: поточна версія, кнопка Check for Updates, прогрес-бар, логи оновлення.



Вкладка Maintenance: кнопки Backup DB, Restore DB, Clear Cache, Disk Cleanup, Smoke Test з описами.

Технічна документація

API налаштувань користувача:

Метод	Endpoint	Опис
GET	/api/user/language/	Отримати мову
POST	/api/user/language/	Встановити мову

Метод	Endpoint	Опис
GET	/api/user/dark-mode/	Отримати темний режим
POST	/api/user/dark-mode/	Встановити темний режим

API ліцензії:

Метод	Endpoint	Опис
GET	/api/license/status/	Статус ліцензії
POST	/api/license/activate/	Активація
POST	/api/license/generate/	Генерація ключа

API оновлень:

Метод	Endpoint	Опис
GET	/api/system/update-status/	Статус оновлення
POST	/api/system/update-check/	Перевірка оновлень
POST	/api/system/update-apply/	Застосувати оновлення
GET	/api/system/update-progress/	Прогрес
POST	/api/system/update-cancel/	Скасування
POST	/api/system/update-rollback/	Відкат
GET	/api/system/update-history/	Історія оновлень
GET	/api/system/update-backups/	Резервні копії
POST	/api/system/update-backups/{filename}/restore/	Відновлення з бекапа
POST	/api/system/disk-cleanup/	Очищення диска
GET	/api/system/smoke-test/	Smoke-тест
GET	/api/system/diagnostics/	Діагностика (завантаження)

WebSocket: Сповіщення про оновлення

```
{
  "type": "system_update",
  "event": "update_progress",
  "progress_percent": 45,
  "current_step": "Applying migrations...",
  "status": "in_progress"
}
```

26. AI-асистент

Для користувача

Вбудований AI-помічник для керування SCADA через природну мову.

Елементи:

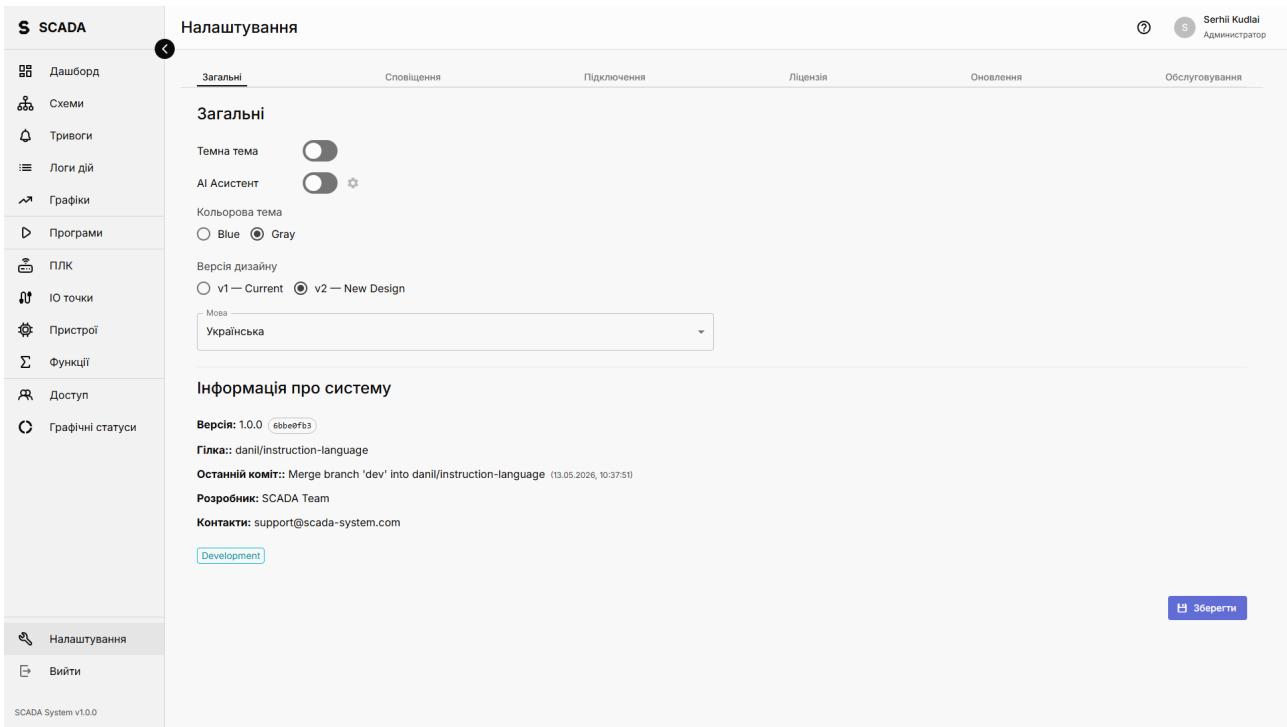
Елемент	Опис
Кнопка AI (FAB)	Плаваюча кнопка в правому нижньому куті екрана
Панель чату	Висувна/перетягувана панель
Список сесій	Бокова панель з історією бесід
Поле введення	Текстове поле для повідомлень
Кнопка надсилання	Надіслати повідомлення
Кнопка прикріпити	Завантажити зображення (P&ID креслення)
Налаштування	Конфігурація AI (модель, API ключ)

Можливості AI:

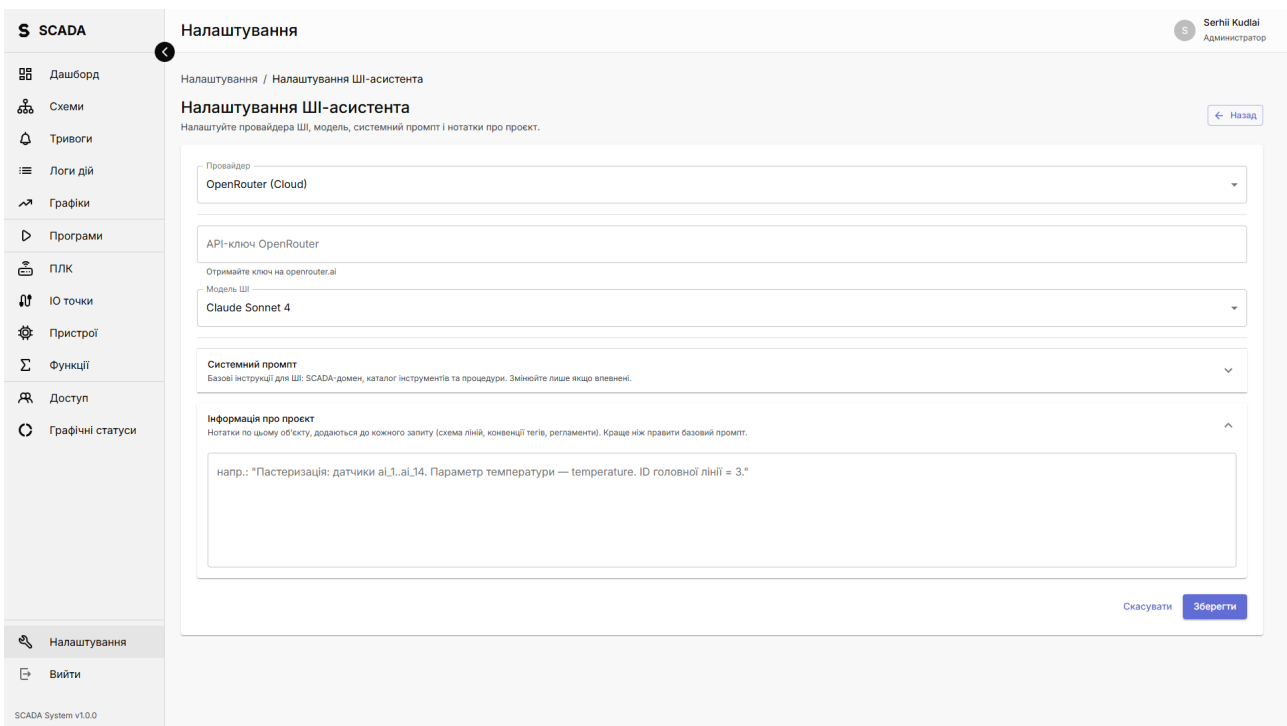
Функція	Опис
Керування пристроями	«Увімкни мотор MOT_001»
Запит інформації	«Які пристрої зараз в аварії?»
Створення елементів	«Створи новий аналоговий вхід AI_010»
Аналіз P&ID	Завантаження зображення → розпізнавання елементів → пропозиція розміщення
Допомога	Відповіді на питання про систему

Налаштування AI (AiSettingsDialog):

Поле	Опис
API ключ	Ключ OpenRouter (шифрується AES/Fernet)
Модель	Вибір LLM моделі
Температура	Креативність відповідей (0-1)
Max tokens	Максимальна довжина відповіді



Плаваюча кнопка AI в правому нижньому куті: кругла, з іконкою робота/чату.



Панель чату AI: ліворуч — список сесій, праворуч — листування (повідомлення користувача та відповіді AI), знизу — поле введення з кнопками Send і Attach.

Технічна документація

AI API (FastAPI microservice, BaseURL: `/ai-api/v1`):

Метод	Endpoint	Опис
POST	/sessions	Створити сесію

Метод	Endpoint	Опис
GET	/sessions	Список сесій
DELETE	/sessions/{id}	Видалити сесію
PATCH	/sessions/{id}	Перейменувати
GET	/sessions/{id}/messages	Повідомлення сесії
POST	/chat/{id}/messages	Надіслати повідомлення (SSE stream)
POST	/images/analyze	Аналіз зображення (multipart)
POST	/images/apply	Застосувати пропозицію
GET	/settings	Отримати налаштування
PATCH	/settings	Оновити налаштування
GET	/health	Перевірка здоров'я

SSE події (при stream=true):

Event	Опис
token	Токен тексту відповіді (стрімінг)
tool_call	AI викликає інструмент (devicetools, schematools, query_tools)
tool_result	Результат виклику інструмента
error	Помилка
done	Відповідь завершено

Формат SSE:

```

event: token
data: {"content": "Мотор "}

event: token
data: {"content": "MOT_001 "}

event: tool_call
data: {"name": "get_device_status", "arguments": {"device_id": "mot_001"}}

event: tool_result
data: {"result": {"status": "running", "value": 1}}

event: token
data: {"content": "зараз працює."}

event: done
data: {}

```

27. Експорт PLC-проєкту

Для користувача

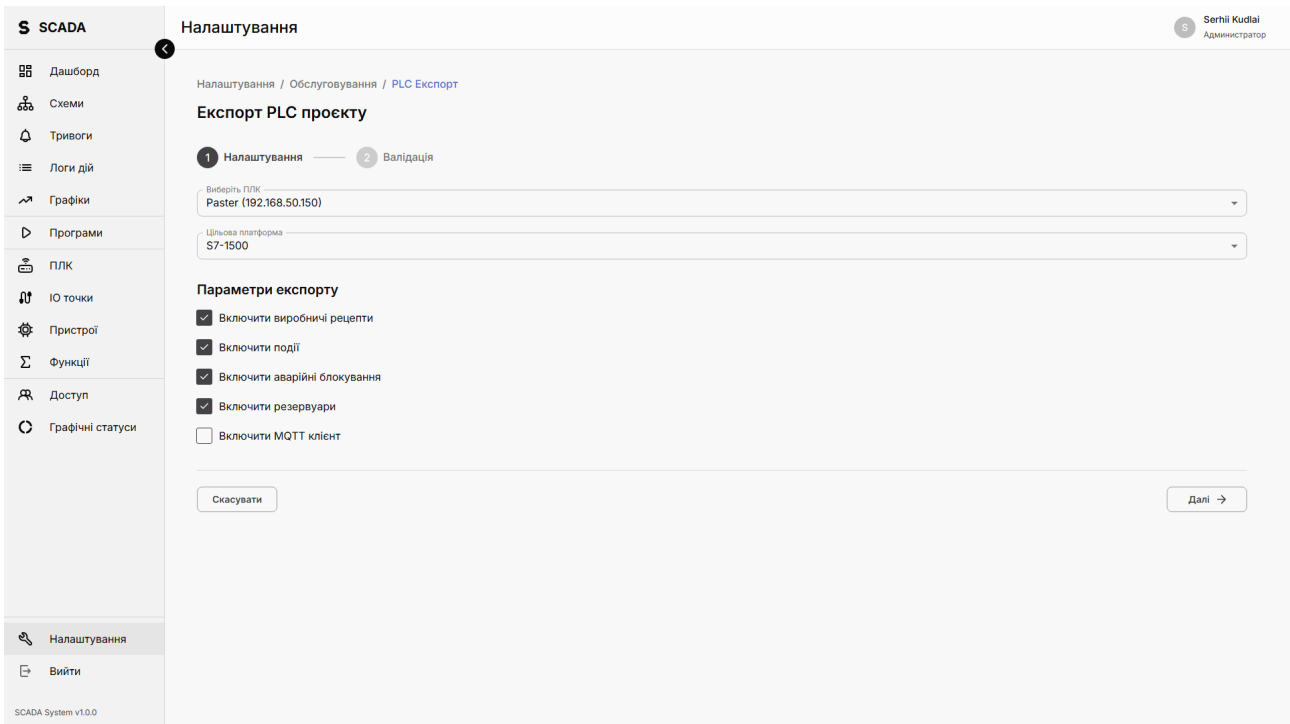
Генерація проєкту для Siemens TIA Portal на основі конфігурації SCADA.

Діалог експорту (PlcExportDialog):

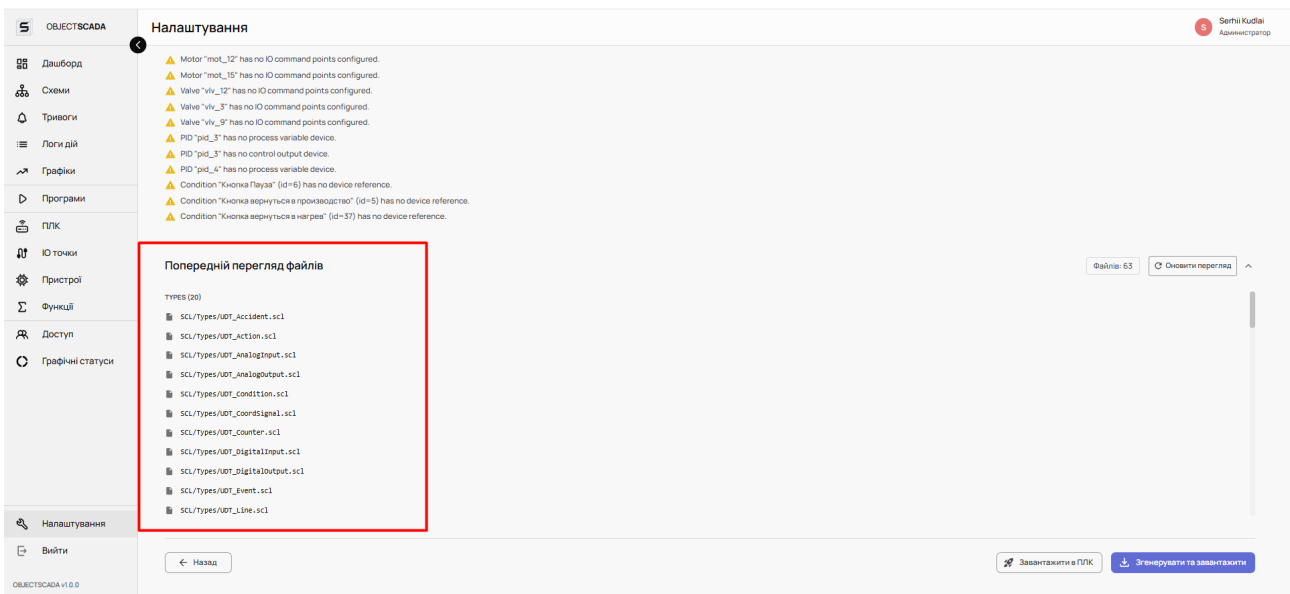
Елемент	Опис
Вибір ПЛК	Цільовий контролер
Тип ПЛК	S7-1200 / S7-1500
Включити рецепти	Чекбокс — додати рецепти
Включити події	Чекбокс — додати події
Включити аварії	Чекбокс — додати аварії
Включити танки	Чекбокс — додати танки
Включити MQTT	Чекбокс — додати MQTT комунікацію
MQTT Broker IP	IP-адреса MQTT брокера
MQTT Topic Prefix	Префікс топіків
Прев'ю	Попередній перегляд коду SCL
Генерувати	Завантажити ZIP-файл з проєктом
Deploy to TIA	Розгорнути через TIA Agent

Склад проєкту, що генерується:

- SCL код функціональних блоків (Motor FB, Valve FB, PID FB, AI FB, AO FB, DI FB, DO FB, COS FB, Counter FB, Timer FB, Tank FB)
- Sequence FB (автомат станів)
- Event/Accident FBs
- Data Blocks (Instance DBs, Recipe DBs)
- OB1 Main (виклик усіх FB)
- XML Tag Tables (SimaticML)



Діалог експорту: dropdown ПЛК, radio S7-1200/S7-1500, чекбоксы Include (Recipes, Events, Accidents, Tanks, MQTT), поля MQTT config, кнопки Preview/Generate/Deploy.



Прев'ю SCL коду: ліворуч — дерево файлів (FBs, DBs, Tag Tables), праворуч — вміст SCL файлу з підсвічуванням синтаксису.

Технічна документація

API PLC Export:

Метод	Endpoint	Опис
GET	/api/plc-export/preview/	Прев'ю коду
POST	/api/plc-export/validate/	Валідація
POST	/api/plc-export/generate/	Генерація ZIP
GET	/api/plc-export/address-map/	Карта адрес

28. Експорт та імпорт проєкту

Для користувача

Повний експорт/імпорт проєкту SCADA для перенесення між установками.

Діалог експорту (ExportProjectDialog):

Елемент	Опис
Чекбокси об'єктів	Вибір типів для експорту: Devices, Schemas, Sequences, Recipes, Events, Accidents, Options, Trends, Alarms, Users, Roles
Select All	Вибрати всі
Включити графіки	Експортувати дані трендів
Пошук	Фільтрація об'єктів
Експорт	Завантажити файл експорту
Прогрес	Індикатор прогресу

Кнопки:

Кнопка	Опис
Експорт проєкту	Почати експорт вибраних об'єктів
Імпорт проєкту	Завантажити файл → прев'ю → застосувати
Бекап БД	Створити резервну копію бази даних
Відновити БД	Відновити з резервної копії

Діалог експорту: список типів з чекбоксами, кнопка *Select All*, toggle «Include Graphs», кнопка *Export*, прогрес-бар.

Технічна документація

API проекту:

Метод	Endpoint	Опис
GET	/api/project/export-models/	Список моделей, що експортуються
POST	/api/project/export/	Експорт
POST	/api/project/import/preview/	Прев'ю імпорту
POST	/api/project/import/	Імпорт
POST	/api/project/backup-db/	Бекап БД
POST	/api/project/restore-db/	Відновлення
POST	/api/project/normalize-ids/	Нормалізація ID
POST	/api/project/clear-database/	Очищення БД
POST	/api/project/export-graph-data/	Експорт графіків
POST	/api/project/import-graph-data/	Імпорт графіків
GET	/api/project/graph-data-stats/	Статистика графіків

29. Танки

Для користувача

Танки — ємності з автоматичним розрахунком об'єму на основі даних сенсорів.

Параметри танка:

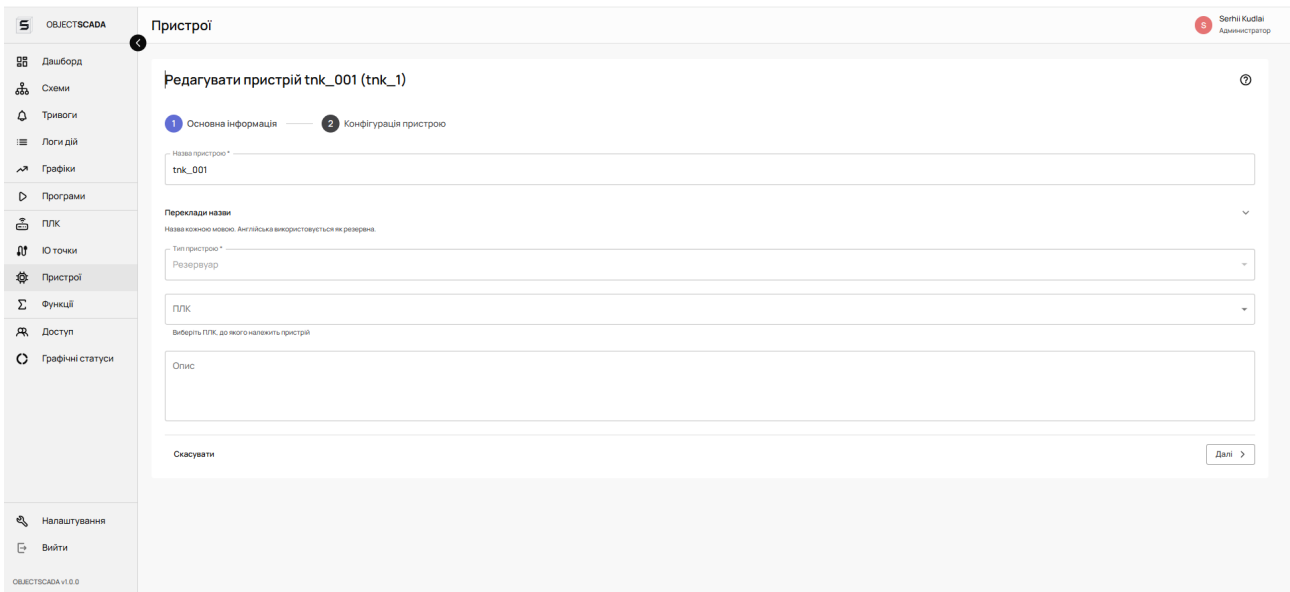
Поле	Опис
Сенсор рівня	Прив'язка до AI пристрою (рівнемір)
Сенсор тиску верх	Тиск у верхній частині (опціонально)
Сенсор тиску низ	Тиск у нижній частині (для ΔP)
Аварійні рівні	HH, H, L, LL — пороги аварійних сигналів

Методи розрахунку об'єму:

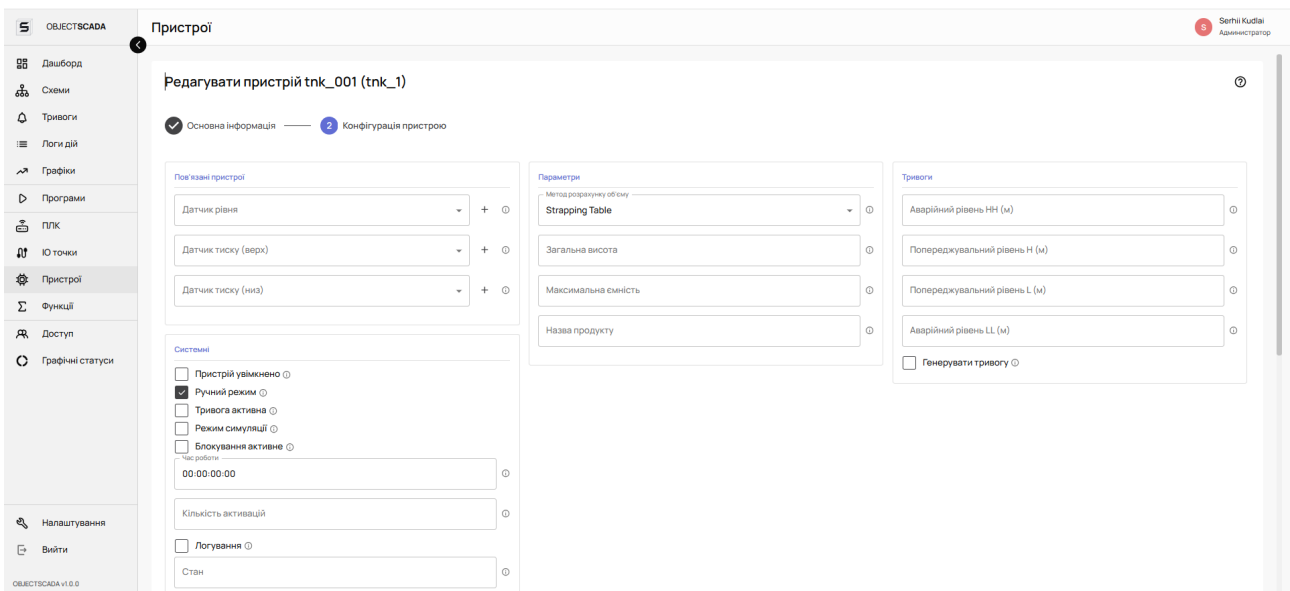
Метод	Опис
Strapping Table	Таблиця градування (інтерполяція <code>sensor_value</code> → <code>volume</code>)
Geometric	Геометричний (секції: циліндр, конус, купол, тарілка)

Конструктор танка на схемі:

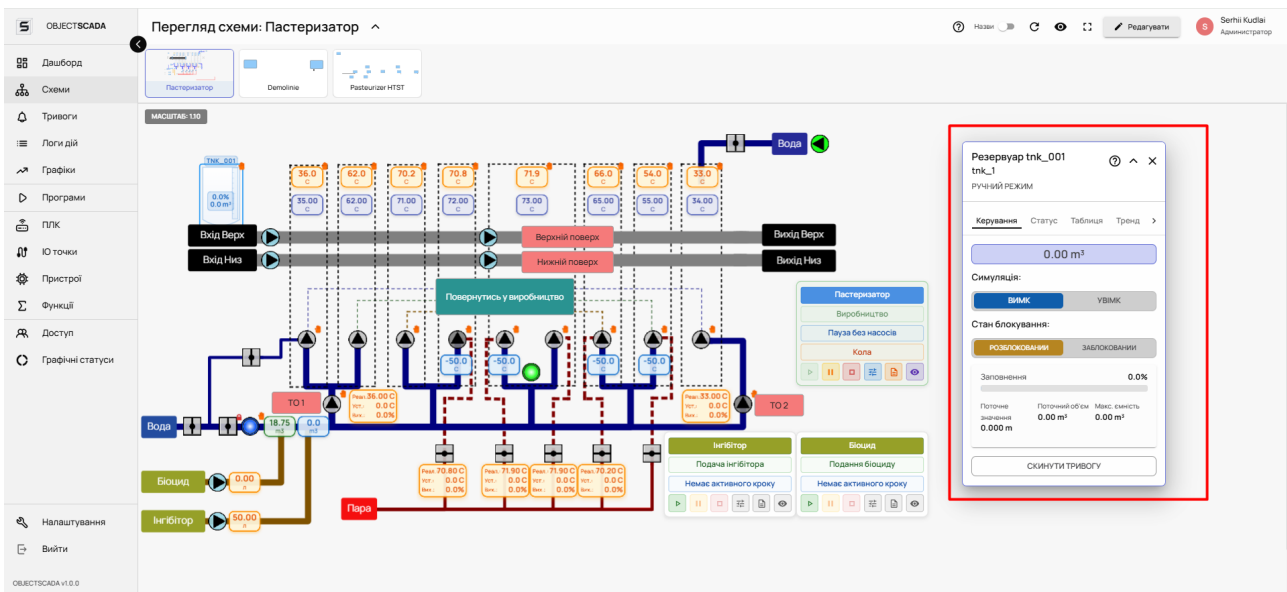
Елемент	Опис
Вибір пристрою	Dropdown існуючих TNK пристроїв
Тип нижньої кришки	FLAT / DOME / DISH / CONE
Тип верхньої кришки	FLAT / DOME / DISH
Ширина / Висота	Візуальні розміри
SVG Прев'ю	Попередній перегляд форми танка
Авто-розміщення сенсорів	Сенсори розміщуються вздовж правої стінки



Форма танка: прив'язка сенсорів (Level Sensor dropdown, Pressure Top, Pressure Bottom), пороги аварій (HH, H, L, LL — числові поля).



Таблиця градування: дві колонки — Sensor Value і Volume, кнопки Add Row, Delete Row, Import CSV, Export CSV.



Конструктор танка: ліворуч — налаштування (пристрій, кришки, розміри), праворуч — SVG прев'ю з вибраною формою.

Технічна документація

API танків:

Метод	Endpoint	Опис
GET	/api/tanks/	Список танків
POST	/api/tanks/	Створити
PUT	/api/tanks/{id}/	Оновити
DELETE	/api/tanks/{id}/	Видалити
POST	/api/tanks/{id}/calculate_volume/	Перерахувати об'єм
POST	/api/tanks/{id}/import_strapping_table/	Імпорт таблиці (CSV)
GET	/api/tanks/{id}/export_strapping_table/	Експорт таблиці
GET	/api/tanks/{id}/volume_at_level/	Об'єм при рівні (?sensor_value=X)
GET	/api/tank-sections/	Секції (?tank=ID)
GET	/api/tank-strapping-entries/	Рядки таблиці (?tank=ID)

Rust Worker — періодичний розрахунок (кожні 1000 мс):

Файл `scada-worker/src/tasks/tank_volume.rs` — опитує сенсори рівня та тиску, розраховує об'єм за градувальною таблицею або геометрією, записує результат у БД і передає через Redis.

Навігація додатка (Sidebar)

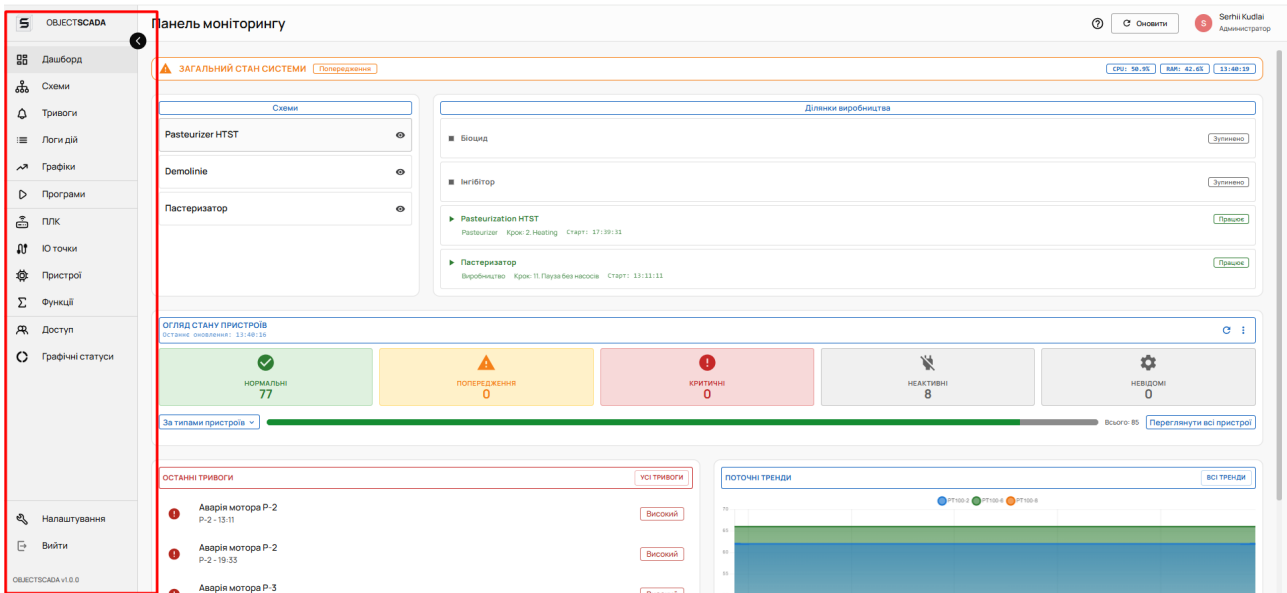
Для користувача

Бокова панель навігації доступна на всіх сторінках. Пункти відображаються залежно від дозволів користувача.

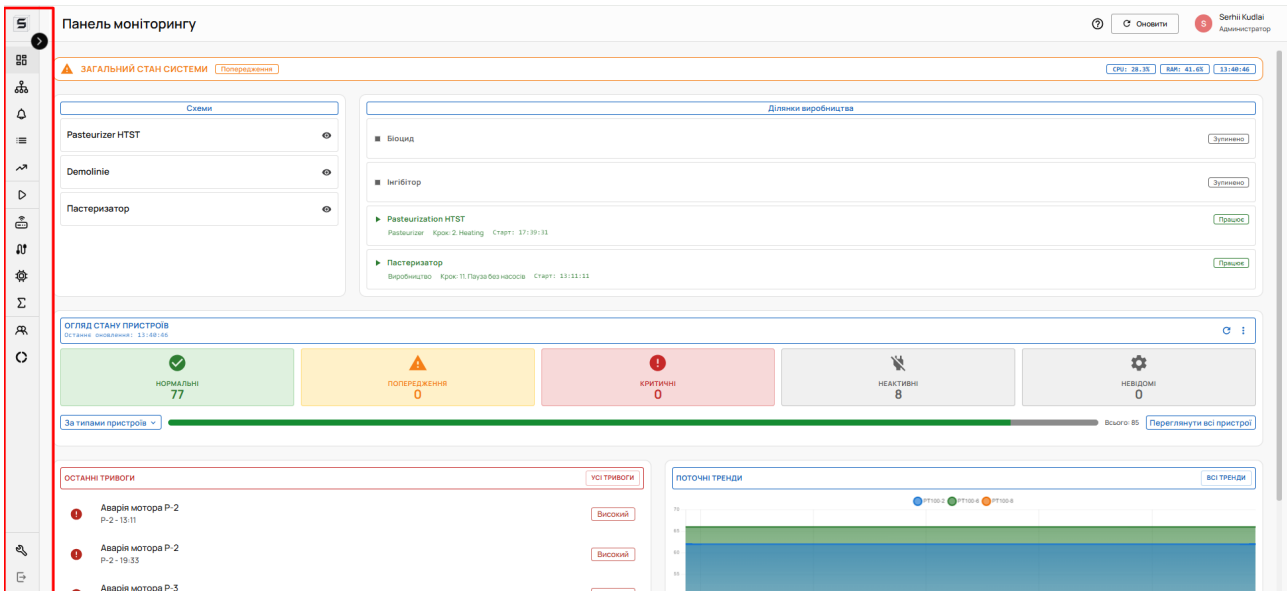
Пункт меню	Сторінка	Потрібний дозвіл
Dashboard	Головна панель	— (доступно всім)
Schemas	Мнемосхеми	—
Alarms	Аварії	—
Activity Logs	Журнал активності	—
Trends	Тренди	—
Sequences	Послідовності	sequences_recipes
PLCs	ПЛК	plcs
I/O Points	Точки вводу-виводу	io_points
Devices	Пристрої	devices
Functions	Мат. функції	devices
Users	Користувачі	users
Roles	Ролі	roles
Graphical Statuses	Граф. статуси	graphical_statuses
Settings	Налаштування	— (доступно всім)
Logout	Вихід	—

Додаткові елементи бокової панелі:

Елемент	Опис
Аватар користувача	Відображення фото та імені
Згорнути/Розгорнути	Зменшити панель до іконок
Номер версії	Поточна версія застосунку



Бокова панель у розгорнутому вигляді: аватар користувача зверху, пункти меню з іконками та підписами, кнопка Logout знизу.



Бокова панель у згорнутому вигляді: тільки іконки пунктів меню.

Глобальні елементи інтерфейсу

Для користувача

Шапка (Header):

Елемент	Опис
Заголовок сторінки	Поточна назва розділу
Кнопка допомоги	Запуск навчального туру
Індикатор оновлень	Бейдж за наявності оновлень
Індикатор з'єднання	Статус WebSocket/MQTT підключення
Контекстні дії	Кнопки, специфічні для поточної сторінки

Сповіщення (Snackbar):

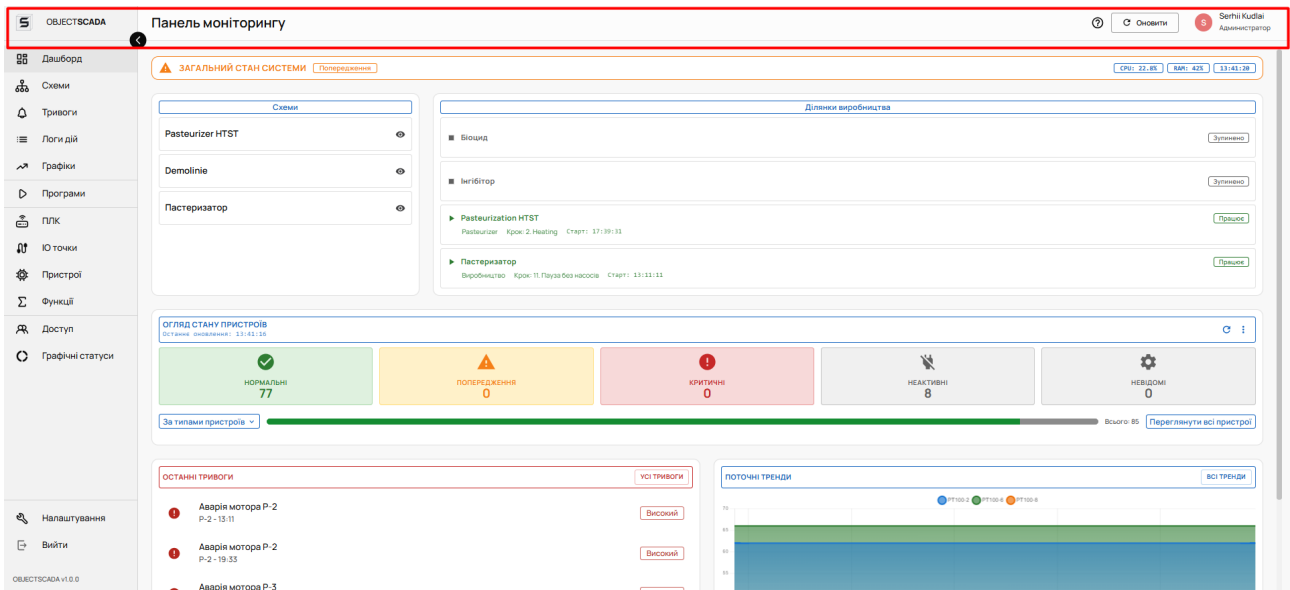
Тип	Опис
Success (зелений)	Операція успішна
Error (червоний)	Помилка
Warning (жовтий)	Попередження
Info (синій)	Інформація

Банери:

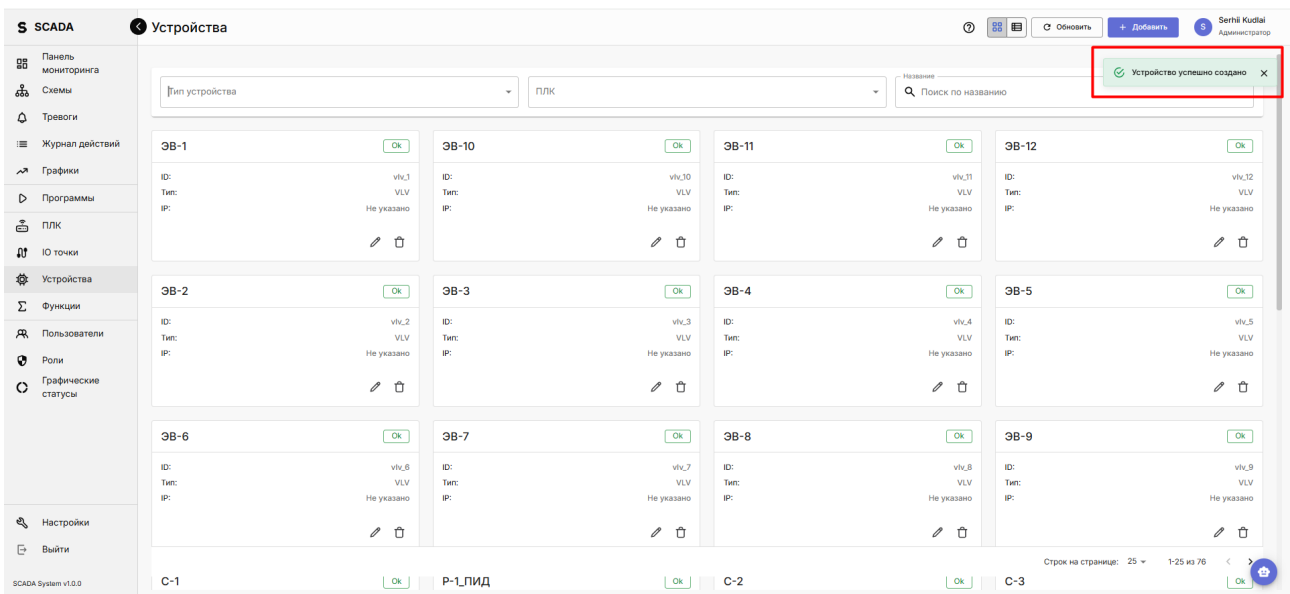
Банер	Опис
Connection Status	Жовтий банер при втраті з'єднання (після 15 с)
Maintenance	Синій банер при увімкненому режимі обслуговування
Version Mismatch	Помаранчевий банер при невідповідності версій frontend/backend
Update Available	Зелений бейдж у шапці при доступному оновленні

Навчальні тури (Guided Tours):

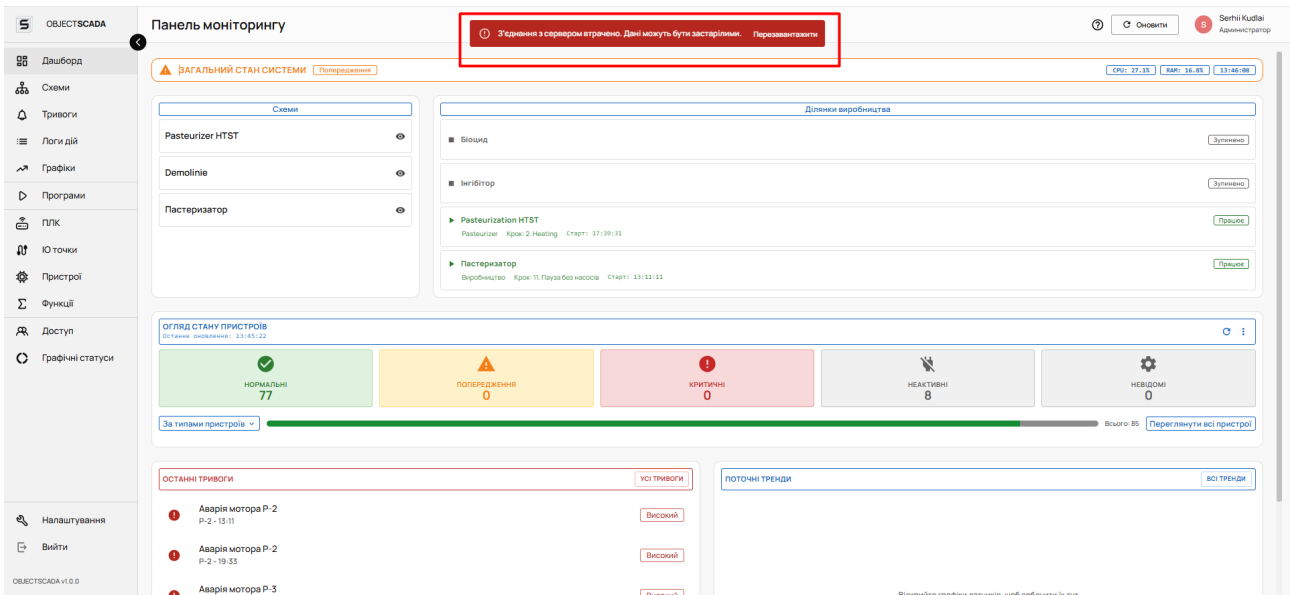
Доступні на кожній сторінці через кнопку «?» у шапці. Покрокове інтерактивне керівництво з підсвічуванням елементів та описом.



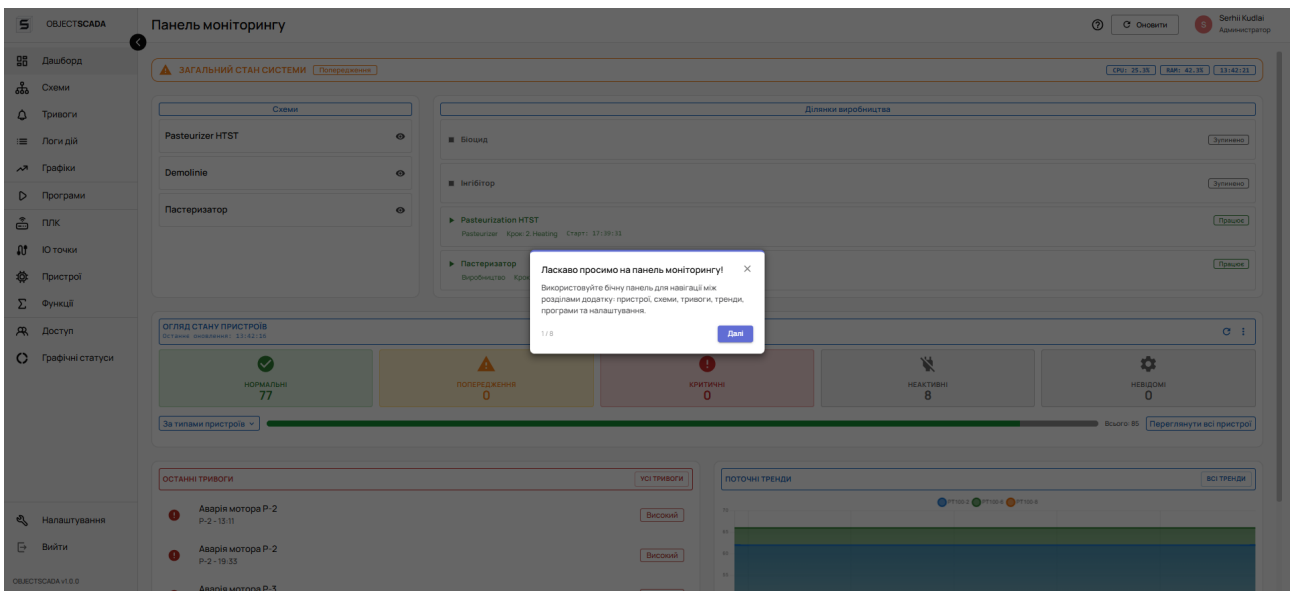
Шапка застосунку: заголовок сторінки ліворуч, кнопка туру (і), індикатор оновлень (червоний бейдж), статус з'єднання (зелена/червона крапка).



Приклади сповіщень: зелене «Device saved successfully», червоне «Error: Connection lost», жовте «Warning: License expires in 7 days».



Жовтий банер зверху: «Connection lost. Attempting to reconnect...» з анімацією.



Навчальний тур: підсвічений елемент з підказкою, кнопки «Next», «Back», «Skip», прогрес-бар.

Гарячі клавіші

Клавіша	Дія	Де працює
Ctrl+Z	Скасувати	Schema Builder
Ctrl+Y	Повторити	Schema Builder
Ctrl+S	Зберегти	Schema Builder
Delete	Видалити елемент	Schema Builder
Ctrl+A	Виділити все	Schema Builder
Ctrl+C	Копіювати	Schema Builder
Ctrl+V	Вставити	Schema Builder
Escape	Скасувати виділення / Закрити діалог	Скрізь
Колесо миші	Масштабування	Schema Builder / Viewer, Trends

Практичні приклади

Розділ містить наскрізні сценарії автоматизації, що показують, як пов'язати воедино пристрої, мнемосхему, послідовність, рецепт і події для реальних технологічних процесів.

Приклад 1. Автоматизація пастеризатора молока

Технологічне завдання. Нагрів молока до 75 °С, витримка 15 секунд, охолодження до 4 °С. Це класичний HTST-процес (High Temperature Short Time), широко застосовуваний у молочній промисловості.

Обладнання процесу:

Пристрій	ID	Тип	Призначення
Насос подачі	mot_00	Motor	Прокачування молока через теплообмінник
Клапан подачі пари	val_001	Valve	Подача пари в нагрівальну секцію
Клапан охолоджувача	val_002	Valve	Подача крижаної води
Датчик t° нагріву	ai_001	Analog Input	Температура після нагрівальної секції
Датчик t° витримки	ai_002	Analog Input	Температура після секції витримки
Датчик t° на виході	ai_003	Analog Input	Температура після охолоджувача
Регулятор нагріву	pid_001	PID Controller	Підтримання t° нагріву (уставка → val_001)
Регулятор охолодження	pid_002	PID Controller	Підтримання t° на виході (уставка → val_002)
Рівень у приймальному танку	tnk_001	Tank	Контроль заповнення
3-ходовий клапан повернення	val_003	Valve	Повернення недогрітого молока на повторний нагрів

Етап 1. Створення пристроїв

Створіть усі пристрої через Devices → «+». Для PID-регулятора pid_001:

```
POST /api/pid-controllers/
{
  "id": "pid_001",
  "name": "Регулятор нагріву",
  "plc": 1,
  "setpoint": 75.0,
  "kp": 2.5,
  "ki": 0.1,
  "kd": 0.05,
  "output_min": 0,
  "output_max": 100
}
```

Для танка з автоматичним розрахунком об'єму через strapping-таблицю:

```
POST /api/tanks/
{
  "id": "tnk_001",
  "name": "Приймальний танк",
  "plc": 1,
  "volume_method": "STRAPPING",
  "level_sensor": "ai_004",
  "alarm_hh": 95.0,
  "alarm_h": 85.0,
}
```

```
"alarm_1": 15.0,  
"alarm_11": 5.0  
}
```

Етап 2. Послідовність (Sequence)

Послідовність з 5 кроків описує повний цикл пастеризації:

№	Крок	Призначення	Умова переходу до наступного кроку
1	Idle	Очікування команди запуску	Опція «Start» (оператор)
2	Heating	Нагрів до уставки	ai_001 >= 75.0 протягом 5 с
3	Holding	Витримка при t°	Таймер 15 с
4	Cooling	Охолодження до 4 °C	ai_003 <= 4.0
5	Complete	Завершення	Повернення на Idle

Створення послідовності через API:

```
POST /api/sequences/
{
  "name": "Pasteurization HTST",
  "name_translations": {"uk": "Пастеризація HTST", "ru": "Пастеризация HTST"},
  "description": "Нагрів 75°C → витримка 15 с → охолодження 4°C"
}
```

Додавання кроків:

```
POST /api/sequence-steps/
{
  "sequence": 1,
  "order": 2,
  "name": "Heating",
  "name_translations": {"uk": "Нагрів"},
  "next_steps": {"3": [10]}
}
```

У полі next_steps ключ "3" — порядковий номер цільового кроку, значення [10] — список ID умов переходу (TransitionCondition).

Умова переходу «температура ≥ 75°C»:

```
POST /api/transition-conditions/
{
  "id": 10,
  "type": "device",
  "device_id": "ai_001",
  "content_type": "analoginput",
  "operator": ">=",
  "value": 75.0,
  "duration_seconds": 5
}
```

Важливо. duration_seconds гарантує, що умова виконується безперервно протягом зазначеного часу — це захищає від хибних переходів через поодинокі викиди сигналу.

Етап 3. Виробничий рецепт

Винесемо параметри, що залежать від конкретного продукту, у змінні рецепта — щоб ту саму послідовність використовувати для різних молочних продуктів (цільне молоко, вершки, знежирене молоко).

Змінні рецепта:

```
POST /api/production-recipe-variables/  
{  
  "name": "target_heating_temp",  
  "name_translations": {"uk": "Цільова t° нагріву"},  
  "sequence": 1,  
  "default_value": 75.0,  
  "min_value": 70.0,  
  "max_value": 90.0  
}
```

```
POST /api/production-recipe-variables/  
{  
  "name": "holding_time_sec",  
  "name_translations": {"uk": "Час витримки, с"},  
  "sequence": 1,  
  "default_value": 15.0,  
  "min_value": 10.0,  
  "max_value": 30.0  
}
```

Рецепт для цільного молока:

```
POST /api/production-recipes/  
{  
  "name": "Whole Milk 3.5%",  
  "name_translations": {"uk": "Молоко цільне 3.5%"},  
  "sequence": 1,  
  "variables": {  
    "target_heating_temp": 75.0,  
    "holding_time_sec": 15.0  
  }  
}
```

Рецепт для вершків (жорсткіший режим):

```
POST /api/production-recipes/  
{  
  "name": "Cream 30%",  
  "name_translations": {"uk": "Вершки 30%"},  
  "sequence": 1,  
  "variables": {  
    "target_heating_temp": 85.0,  
    "holding_time_sec": 20.0  
  }  
}
```

Після активації рецепта уставка PID `pid_001.setpoint` автоматично прив'язується до змінної `target_heating_temp`, а таймер кроку Holding — до `holding_time_sec`.

Етап 4. Аварійні події

Визначимо дві критичні аварії:

1. Перегрів (понад 90 °C):

```
POST /api/accident-events/
{
  "name": "Overheat protection",
  "name_translations": {"uk": "Захист від перегріву"},
  "device_id": "ai_001",
  "content_type": "analoginput",
  "operator": ">",
  "value": 90.0,
  "severity_level": 4,
  "action": "STOP_SEQUENCE"
}
```

severity_level: 4 — CRITICAL, негайне припинення послідовності.

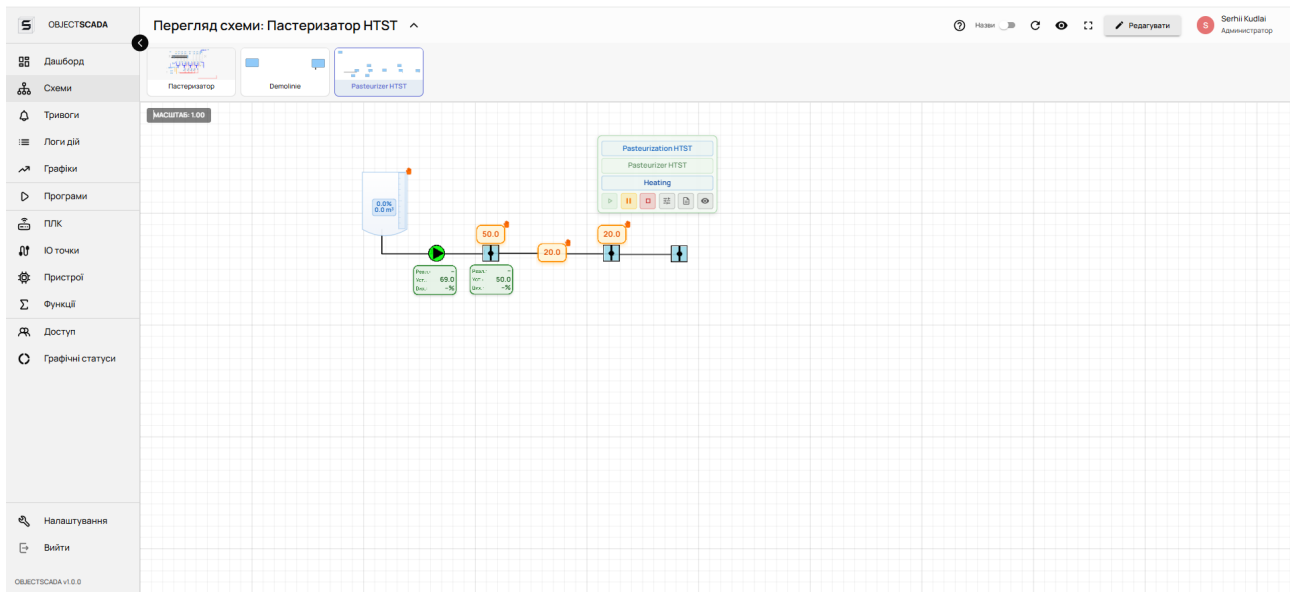
2. Недогрів на витримці (нижче 73 °C у режимі Holding):

```
POST /api/accident-events/
{
  "name": "Insufficient pasteurization",
  "name_translations": {"uk": "Недостатня пастеризація"},
  "device_id": "ai_002",
  "content_type": "analoginput",
  "operator": "<",
  "value": 73.0,
  "severity_level": 3,
  "action": "TRIGGER_DIVERT"
}
```

Коли спрацює ця аварія, система автоматично вмикає клапан повернення val_003 — недопастеризоване молоко піде на повторний нагрів.

Етап 5. Мнемосхема

- Створіть схему «Пастеризатор HTST».
- У Schema Builder розмістіть:
- Приймальний танк tnk_001 через Танк — у конструкторі оберіть форму FLAT / DOME і розміри 80×120.
- Насос mot_001, клапани val_001, val_002, val_003, PID pid_001, pid_002.
- Датчики ai_001, ai_002, ai_003.
- З'єднайте трубами: танк → насос → нагрівач (з val_001 і ai_001) → секція витримки (ai_002) → охолоджувач (val_002, ai_003) → вихід.
- Прив'яжіть послідовність: Послідовність → обрати «Pasteurization HTST».
- Додайте кнопки-опції «Start», «Pause», «Stop» (окремий блок Опція).



Мнемосхема пастеризатора: ліворуч приймальний танк з рівнем, від нього труба йде до насоса, далі через нагрівальну секцію з датчиком температури та клапаном пари, секцію витримки, охолоджувальну секцію з клапаном холодоносія. Праворуч — блок послідовності з підсвічуванням активного кроку «Heating».

Етап 6. Запуск процесу

- Відкрийте схему в Schema Viewer.
- На блоці послідовності оберіть активний рецепт — наприклад, «Whole Milk 3.5%».
- Натисніть Start — послідовність перейде з кроку Idle на Heating.
- Спостерігайте за процесом у реальному часі:
- Підсвічування активного кроку (жовтий → зелений при успішному переході).
- Поточне значення t° на датчику у рорірі.
- Відсоток відкриття val_001 (керується PID).
- При досягненні 75°C та утриманні 5 с відбудеться автоматичний перехід на Holding.
- Після закінчення 15 с — перехід на Cooling.
- При t° на виході $\leq 4^{\circ}\text{C}$ — Complete → повернення на Idle.

Етап 7. Аналіз після партії

- Відкрийте Trends → створіть графік.
- Додайте трендові лінії: ai_001, ai_002, ai_003, pid_001.output.
- Встановіть інтервал — останні 30 хвилин.
- Перевірте:
- Форму кривої нагріву (час виходу на режим).
- Стабільність температури витримки (коливання мають бути $\leq 0.5^{\circ}\text{C}$).
- Швидкість охолодження.
- Експортуйте графік у PNG або CSV для звіту.

Етап 8. Генерація PLC-проекту

- Відкрийте Dashboard → кнопка Експорт PLC (або Settings → PLC Export).
- У діалозі оберіть:
- ПЛК: PLC_DEMO
- Пристрої, послідовності, рецепти та аварії — усі.
- Натисніть Preview — система покаже список файлів, що генеруються:
- Motor_FB.scl, Valve_FB.scl, PID_FB.scl — функціональні блоки пристроїв

- Sequence_Pasteurization_FB.scl — автомат станів
- Accident_FB.scl — обробники аварій
- OB1_Main.scl — головна програма
- Tags_Inputs.xml, Tags_Outputs.xml — таблиці тегів для TIA Portal
- Validate — перевірить коректність конфігурації.
- Generate — сформує ZIP для імпорту в TIA Portal.

Приклад 2. CIP-мийка з керуванням партіями

Технологічне завдання. Автоматичне миття технологічного обладнання (танків, трубопроводів) після виробничої зміни. Багатоступеневий процес: попереднє обполіскування → лужна мийка → проміжне обполіскування → кислотна мийка → фінальне обполіскування.

Особливості прикладу:

- Використовується Batch Management — кожен цикл мийки фіксується як партія.
- Застосовується Production Line з етапами (stages).
- Оператор може пропустити окремі етапи через опції.
- Демонструється використання таймерів і лічильників.

Етап 1. Обладнання

ID	Тип	Призначення
tnk_cip_01	Tank	Танк лужного розчину (NaOH 2%)
tnk_cip_02	Tank	Танк кислотного розчину (HNO 1.5%)
tnk_water	Tank	Танк оборотної води
mot_cip_pump	Motor	Циркуляційний насос CIP
val_cip_supply	Valve	Клапан подачі в мийний контур
val_cip_return	Valve	Клапан зворотного потоку
val_alkali	Valve	Клапан подачі лугу
val_acid	Valve	Клапан подачі кислоти
val_water	Valve	Клапан подачі води
ai_cip_temp	Analog Input	Температура розчину
ai_cip_cond	Analog Input	Електропровідність (контроль концентрації)
cnt_cip_flow	Counter	Витратомір на подачі
tim_cip_stage	Timer	Таймер поточного етапу

Етап 2. Виробнича лінія та етапи

Створюємо Production Line з п'ятьма етапами — кожен матиме свою послідовність.

```
POST /api/production-lines/
{
  "name": "CIP Line 1",
  "name_translations": {"uk": "CIP-лінія №1"},
  "description": "Мийка танків виробництва 1"
}
```

Етапи (ProductionLineStage):

№	Назва	Послідовність	Дія з партією
1	Pre-rinse	seq_cip_rinse (вода 90 с)	TRANSFER на наступний етап
2	Alkali wash	seq_cip_alkali (NaOH 20 хв при 75 °C)	TRANSFER
3	Intermediate rinse	seq_cip_rinse	TRANSFER
4	Acid wash	seq_cip_acid (HNO 15 хв при 65 °C)	TRANSFER
5	Final rinse	seq_cip_rinse	COMPLETE

Поле `batch_transfer_action` на останньому кроці кожної послідовності керує переходом:

- TRANSFER — партія передається на наступний етап автоматично.
- COMPLETE — партія завершується.
- NONE — партія залишається на поточному етапі (ручне керування).

Етап 3. Послідовність «Лужна мийка»

Кроки:

- Fill — наповнення контуру лугом (`val_alkali = OPEN`, `mot_cip_pump = ON`).
- Heat — нагрів до 75 °C (`ai_cip_temp >= 75.0`).
- Circulate — циркуляція 20 хв (таймер `tim_cip_stage`).
- Drain — злив (`val_cip_return = OPEN` до каналізації).

Рецепт «Лужна мийка стандартна»:

```
POST /api/production-recipes/
{
  "name": "Alkali wash - standard",
  "production_line": 1,
  "variables": {
    "alkali_temp": 75.0,
    "alkali_duration_min": 20,
    "acid_temp": 65.0,
    "acid_duration_min": 15
  }
}
```

Зверніть увагу: рецепт прив'язаний не до послідовності, а до виробничої лінії (`production_line: 1`). Змінні доступні в усіх послідовностях етапів цієї лінії.

Етап 4. Опції для оператора

Додамо кнопки опцій, що дозволяють прискорити або модифікувати процес:

Опція	Опис	Ефект
SKIP_ACID	Пропустити кислотну мийку	При активації етап 4 пропускається (перехід одразу на етап 5)
DOUBLE_RINSE	Подвійне фінальне обполіскування	Етап 5 повторюється двічі
LOW_TEMP	М'який режим (знижена t°)	Змінні <code>alkali_temp = 60</code> , <code>acid_temp = 50</code>

Опції додаються через Sequence Options і розміщуються на мнемосхемі як кнопки.

Етап 5. Створення та запуск партії

- Відкрийте схему CIP-лінії в Schema Viewer.
- На елементі Production Line натисніть Створити партію — відкриється CreateBatchDialog.

- Заповніть:
- Batch name: CIP-2026-04-18-001
- Recipe: «Alkali wash — standard»
- Stage: Pre-rinse (початковий етап)
- Priority: Normal
- Create — партія потрапляє в чергу.
- Активуйте партію з панелі BatchManagementPanel → Start.

API-еквівалент:

```
POST /api/production-lines/1/create_batch/  
{  
  "name": "CIP-2026-04-18-001",  
  "recipe_id": 5,  
  "starting_stage_id": 1,  
  "priority": "NORMAL"  
}
```

Етап 6. Моніторинг партії

У панелі BatchManagementPanel відображається:

- Поточний етап (наприклад, «Alkali wash»).
- Прогрес усередині етапу (таймер зворотного відліку).
- Поточні показання ключових датчиків (t°, провідність, витрата).
- Загальний прогрес партії по всіх етапах.

Історія переходів зберігається в BatchStageHistory — пізніше можна відновити повну хронологію:

```
GET /api/batches/{id}/history/
```

Етап 7. Валідація успішної мийки

За стандартами HACCP/ISO 22000 потрібно документально підтвердити факт мийки. Використаємо аварійну подію у зворотний бік:

```
POST /api/events/  
{  
  "name": "CIP validation - conductivity check",  
  "device_id": "ai_cip_cond",  
  "operator": "<",  
  "value": 0.5,  
  "duration_seconds": 30,  
  "action": "LOG_EVENT"  
}
```

Ця подія логує в журнал момент, коли провідність фінального обполіскування впала нижче 0.5 мСм/см протягом 30 секунд — ознака повного видалення мийного засобу.

Журнал доступний у Activity Logs з фільтром за послідовністю/партією та може бути експортований у CSV для звітності.

Чек-ліст налаштування нового технологічного процесу

Використовуйте як шаблон при налаштуванні будь-якого нового процесу:

1. Проектування (до введення в систему):

- Намальовано P&ID схему процесу
- Складено список пристроїв з унікальними ID
- Визначено всі аналогові та дискретні сигнали
- Описано логіку роботи (кроки та умови переходів)
- Визначено аварійні пороги та дії при спрацюванні

2. Налаштування в SCADA:

- Створено ПЛК з коректною адресою
- Створено всі пристрої з прив'язкою до ПЛК
- Налаштовано параметри PID (Kp, Ki, Kd)
- Налаштовано Tank strapping-таблиці (якщо застосовно)
- Створено мнемосхему з розміщенням пристроїв та трубами
- Створено послідовність з кроками
- Створено умови переходів між кроками
- Створено змінні рецепта
- Створено хоча б один рецепт з дефолтними значеннями
- Налаштовано аварійні події з коректними severity

3. Тестування (до введення в промислову експлуатацію):

- Кожен пристрій керується вручну (Manual mode)
- Умови переходів спрацьовують коректно (використовуйте симуляцію — `POST /api/transition-conditions/{id}/simulate/`)
- Послідовність проходить усі кроки end-to-end у режимі Manual
- Аварії спрацьовують та зупиняють процес коректно
- Рецепти застосовуються — значення уставок оновлюються
- Тренди пишуть дані по всіх ключових змінних
- PLC-експорт проходить валідацію без помилок

4. Введення в експлуатацію:

- Операторам проведено інструктаж
- Виконано резервне копіювання проєкту (`POST /api/project/export/`)
- Призначено ролі та дозволи (оператор, майстер, інженер)
- Запущено першу партію на знижених параметрах для перевірки
- Оформлено документацію на робочому місці

Усунення несправностей і FAQ

Розділ містить типові проблеми, з якими стикаються користувачі, та способи їх вирішення.

З'єднання та автентифікація

Q: Постійно показується жовтий банер «Connection lost».

Можливі причини:

- Зупинився Rust-worker або Django-backend — перевірте: `docker-compose ps`, усі 6 контейнерів мають бути у статусі Up.
- Nginx не проксує WebSocket — відкрийте DevTools браузера → Network → WS. Якщо з'єднання закривається з кодом 1006 або 502 — проблема у nginx. Перевірте `nginx.conf`: мають бути `proxy_http_version 1.1` та заголовки `Upgrade / Connection`.
- Закінчився JWT-токен — вийдіть і зайдіть знову. У нормі токен оновлюється автоматично за 1 хвилину до закінчення.
- Мережевий фаєрвол блокує порт 8001 (Rust WebSocket). Система перейде на fallback Django WebSocket, але з затримкою. Відкрийте порт.

Q: Після оновлення з'являється помаранчевий банер «Version mismatch».

Frontend і backend розгорнуто різних версій. Оновіть браузер з очищенням кешу (Ctrl+Shift+R). Якщо не допомогло — перевірте `docker-compose.yml`: tag образу для frontend має збігатися з tag для backend та `scada-worker`.

Q: Не можу увійти — пише «Invalid credentials».

- Перевірте розкладку клавіатури (CAPS LOCK).
- Переконайтесь, що використовуєте `username`, а не `email` (якщо налаштування вимагає конкретного).
- Якщо пароль забуто та SMTP не налаштовано — увійдіть на сервер і скиньте пароль командою: ``bash docker-compose exec backend python manage.py changepassword``
- Перевірте, чи не заблоковано вашу IP-адресу в IP Access (адміністратор може увімкнути `whitelist`).

Мнемосхема та пристрої

Q: Пристрій на схемі залишається сірим, значення не оновлюються.

- PLC offline — перевірте статус у розділі PLCs. Якщо статус не Online → пристрій не отримує дані.
- Пристрій не прив'язано до PLC — відкрийте картку пристрою, перевірте поле PLC.
- WebSocket канал не підписано — перевірте в DevTools → Network → WS: має бути підключення `ws/schema/{id}/devices`. Якщо його немає — оновіть сторінку.
- У вас немає дозволу на пристрій — попросіть адміністратора перевірити ваші ролі.

Q: Танк показує об'єм 0, хоча датчик рівня працює.

- Не налаштовано `strapping`-таблицю (для методу STRAPPING). Відкрийте танк → вкладка «Strapping Table» → додайте мінімум 2 точки (`sensor_value` → `volume`).
- Невірно обрано метод розрахунку — перевірте поле `volume_method`: STRAPPING або GEOMETRIC. Для геометричного мають бути задані `TankSection` (циліндр, конус тощо).
- Rust-worker не оновлює об'єм — перевірте логи `docker-compose logs scada-worker | grep tank_volume`. За замовчуванням розрахунок раз на 1 секунду (`TANK_VOLUME_INTERVAL_MS`).

Q: Труба не анімує потік, хоча двигун увімкнено.

Увімкніть властивість Моніторинг процесу (Process Monitoring) у труби в правій панелі властивостей

Schema Builder. За замовчуванням вона вимкнена.

Q: При перетягуванні пристрій «стрибає» назад.

Це відбувається при одночасному WebSocket-оновленні. Якщо ви в Schema Builder — редагування блокує live-updates. Якщо ні — збережіть зміну (Ctrl+S) одразу після переміщення.

Послідовності та рецепти

Q: Послідовність не переходить на наступний крок, хоча умова виконана.

- Умова не прив'язана до кроку — відкрийте поточний крок, перевірте next_steps. Для цільового кроку має бути вказано ID умови.
- `duration_seconds` ще не закінчився — умова має триматись безперервно зазначений час. Подивіться в роруп умови поле «Continuous for».
- Умова симулюється з іншим значенням — перевірте прапор is_simulated у WebSocket-даних умови. Скиньте: POST /api/transition-conditions/clear_simulation/ з {"step_id": }.
- Поточне значення пристрою не надходить — пристрій offline (див. вище).

Q: Змінив рецепт, але уставка PID не змінилася.

Переконайтесь, що:

- Рецепт активовано — POST /api/production-recipes/{id}/select-production-recipe/.
- Змінна рецепта прив'язана до поля пристрою в редакторі послідовності (вкладка «Кроки» → для кожного кроку перевірте значення пристроїв).
- Послідовність не в стані is_raise = true — raise заморожує всі переходи та застосування значень.

Q: Хочу перевірити логіку без реального обладнання.

- Використовуйте симуляцію умов переходу — POST /api/transition-conditions/{id}/simulate/ з {"simulated_value": 80.0}. Умова оцінюватиме передане значення замість реального.
- Для аналогових входів — увімкніть Simulation mode: POST /api/analog-inputs/{id}/set_simulation/ з {"is_simulated": true, "simulated_value": 75.0}.
- Після тестування обов'язково вимкніть симуляцію — у промисловій експлуатації це небезпечно.

Партії (Batches)

Q: Партія «висить» на етапі і не переходить далі.

- Перевірте batch_transfer_action на останньому кроці послідовності поточного етапу — має бути TRANSFER (для переходу) або COMPLETE (для завершення).
- Перевірте, що наступний етап існує та активний (is_active: true).
- Перевірте логи Rust-worker — повідомлення batch transfer failed вкаже причину.
- Ручний трансфер: POST /api/batches/{id}/transfer/ з {"next_stage_id": }.

Q: Черга партій переплуталась за пріоритетом.

Сортування черги — за пріоритетом, потім за часом створення. Змінити пріоритет: POST /api/batch-queue/{id}/set_priority/ з {"priority": "HIGH"}. Можливі значення: LOW, NORMAL, HIGH, URGENT.

PLC-експорт

Q: При натисканні «Validate» з'являється список помилок.

Типові помилки валідації:

- Device ID contains invalid characters — ID має бути лише [a-z0-9_], без дефісів і пробілів.

- Sequence references missing device — пристрій видалено, але посилання в кроці залишилось. Перейдіть у редактор послідовності і виправте.
- PID output_max <= output_min — перевірте межі PID.
- Tank strapping entries must be monotonic — значення sensor_value мають зростати.
- No PLC assigned to device X — прив'яжіть пристрій до ПЛК.

Q: Згенерований SCL не компілюється в TIA Portal.

- Перевірте версію TIA Portal — генератор підтримує TIA Portal v16+.
- Переконайтесь, що імпортовано обидві XML-таблиці тегів (Inputs і Outputs) до імпорту SCL.
- Подивіться лог компіляції TIA Portal — помилка зазвичай вказує конкретний FB та рядок.

Тренди

Q: Графік порожній, хоча пристрій працює.

- Часовий діапазон замалий або в майбутньому — скиньте на «Last 1 hour».
- Timescale offline — перевірте GET /api/graphs/timescale-status/. Якщо повертає disconnected — TimescaleDB не запущено або недоступно.
- Data points не збираються — розділ Data Points → пристрій має мати активну підписку на збір.

Q: Тренд сильно гальмує при великому інтервалі.

Використовує даунсемплінг. Якщо даних забагато — зменште діапазон або інтервал агрегації. Для довгострокового аналізу експортуйте дані в CSV і обробіть поза системою.

AI-асистент

Q: AI-асистент повертає «Authentication failed».

- Перевірте API-ключ OpenRouter у Settings → AI Assistant. Ключі шифруються Fernet при збереженні — помилка означає або прострочений ключ, або пошкодження БД.
- Перевірте баланс на OpenRouter — при вичерпанні коштів запити відхиляються.
- Перевірте логи: docker-compose logs ai-assistant | tail -50.

Q: AI довго відповідає.

Модель виконує агентний цикл до 15 ітерацій з викликами інструментів. Складні запити («проаналізуй усі аварії за тиждень і запропонуй оптимізацію») можуть займати 30-60 секунд. Для швидких відповідей — короткі та конкретні питання.

Продуктивність

Q: Інтерфейс гальмує при роботі зі схемою з 100+ пристроями.

- Вимкніть анімацію потоку в труб, що не використовуються для процесу.
- У Settings → Performance увімкніть «Throttle non-critical updates» (оновлення не-активних пристроїв раз на 2 секунди замість 250 мс).
- Збільште RAM Docker (мінімум 4 ГБ для комфортної роботи).

Q: База даних швидко росте.

Основне джерело — journal активності та історичні тренди. Налаштуйте retention у Settings → Data Management:

- Activity Logs: за замовчуванням 90 днів.
- Trend data: налаштовується per-device через Data Points.

Безпека та найкращі практики

Цей розділ містить рекомендації щодо безпечної експлуатації системи. Дотримання знижує ризик інцидентів на виробництві та забезпечує відповідність стандартам якості (ISO 9001, HACCP, GMP).

Правила роботи з режимами Auto / Manual

Manual mode дає оператору пряме керування пристроєм в обхід послідовностей та аварійних блокувань. Це потужний інструмент — і небезпечний.

Правильне використання Manual:

- Тільки для пусконаладження та діагностики.
- Тільки на зупиненому обладнанні або в спеціально передбачених для цього станах.
- З обов'язковим усним узгодженням з начальником зміни на працюючому виробництві.
- З записом у журнал активності (це відбувається автоматично, але оператор має знати).

Неправильне використання Manual:

- Обхід захисних умов для прискорення процесу.
- Запуск обладнання у ручному режимі без знання поточного стану партії.
- Залишення обладнання у Manual після закінчення зміни.

Правило. По закінченні роботи з пристроєм у Manual — завжди повертайте його в Auto. Система не робить це автоматично.

Симуляція: тільки для тестів

Симуляція умов, аналогових входів та таймерів — інструмент для проєктувальника, не для оператора.

Сценарій	Дозволено?
Тестування логіки послідовності на стенді	Так
Перевірка спрацювання аварій у симуляторі	Так
«Пропуск» умови під час виробництва, щоб прискорити запуск	Категорично ні
Симуляція датчика, коли реальний зламався	Ні — виправте датчик або зупиніть процес

Будь-яка активна симуляція при працюючому виробництві — потенційна причина інциденту. Перед запуском промислової партії запустіть перевірку: `GET /api/transition-conditions/?is_simulated=true` — список не повинен містити нічого, що стосується активної послідовності.

Резервні копії та відновлення

Що резервувати:

- Конфігурація проєкту — пристрої, схеми, послідовності, рецепти, аварії. Робиться через `POST /api/project/export/` → ZIP-файл.
- База даних — повний дамп PostgreSQL (включає journal, історичні тренди, дані партій). Команда: ``bash docker-compose exec postgres pgdump -U scada scadadb > backup_$(date +%Y%m%d).sql``
- Користувачські завантаження — volume ai_uploads (зображення для AI-аналізу).
- Налаштування nginx та docker-compose.yml — зберігайте у Git.

Як часто:

- Конфігурація проєкту — перед будь-якою значущою зміною.
- База даних — щодня автоматично + перед оновленнями системи.
- Перед оновленням версії SCADA — обов'язково обидва типи бекапів.

Перевірка відновлення. Раз на квартал розгорніть бекап на тестовому стенді та переконайтеся, що він коректний. Бекап, який ніколи не перевірявся, — ненадійний бекап.

Керування доступом

Принцип мінімальних привілеїв. Кожен користувач отримує лише ті дозволи, які потрібні для його роботи.

Типова матриця ролей:

Роль	Дозволи
Оператор	Перегляд схем + керування пристроями в Auto + підтвердження аварій. Немає доступу до редагування.
Майстер	Все, що в оператора + запуск/зупинка партій + зміна рецептів зі списку дозволених.
Технолог	Все, що у майстра + створення/редагування рецептів та змінних.
Інженер АСУ	Все, що у технолога + редагування схем, послідовностей, пристроїв.
Адміністратор	Повний доступ, включно з керуванням користувачами та системними налаштуваннями.

р

Додатково:

- Per-schema access — окремі схеми можна закрити від користувачів, які не працюють на даній ділянці.
- IP-фільтрація — обмежте доступ лише з внутрішньої мережі підприємства (діапазон 192.168.0.0/16 або VPN).
- Регулярний аудит — раз на квартал перевіряйте список активних користувачів у Users. Видаляйте звільнених співробітників.

Оновлення системи

Послідовність коректного оновлення:

- Заплануйте вікно — оновлення потребує зупинки виробництва на 5-15 хвилин.
- Зробіть бекап (конфігурація + БД).
- Прочитайте CHANGELOG оновлення — можуть бути breaking changes в API або міграції.
- Запустіть dry-run — система перевірить міграції без застосування.
- Застосуйте оновлення — через Settings → System → Updates.
- Дочекайтеся smoke-тестів — система автоматично прожоне базові перевірки.
- Якщо щось зламалось — використайте Rollback (доступний протягом 24 годин після оновлення).

Ніколи не оновлюйте систему під час активної партії або за наявності нерозв'язаних критичних аварій.

Робота з аваріями

Що РОБИТИ при спрацюванні аварії:

- Не панікувати. Прочитайте опис аварії в Alarm Bar.
- Перейти до обладнання, на якому спрацювала аварія.
- Візуально оцінити стан (витік, перегрів, сторонній шум).
- Якщо небезпеки немає — натиснути Acknowledge в Alarms.

- Усунути причину.
- Натиснути Resolve після усунення.

Що НЕ РОБИТИ:

- Масово «квитувати» всі аварії без розгляду (resolve-all — тільки у крайніх випадках за рішенням майстра).
- Видаляти аварійні події з налаштувань під час роботи.
- Знижувати severity-level, щоб позбутися набридливого алярму — натомість налаштуйте коректні пороги.

Журнал активності

Журнал (Activity Log) — ваш захист та джерело істини. Він автоматично фіксує:

- Входи та виходи користувачів.
- Усі команди керування пристроями (Start, Stop, Open, Close).
- Зміни рецептів та послідовностей.
- Спрацювання аварій.
- Запуски/зупинки/трансфери партій.

Рекомендації:

- Зберігайте журнал мінімум 1 рік (вимога багатьох стандартів якості).
- Регулярно експортуйте в CSV для архіву.
- При розслідуванні інциденту — першою справою дивіться журнал за період ± 1 година навколо події.

Рекомендації з іменування

Єдина конвенція іменування сильно спрощує життя — особливо при зростанні проекту.

Пристрої:

- Префікс за типом: mot_ (motor), val_ (valve), ai_ (analog input), pid_, tnk_ (tank).
- Нумерація з провідними нулями: mot_001, mot_002 (до mot_099).
- Групування за ділянкою: val_tank1_in, val_tank1_out.

Послідовності:

- Префікс процесу: seq_pasteurize_htst, seq_cip_alkali.

Рецепти:

- З ім'ям продукту та параметрами: Milk_Whole_3.5%, Cream_30%_HotFill.

Глосарій

Довідник термінів, що використовуються в системі та в цій документації.

Загальна автоматизація

SCADA (Supervisory Control And Data Acquisition) — система диспетчерського керування та збору даних. Клас програмних систем для моніторингу та керування технологічними процесами.

ПЛК / PLC (Programmable Logic Controller) — програмований логічний контролер. Промисловий комп'ютер, що виконує логіку керування обладнанням у реальному часі.

HMI (Human-Machine Interface) — інтерфейс «людина-машина». У цій системі HMI — це Schema Viewer.

P&ID (Piping and Instrumentation Diagram) — технологічна схема з позначенням трубопроводів та контрольно-вимірювальних приладів.

I/O — Input/Output, точки вводу-виводу. Фізичні або логічні канали зв'язку між ПЛК та обладнанням.

Tag — тег, іменований сигнал ПЛК. У SCADA відповідає адресі, за якою читається/записується значення (наприклад, DB10.DBDO для Siemens).

Пристрої

AI (Analog Input) — аналоговий вхід. Приймає безперервний сигнал (наприклад, 4-20 мА, 0-10 В) та перетворює в інженерну величину (температура, тиск).

AO (Analog Output) — аналоговий вихід. Видає безперервний сигнал для керування (уставка ЧРП, положення клапана).

DI / DO (Digital Input / Output) — дискретний вхід/вихід. Логічний сигнал 0/1 (кнопка, кінцевик, лампа, пускач).

PID (Proportional-Integral-Derivative) — ПІД-регулятор. Алгоритм автоматичного регулювання з трьома складовими: пропорційною (Kp), інтегральною (Ki) та диференційною (Kd).

Setpoint (SP) — уставка, цільове значення регульованої величини.

Process Value (PV) — поточне значення регульованої величини, виміряне датчиком.

VSD (Variable Speed Drive) — частотно-регульований привод (ЧРП). Керує швидкістю обертання електродвигуна.

COS (Coordination Signal) — координаційний сигнал. Логічна змінна, що використовується для міжпроцесної синхронізації (наприклад, «лінія 1 готова до передачі партії»).

Counter — лічильник (імпульсів, подій). Використовується для підрахунку витрати, обертів, циклів.

Interlock — блокування. Умова, що забороняє виконання дії (наприклад, «не запускати насос при закритому клапані»).

Танки

Strapping table (калібрувальна таблиця) — таблиця відповідності показань датчика рівня об'єму рідини в танку. Використовується для танків нестандартної форми.

HH / H / L / LL — High High / High / Low / Low Low. Чотири рівні аварійних порогів для танків: критично

високий, високий, низький, критично низький.

Hydrostatic level — гідростатичний розрахунок рівня за перепадом тисків (ΔP) між верхнім та нижнім датчиком тиску в танку.

Послідовності та рецепти

Sequence — послідовність. Автомат станів, що описує послідовність кроків технологічного процесу.

Step — крок. Один стан послідовності.

Transition — перехід. Зв'язок між двома кроками з умовою виконання.

Transition Condition — умова переходу. Тип: Device (за значенням пристрою), Timer (за таймером), Option (за опцією оператора).

Recipe — рецепт. Конкретні значення пристроїв для кожного кроку послідовності.

Production Recipe — виробничий рецепт. Набір змінних для конкретного продукту.

Recipe Variable — змінна рецепта. Параметр, значення якого залежить від типу продукту.

Option — опція. Керуючий сигнал від оператора (кнопка Start / Pause / Skip).

Партії та виробничі лінії

Batch (партія) — одиниця продукції, що проходить через виробничий процес. Принцип взято зі стандарту ISA-88.

Production Line (виробнича лінія) — набір обладнання для виробництва продукції.

Production Line Stage (етап лінії) — один етап обробки на лінії (наприклад, «мийка» → «сушка» → «пакування»).

Batch Transfer — передача партії з одного етапу на наступний. Керується полем `batch_transfer_action` на кроці послідовності.

Batch Queue — черга партій, що очікують запуску на лінії.

ISA-88 — міжнародний стандарт для batch control у періодичних виробництвах.

PLC-експорт та Siemens

TIA Portal — Totally Integrated Automation Portal. Середовище розробки Siemens для конфігурування ПЛК.

SCL (Structured Control Language) — мова програмування ПЛК Siemens (аналог Pascal за стандартом IEC 61131-3).

FB (Function Block) — функціональний блок. Програмна одиниця в ПЛК зі своєю логікою та екземплярними даними.

DB (Data Block) — блок даних. Структура для зберігання змінних у пам'яті ПЛК.

OB (Organization Block) — організаційний блок. Точка входу програми ПЛК.

OB1 — головний циклічний OB, викликається кожен скан-цикл.

SimaticML — XML-формат для імпорту/експорту проектів у TIA Portal.

GSD-файл (Generic Station Description) — описувач апаратного модуля PROFIBUS/PROFINET.

WebSocket та API

REST API — Representational State Transfer. Протокол HTTP-запитів для CRUD-операцій з даними.

WebSocket — двонаправлений протокол поверх TCP, що дозволяє push-оновлення від сервера до клієнта.

JWT (JSON Web Token) — токен автентифікації. Містить дані користувача у підписаному вигляді.

SSE (Server-Sent Events) — технологія односторонньої потокової передачі від сервера до клієнта. Використовується AI-асистентом для поточкових відповідей.

Delta update — дельта-оновлення. Передача лише змінених полів об'єкта замість повного стану.

Batching — пакетування. Групування кількох оновлень в одне повідомлення для зниження мережевого навантаження.

Виробничі стандарти

HACCP (Hazard Analysis and Critical Control Points) — аналіз ризиків та критичні контрольні точки. Методологія забезпечення безпеки харчової продукції.

GMP (Good Manufacturing Practice) — належна виробнича практика. Стандарт керування виробництвом, особливо у фармацевтиці та харчовому виробництві.

CIP (Cleaning In Place) — мийка на місці. Автоматичне миття технологічного обладнання без розбирання.

HTST (High Temperature Short Time) — високотемпературна короткочасна пастеризація (72-75°C, 15-20 секунд).

ISO 9001 — міжнародний стандарт системи менеджменту якості.

Додаток А: Довідник WebSocket-каналів

Канал	Опис	Напрямок
ws/device/{id}	Статус пристрою	Server → Client
ws/device/{id}/control	Команди пристрою	Bidirectional
ws/schema/{id}/devices	Усі пристрої схеми	Server → Client
ws/schema/{id}/pipes	Трубопроводи схеми	Server → Client
ws/schema/{id}/sequences	Послідовності схеми	Server → Client
ws/schema/{id}/batches	Партії схеми	Bidirectional
ws/schema/{id}/accidents	Аварії схеми	Server → Client
ws/sequence/{id}	Статус послідовності	Server → Client
ws/sequences/all	Усі послідовності	Server → Client
ws/pipe/{id}	Статус трубопроводу	Server → Client
ws/system	Системні метрики	Server → Client
ws/plc/status	Статус ПЛК	Server → Client
ws/device-status	Глобальний статус	Server → Client
ws/cos/all	Координаційні сигнали	Server → Client
ws/math-functions	Мат. функції	Server → Client
ws/activity-log	Журнал активності	Server → Client
ws/accidents	Усі аварії	Server → Client

Rust WebSocket сервер (порт 8001, проксі `/rws/ws/`):

- Read-only для масових оновлень пристроїв
- Автоматичний fallback на Django WebSocket
- Автентифікація через JWT у query-параметрі URL

Додаток В: Довідник API (зведена таблиця)

Ресурс	Base URL	CRUD	Дод. дії
Auth	/api/auth/	—	login, register, me, refresh
Users	/api/users/		set_password, upload-avatar, roles
Roles	/api/roles/		permissions
Permissions	/api/permissions/		categories, actions
Devices (all)	/api/devices/	—	alldevices, batchwebsocket_update
Analog Inputs	/api/analog-inputs/		setmode, setsimulation, setinterlock, setmanual_value
Analog Outputs	/api/analog-outputs/		(аналогічно)
Digital Inputs	/api/digital-inputs/		(аналогічно)
Digital Outputs	/api/digital-outputs/		(аналогічно)
Motors	/api/motors/		(аналогічно)
Valves	/api/valves/		(аналогічно)
PID Controllers	/api/pid-controllers/		(аналогічно)
Counters	/api/counters/		(аналогічно)
Timers	/api/timers/		(аналогічно)
COS	/api/coordination-signals/		(аналогічно)
VSD	/api/variable-speed-drives/		(аналогічно)
Tanks	/api/tanks/		calculatevolume, import/exportstrapping
Tank Sections	/api/tank-sections/		—
Tank Strapping	/api/tank-strapping-entries/		—
PLCs	/api/plcs/		status, set-run-mode
Schemas	/api/schemas/		add/remove devices, sequences, options, production lines
Pipes	/api/pipes/		—
Sequences	/api/sequences/		steps, transitions, positions
Sequence Steps	/api/sequence-steps/		—
Transition Conditions	/api/transition-conditions/		simulate, clear_simulation
Recipes	/api/recipes/		—
Production Recipes	/api/production-recipes/		select-production-recipe
Recipe Variables	/api/production-recipe-variables/		—
Events	/api/events/		set_active
Accident Events	/api/accident-events/		—
Accidents	/api/accidents/		acknowledge, resolve, resolve-all
Sequence Options	/api/sequence-options/		—
Production Lines	/api/production-lines/		activate, deactivate, create_batch
Production Stages	/api/production-line-stages/		—
Batches	/api/batches/		start, pause, resume, cancel, transfer
Batch Queue	/api/batch-queue/		reorder, set_priority
Pulse Modules	/api/pulse-modules/		start, stop

Ресурс	Base URL	CRUD	Дод. дії
Math Functions	/api/math-functions/		calculate, validate, test
I/O Points (AI)	/api/points/analog-inputs/		validate, test
I/O Points (AO)	/api/points/analog-outputs/		(аналогічно)
I/O Points (DI)	/api/points/digital-inputs/		(аналогічно)
I/O Points (DO)	/api/points/digital-outputs/		(аналогічно)
Graphical Statuses	/api/graphical-statuses/		upload-icon, list-icons
Graphs	/api/graphs/		data, timescale-status
Data Points	/api/data-points/		—
HW Catalog	/api/hw-catalog/		import-gsd
HW Stations	/api/hw-stations/		auto-assign-addresses
HW Slots	/api/hw-slots/		bulk-update
HW Network	/api/hw-network/		—
TIA Deploy	/api/tia/	—	deploy, auto-addresses, address-mapping
TIA Agent Config	/api/tia-agent-config/		check-connection, agent-status
Activity Logs	/api/user-activity-logs/	R/D	recent, statistics, clear
IP Access	/api/ip-access/allowed-ips/		checkip, myips
License	/api/license/	—	status, activate, generate
System	/api/system/	—	info, health, updates, maintenance
Project	/api/project/	—	export, import, backup, restore
PLC Export	/api/plc-export/	—	preview, validate, generate
AI Sessions	/ai-api/v1/sessions		messages
AI Chat	/ai-api/v1/chat	—	messages (SSE)
AI Images	/ai-api/v1/images	—	analyze, apply
AI Settings	/ai-api/v1/settings	R/U	—

Додаток С: Коди помилок і аварій

Довідник кодів та статусів, з якими взаємодіє користувач і які трапляються в журналах, WebSocket-повідомленнях та API-відповідях.

Рівні серйозності аварій (Severity levels)

Поле severity_level моделі AccidentType та accident_event.severity_type.

Код	Найменування	Колір	Рекомендована реакція
1 / LOW	Низький	Блакитни	Інформаційне повідомлення. Не потребує негайних дій. Записується в журнал для статистики.
2 / MEDIUM	Середній	Жовтий	Потребує уваги оператора найближчим часом. Виробництво може продовжуватися.
3 / HIGH	Високий	Помаранчевий	Потребує негайної уваги. Можлива деградація якості продукту або ризик пошкодження обладнання.
4 / CRITICAL	Критичний	Червоний	Негайна зупинка послідовності. Можливий ризик для людей або обладнання.

L

Типи серйозності подій (`SeverityType` для Events):

Код	Найменування	Застосування
INFO	Інформаційне	Статусні повідомлення (старт, завершення етапу)
WARNING	Попередження	Відхилення від норми без загрози
ERROR	Помилка	Критична проблема, що потребує втручання

Статуси аварій (Accident Status)

Поле status моделі Accident. Життєвий цикл аварії:

NEW → ACKNOWLEDGED → IN_PROGRESS → RESOLVED → CLOSED

Код	Найменування	Опис
NEW	Нова	Щойно виникла, не підтверджена. Відображається в Alarm Bar.
ACKNOWLEDGED	Підтверджена	Оператор підтвердив отримання (POST /api/accidents/{id}/acknowledge/). Прибирається з топ-пріоритету, але залишається активною.
IN_PROGRESS	В роботі	Ведеться усунення причини.
RESOLVED	Усунена	Причину усунено (POST /api/accidents/{id}/resolve/).
CLOSED	Закрит	Фінальний статус після RESOLVED + певний час (для звітності).

D

Статуси послідовностей

Поля моделі Sequence:

Поле	Тип	Значення	Опис
is_running	bool	true	Послідовність активна, кроки перемикаються.
is_pause	bool	true	Призупинена, переходи заморожено, пристрої утримують поточний стан.

Поле	Тип	Значення	Опис
is_stop	bool	true	Зупинена, всі пристрої повернуто в безпечний стан.

Комбінації:

- is_running=true, is_pause=false, is_stop=false — робота.
- is_running=true, is_pause=true, is_stop=false — на паузі (можна відновити).
- is_running=false, is_pause=false, is_stop=true — повністю зупинена.

Статуси партій (Batch Status)

Код	Найменування	Опис
PENDING	В очікуванні	У черзі, не запущена
RUNNING	Виконується	Активна партія
PAUSED	Призупинена	Тимчасово зупинена оператором
COMPLETED	Завершена	Успішно пройшла всі етапи
CANCELLED	Скасована	Перервана вручну
FAILED	Помилка	Перервана через критичну аварію

Пріоритети (Priority):

Код	Порядок у черзі
URGENT	Обробляється першою
HIGH	2
NORMAL	3 (за замовчуванням)
LOW	4

Режими пристроїв (Device Mode)

Застосовуються до двигунів, клапанів, PID, AI/AO/DI/DO:

Код	Опис
AUTO	Керування через послідовність / логіку системи
MANUAL	Пряме керування оператором
SIMULATION	Симульоване значення (тільки для тестування)
INTERLOCK	Заблоковано блокуванням (не можна змінювати значення)

Коди помилок API

Типові HTTP-коди та їх значення в контексті системи:

Код	Значення	Типова причина
200 OK	Успіх	Запит виконано
201 Created	Створено	Створено новий об'єкт
204 No Content	Успіх без тіла	Видалення
400 Bad Request	Невірний	

запит

Валідація не пройшла. Тіло відповіді містить {"field": ["error msg"]}

Код	Значення	Типова причина
401 Unauthorized	Не авторизовано	JWT-токен відсутній або невалідний
403 Forbidden	Доступ заборонено	Автентифікований, але немає прав на дію
404 Not Found	Не знайдено	ID не існує
409 Conflict	Конфлікт	Наприклад, спроба створити пристрій з існуючим ID
423 Locked	Заблоковано	Об'єкт заблоковано іншим процесом (наприклад, збереження схеми під час запущеної послідовності)
500 Internal Error	Помилка сервера	Баг або виняток. Перевірте логи docker-compose logs backend
502 Bad Gateway	Помилка	Nginx не може достукатися до backend / Rust worker
503 Service Unavailable	Недоступно	Режим обслуговування або перевантаження

Коди помилок WebSocket

При закритті WebSocket-з'єднання можуть повертатися коди:

Код	Опис
1000	Нормальне закриття
1001	Сервер йде (перезапуск)
1006	Аномальне закриття (втрата зв'язку). Frontend автоматично перепідключається.
1008	Порушення політики (наприклад, невалідний JWT у query-параметрі)
1011	Внутрішня помилка сервера
4001	Custom: Authentication failed
4003	Custom: Forbidden (немає прав на канал)
4004	Custom: Channel not found

Типи умов переходу (Transition Condition Type)

Код	Опис
device	За значенням пристрою (порівняння з порогом)
timer	За тривалістю (після входу в крок)
option	За натисканням опції оператором

Оператори порівняння (для type=`device`):

Символ	Значення
==	Дорівнює
!=	Не дорівнює
>	Більше
<	Менше
>=	Більше або дорівнює
<=	Менше або дорівнює

Дії з партією на кроці (batch_transfer_action)

Код	Опис
NONE	Без дії, партія залишається на поточному етапі
TRANSFER	Передати партію на наступний етап лінії
COMPLETE	Завершити партію (фінальний статус)

Типи форм кришок танків

Для поля SchemaDevice.configuration.tank_visual:

Нижні (`bottom_type`):

Код	Опис
FLAT	Плоске
DOME	Сферичне (випукле донизу)
DISH	Торосферичне (еліптичне)
CONE	Конічне

Верхні (`top_type`):

Код	Опис
FLAT	Плоске
DOME	Сферичне
DISH	Торосферичне

Документ створено автоматично на основі аналізу вихідного коду. Для заповнення скриншотів: відкрийте застосунок, перейдіть на зазначений екран і зробіть скриншот з описаними елементами.