

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики та комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та
робототехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)
Розроблення підсистеми автоматизації для управління якістю продукції на всіх
етапах виробництва
(тема)

Виконав:

здобувачка 2 року навчання,

групи КІТПВМ-24-2

Софія ВІННИЧЕНКО

(власне ім'я, прізвище)

Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка

(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Комп'ютерно-інтегровані технологічні процеси та виробництва

(повна назва освітньої програми)

Керівник доц. Софія ХРУСТАЛЬОВА

(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри _____

(підпис)

2025 р.

Ігор НЕВЛЮДОВ

(власне ім'я, прізвище)

Я, Вінниченко Софія Олександрівна, як здобувачка вищої освіти ХНУРЕ, розумію та підтримую політику закладу з академічної доброчесності. Я не надавала і не одержувала недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовувала штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«02» грудня 2025 р.

 Софія ВІННИЧЕНКО

Харківський національний університет радіоелектроніки

Факультет _____ АКТ _____

Кафедра _____ КІТАР _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології _____

та робототехніка _____

(код і повна назва)

Тип програми _____ освітньо-професійна _____

Освітня програма Комп'ютерно-інтегровані технологічні процеси та виробництва _____

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« _____ » _____

20 ____ р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

здобувачі _____ Вінниченко Софії Олександрівні _____

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення підсистеми автоматизації для управління якістю продукції на всіх етапах виробництва
затверджена наказом університету від 10 листопада 2025 р. № 1029Ст
2. Термін подання здобувачкою роботи до екзаменаційної комісії 22 грудня 2025 р.
3. Вихідні дані до роботи Форма діалогу: вебдодаток. Перелік програмних засобів, що використовуються: IntelliJ IDEA, Java, Spring Boot, Thymeleaf, MySQL, Bootstrap, Python, PyCharm. Технічне обладнання: комп'ютер або ноутбук з веббраузером Google Chrome.
4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ; 4.2 Огляд і аналіз існуючих методів, засобів, та систем автоматизації; 4.3 Розроблення структурної схеми та алгоритму роботи підсистеми; 4.4 Проектування бази даних підсистеми; 4.5 Реалізація підсистеми автоматизації у вигляді програмного засобу; 4.6 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Демонстраційний матеріал, представлений у форматі презентації PowerPoint (*.pptx) – 15 сторінок формату А4

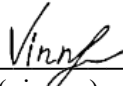
6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Огляд і аналіз існуючих методів та систем	05.09 – 15.09.2025	Виконано
2	Розроблення структурної схеми та алгоритму	15.09 – 20.09.2025	Виконано
3	Проектування бази даних підсистеми	26.09 – 01.10.2025	Виконано
4	Реалізація підсистеми у вигляді програмного засобу	01.10 – 30.11.2025	Виконано
4	Тестування роботи підсистеми автоматизації	30.11 – 02.12.2025	Виконано
5	Оформлення пояснювальної записки	02.12 – 05.12.2025	Виконано
6	Подання роботи на нормоконтроль	05.12 – 14.12.2025	Виконано
7	Подання роботи на перевірку Інтернет-сервісом Strikeplagiarism	14.12 – 15.12.2025	Виконано
8	Подання роботи на рецензування	15.12 – 19.12.2025	Виконано
9	Подання роботи на підпис зав. кафедри	19.12 – 22.12.2025	Виконано
10	Подання роботи до ЕК	22.12.2025	Виконано

Дата видачі завдання 01 вересня 2025 р.

Здобувачка _____  _____ Софія ВІННИЧЕНКО
(підпис)

Керівник роботи _____ доц. Софія ХРУСТАЛЬОВА
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка містить: 91 с., 1 табл., 49 рис., 3 дод., 30 джерел.

СИСТЕМА АВТОМАТИЗАЦІЇ, ЯКІСТЬ, ВИРОБНИЦТВО, ТЕЛЕФОННІ АКУМУЛЯТОРИ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Мета дослідження – підвищення ефективності управління якістю продукції на виробництві телефонних акумуляторів за рахунок розроблення MES-підсистеми.

Об'єкт дослідження – процес управління якістю виготовлення телефонних акумуляторів.

Предмет дослідження – підсистема автоматизації для підвищення якості продукції, реалізована у вигляді програмного засобу.

Методи дослідження – ознайомлення з вимогами підприємства, моделювання виконання процесів за допомогою UML-діаграм, створення реляційної бази даних MySQL, розробка вебдодатку з використанням мов Java, HTML, CSS, JavaScript та додаткових фреймворків.

В роботі досліджено питання автоматизації управління якістю продукції на сучасному виробництві. В результаті розроблено підсистему MES-системи, яку реалізовано у вигляді програмного засобу.

Запропоноване програмне рішення слугує основою для подальшої інтеграції у виробничу інфраструктуру підприємства. Його впровадження сприятиме мінімізації відсотка браку та дозволить підвищити загальну ефективність та конкурентноспроможність виробництва на ринку.

ABSTRACT

The explanatory note contains: 91 pp., 1 tab., 49 figs, 3 appendices, 30 sources.

SYSTEM OF AUTOMATION, QUALITY, MANUFACTURING, TELEPHONE BATTERIES, SOFTWARE.

The goal of research is the development of an MES-subsystem aimed at improving product quality in the manufacturing of telephone batteries.

The object of research is the process of manufacturing telephone batteries.

The subject of research is the automation subsystem for improving product quality.

Research methods – familiarization with the requirements of the enterprise, modeling the execution of processes using UML diagrams, creating a MySQL relational database, developing a web application using Java, HTML, CSS, JavaScript, and additional frameworks.

The certification work investigates the issue of automation of product quality management in modern production. A web application was designed, which is a demonstration model of the MES subsystem.

The proposed software solution serves as the basis for further integration into the production infrastructure of the enterprise. Its implementation will help minimize the percentage of defects and will increase the overall efficiency and competitiveness of production in the market.

ЗМІСТ

Перелік умовних скорочень	9
Вступ.....	10
1 Огляд і аналіз вснуючих методів, засобів та систем автоматизації для управління якістю продукції на всіх етапах виробництва	12
1.1 Опис та аналіз поточного стану проблеми якості продукції, що виготовляється	12
1.2 Аналіз існуючих систем автоматизації управління якістю	14
1.3 Недоліки існуючих систем автоматизації управління якістю	18
2 Розроблення структурної схеми та алгоритму роботи підсистеми автоматизації для управління якістю продукції на всіх етапах виготовлення телефонних акумуляторів	22
2.1 Опис архітектури розроблюваної підсистеми автоматизації для управління якістю виготовлення телефонних акумуляторів	22
2.2 Визначення функціональних та нефункціональних вимог до підсистеми	26
2.3 Розроблення структурної схеми підсистеми автоматизації для управління якістю виготовлення телефонних акумуляторів за допомогою мови UML.....	29
2.3.1 Розроблення діаграми варіантів використання.....	29
2.3.2 Розроблення діаграми класів для підсистеми	34
3 Проектування бази даних підсистеми автоматизації для управління якістю виготовлення телефонних акумуляторів	37
3.1 Обґрунтування вибору способу зберігання даних та архітектури бази даних	37
3.2 Нормалізація таблиць бази даних.....	38
3.3 Опис сутностей та зв'язків між ними.....	42

4 Реалізація підсистеми автоматизації для управління якістю виготовлення телефонних акумуляторів у вигляді програмного засобу	63
4.1 Вибір мов програмування, середовищ розробки та додаткових інструментів	63
4.2 Принцип роботи AI-модуля	66
4.2.1 Інтеграція AI-модуля у основний застосунок	69
4.2.2 Створення датасету у базі даних для навчання моделі	71
4.3 Реалізація імітації надходження параметрів з обладнання.....	72
4.4 Опис реалізації інтерфейсу підсистеми та тестування програмного засобу	74
4.5 Охорона праці	83
Висновки	85
Перелік джерел посилання	87
Додаток А Апробація наукових результатів дослідження.....	92
Додаток Б Лістинг програми.....	102
Додаток В Демонстраційний матеріал.....	115

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

НПАОП – нормативно-правовий акт з охорони праці;
ПЗ – програмне забезпечення;
ПЛК – програмований логічний контролер;
СУБД – система управління базами даних;
ACID – Atomicity, Consistency, Isolation, Durability;
AI – Artificial Intelligence;
ERP – Enterprise Resource Planning;
HTTP – HyperText Transfer Protocol;
JSON – JavaScript Object Notation;
JVM – Java Virtual Machine;
MES – Manufacturing Execution System;
ORM – Object-Relational Mapping;
REST – Representational State Transfer;
RFID – Radio-Frequency Identification;
SCADA – Supervisory Control and Data Acquisition;
SQL – Structured Query Language;
UML – Unified Modeling Language;
URL – Uniform Resource Locator.

ВСТУП

В умовах цифрової трансформації промисловості впровадження MES-підсистем для моніторингу та управління якістю дозволяє не лише підвищити ефективність виробництва, а й забезпечити стабільність технологічних процесів. Такий підхід відповідає цілям сталого розвитку ООН, зокрема Цілі 9 «Індустріалізація, інновації та інфраструктура» та Цілі 12 «Відповідальне споживання і виробництво», адже сприяє впровадженню інноваційних технологій, оптимізації використання ресурсів і зменшенню кількості виробничих відходів. Актуальність обраної теми зумовлена зростаючими вимогами до якості та надійності акумуляторної продукції, що є критично важливою складовою сучасних мобільних технологій.

Мета кваліфікаційної роботи – підвищення ефективності управління якістю продукції на виробництві телефонних акумуляторів за рахунок розроблення MES-підсистеми.

Об'єкт дослідження – процес управління якістю виготовлення телефонних акумуляторів.

Предмет дослідження – підсистема автоматизації для підвищення якості продукції, реалізована у вигляді програмного засобу.

Для реалізації поставленої мети необхідно виконати наступні завдання:

- провести огляд та аналіз існуючих методів, засобів та автоматизованих систем управління якістю продукції;
- розробити структурну схему та алгоритм роботи підсистеми управління якістю продукції;
- спроектувати базу даних підсистеми автоматизації для управління якістю виготовлення телефонних акумуляторів;

– реалізувати підсистему автоматизації для управління якістю виготовлення телефонних акумуляторів у вигляді програмного засобу;

– оформити кваліфікаційну роботу згідно з рекомендаціями [1-6], та вимогами ДСТУ 3008:2015 [7]. За результатами роботи було опубліковано наукову статтю у фаховому журналі, наведено її текст у Додатку А [8].

1 ОГЛЯД І АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ, ЗАСОБІВ ТА СИСТЕМ АВТОМАТИЗАЦІЇ ДЛЯ УПРАВЛІННЯ ЯКІСТЮ ПРОДУКЦІЇ НА ВСІХ ЕТАПАХ ВИРОБНИЦТВА

1.1 Опис та аналіз поточного стану проблеми якості продукції, що виготовляється

Сучасні промислові підприємства функціонують в умовах високої конкуренції, глобалізації ринку та постійного зростання вимог до якості продукції, що випускається. Для виробників високотехнологічних виробів, до яких відносяться телефонні акумулятори, забезпечення якості на всіх етапах життєвого циклу продукції стає ключовим фактором економічної ефективності. Незначні відхилення технологічних параметрів при виготовленні акумуляторів, наприклад, при нанесенні електродної суспензії, сушінні або герметизації, можуть призвести до погіршення характеристик батареї, зниження її ємності та рівня безпеки експлуатації. Тому задача постійного моніторингу та управління якістю в реальному часі є критично важливими для сучасного підприємства [9].

Традиційні підходи до забезпечення якості, які ґрунтуються на вибірковому контролі готової продукції, вже не відповідають вимогам сучасної промисловості. Вони не дозволяють вчасно виявляти причини дефектів та попереджати їх на ранніх стадіях виробничого процесу. У зв'язку з цим підприємства переходять до концепції інтегрованого управління якістю, яка передбачає використання автоматизованих систем, здатних об'єднувати дані з різних рівнів виробництва – від обладнання та датчиків до систем планування ресурсів.

Особливу роль у розв'язанні даних задач відіграє MES – система управління виробничими процесами на рівні цеху. Вона є сполучною ланкою між

корпоративним рівнем (ERP-системи) та рівнем обладнання (SCADA/PLC), забезпечуючи збір, аналіз та візуалізацію даних в реальному часі.

В контексті виготовлення телефонних акумуляторів впровадження MES є особливо актуальним. Даний тип продукції вимагає суворого дотримання технологічних параметрів, незначне відхилення може призвести до зростання внутрішнього опору, зменшення часу стабільної роботи або ризику термічного розгону. Впровадження MES надає можливість забезпечити відстеження якості на кожному етапі виготовлення продукту: від прийому сировини до фінального тестування готових виробів [10].

Сучасні тенденції промисловості направлені на реалізацію концепцій Індустрії 4.0 та цифрового виробництва, які передбачають інтеграцію інформаційних систем, аналітики Big Data та штучного інтелекту. Актуальність впровадження MES в рамках Індустрії 4.0 обумовлена необхідністю оперативного обміну інформацією між всіма рівнями виробничої ієрархії. Система забезпечує наскрізну цифрову інтеграцію, зв'язуючи дані з рівня обладнання з корпоративними системами. Це надає можливість отримати цілісне уявлення про стан виробничого процесу, що є основою для його подальшої оптимізації та управління. Впровадження MES надає підприємству ряд переваг, серед яких можна виділити:

- забезпечення доступу до достовірних та актуальних даних про хід виробництва, завантаження обладнання, виконання технологічних операцій та якість продукції;

- фіксування походження партій сировини, параметрів кожної операції та результатів контрольних процедур, що робить можливим своєчасне виявлення причин невідповідностей;

- оперативне відстеження технологічних порушень та запуск коригуючих дій до того, як проблема вплине на якість кінцевого продукту;

– документування технологічних процесів, що полегшує проходження сертифікацій та відповідає міжнародним стандартам, таким як ISO 9001 [11].

1.2 Аналіз існуючих систем автоматизації управління якістю

Для визначення переваг та недоліків різних підходів, глибшого розуміння ринку та формування власної стратегії розробки підсистеми управління якістю було проаналізовано функціонал MES від провідних виробників.

Компанія SAP – німецька компанія і постачальник програмного забезпечення для управління бізнес-процесами та операційною діяльністю компаній [12]. Особливістю рішень SAP є створення єдиного джерела даних на базі ERP-системи для всіх підрозділів – від виробництва та логістики до продажів. До готової системи є можливість додавати додаткові модулі, такі як: QM (Quality Management), MM (Materials Management), SD (Sales and Distribution), PP (Production Planning) [13]. Модуль QM допомагає оптимізувати використання ресурсів, підвищувати ефективність виробництва та відповідність продукції стандартам. На рисунку 1.1 відображено приклад інтерфейсу модулю для управління якістю, який використовується для введення та керування параметрами для певного матеріалу на конкретному заводі.

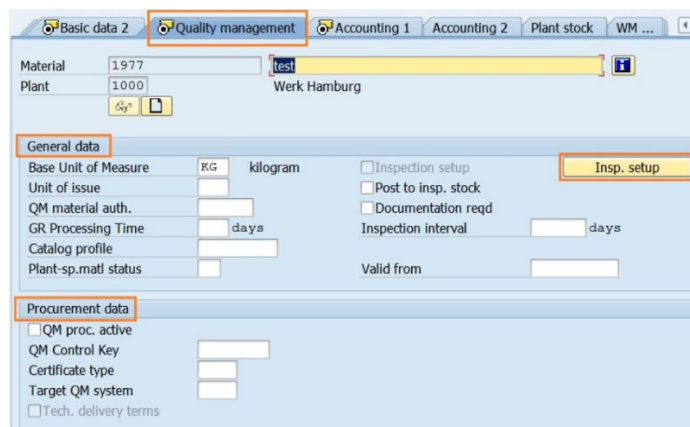


Рисунок 1.1 – Інтерфейс модулю QM [14]

Головним недоліком рішень компанії є те, що інтеграція SAP з великою кількістю корпоративних бізнес-процесів одночасно може призводити до менш оперативного реагування на зміни у виробництві та обмежених можливостей налаштування системи під специфічні потреби окремих цехів у порівнянні зі спеціалізованими MES.

Рішення компанії Siemens SIMATIC IT MES надає ключові функції, що дозволяють контролювати критично важливі виробничі операції в режимі реального часу завдяки наскрізній видимості. Поєднуючи планування ресурсів підприємства (ERP), управління життєвим циклом продукції та інші корпоративні системи, SIMATIC IT надає функціональність для своєчасного відстеження дефектів, управління невідповідностями та контролю виконання робочих операцій [15]. Функціональність програмного забезпечення включає в себе:

- розподіл робочого навантаження між операторами на початку зміни в залежності від їх навичок;
- перевірку виконання етапів процесу виготовлення, керування витратами ресурсів, перевірку валідності деталей та інструментів;
- відстеження дефектів, керування процесами переробки браку;
- генерацію та надання електронних робочих інструкцій, керування пунктами збору даних, створення звітності.

На платформі для огляду програмного забезпечення SoftwareReviews користувачі високо оцінюють SIMATIC IT за легкість інтеграції у систему (74/100), управління документообігом (73/100), цінність для бізнесу (77/100). В той же час інтуїтивність та зручність використання є однією з найслабших сторін (65/100), користувачі наголошують на тому, що працівник повинен мати технічні навички, щоб розпочати роботу з інструментом. Також не високо оцінені функції аналізу даних та формування звітності (64/100), простота налаштування та інтеграції даних

(66/100), широта функціоналу ПЗ (64/100). На рисунку 1.2 наведені всі критерії для оцінювання та результати.

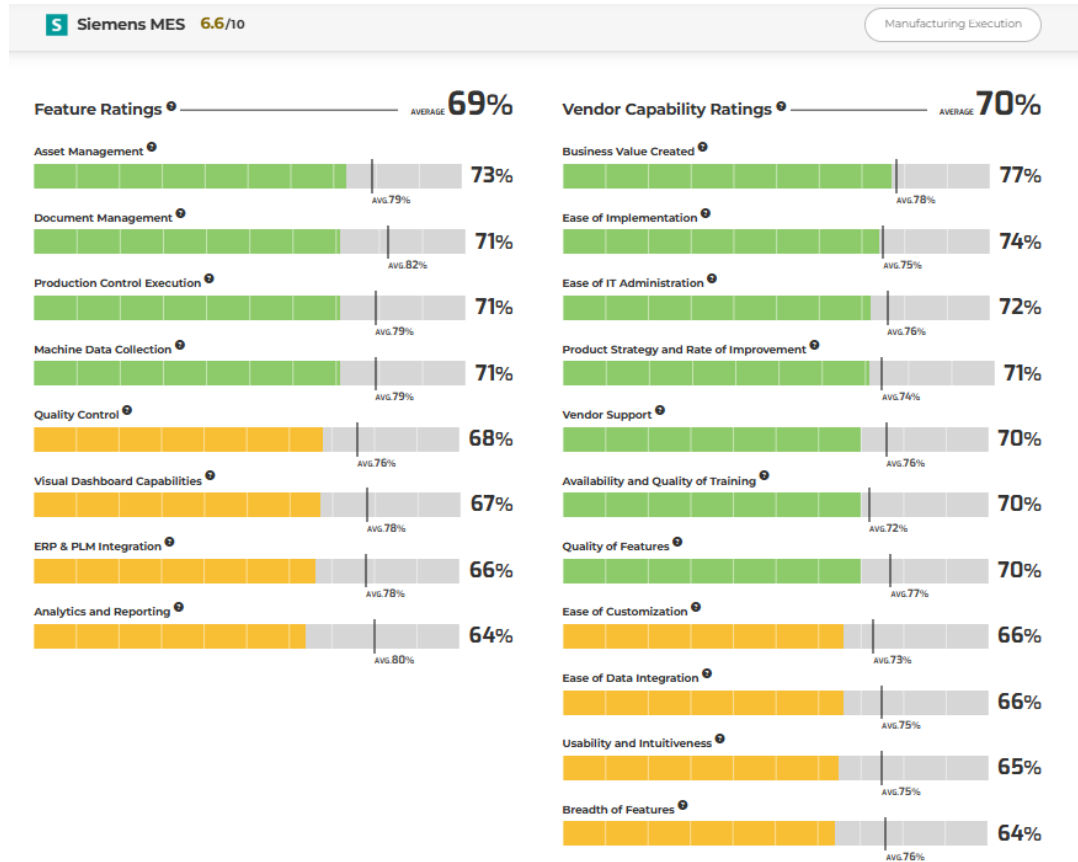


Рисунок 1.2 – Результати оцінювання Siemens SIMATIC IT MES [16]

Rockwell Automation – це американська компанія, що є одним зі світових лідерів у сфері промислової автоматизації та інформаційних технологій [17].

Їх ПЗ Plex QMS є хмарним цифровим рішенням, тому вимоги до виготовлення та технічна документація постійно доступні на усіх ділянках підприємства. Також програма збирає дані у режимі реального часу у процесі виробництва, підключаючись безпосередньо до приладів для усунення ручного введення даних.

На відміну від інших рішень, Plex QMS інтегрує CAPA (Corrective and Preventive Action) з моніторингом постачання та запасів ресурсів, готових партій, дозволяючи відстежувати дефекти до рівня окремої деталі.

Завдяки цифровій інфраструктурі та централізованому управлінню документами система забезпечує аудиторам миттєвий доступ до всіх необхідних показників та перевірок якості [18].

На рисунку 1.3 наведено інтерфейс Plex QMS, що використовується для цифрового керування та проведення аналізу PFMEA (Process Failure Mode and Effects Analysis), об'єднуючи структурний, функціональний та ризиковий аналіз в одному інтегрованому робочому просторі.

The screenshot displays the Plex QMS interface for a PFMEA analysis. At the top, there is a navigation bar with tabs for Planning and Preparation, Structure Analysis, Function Analysis, Failure Analysis, Risk Analysis, Optimization, Delete, Print, Revision History, Foundation Changes (0), Control Plan, Associated Parts, and PFMEA T. Below this is a metadata table with the following data:

Field	Value	Field	Value	Field	Value
PFMEA Number	16	PFMEA ID Number	PFMEA Linkage Trial	Subject	Linkage Trial
PFMEA Status	Approved	Part Number	PFMEA Linkage Trial Part Rev AA	Company Name	Ed Svcs BC 2291 Martensen
Plant Location	Main	PFMEA Start Date	1/30/2023	Process Responsibility	Brian Martensen
Customer Name		PFMEA Revision Date	1/30/2023	Confidentiality Level	
Model Year / Program	2023 / PFMEA Linkage	Cross-functional Team	Cross-Functional Team	Note	
Active	Yes				

Below the metadata table, there are checkboxes for 'Linked' and 'Unlinked' with a link to 'Process standards, controls, and preparation'. The main data table is organized into four columns: Structure Analysis, Function Analysis, Failure Analysis, and Risk Analysis. The first row of data is as follows:

Foundation PFMEA Link	Process Item	Process Step	Process Work Element	Function of the Process Item	Function of the Process Step	Function of the Process Work Element	Your Plant Failure Effect	Ship To Plant Failure Effect	End User Failure Effect	S	Failure Mode	Failure Cause	Current Prevention Controls	O	Current Detection Controls	D	Special Characteristics	PFMEA AP	RPN	Is Num
Receiving	Receiving	10. Receiving	Man. Verify Correct Material Grade	Your Plant: Receive material and confirm material grade Ship to Plant: Confirm correct material grade End User: Confirm correct material grade	Verify material grade Material Grade	Review material cert and tag to verify correct grade 409 SS	Wrong material run in press	No discernible effect	Washer lifespan is compromised	7	Wrong material shipped	Material mislabeled	Color code of material from mill by grade	2	Microwave Plasma Atomic Emission Spectroscopy of each coil	2		L	28	

Рисунок 1.3 – Інтерфейс програмного забезпечення Plex QMS [19]

Одним з головних недоліків даного рішення є залежність від наявності на виробництві сучасного обладнання з можливістю інтеграції з хмарними системами.

1.3 Недоліки існуючих систем автоматизації управління якістю

Попри значні функціональні можливості, аналіз існуючих рішень виявляє ряд ключових обмежень, які заважають повністю реалізувати потенціал систем у сфері управління якістю. Виходячи з цього, було виділено наступні недоліки:

- проблема універсальності. Системи, що намагаються охопити усі бізнес-процеси на підприємстві часто є громіздкими та негнучкими. Вони не встигають оперативно реагувати на динамічні зміни безпосередньо на виробничій лінії та погано адаптуються під потреби окремих цехів, поступаючись спеціалізованим рішенням;

- проблема складності використання. Через багатоетапність виробничих процесів та впровадження великої кількості функціоналу інтерфейс створених систем може бути неінтуїтивним та складним для розуміння працівниками. Це ускладнює впровадження та налаштування, а також вимагає часу та додаткових грошових ресурсів компанії на навчання персоналу;

- проблема залежності від сучасної інфраструктури. Особливо це стосується хмарних рішень. Їхня ефективність напряду залежить від наявності на виробництві сучасного обладнання, здатного підключатися до мережі та передавати дані у реальному часі. Це створює технологічний бар'єр для підприємств зокрема через високу вартість пристроїв;

- проблема реактивного підходу до контролю якості. Всі описані вище MES значною мірою зосереджені на реактивному управлінні. Вони можуть зупинити виготовлення продукту або попередити про відхилення коли дефект вже виявлено, але менш ефективні у його запобіганні на ранніх стадіях. Звітність найчастіше формується після закінчення зміни, що обмежує можливість вчасно вжити коригувальних заходів.

Подолання вищеописаних проблем вимагає інтеграції сучасних технологій, що здатні змінити підхід до управління якістю.

Для злагодженої роботи підприємства варто дотримуватися чіткої ієрархії систем управління. Система ERP має відповідати лише за бізнес-планування (замовлення, фінанси, тощо) та передавати їх до MES, яка, в свою чергу, перетворюватиме їх на детальні, покрокові завдання для виробництва. Варто зазначити, що MES повинна бути побудована за модульним принципом, тому підприємство має право обирати лише ті функції, які необхідні йому на конкретній виробничій лінії.

З метою уникнення створення складних для взаємодії інтерфейсів варто впроваджувати їх сегрегацію за ролями. Диспетчер, проєктувальник та лаборант повинні мати доступ до різних сторінок, відповідно до їх компетенцій. Також замість відображення даних у вигляді таблиць або багатосторінкових меню варто використовувати інтерактивні панелі з кольоровим відображенням різних станів та функцій. З метою зниження кількості помилок система повинна забороняти доступ до подальших кроків, поки коректно не виконано попередній.

Інтернет Речей (IoT) є однією з ключових технологій для створення інтелектуальних систем управління якістю. IoT-датчики, які встановлюються на обладнанні, зчитують безперервний потік даних про температуру, тиск, товщину, тощо. Виходячи з цього, система не має потреби підключатися безпосередньо до машини, оскільки датчики своєчасно передають всі параметри. Це вирішує проблему фрагментованості інформації та її витоку, адже пристрої фіксують та передають дані у систему в режимі реального часу, створюючи «цифрового двійника» виробничого процесу. Це надає програмі можливість приймати рішення на основі щойно отриманих даних, що мінімізує ризик втрати важливої інформації.

Зібрані за допомогою IoT масиви даних можуть бути застосовані для аналітичних інструментів вищого рівня, зокрема алгоритмів машинного навчання (ML) та штучного інтелекту (AI). Алгоритми AI/ML здатні аналізувати великі обсяги інформації, зібрані з виробничого процесу, для прогнозування можливих дефектів до їх виникнення. Також системи комп'ютерного зору на основі AI можуть проводити інспекцію продукції на предмет дефектів з високою точністю, що усуває такі проблеми як людський фактор або суб'єктивність. Навчена система може не лише виявляти неточності, а й відбракувати дефектну продукцію, запобігаючи її просуванню по виробничій лінії. AI/ML можуть аналізувати дані з датчиків для прогнозування збоїв обладнання. Виявлення потенційних несправностей у роботі машин до їх настання дозволяє запланувати технічне обслуговування заздалегідь, мінімізуючи можливість простоїв пристрою у майбутньому.

Ефективне управління якістю є неможливим без застосування науково обґрунтованих методів контролю та аналізу. Нині сформовано цілий комплекс інструментів, об'єднаних єдиною метою – забезпечити стабільність процесів та підвищити рівень якості продукції. Найбільш відомими та використовуваними є статистичні, аналітичні та графічні методи, серед яких:

- контрольні карти Шухарта. Суть методу полягає у побудові графіків, які відображають зміну ключових параметрів процесу в часі з нанесенням верхньої та нижньої контрольної межі. Це дозволяє відстежувати стабільність процесу та відрізнити випадкові коливання від систематичних відхилень;

- діаграма Ішікави. Призначена для аналізу причинно-наслідкових зв'язків, які призводять до виникнення дефекту. Вона візуалізує вплив різних факторів – таких як людина, машина, матеріали, метод, середовище, вимірювання – на якість кінцевого продукту;

- метод «5 Чому». Направлений на пошук першопричини проблеми. Шляхом послідовного ставлення запитання «чому?» можна виявити глобальні причини

невідповідностей. У виробничих умовах часто використовується разом з діаграмою Ішікави або в рамках процедур коригуючих дій;

– статистичне керування процесами (SPS). Являє собою комплекс інструментів, включаючи контрольні карти, аналіз варіацій і оцінку можливостей процесів. Метою є підтримка процесів в стані статистичного контролю та мінімізації розкиду параметрів.

Підсистема управління якістю має реалізовувати принципи та підходи, які лежать в основі класичних принципів та підходів, але на новому, інтелектуальному рівні. Її архітектура має забезпечувати перехід від традиційного контролю до предиктивного керування якістю на базі даних та алгоритмів штучного інтелекту.

Використання моделі AI, навченої на виробничих даних, відповідає концепції SPS та карт Шухарта. Модель виступає цифровим аналогом контрольної карти: вона аналізує діапазони технологічних параметрів, виявляє потенційно нестабільні поєднання та формує попередження про ризик браку до запуску виготовлення. Функція аналізу причин відхилень, заснована на накопичених даних про виробничий процес, схожа на підходи діаграми Ішікави та методу «5 Чому». На основі історії партій та зафіксованих відхилень підсистема може виявляти залежності між параметрами та якістю продукції, допомагаючи визначати фактори, які найбільш впливають на появу дефектів. Це створює основу для інтелектуального аналізу причинно-наслідкових зв'язків без участі працівників.

Також кожна нова партія даних використовується не лише для аналізу, а для перенавчання моделі, що забезпечує поступове підвищення точності прогнозів та адаптацію до змін робочих умов виробництва.

У даному розділі проаналізовано існуючі методи, засоби та системи автоматизації управління якістю продукції, а також визначено їхні основні переваги й недоліки. На основі проведеного аналізу сформовано підхід до розроблення власної підсистеми, спрямований на усунення виявлених обмежень.

2 РОЗРОБЛЕННЯ СТРУКТУРНОЇ СХЕМИ ТА АЛГОРИТМУ РОБОТИ ПІДСИСТЕМИ АВТОМАТИЗАЦІЇ ДЛЯ УПРАВЛІННЯ ЯКІСТЮ ПРОДУКЦІЇ НА ВСІХ ЕТАПАХ ВИГОТОВЛЕННЯ ТЕЛЕФОННИХ АКУМУЛЯТОРІВ

Головною метою кваліфікаційної роботи є розробка підсистеми MES з вбудованою моделлю штучного інтелекту для керування якістю продукції на всіх етапах виробництва телефонних акумуляторів. Для створення та успішної інтеграції підсистеми у виробництво необхідно проаналізувати потреби підприємства, визначити функціональні та не функціональні вимоги, розробити архітектуру програмного забезпечення, створити програмний макет для демонстрації замовнику, налаштувати підсистему на підприємстві з урахуванням специфіки виробничих процесів, провести інструктаж з використання ПЗ для працівників, забезпечити підтримку, моніторинг та оновлення компонентів підсистеми після успішної інтеграції.

2.1 Опис архітектури розроблюваної підсистеми автоматизації для управління якістю виготовлення телефонних акумуляторів

Підсистема має інтерфейс для одинадцяти користувачів: лаборанта, який перевіряє наявність потрібних характеристик інгредієнтів у підсистемі та створює технічні карти на виробництво партій, диспетчера станції зважування, який запускає процес зважування та відслідковує правильність вхідних даних, диспетчера станції замішування, який запускає процес замішування і, після його завершення, ініціює відправку даних на станцію нанесення пасти. Диспетчери станцій нанесення пасти, сушіння, каландрування, нарізання, вкладання, заливки та

запаювання відслідковують правильність вхідних параметрів від датчиків та ініціюють подачу рапорту на ремонт у разі несправностей. Рапорт отримує бригадир ремонтної бригади, який відправляє працівників на перевірку машини. Після отримання партії акумуляторів з попередніх етапів диспетчер станції тестування запускає перевірки на цикл заряду/розряду, перегрів та коротке замикання по черзі для кожного акумулятора партії, спостерігаючи за показниками. У разі виявлення відхилення батарея відбраковується.

Для проєктування функціоналу підсистеми було виділено наступні процеси виробництва:

- прийом сировини – процес включає в себе отримання виробництвом сировини та занесення її характеристик (кількість, дані про постачальника, тощо) у базу даних;

- зважування інгредієнтів – процес включає у себе формування необхідної кількості інгредієнтів для виготовлення конкретної партії акумуляторів;

- замішування суспензії – включає у себе замішування активних речовин (наприклад, літій-кобальтового оксиду) з провідниками та зв'язуючими компонентами;

- замішування електроліту – включає в себе замішування солей літію з органічними розчинниками;

- нанесення пасти на фольгу – включає в себе нанесення приготованої пасти тонким шаром на алюмінієву (для катода) та мідну (для анода) фольги;

- сушіння електродів – включає в себе проходження рулонів фольги через печі з метою видалення залишків розчинника;

- каландрування електродів – включає в себе процес ущільнення електродів за допомогою сталевих валів для покращення провідності та забезпечення рівномірності його структури;

– нарізання електродів – на даному етапі електроди нарізаються на рівні частини та підготовлюються до збирання;

– вкладання елементів до корпусу – на даному етапі формується електродний блок, у якому катод та анод розділені пористим сепаратором. Далі він вкладається до полімерного пакету або металічного корпусу, до його струмовиводів приварюються контактні пластини;

– заливання електроліту – на даному етапі у герметичну оболонку під вакуумом вводиться електроліт. Дана операція повинна проводитися в умовах мінімальної вологості. Після успішного виконання процесу пакет запаюється;

– проведення тестів. На даному етапі акумулятори проходять три етапи тестування. Спочатку акумулятор проходить декілька циклів розрядження та зарядження, під час яких перевіряється напруга, ємність та внутрішній опір. Наступними є стрес-тести на перегрів та коротке замикання.

Для підвищення якості шляхом введення в підприємство підсистеми MES було виділено показники для оцінки якості телефонних акумуляторів, за якими покращується ефективність виробництва:

– рівень дефектності (DR). Показник, який відображає рівень дефектної продукції у порівнянні з загальною кількістю. Розраховується за формулою:

$$DR = \frac{a}{b} \cdot 100\%, \quad (2.1)$$

де a – кількість дефектних одиниць;

b – загальна кількість виготовлених виробів.

Показник розраховується до та після впровадження системи, потім дані порівнюються;

– частина дефектів на мільйон операцій (DPMO). DPMO оцінює кількість дефектів у порівнянні з можливостями їх виникнення. За його допомогою можна

виявити області з найвищими показниками дефектів для вживання коригувальних заходів. Формула:

$$DR = \frac{a}{b \times c} \cdot 1000000, \quad (2.2)$$

де a – кількість дефектних одиниць;

b – загальна кількість виготовлених виробів;

c – кількість потенційних можливостей дефектів на одиницю [20].

MES автоматично збирає та зберігає дані про параметри процесу, що робить розрахунок точнішим;

– індекс можливостей процесу (C_p). Даний індекс показує наскільки виробничий процес здатний виробляти продукцію в межах допусків (технічних вимог). Розраховується за формулою:

$$C_p = \frac{USL - LSL}{6 \times \sigma}, \quad (2.3)$$

де USL – верхня межа специфікації;

LSL – нижня межа специфікації;

σ – стандартне відхилення (від зразка) [21].

Автоматизований збір даних, який забезпечує MES, дозволяє зменшити стандартне відхилення, що веде до збільшення C_p та покращення якості.

Розроблена підсистема реалізує наступний функціонал:

- можливість перегляду характеристик сировини в системі;
- можливість заповнення користувачем полів для створення технічної карти та відправки її на перевірку модулю AI;
- можливість отримання результату перевірки від модулю та відображення його на інтерфейсі користувача;

- можливість збереження валідної технічної карти до бази даних;
- можливість відображення на інтерфейсі диспетчерів технічних карт для тих партій, які пройшли попередній етап виготовлення;
- можливість керування диспетчерами віддаленим запуском машин на станції;
- можливість прийому параметрів виготовлення та порівняння їх з референтними у технічній карті;
- можливість створення рапорту на ремонт у разі виявлення дефекту та відправки бригадиру ремонтної бригади;
- можливість генерації кінцевого звіту по кожній партії з деталями про технічні характеристики та кількість браку.

2.2 Визначення функціональних та нефункціональних вимог до підсистеми

Для створення стабільної та надійної підсистеми для управління якістю виготовлення телефонних акумуляторів варто окреслити основні вимоги, відповідно до яких вона буде розроблятися.

Підсистема повинна (функціональні вимоги):

- містити у базі даних інформацію про характеристики сировини. Редагувати дані має право лише лаборант;
- містити у базі даних інформацію про референтні значення маси інгредієнтів для суспензії та електроліту. Редагувати дані має право лише лаборант;
- фіксувати час початку та час закінчення поточного процесу з точністю до секунди;
- вміти порівнювати отримані значення з референтними;

- при порушенні норм виконання поточного процесу повідомляти користувача спливаючим вікном з текстовим повідомленням. Має перший пріоритет;

- створювати рапорт на ремонт, який містить причину створення задачі, процес, модель машини, діапазон прийнятних значень, отримане значення, можливу причину несправності;

- повідомляти користувача про готовність до наступного процесу коли всі попередні етапи успішно завершені і всі дані внесені;

- запускати потрібну виробничу станцію при натисканні користувачем відповідної кнопки;

- приймати інформацію про завершення поточного процесу від програмованого логічного контролера та повідомляти користувача про завершення процесу спливаючим вікном;

- фіксувати та зберігати параметри процесу та відхилення (вихід за допустимий діапазон) при виконанні процесів;

- приймати дані про поточний процес від програмованого логічного контролера з частотою оновлення не менше 1 секунди;

- програмно генерувати дані, що імітують цифровий вихід вагів із точністю 0,001 г;

- генерувати кінцеві звіти для кожної партії починаючи від характеристик сировини і закінчуючи результатами тестування готових акумуляторів. Звіти у форматі PDF;

- приймати і демонструвати повідомлення від AI-моделі. Має другий пріоритет.

Підсистема має (нефункціональні вимоги):

- надавати обмежений доступ для різних ролей;

- архітектура повинна дозволяти додавання нової виробничої лінії без значних змін;

- мати зрозумілий інтерфейс, щоб новий користувач міг виконати базові операції після 15-хвилинного навчання;

- повинна мати 11 рівнів доступу (лаборант, диспетчер станції зважування, диспетчер станції замішування, диспетчер станції нанесення пасти, диспетчер станції сушіння, диспетчер станції каландрування, диспетчер станції нарізання, диспетчер станції вкладання, диспетчер станції заливки електроліту, диспетчер станції тестування, бригадир ремонтної бригади);

- інтегрована з AI-моделлю для прогнозування можливих дефектів.

Для побудови правильної архітектури AI-моделі до неї також було сформовано два види вимог.

Модель AI повинна (функціональні вимоги):

- аналізувати інформацію про наявні історичні дані про виготовлення акумуляторів та робити висновок про правильність створення технічної карти;

- мати можливість перенавчання раз на 5 днів з метою обробки нових даних;

- надсилати рекомендації в систему;

Модель ШІ має (не функціональні вимоги):

- мати модульну структуру для оновлювання окремих компонентів без необхідності повного переналаштування всієї системи;

- мати механізм безперервного навчання та перенавчання на нових даних без затримки роботи системи;

- візуалізувати результати аналізу у форму текстового повідомлення до 250 знаків.

2.3 Розроблення структурної схеми підсистеми автоматизації для управління якістю виготовлення телефонних акумуляторів за допомогою мови UML

UML – це уніфікована мова моделювання, призначена для візуального представлення, проектування та документування програмних систем. Головною метою UML є створення наочних моделей складних систем, які описують структуру, поведінку, взаємодію компонентів між собою та життєвий цикл об'єктів. Вона не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація [22].

UML включає декілька типів діаграм, зокрема:

- діаграми класів. Відображають класи системи, їх поля, методи та зв'язки. Використовується для опису внутрішньої структури модулів;
 - діаграми компонентів. Демонструють, з яких програмних блоків складається система (модулі, сервіси, підсистеми) та як вони взаємодіють між собою;
 - діаграми варіантів використання. Демонструють, як користувачі (актори) взаємодіють з системою. Найчастіше використовується для аналізу вимог;
 - діаграми послідовностей. Описують порядок взаємодії об'єктів у часі.
- Для проектування підсистеми MES буде задіяно діаграми варіантів використання та класів.

2.3.1 Розроблення діаграми варіантів використання

Даний вид діаграми призначений для опису функціональних вимог до системи з точки зору користувачів. Вона демонструє, хто взаємодіє з системою і які дії він може виконувати. Основні елементи: актор (користувач або зовнішня система, яка взаємодіє з ПЗ), варіант використання (дія, яку виконує актор у

системі), зв'язки (лінії, які показують, як актори задіяні у конкретних сценаріях). На рисунку 2.1 наведено діаграму варіантів використання для підсистеми MES.

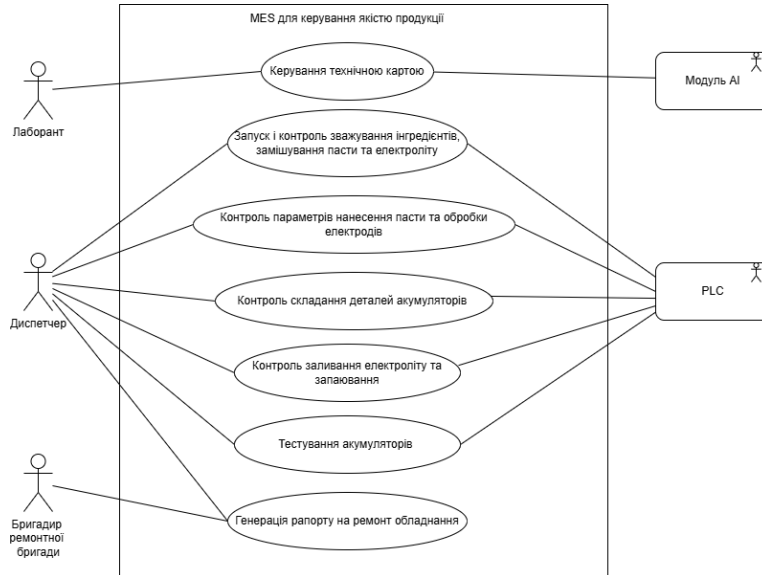


Рисунок 2.1 – Розроблена діаграма варіантів використання

При створенні діаграми було виділено наступних акторів:

- лаборант: фахівець, який відповідає за створення технічних карт та перегляд характеристик інгредієнтів;
- диспетчер: фахівець, який відповідає за моніторинг параметрів процесу на його станції;
- бригадир ремонтної бригади: відповідає за отримання рапорту на ремонт та відправку працівників на необхідну виробничу ланку;
- ПЛК налаштований на взаємодію з датчиками, зчитувачами, вагами, машинами для тестування та рентгенівською установкою і передачу параметрів до системи;
- модуль AI: модель штучного інтелекту, навчена на історичних даних про виготовлення акумуляторів для перевірки валідності технічної карти.

Також було виділено наступні прецеденти:

- керування технічною картою: прецедент дозволяє лаборанту створювати технічні карти та направляти їх моделі AI на перевірку;

- запуск і контроль зважування інгредієнтів, замішування пасти та електроліту: прецедент дозволяє диспетчерам станцій запустити процеси та бачити у своєму інтерфейсі дані з виробничого процесу, передані від ПЛК;

- контроль параметрів нанесення пасти та обробки електродів: дозволяє диспетчеру відслідковувати вихідні параметри на станціях з нанесення пасти, сушіння, каландрування, нарізання електродів;

- контроль складання деталей акумуляторів: дозволяє диспетчеру отримати дані про точність вкладання від ПЛК, який під'єднано до рентгенівської установки;

- контроль заливання електроліту та запаювання: дозволяє диспетчеру отримати параметри про заливку електроліту, і, після перевірки їх валідності, запустити процес остаточного запаювання;

- тестування акумуляторів: дозволяє диспетчеру запустити три тести по черзі отримуючи параметри акумуляторів для кожного.

- створення рапорту на ремонт обладнання: прецедент дозволяє генерувати рапорт на ремонт у разі потреби і надавати його на розгляд диспетчеру. Після погодження відправляти на інтерфейс бригадира.

З огляду на діаграму варіантів використання було побудовано успішний сценарій з розширеннями.

Сценарій «Керування якістю на виробництві».

Назва: обробка партії акумуляторів.

Актор: лаборант, диспетчер, бригадир ремонтної бригади.

Успішний сценарій:

- а) лаборант перевіряє наявність потрібних характеристик сировини у підсистемі;

б) лаборант заповнює технічну карту на виготовлення акумуляторів або обирає існуючу для даної партії. Відправляє на перевірку валідності внесених даних підсистемою. Перевірка проходить і карта зберігається підсистемою;

в) система відображає карту на інтерфейсі диспетчера;

г) диспетчер на станції зважування переглядає технічну карту та запускає процес зважування. Підсистема передає запит на запуск на програмований логічний контролер (ПЛК). Підсистема фіксує виміряне на вагах значення та порівнює з референтним значенням. Після зважування всіх інгредієнтів підсистема отримує повідомлення від ПЛК і повідомляє диспетчера. Диспетчер ініціює відправку інгредієнтів на станцію замісу;

д) диспетчер станції замісу переглядає карту і погоджує початок замісу пасти та електроліту міксерами. Час початку заноситься у підсистему, потім і температура при замішуванні. Після завершення замісу ПЛК передає інформацію і час завершення до системи. Підсистема фіксує час і перевіряє, чи достатній проміжок часу, відведений на заміс. Диспетчер ініціює відправку суспензії на станцію дозування;

е) система отримує технічну карту на партію та ідентифікатор акумулятора з ПЛК, який приймає дані з RFID-зчитувача на фользі [23]. ПЛК отримує товщину нанесеної речовини та передає в систему. Підсистема порівнює значення з референтним і приймає рішення про норму. Оброблені колектори проходять на наступний етап;

є) підсистема отримує технічну карту на партію та ідентифікатор акумулятора з ПЛК який приймає дані з RFID-зчитувача на фользі. Фіксується час початку. Після просушування зчитувач зчитує мітку на кінці фольги акумулятора і передає час закінчення сушіння до ПЛК, який надає його системі. Також ПЛК передає інформацію про температурний режим печі. Підсистема порівнює значення

з референтними і приймає рішення про норму. Оброблені електроди проходять на пресування;

ж) підсистема отримує технічну карту на партію та ідентифікатор акумулятора з ПЛК, який приймає дані з RFID-зчитувача на фользі. ПЛК передає інформацію про ступінь стискання до підсистеми. Підсистема порівнює значення з референтним і приймає рішення про норму. Оброблені електроди проходять на станцію нарізання;

з) підсистема отримує технічну карту на партію та ідентифікатор акумулятора з ПЛК який приймає дані з RFID-зчитувача на корпусі акумулятора. ПЛК передає інформацію про точність нарізання у підсистему. Підсистема порівнює значення з референтним і приймає рішення про норму. Оброблені електроди складаються у лоток і проходять на станцію вкладання;

и) підсистема отримує ідентифікатор акумулятора з ПЛК який приймає дані з RFID-зчитувача на корпусі акумулятора. Рентгенівські установки фіксують зображення кожного акумулятору у процесі вкладання та передають точність вкладання у підсистему, вона перевіряє їх відповідність референтному значенню. Оброблені акумулятори складаються у лоток і проходять на станцію запаювання та заливки;

і) підсистема отримує технічну карту на партію та ідентифікатор акумулятора з ПЛК який приймає дані з RFID-зчитувача на корпусі акумулятора. Після запаювання перевіряється герметичність акумулятора, через ПЛК дані передаються у систему. До та після заливки перевіряється вага кожного акумулятора, підсистема перевіряє їх відповідність референтним значенням. Далі отвір запаюється. Оброблені акумулятори складаються у лоток і проходять на станцію тестування;

ї) підсистема отримує технічну карту на партію та ідентифікатор акумулятора з ПЛК, який приймає дані з RFID-зчитувача на корпусі акумулятора. Спочатку кожен акумулятор проходить кілька циклів заряду і розряду. Протягом цього

процесу відстежуються такі параметри, як напруга, ємність та внутрішній опір (дані від ПЛК надсилаються у підсистему та порівнюються з референтними). Далі проходять тестування на перегрів (відстежуються температура корпусу, напруга елементів акумулятора, внутрішній опір) та коротке замикання (відстежуються напруга та струм, температура корпусу, ємність, внутрішній опір). Оброблені акумулятори складаються у лоток і проходять на маркування та пакування.

Розширення до пункту б. Перевірка модулем AI:

а) підсистема отримує текстове повідомлення про можливість браку від модуля;

б) підсистема передає повідомлення на інтерфейс лаборанта і пропонує змінити введені параметри;

в) лаборант змінює дані відповідно до порад та зберігає карту.

Розширення до пунктів г-і. Виміряне значення більше за референтне:

а) підсистема фіксує його як відхилення;

б) виводить повідомлення про проблему, генерує рапорт на ремонт для ремонтної бригади і запитує про правильність його. Після відправляє бригаді.

2.3.2 Розроблення діаграми класів для підсистеми

Даний вид діаграми описує внутрішню архітектуру програмної підсистеми, демонструючи, з яких класів вона складається, які між ними зв'язки та які атрибути і методи реалізовані у кожному класі.

Існують такі типи зв'язків:

– асоціація (означає, що один клас знає про існування іншого та може з ним взаємодіяти. Позначається як суцільна лінія між класами);

– агрегація (означає, що ціле має частини, але вони можуть існувати роздільно один від одного. Позначається суцільною лінією з порожнім ромбом поряд з цілим);

- композиція (частини не можуть існувати без цілого. Позначається суцільною лінією з чорним ромбом біля цілого);
- наслідування (відображає ієрархію класів, коли один клас наслідує властивості та методи іншого. Позначається як стрілка з порожнім трикутником, направлена від підкласу до суперкласу);
- залежність (коли один клас тимчасово використовує інший, але не має постійного зв'язку. Позначається як пунктирна стрілка від класу-користувача до іншого);
- реалізація (коли клас реалізує інтерфейс. Позначається як пунктирна лінія зі стрілкою від класу до інтерфейсу).

Діаграма класів для підсистеми MES наведена на рисунку 2.2.

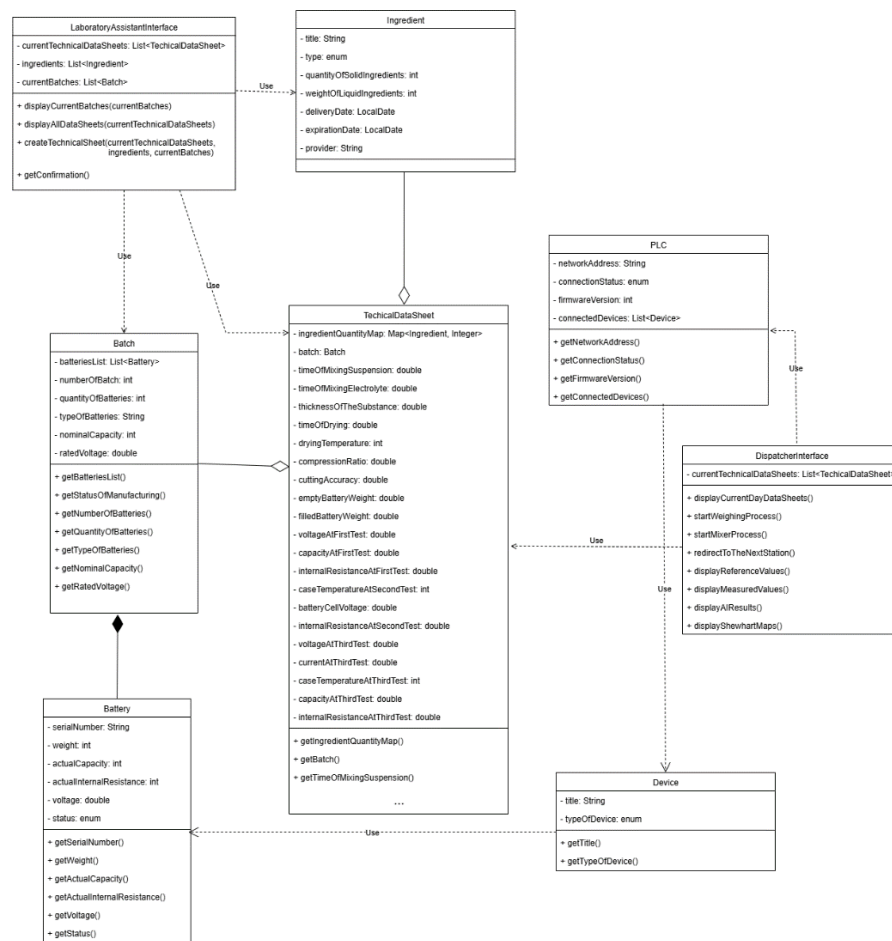


Рисунок 2.2 – Розроблена діаграма класів

У даному розділі сформовано сукупність функціональних і нефункціональних вимог до підсистеми автоматизації управління якістю продукції на всіх етапах виготовлення телефонних акумуляторів. Розроблені структурні схеми мовою UML відображають логіку роботи підсистеми, взаємодію її компонентів та слугують основою для подальшої реалізації й проєктування програмних і інформаційних складових.

3 ПРОЄКТУВАННЯ БАЗИ ДАНИХ ПІДСИСТЕМИ АВТОМАТИЗАЦІЇ ДЛЯ УПРАВЛІННЯ ЯКІСТЮ ВИГОТОВЛЕННЯ ТЕЛЕФОННИХ АКУМУЛЯТОРІВ

3.1 Обґрунтування вибору способу зберігання даних та архітектури бази даних

Проєктування підсистеми автоматизації управління якістю в рамках MES-системи потребує створення надійного сховища даних. Специфіка предметної області передбачає роботу з великими об'ємами структурованої інформації, що включає дані технічних карт, параметри виробничих партій, результати перевірок, тощо. Для реалізації даних задач було обрано реляційну модель даних, оскільки вона забезпечує цілісність інформації та надає можливість будувати складні логічні зв'язки між сутностями виробничого процесу (наприклад, зв'язок між конкретним дефектом акумулятора, партією та технічною картою), що є критичною вимогою для забезпечення простежуваності.

В якості СУБД для реалізації проєкту була обрана MySQL. Даний вибір обумовлено сукупністю технічних та економічних факторів, які відповідають вимогам до сучасних промислових інформаційних систем. MySQL є однією з найбільш поширених реляційних СУБД у світі, що володіє високою продуктивністю при обробці транзакційних запитів, необхідною для реєстрації даних з виробничих ліній в режимі реального часу. Вона підтримує принципи ACID, гарантуючи безпечне зберігання критично важливих даних навіть при апаратних збоях або мережевих помилках. Важливим аргументом на користь використання MySQL є наявність розвинених механізмів індексації, підтримки зовнішніх ключів та збережених процедур, що дозволяє перенести частину логіки

по забезпеченню валідності даних на рівень бази даних, розвантажуючи при цьому обчислювальні потужності сервера додатків.

3.2 Нормалізація таблиць бази даних

Ключовим етапом проєктування бази даних є процес нормалізації. Дана процедура являє собою формальний метод декомпозиції відношень (таблиць) з ціллю мінімізації інформаційної перенасиченості та виключення потенційних аномалій при модифікації даних.

Теоретичним фундаментом нормалізації слугує концепція нормальних форм (НФ). Кожна нормальна форма являє собою визначений набір вимог, яким повинна відповідати структура таблиці. Ці форми мають наступну ієрархію: умовою приведення таблиці до визначеної НФ є виконання всіх вимог попередніх рівнів. На практиці для реляційних баз даних, орієнтованих на транзакційну обробку, достатнім та необхідним є приведення схеми до третьої нормальної форми (3НФ) [24]. Даний рівень забезпечує оптимальний баланс між складністю структури даних та швидкістю виконання запитів.

Розглянемо процес приведення до третьої нормальної форми на прикладі групи сутностей, які відповідають за реєстрацію результатів зважування компонентів для виготовлення акумуляторів. Даний фрагмент предметної області включає в себе інформацію про виробничу партію (batch), акт зважування (weighing_data) та деталізацію використаних інгредієнтів (weighing_item).

На початковому етапі аналізу дані про зважування можна було подати у вигляді універсальної таблиці, де для кожної партії було би перераховано всі використані компоненти. Така структура містила би у собі елементи, що повторюються (наприклад, «Інгредієнт 1», «Кількість 1», «Інгредієнт 2», «Кількість

2»), що порушує вимогу атомарності та унеможливлює гнучке додавання нових компонентів без зміни структури таблиці. Вигляд таблиці наведено на рисунку 3.1.

Таблиця weighing_data						
ID	Batch_ID	Ingredient_1	Quantity_1	Ingredient_2	Quantity_2	...

Рисунок 3.1 – Вигляд таблиці weighing_data на початковому етапі

Для приведення до 1НФ дані були розділені таким чином, щоб кожен запис у таблицю зважування відповідав одній конкретній порції одного інгредієнта. Проте, якщо би в одній таблиці зберігалася інформація і про дату зважування, і про інгредієнт, то це спричинило би дублювання даних про дату для кожного зваженого компонента. Вигляд таблиці після приведення до 1НФ наведено на рисунку 3.2.

Таблиця weighing_data			
ID	Batch_ID	Ingredient_ID	Quantity

Рисунок 3.2 – Вигляд таблиці weighing_data після приведення до 1НФ

Для усунення часткових та транзитивних залежностей структуру було розділено на декілька зв'язаних таблиць:

- таблиця weighing_data – виділена у окрему сутність, пов'язану з batch зв'язком 1:1. Це надає можливість відділити факт проведення технологічної операції зважування від загальних даних партії;

- таблиця weighing_item – зберігає результати вимірювань. Тут було усунено транзитивну залежність: назва інгредієнту та його властивості не зберігаються у даній таблиці. Натомість використовується зовнішній ключ ingredient_id, який

посилається на таблицю інгредієнтів. Сама таблиця `weighing_item` має лише атрибут `real_quantity`, який функціонально залежить від складеного ключа (конкретного процесу зважування та конкретного інгредієнта).

В результаті було отримано наступний вигляд таблиць (рис. 3.3).

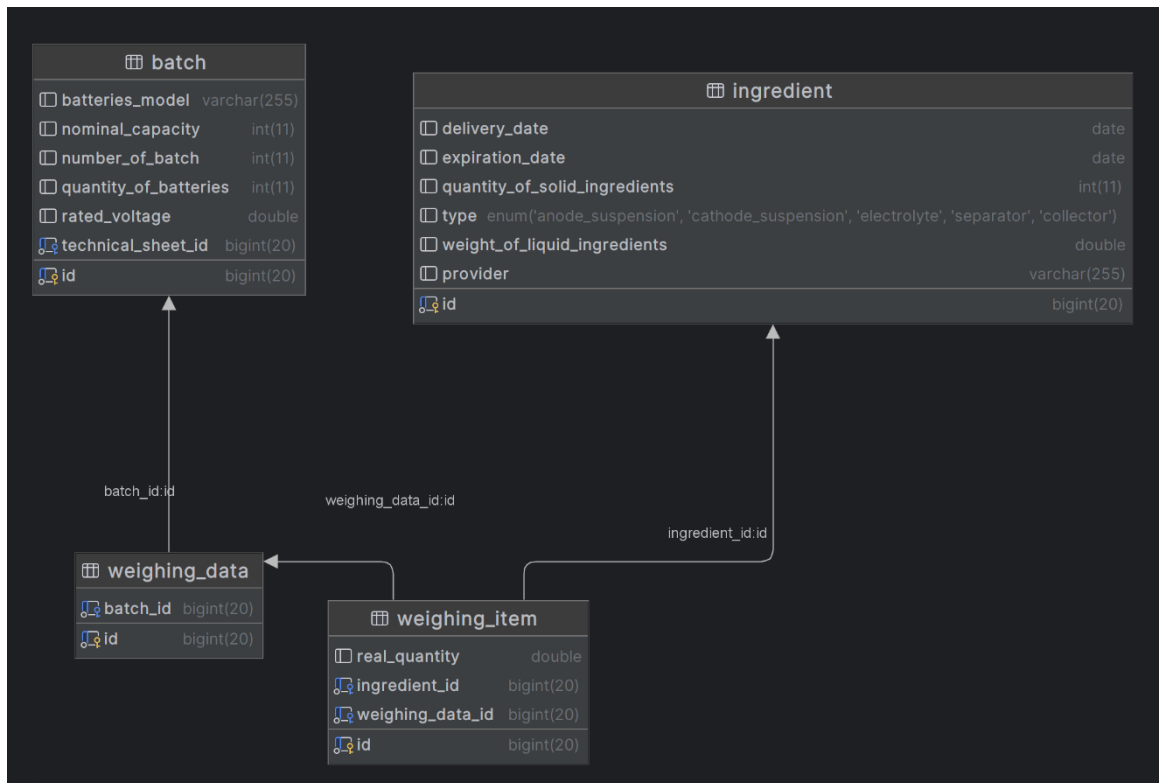


Рисунок 3.3 – Результат приведення до 3НФ

Така структура унеможливорює проблеми при оновленні даних (при зміні назви інгредієнта вона змінюється лише у одній таблиці)) та забезпечує цілісність даних при видаленні партії.

Відповідно до описаних у розділі 2 функціональних вимог та UML-діаграм було створено сутності та проведено зв'язки між ними, аналогічним чином було нормалізовано інші сутності системи. Повна логічна схема бази даних наведена на рисунку 3.4.

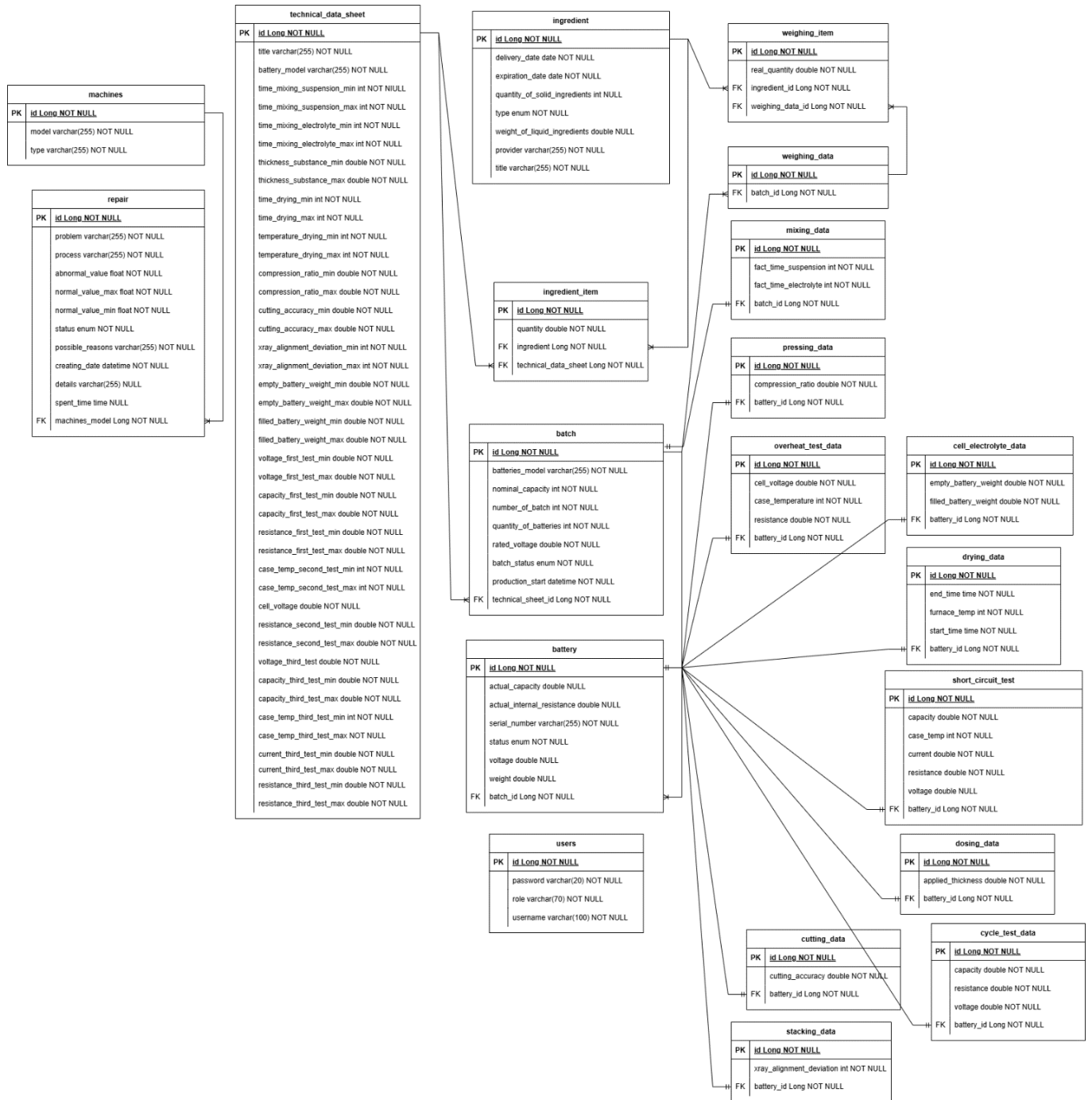


Рисунок 3.4 – Модель «сутність-зв'язок»

3.3 Опис сутностей та зв'язків між ними

Короткий опис зв'язків з їх типами наведено у таблиці 3.1.

Таблиця 3.1 – Короткий опис зв'язків

Сутність 1	Зв'язок	Сутність 2	Тип зв'язку
machines	має	repair	1:M
technical_data_sheet	пов'язана з	batch	1:M
technical_data_sheet	містить список	ingredient_item	1:M
ingredient	міститься у	weighing_item	1:M
ingredient	міститься у	ingredient_item	1:M
weighing_data	містить список	weighing_item	1:M
batch	взаємодіє з	weighing_data	1:1
batch	містить список	battery	1:M
batch	взаємодіє з	mixing_data	1:1
battery	взаємодіє з	stacking_data	1:1
battery	взаємодіє з	cutting_data	1:1
battery	взаємодіє з	cycle_test_data	1:1
battery	взаємодіє з	dosing_data	1:1
battery	взаємодіє з	short_circuit_data	1:1
battery	взаємодіє з	drying_data	1:1
battery	взаємодіє з	cell_electrolyte_data	1:1
battery	взаємодіє з	overheat_test_data	1:1
battery	взаємодіє з	pressing_data	1:1

Опис зв'язків:

- одна модель машини може мати декілька рапортів на ремонт, один рапорт на ремонт може бути лише для однієї машини;
- одна технічна карта може бути для декількох партій, одна партія може мати лише одну технічну карту;
- одна технічна карта може мати список `ingredient_item`, тоді як один `ingredient_item` відноситься лише до однієї технічної карти;
- один інгредієнт може міститися у декількох `weighing_item`, тоді як один `weighing_item` містить лише один інгредієнт;
- один інгредієнт може міститися у декількох `ingredient_item`, тоді як один `ingredient_item` містить лише один інгредієнт;
- один `weighing_data` може містити список з `weighing_item`, тоді як один `weighing_item` може міститися лише в одному `weighing_data`;
- одна партія може взаємодіяти з одним `weighing_data`, тоді як один `weighing_data` може бути у однієї партії;
- одна партія має список з батарей, тоді як одна батарея міститься лише у одній партії;
- одна партія може мати лише один набір даних про замішування, тоді як один набір даних створюється лише для однієї партії;
- одна батарея має лише один набір даних про вкладання, тоді як один набір даних створюється лише для однієї батареї;
- одна батарея має лише один набір даних про нарізання, тоді як один набір даних створюється лише для однієї батареї;
- одна батарея має лише один набір даних про тестування циклами, тоді як один набір даних створюється лише для однієї батареї;
- одна батарея має лише один набір даних про нанесення пасти, тоді як один набір даних створюється лише для однієї батареї;

- одна батарея має лише один набір даних про тестування короткого замикання, тоді як один набір даних створюється лише для однієї батареї;
- одна батарея має лише один набір даних про висушування, тоді як один набір даних створюється лише для однієї батареї;
- одна батарея має лише один набір даних про заливання електроліту, тоді як один набір даних створюється лише для однієї батареї;
- одна батарея має лише один набір даних про тестування на перегрів, тоді як один набір даних створюється лише для однієї батареї;
- одна батарея має лише один набір даних про пресування, тоді як один набір даних створюється лише для однієї батареї.

Таблиця `batch` містить інформацію про створені партії акумуляторів, їх основні характеристики та бажану кількість батарей. Її атрибути:

- `id` (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- `production_start` (datetime) – дата та час початку виготовлення партії;
- `number_of_batch` (Integer) – ідентифікаційний номер партії;
- `quantity_of_batteries` (Integer) – кількість акумуляторів, заявлена клієнтом;
- `nominal_capacity` (Integer) – заявлена (референсна) ємність батарей;
- `batch_status` (enum) – статус партії протягом виробництва. Може мати значення: `PLANNED`, `READY_FOR_PRODUCTION`, `PRESSING_COMPLETED`, `ERROR`, `WEIGHING_CHECKED`, `WEIGHING_COMPLETED`, `MIXING_CHECKED`, `MIXING_COMPLETED`, `DOSING_COMPLETED`, `CUTTING_COMPLETED`, `STACKING_COMPLETED`, `CELL_ELECTROLYTE_COMPLETED`, `TESTING_CYCLE_COMPLETED`, `TESTING_OVERHEAT_COMPLETED`, `DRYING_COMPLETED`, `COMPLETED`.
- `batteries_model` (String) – модель батарей партії (їх тип);
- `rated_voltage` (Double) – заявлена номінальна напруга для всіх батарей партії;

– `technical_sheet_id` (Long) – зовнішній ключ таблиці `technical_data_sheet`.

На рисунку 3.5 наведена структура таблиці.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<code>id</code> 🔑	<code>bigint(20)</code>			No	<i>None</i>		<code>AUTO_INCREMENT</code>
2	<code>batteries_model</code>	<code>varchar(255)</code>	<code>utf8_general_ci</code>		Yes	<i>NULL</i>		
3	<code>nominal_capacity</code>	<code>int(11)</code>			Yes	<i>NULL</i>		
4	<code>number_of_batch</code> 🧠	<code>int(11)</code>			Yes	<i>NULL</i>		
5	<code>quantity_of_batteries</code>	<code>int(11)</code>			Yes	<i>NULL</i>		
6	<code>rated_voltage</code>	<code>double</code>			Yes	<i>NULL</i>		
7	<code>technical_sheet_id</code> 🧠	<code>bigint(20)</code>			Yes	<i>NULL</i>		
8	<code>batch_status</code>	<code>enum('PLANNED','ERROR','WEIGHING_CHECKED','WEIG...</code>	<code>utf8_general_ci</code>		Yes	<i>NULL</i>		
9	<code>production_start</code>	<code>datetime(6)</code>			Yes	<i>NULL</i>		

Рисунок 3.5 – Структура таблиці `batch`

Таблиця `battery` містить інформацію про конкретні батареї, які було виготовлено (або заплановано до виготовлення) на виробництві. Атрибути:

- `id` (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- `actual_capacity` (double) – фактична ємність виготовленої батареї;
- `actual_internal_resistance` (double) – фактичний внутрішній опір батареї;
- `serial_number` (varchar(255)) – серійний номер батареї;
- `status` (enum) – статус батареї протягом виготовлення. Може мати значення: `DEFAULT`, `DOSING_COMPLETED`, `DRYING_DEFECTIVE`, `DRYING_COMPLETED`, `DOSING_DEFECTIVE`, `PRESSING_COMPLETED`, `PRESSING_DEFECTIVE`, `CUTTING_COMPLETED`, `CUTTING_DEFECTIVE`, `STACKING_COMPLETED`, `STACKING_DEFECTIVE`, `CELL_ELECTROLYTE_COMPLETED`, `CELL_ELECTROLYTE_DEFECTIVE`, `TESTING_CYCLE_COMPLETED`, `TESTING_CYCLE_DEFECTIVE`, `TESTING_OVERHEAT_COMPLETED`, `TESTING_OVERHEAT_DEFECTIVE`, `TESTING_SHORT_CIRCUIT_DEFECTIVE`, `WORKING`;
- `voltage` (double) – фактична напруга батареї;
- `weight` (double) – фактична вага батареї;

– batch_id (Long) – зовнішній ключ таблиці batch.

На рисунку 3.6 наведено структуру таблиці battery.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 actual_capacity	double			Yes	NULL		
<input type="checkbox"/>	3 actual_internal_resistance	double			Yes	NULL		
<input type="checkbox"/>	4 serial_number	varchar(255)	utf8_general_ci		Yes	NULL		
<input type="checkbox"/>	5 status	enum('WORKING','DOSING_COMPLETED','DRYING_DEFECT')	utf8_general_ci		Yes	NULL		
<input type="checkbox"/>	6 voltage	double			Yes	NULL		
<input type="checkbox"/>	7 weight	double			Yes	NULL		
<input type="checkbox"/>	8 batch_id	bigint(20)			No	None		

Рисунок 3.6 – Структура таблиці battery

Таблиця ingredient містить інформацію про рідкі та тверді інгредієнти, необхідні для виготовлення акумуляторів:

- id (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- title (varchar(255)) – назва інгредієнта;
- delivery_date (date) – дата прийому інгредієнта виробництвом;
- expiration_date (date) – дата закінчення терміну придатності інгредієнта;
- quantity_of_solid_ingredients (integer) – кількість (у штуках) одиниць твердого інгредієнту, що отримало виробництво;
- quantity_of_liquid_ingredients (double) – кількість (у грамах) одиниць рідкого інгредієнту, що отримало виробництво;
- provider (varchar(255)) – назва постачальника інгредієнта;
- type (enum) – тип інгредієнта (його призначення). Може приймати значення: ANODE_SUSPENSION, CATHODE_SUSPENSION, ELECTROLYTE, SEPARATOR, COLLECTOR.

На рисунку 3.7 наведено структуру таблиці ingredient.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id 🗝️	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 delivery_date	date			Yes	NULL		
<input type="checkbox"/>	3 expiration_date	date			Yes	NULL		
<input type="checkbox"/>	4 quantity_of_solid_ingredients	int(11)			Yes	NULL		
<input type="checkbox"/>	5 type	enum('ANODE_SUSPENSION', 'CATHODE_SUSPENSION', 'EL...')	utf8_general_ci		Yes	NULL		
<input type="checkbox"/>	6 weight_of_liquid_ingredients	double			Yes	NULL		
<input type="checkbox"/>	7 provider	varchar(255)	utf8_general_ci		Yes	NULL		
<input type="checkbox"/>	8 title	varchar(255)	utf8_general_ci		Yes	NULL		

Рисунок 3.7 – Структура таблиці ingredient

Таблиця gearair містить інформацію про створені диспетчерами рапорти на ремонт обладнання. Атрибути:

- id (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- problem (varchar(255)) – містить інформацію про проблему, через яку створюється даний рапорт. Заповнюється диспетчером;
- machines_model (Long) – зовнішній ключ таблиці machines;
- process (varchar(255)) – містить інформацію про процес, при виконанні якого сталась поломка. Заповнюється диспетчером;
- abnormal_value (float) – містить інформацію про числовий параметр, який не є прийнятним для технічної карти. Заповнюється диспетчером;
- normal_value_min (float) – містить інформацію про нижню межу діапазону, який є прийнятним для технічної карти. Заповнюється диспетчером;
- normal_value_max (float) – містить інформацію про верхню межу діапазону, який є прийнятним для технічної карти. Заповнюється диспетчером;
- possible_reason (varchar(255)) – містить інформацію про процес, при виконанні якого сталась поломка. Заповнюється диспетчером;
- details (varchar(255)) – заповнюється бригадиром після проведення ремонту. Містить деталі ремонту;

– spent_time (time(6)) – заповнюється бригадиром після проведення ремонту.

Містить дані про те, скільки часу було витрачено на ремонт;

– status (enum) – статус рапорту. При створенні його диспетчером встановлюється статус SENT. Після заповнення бригадиром усіх незаповнених полів статус стає COMPLETED;

– creating_date (datetime) – дата створення рапорту.

На рисунку 3.8 наведено структуру таблиці repair.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id 🗝️	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 problem	varchar(255)	utf8_general_ci		Yes	NULL		
<input type="checkbox"/>	3 machines_model 🧠	bigint(20)			No	None		
<input type="checkbox"/>	4 process	varchar(255)	utf8_general_ci		Yes	NULL		
<input type="checkbox"/>	5 abnormal_value	float			No	None		
<input type="checkbox"/>	6 normal_value_max	float			No	None		
<input type="checkbox"/>	7 normal_value_min	float			No	None		
<input type="checkbox"/>	8 possible_reasons	varchar(255)	utf8_general_ci		Yes	NULL		
<input type="checkbox"/>	9 details	varchar(255)	utf8_general_ci		Yes	NULL		
<input type="checkbox"/>	10 status	enum('SENT', 'COMPLETED')	utf8_general_ci		Yes	NULL		
<input type="checkbox"/>	11 spent_time	time(6)			Yes	NULL		
<input type="checkbox"/>	12 creating_date	datetime(6)			Yes	NULL		

Рисунок 3.8 – Структура таблиці repair

Таблиця machines містить дані про всі моделі машин на виробництві та їх тип (призначення). Атрибути:

– id (Long) – унікальний ідентифікатор таблиці, первинний ключ;

– model (varchar(255)) – описує моделі машин, що використовуються у підприємстві;

– type (varchar(255)) – описує призначення машин, яке вони виконують.

На рисунках 3.9-3.10 наведено структуру та вміст таблиці machines.


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id 	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 model	varchar(255)	utf8_general_ci		Yes	NULL		
<input type="checkbox"/>	3 type	varchar(255)	utf8_general_ci		Yes	NULL		

Рисунок 3.9 – Структура таблиці machines

id	model	type
1	TOB-XFZH01	mixerpasta
2	ElectroMixSBM-500	mixerelectrolyte
3	TOB-JS350-3.0	syringemachines
4	DR-H150-200	hydrpress
5	TOB-MSK-300	lasercutter
6	TOB-YF200-JZ	sealingchamber
7	TOB-ZYJ-02	cellelectrolytem
8	DZF-6050	dryoven
9	TOB-ADP-300B	stackmash
10	SuperMix1890	mixerpasta
11	MonstaX1900	mixerpasta
12	EleSuper100	mixerelectrolyte
14	TOB-HRP-300	hydrpress
15	MTechFL3019	lasercutter
16	TOB-BFZ-200-G	sealingchamber
17	TOB-ZY300	cellelectrolytem
18	TOB-DZF-6510	dryoven
19	TOB-S-DP-150	stackmash
20	AKN800NH0385	syringemachines
21	SCALES-2000	scale

Рисунок 3.10 – Вміст таблиці machines

Таблиця users містить інформацію про користувачів системи, їх логіни, паролі та роль. Атрибути:

- id (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- password (varchar(20)) – пароль користувача;
- username (varchar(100)) – логін користувача;
- role (varchar(20)) – роль, яка перевіряється при авторизації користувача.

На рисунках 3.11-3.12 наведено структуру та вміст таблиці users.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 password	varchar(20)	utf8_general_ci		No	None		
<input type="checkbox"/>	3 role	varchar(70)	utf8_general_ci		No	None		
<input type="checkbox"/>	4 username	varchar(100)	utf8_general_ci		No	None		

Рисунок 3.11 – Структура таблиці users

id	password	role	username
1	123456	ROLE_PLANNER	sofia@vt
2	234567	ROLE_DISPATCHER	elisa@vt
3	345678	ROLE_REPAIRER	oleh@vt
4	070103	ROLE_DISPATCHER_WEIGHING	olena@vt
5	050670	ROLE_DISPATCHER_MIXING	lina@vt
6	310802	ROLE_DISPATCHER_DOSING	jack@vt
7	220868	ROLE_DISPATCHER_DRYING	kessy@vt
8	160391	ROLE_DISPATCHER_PRESSING	eugen@vt
9	120693	ROLE_DISPATCHER_CUTTING	jane@vt
10	080912	ROLE_DISPATCHER_STACKING	gregory@vt
11	567089	ROLE_DISPATCHER_FILLING	helen@vt
12	431267	ROLE_DISPATCHER_TESTING	tori@vt
13	347823	ROLE_LABORANT	derek@vt

Рисунок 3.12 – Вміст таблиці users

Таблиця `technical_data_sheet` містить дані про технічну карту, в якій містяться референтні значення для виготовлення партії акумуляторів. Атрибути:

- `id` (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- `title` (varchar(255)) – назва технічної карти;
- `battery_model` (varchar(255)) – модель батарей, для якої створюється карта;
- `time_mixing_suspension_min` (integer) – мінімальне значення часу, за який міксер повинен створити суспензію;
- `time_mixing_suspension_max` (integer) – максимальне значення часу, за який міксер повинен створити суспензію;

- `time_mixing_electrolyte_min` (integer) – мінімальне значення часу, за який міксер повинен створити електроліт;
- `time_mixing_electrolyte_max` (integer) – максимальне значення часу, за який міксер повинен створити електроліт;
- `thickness_substance_min` (double) – мінімальне значення товщини суспензії, що має бути нанесено на електроди;
- `thickness_substance_max` (double) – максимальне значення товщини суспензії, що має бути нанесено на електроди;
- `time_of_drying_min` (integer) – мінімальне значення часу висушування електродів у печі;
- `time_of_drying_max` (integer) – максимальне значення часу висушування електродів у печі;
- `temperature_of_drying_min` (integer) – мінімальне значення температури висушування електродів у печі;
- `temperature_of_drying_max` (integer) – максимальне значення температури висушування електродів у печі;
- `compression_ratio_min` (double) – мінімальне значення коефіцієнту стискання електродів пресом;
- `compression_ratio_max` (double) – максимальне значення коефіцієнту стискання електродів пресом;
- `xray_alignment_deviation_min` (integer) – мінімальне значення відхилення при вкладанні елементів у корпус;
- `xray_alignment_deviation_max` (integer) – максимальне значення відхилення при вкладанні елементів у корпус;
- `empty_battery_weight_min` (double) – мінімальне значення ваги порожнього корпусу;

- empty_battery_weight_max (double) – максимальне значення ваги порожнього корпусу;
- filled_battery_weight_min (double) – мінімальне значення ваги заповненого корпусу;
- filled_battery_weight_max (double) – максимальне значення ваги заповненого корпусу;
- voltage_first_test_min (double) – мінімальне значення напруги батареї при тестуванні на повне зарядження/розрядження;
- voltage_first_test_max (double) – максимальне значення напруги батареї при тестуванні на повне зарядження/розрядження;
- capacity_first_test_min (double) – мінімальне значення ємності батареї при тестуванні на повне зарядження/розрядження;
- capacity_first_test_max (double) – максимальне значення ємності батареї при тестуванні на повне зарядження/розрядження;
- resistance_first_test_min (double) – мінімальне значення внутрішнього опору батареї при тестуванні на повне зарядження/розрядження;
- resistance_first_test_max (double) – максимальне значення внутрішнього опору батареї при тестуванні на повне зарядження/розрядження;
- case_temp_second_test_min (integer) – мінімальне значення температури корпусу при тестуванні на перегрів;
- case_temp_second_test_max (integer) – максимальне значення температури корпусу при тестуванні на перегрів;
- cell_voltage (double) – значення напруги елементів при тестуванні на перегрів;
- resistance_second_test_min (double) – мінімальне значення внутрішнього опору батареї при тестуванні на перегрів;

- `resistance_second_test_max` (double) – максимальне значення внутрішнього опору батареї при тестуванні на перегрів;
- `voltage_third_test` (double) – значення напруги при тестуванні на коротке замикання;
- `capacity_third_test_min` (double) – мінімальне значення ємності батареї при тестуванні на коротке замикання;
- `capacity_third_test_max` (double) – максимальне значення ємності батареї при тестуванні на коротке замикання;
- `case_temp_third_test_min` (integer) – мінімальне значення температури корпусу при тестуванні на коротке замикання;
- `case_temp_third_test_max` (integer) – максимальне значення температури корпусу при тестуванні на коротке замикання;
- `current_third_test_min` (double) – мінімальне значення струму при тестуванні на коротке замикання;
- `current_third_test_max` (double) – максимальне значення струму при тестуванні на коротке замикання;
- `resistance_third_test_min` (double) – мінімальне значення внутрішнього опору батареї при тестуванні на коротке замикання;
- `resistance_third_test_max` (double) – максимальне значення внутрішнього опору батареї при тестуванні на коротке замикання.

На рисунку 3.13 наведено структуру та вміст таблиці `technical_data_sheet`.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id 🔑	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 title	varchar(255)	utf8_general_ci		Yes	NULL		
<input type="checkbox"/>	3 battery_model	varchar(255)	utf8_general_ci		Yes	NULL		
<input type="checkbox"/>	4 time_mixing_suspension_min	bigint(20)			Yes	NULL		
<input type="checkbox"/>	5 time_mixing_suspension_max	bigint(20)			Yes	NULL		
<input type="checkbox"/>	6 time_mixing_electrolyte_min	bigint(20)			Yes	NULL		
<input type="checkbox"/>	7 time_mixing_electrolyte_max	bigint(20)			Yes	NULL		
<input type="checkbox"/>	8 thickness_substance_min	double			Yes	NULL		
<input type="checkbox"/>	9 thickness_substance_max	double			Yes	NULL		
<input type="checkbox"/>	10 time_drying_min	bigint(20)			Yes	NULL		
<input type="checkbox"/>	11 time_drying_max	bigint(20)			Yes	NULL		
<input type="checkbox"/>	12 temperature_drying_min	int(11)			Yes	NULL		
<input type="checkbox"/>	13 temperature_drying_max	int(11)			Yes	NULL		
<input type="checkbox"/>	14 compression_ratio_min	double			Yes	NULL		
<input type="checkbox"/>	15 compression_ratio_max	double			Yes	NULL		
<input type="checkbox"/>	16 cutting_accuracy_min	double			Yes	NULL		
<input type="checkbox"/>	17 cutting_accuracy_max	double			Yes	NULL		
<input type="checkbox"/>	18 xray_alignment_deviation_min	int(11)			Yes	NULL		
<input type="checkbox"/>	19 xray_alignment_deviation_max	int(11)			Yes	NULL		
<input type="checkbox"/>	20 empty_battery_weight_min	double			Yes	NULL		
<input type="checkbox"/>	21 empty_battery_weight_max	double			Yes	NULL		
<input type="checkbox"/>	22 filled_battery_weight_min	double			Yes	NULL		
<input type="checkbox"/>	23 filled_battery_weight_max	double			Yes	NULL		
<input type="checkbox"/>	24 voltage_first_test_min	double			Yes	NULL		
<input type="checkbox"/>	25 voltage_first_test_max	double			Yes	NULL		
<input type="checkbox"/>	26 capacity_first_test_min	double			Yes	NULL		
<input type="checkbox"/>	27 capacity_first_test_max	double			Yes	NULL		
<input type="checkbox"/>	28 resistance_first_test_min	double			Yes	NULL		
<input type="checkbox"/>	29 resistance_first_test_max	double			Yes	NULL		
<input type="checkbox"/>	30 case_temp_second_test_min	int(11)			Yes	NULL		
<input type="checkbox"/>	31 case_temp_second_test_max	int(11)			Yes	NULL		
<input type="checkbox"/>	32 cell_voltage	double			Yes	NULL		
<input type="checkbox"/>	33 resistance_second_test_min	double			Yes	NULL		
<input type="checkbox"/>	34 resistance_second_test_max	double			Yes	NULL		
<input type="checkbox"/>	35 voltage_third_test	double			Yes	NULL		
<input type="checkbox"/>	36 capacity_third_test_max	double			Yes	NULL		
<input type="checkbox"/>	37 capacity_third_test_min	double			Yes	NULL		

Рисунок 3.13 – Структура таблиці technical_data_sheet, аркуш 1

<input type="checkbox"/>	38	case_temp_third_test_min	int(11)	Yes	NULL
<input type="checkbox"/>	39	case_temp_third_test_max	int(11)	Yes	NULL
<input type="checkbox"/>	40	current_third_test_min	double	Yes	NULL
<input type="checkbox"/>	41	current_third_test_max	double	Yes	NULL
<input type="checkbox"/>	42	resistance_third_test_max	double	Yes	NULL
<input type="checkbox"/>	43	resistance_third_test_min	double	Yes	NULL

Рисунок 3.13, аркуш 2

Таблиця `ingredient_item` призначена для зберігання інгредієнтів та їх кількості, необхідних для конкретної технічної карти. Атрибути:

- `id` (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- `quantity` (double) – кількість інгредієнту, необхідна для виготовлення партії;
- `ingredient` (Long) – зовнішній ключ таблиці `ingredient`;
- `technical_data_sheet` (Long) – зовнішній ключ таблиці `technical_data_sheet`.

На рисунку 3.14 наведено структуру таблиці `ingredient_item`.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id 🔑	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 quantity	double			Yes	NULL		
<input type="checkbox"/>	3 ingredient 🔑	bigint(20)			Yes	NULL		
<input type="checkbox"/>	4 technical_data_sheet 🔑	bigint(20)			Yes	NULL		

Рисунок 3.14 – Структура таблиці `ingredient_item`

Таблиця `weighing_item` призначена для зберігання інгредієнтів та їх фактичної кількості, вимірних за допомогою промислових вагів. Атрибути:

- `id` (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- `real_quantity` (double) – фактична кількість інгредієнту, виміряна для виготовлення партії;
- `ingredient_id` (Long) – зовнішній ключ таблиці `ingredient`;
- `weighing_data_id` (Long) – зовнішній ключ таблиці `weighing_data`.

На рисунку 3.15 наведено структуру таблиці `weighing_item`.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 real_quantity	double			Yes	NULL		
<input type="checkbox"/>	3 ingredient_id	bigint(20)			No	None		
<input type="checkbox"/>	4 weighing_data_id	bigint(20)			No	None		

Рисунок 3.15 – Структура таблиці `weighing_item`

Таблиця `weighing_data` необхідна для зв'язку між вимірними значеннями кількості інгредієнтів з конкретною партією. Атрибути:

- `id` (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- `batch_id` (Long) – зовнішній ключ таблиці `batch_id`.

На рисунку 3.16 наведено структуру таблиці `weighing_data`.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 batch_id	bigint(20)			No	None		

Рисунок 3.16 – Структура таблиці `weighing_data`

Таблиця `mixing_data` призначена для зберігання фактичних даних про результати замішування суспензії та електроліту відповідними міксерами.

Атрибути:

- `id` (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- `fact_time_suspension` (integer) – фактичний час замішування суспензії міксером;
- `fact_time_electrolyte` (integer) – фактичний час замішування електроліту міксером;
- `batch_id` (Long) – зовнішній ключ таблиці `batch_id`.

На рисунку 3.17 наведено структуру таблиці `mixing_data`.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 fact_time_suspension	bigint(6)			Yes	NULL		
<input type="checkbox"/>	3 batch_id	bigint(20)			No	None		
<input type="checkbox"/>	4 fact_time_electrolyte	bigint(20)			No	None		

Рисунок 3.17 – Структура таблиці `mixing_data`

Таблиця `dosing_data` призначена для зберігання фактичних даних про результати нанесення суспензії на електроди машиною. Атрибути:

- `id` (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- `applied_thickness` (double) – фактичне значення товщини нанесеної суспензії;
- `battery_id` (Long) – зовнішній ключ таблиці `battery_id`.

На рисунку 3.18 наведено структуру таблиці `dosing_data`.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 applied_thickness	double			Yes	NULL		
<input type="checkbox"/>	3 battery_id	bigint(20)			No	None		

Рисунок 3.18 – Структура таблиці `dosing_data`

Таблиця `drying_data` призначена для зберігання фактичних даних про результати сушіння електродів піччю. Атрибути:

- `id` (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- `start_time` (time) – зафіксований час початку сушіння електродів;
- `end_time` (time) – зафіксований час закінчення сушіння електродів;
- `furnace_temp` (integer) – фактичне значення температури сушіння електродів;
- `battery_id` (Long) – зовнішній ключ таблиці `battery_id`.

На рисунку 3.19 наведено структуру таблиці `drying_data`.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 end_time	time(6)			Yes	NULL		
<input type="checkbox"/>	3 furnace_temp	int(11)			Yes	NULL		
<input type="checkbox"/>	4 start_time	time(6)			Yes	NULL		
<input type="checkbox"/>	5 battery_id	bigint(20)			No	None		

Рисунок 3.19 – Структура таблиці `drying_data`

Таблиця `pressing_data` призначена для зберігання фактичних даних про результати пресування електродів машиною. Атрибути:

- `id` (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- `compression_ratio` (double) – фактичний коефіцієнт стискання електродів;
- `battery_id` (Long) – зовнішній ключ таблиці `battery_id`.

На рисунку 3.20 наведено структуру таблиці `pressing_data`.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 compression_ratio	double			Yes	NULL		
<input type="checkbox"/>	3 battery_id	bigint(20)			No	None		

Рисунок 3.20 – Структура таблиці `pressing_data`

Таблиця `cutting_data` призначена для зберігання фактичних даних про результати нарізання електродів машиною. Атрибути:

- `id` (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- `cutting_accuracy` (double) – фактична точність нарізання електродів;
- `battery_id` (Long) – зовнішній ключ таблиці `battery_id`.

На рисунку 3.21 наведено структуру таблиці `cutting_data`.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 cutting_accuracy	double			Yes	NULL		
<input type="checkbox"/>	3 battery_id	bigint(20)			No	None		

Рисунок 3.21 – Структура таблиці cutting_data

Таблиця stacking_data призначена для зберігання фактичних даних про результати вкладання елементів у корпус машиною. Атрибути:

- id (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- xray_alignment_deviation (integer) – фактична точність вкладання елементів у корпус;
- battery_id (Long) – зовнішній ключ таблиці battery_id.

На рисунку 3.22 наведено структуру таблиці stacking_data.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 xray_alignment_deviation	int(11)			Yes	NULL		
<input type="checkbox"/>	3 battery_id	bigint(20)			No	None		

Рисунок 3.22 – Структура таблиці stacking_data

Таблиця cell_electrolyte_data призначена для зберігання фактичних даних про результати заливки електроліту у корпус машиною. Атрибути:

- id (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- empty_battery_weight (double) – вага корпусу до заливки електроліту;
- filled_battery_weight (double) – вага корпусу після заливки електроліту;
- battery_id (Long) – зовнішній ключ таблиці battery_id;

На рисунку 3.23 наведено структуру таблиці cell_electrolyte_data:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 empty_battery_weight	double			Yes	NULL		
<input type="checkbox"/>	3 filled_battery_weight	double			Yes	NULL		
<input type="checkbox"/>	4 battery_id	bigint(20)			No	None		

Рисунок 3.23 – Структура таблиці cell_electrolyte_data

Таблиця cycle_test_data призначена для зберігання фактичних даних про результати тестування на цикл заряджання/розряджання акумулятора. Атрибути:

- id (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- capacity (double) – зафіксована ємність батареї;
- resistance (double) – зафіксований внутрішній опір батареї;
- voltage (double) – зафіксована напруга батареї;
- battery_id (Long) – зовнішній ключ таблиці battery_id.

На рисунку 3.24 наведено структуру таблиці cycle_test_data.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 capacity	double			Yes	NULL		
<input type="checkbox"/>	3 resistance	double			Yes	NULL		
<input type="checkbox"/>	4 voltage	double			Yes	NULL		
<input type="checkbox"/>	5 battery_id	bigint(20)			No	None		

Рисунок 3.24 – Структура таблиці cycle_test_data

Таблиця overheat_test_data призначена для зберігання фактичних даних про результати тестування на перегрів. Атрибути:

- id (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- cell_voltage (double) – зафіксована напруга елементів всередині батареї;
- case_temperature (integer) – зафіксована температура корпусу батареї;
- resistance (double) – зафіксований внутрішній опір батареї;

– battery_id (Long) – зовнішній ключ таблиці battery_id.

На рисунку 3.25 наведено структуру таблиці overhear_test_data.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id 🔑	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 cell_voltage	double			Yes	NULL		
<input type="checkbox"/>	3 case_temperature	int(11)			Yes	NULL		
<input type="checkbox"/>	4 resistance	double			Yes	NULL		
<input type="checkbox"/>	5 battery_id 🔑	bigint(20)			No	None		

Рисунок 3.25 – Структура таблиці overhear_test_data

Таблиця short_circuit_test призначена для зберігання фактичних даних про результати тестування на коротке замикання. Атрибути:

- id (Long) – унікальний ідентифікатор таблиці, первинний ключ;
- capacity (double) – зафіксована ємність елементів всередині батареї;
- case_temp (integer) – зафіксована температура корпусу батареї;
- current (integer) – зафіксоване значення струму батареї;
- resistance (double) – зафіксований внутрішній опір батареї;
- voltage (double) – зафіксована напруга батареї;
- battery_id (Long) – зовнішній ключ таблиці battery_id.

На рисунку 3.26 наведено структуру таблиці short_circuit_test.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1 id 🔑	bigint(20)			No	None		AUTO_INCREMENT
<input type="checkbox"/>	2 capacity	double			Yes	NULL		
<input type="checkbox"/>	3 case_temp	int(11)			Yes	NULL		
<input type="checkbox"/>	4 current	double			Yes	NULL		
<input type="checkbox"/>	5 resistance	double			Yes	NULL		
<input type="checkbox"/>	6 voltage	double			Yes	NULL		
<input type="checkbox"/>	7 battery_id 🔑	bigint(20)			No	None		

Рисунок 3.26 – Структура таблиці short_circuit_test

У даному розділі спроектовано базу даних підсистеми автоматизації управління якістю виготовлення телефонних акумуляторів із урахуванням вимог до цілісності та узгодженості даних. Виконана нормалізація таблиць та структура зберігання інформації забезпечують зменшення надлишковості даних і створюють основу для ефективного контролю та аналізу якості виробничих процесів.

4 РЕАЛІЗАЦІЯ ПІДСИСТЕМИ АВТОМАТИЗАЦІЇ ДЛЯ УПРАВЛІННЯ ЯКІСТЮ ВИГОТОВЛЕННЯ ТЕЛЕФОННИХ АКУМУЛЯТОРІВ У ВИГЛЯДІ ПРОГРАМНОГО ЗАСОБУ

4.1 Вибір мов програмування, середовищ розробки та додаткових інструментів

Виходячи з аналізу технологій розробки корпоративних систем, в якості основної мови програмування було обрано Java, а в якості архітектурного каркасу – Spring Framework. Java – об'єктно-орієнтована та крос-платформна мова програмування, випущена компанією Sun Microsystems у 1995 році (з 2009 року правами володіє компанія Oracle) [25]. Для задач автоматизації управління якістю на виробництві ключовими факторами вибору стали:

- статична типізація. Механізми типізації мови Java надають можливість виявляти більшість помилок на етапі компіляції, що є критично важливим для системи, яка керує даними з виробництва. Це знижує ризик появи помилок часу виконання при обробці параметрів;

- кросплатформність. Завдяки використанню JVM розроблений програмний модуль може бути розгорнуто на різних серверних інфраструктурах (Linux, Windows), які використовуються на підприємстві, без перекомпіляції коду;

- керування пам'яттю. Вбудований механізм автоматичного керування пам'яттю Garbage Collector запобігає витокам інформації при довготривалій та безперервній роботі сервера підсистеми.

Розробка програми без використання допоміжних інструментів потребує значних трудовитрат та часу на реалізацію коду. Тому було прийнято рішення

використовувати Spring Framework у конфігурації Spring Boot. Даний вибір забезпечує наступні архітектурні переваги:

- інверсію управління та впровадження залежностей. Spring контейнер керує життєвим циклом об'єктів додатку. Це забезпечує слабкий зв'язок між компонентами, що дозволяє швидко модифікувати бізнес-логіку або проводити модульне тестування компонентів ізольовано один від одного;

- доступ до екосистеми Spring Data JPA. Для взаємодії з обраною СУБД MySQL використовується модуль Spring Data JPA, що реалізує специфікацію ORM. Це надає можливість працювати із записами в базі даних як з Java-об'єктами, виключаючи необхідність писати SQL-код власноруч та захищати підсистему від SQL-ін'єкцій;

- Spring Security. Промислова підсистема вимагає розмежування прав доступу до сторінок. Інтегрований фреймворк безпеки дозволяє гнучко налаштувати аутентифікацію та авторизацію користувачів на рівні окремих методів та URL-адрес.

- архітектурний стиль REST. Spring Web MVC надає інструменти для створення RESTful API, що спрощує інтеграцію розроблюваної підсистеми з іншими модулями MES або зовнішніми системами шляхом використання протоколу HTTP та формату JSON.

Таким чином, поєднання технологій Java та Spring забезпечує створення підтримуваної та розширюваної підсистеми, яка цілком відповідає вимогам сучасного виробництва.

Для розробки модуля AI було обрано мову програмування Python, який на сьогодні є загально визнаним промисловим стандартом у галузях аналітики даних та машинного навчання. Ключовим фактором вибору стала наявність розвиненої екосистеми спеціалізованих бібліотек, таких як pandas для обробки та аналізу структурованих даних, numpy для виконання високоефективних математичних

операцій та scikit-learn, що надає оптимізовані реалізації алгоритмів класифікації, зокрема Random Forest. Використання Python дозволило зосередити увагу на логіці побудови моделі та налаштуванні її параметрів, мінімізуючи витрати часу на реалізацію низькорівневих математичних обчислень.

Для реалізації вебінтерфейсу модульної взаємодії між моделлю та основною системою було обрано фреймворк Flask. Вибір на його користь пояснюється архітектурними вимогами проекту, які передбачають використання мікросервісного підходу. Оскільки роль модуля штучного інтелекту в системі зводиться до отримання вхідних даних у форматі JSON, виконання прогнозу та повернення результату, використання компактного фреймворку є найбільш доцільним рішенням. Flask забезпечує мінімальні накладні витрати ресурсів, простоту розгортання REST API ендпоінтів та високу швидкість обробки HTTP-запитів. Така архітектура дозволила чітко розмежувати відповідальність компонентів системи, де Java відповідає за бізнес-логіку та роботу з базою даних, а зв'язка Python та Flask забезпечує ізольоване виконання задач інтелектуального аналізу даних.

Для реалізації основної серверної частини додатку було обрано середовище IntelliJ IDEA, яке на сьогодні є стандартом у галузі корпоративної Java-розробки. Вирішальним фактором на користь цього середовища стала його глибока інтеграція з екосистемою Spring Framework. IntelliJ IDEA надає розширені можливості для роботи з контекстом Spring Boot, дозволяє візуалізувати залежності між бінами, автоматизувати конфігурацію проекту через Maven та забезпечує зручні вбудовані інструменти для роботи з базами даних.

Для розробки модуля інтелектуального аналізу даних було використано середовище PyCharm. Його вибір зумовлений наявністю специфічних інструментів для проєктів, зокрема зручним інтерфейсом для управління віртуальними середовищами та пакетами бібліотек машинного навчання. PyCharm забезпечує

ефективну підтримку веб-фреймворку Flask, що дозволило спростити процес налагодження REST API ендпоінтів та тестування моделі прогнозування.

4.2 Принцип роботи AI-модуля

Відповідно до вимог, описаних у розділі 2, було створено AI-модуль, задачею якого є автоматичне визначення коректності технологічних параметрів сушіння електродів (температури печі, тривалості сушіння та товщини нанесеного покриття). Система підтримує підключення AI для будь-якого етапу, але в рамках програмного макету було навчено модель для найбільш критичної частини виробництва, адже етап нанесення пасти та сушіння має безпосередній вплив на ємність та довговічність батареї. Модель виступає у якості експертної надбудови, що імітує поведінку лаборанта, і оцінює, наскільки поточні параметри є валідними з огляду на досвід попередніх партій.

Вона навчається на комбінації двох типів даних:

- реальних записів виробничого процесу, отриманих з бази даних, які містять фактичну товщину суспензії, температуру печі та час сушіння для кожної батареї;
- синтетичних даних, штучно згенерованих на основі фізичних залежностей процесу. Вони необхідні для того, щоб модель розрізняла граничні випадки, помилки лаборантів при занесенні даних та аномальні значення.

Для кожного навчального прикладу розраховується цільовий статус, який являє собою один з трьох класів: 0 – ОК (параметри відповідають нормі); 1 – OVERHEAT (пересушування або надлишкове значення температури, що призводить до розтріскування покриття); 2 – UNDERHEAT (недосушування або низька температура, що призводить до надмірного рівня зволоженості покриття).

Визначення статусу виконується на основі інженерної моделі:

– товщина покриття впливає на оптимальну температуру: чим більше товщина, тим нижче повинна бути температура;

– товщина покриття впливає на оптимальний час сушіння: чим більше товщина, тим більше повинен бути час [26].

Для кожної партії розраховуються ідеальні значення температури та часу, а потім аналізуються відхилення фактичних параметрів від цих значень з урахуванням технологічних допусків. Якщо час або температура виходять за допустимі межі – автоматично формується відповідна мітка (OVERHEAT/UNDERHEAT).

В якості референтних (середніх) значень та допусків для меж було взято наступні параметри: товщина покриття – 117,5 мкм; температура – 115 °C; час – 2700 с [27].

Для розв’язання задачі автоматичної оцінки використовується алгоритм машинного навчання Random Forest Classifier. Даний метод відноситься до класу ансамблевих моделей на основі дерев, є стійким до шуму та здатний моделювати нелінійні залежності. Random Forest являє собою ансамбль з великої кількості незалежних дерев, які приймають рішення. Кожне дерево навчається не на усій вибірці, а на випадково сформованому наборі рядків. Потім вони видають свої вердикти щодо даних у вигляді чисел (0, 1, 2). Після цього ухвалюється остаточне рішення, яке визначається більшістю голосів.

В контексті проєкту під час формування виборки даних до неї включаються реальні записи з бази даних, синтетичні випадки коректних параметрів, пересушування, недосушування, екстремальні точки. В ході навчання алгоритм будує 200 дерев, надає їм різні набори даних, вузли кожного дерева будуються на основі випадкових підмножин ознак. В результаті дерева вчаться розрізняти класи на основі реальних закономірностей. Коли під час експлуатації підсистема отримує

нові вхідні параметри технологічної карти, процедура класифікації відбувається наступним чином:

- кожен з 200 дерев розраховує свій прогноз;
- прогнози об'єднуються голосуванням;
- у результаті клас, який набрав більше за всіх голосів стає результатом перевірки. Якщо більшість дерев сигналізують про підвищення часу або температури – модель видасть OVERHEAT, якщо навпаки – UNDERHEAT. У випадку, коли параметри не перевищують допустимі межі – ОК.

Також у проєкті реалізовано модуль рекомендацій, який отримує прогноз Rando Forest, досліджує, які мінімальні зміни параметрів могли би привести до класу ОК, покроково змінює температуру або час та пропускає скориговане значення через модель. Якщо вона видає необхідний статус – система повертає рекомендацію користувачеві.

Після навчання модель перевіряється на тестовій виборці для оцінки точності (precision), повноти (recall), збалансованості (f1-score), загальної точності (accuracy), середнього значення по класах (macro avg), середнього значення (weighted avg) з урахуванням кількості прикладів кожного класу (support). Результати перевірки наведені на рисунку 4.1.

```

Завантажено реальних записів з БД: 500
Даних після додавання синтетики: 1100

Звіт по якості моделі:

```

	precision	recall	f1-score	support
0	0.99	0.99	0.99	106
1	0.99	0.99	0.99	97
2	0.94	0.94	0.94	17
accuracy			0.99	220
macro avg	0.97	0.97	0.97	220
weighted avg	0.99	0.99	0.99	220

```

Модель збережено у drying_quality_model.pkl

```

Рисунок 4.1 – Результат перевірки навченої моделі

Видно, що загальна точність моделі складає 99%, що свідчить про здатність системи коректно розрізняти технологічні режими. Класи ОК та OVERHEAT розпізнаються на високому рівні ($\text{precision} = 0,99$, $\text{recall} = 0,99$). Клас UNDERHEAT має більш низькі метрики ($\text{precision} = 0,94$), що можна пояснити меншою кількістю прикладів даного типу у навчальній вибірці ($\text{support} = 17$). Незважаючи на це, значення f1-score також є високим та говорить про те, що модель коректно виявляє випадки недостатнього рівня просушування. Результуючі агреговані метрики ($\text{macro avg} = 0,97$, $\text{weighted avg} = 0,99$) підтверджують здатність класифікатора до узагальнення закономірностей процесу.

Отримані результати свідчать про те, що створена модель Random Forest є ефективним інструментом для автоматизованого контролю параметрів сушіння електродів та забезпечує надійність діагностики технологічних порушень.

4.2.1 Інтеграція AI-модуля у основний застосунок

AI-модуль було реалізовано мовою програмування Python з використанням вебфреймворку Flask та алгоритму машинного навчання Random Forest, що забезпечує гнучку обробку виробничих даних і формування прогнозів щодо якості продукції. Взаємодія модуля з основним Java-додатком здійснюється через REST API, що дозволяє організувати надійний обмін даними, забезпечити модульність архітектури та спростити подальше масштабування й інтеграцію підсистеми. Структура взаємодії компонентів представлена на рисунку 4.2.

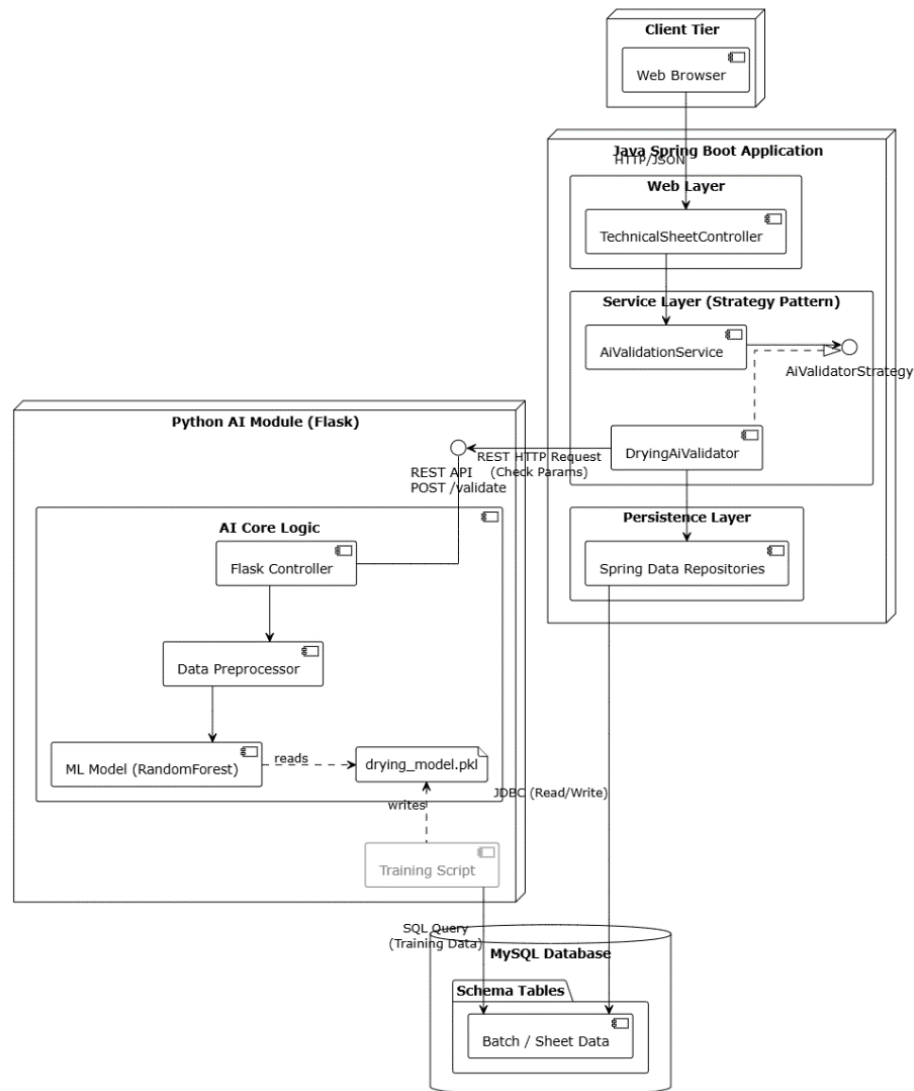


Рисунок 4.2 – Структурна схема взаємодії AI-модуля з основним додатком

Python-модуль виконує функції обчислювального ядра системи контролю якості. Він забезпечує попередню обробку вхідних параметрів, виклик навченої моделі для класифікації даних, генерацію прогнозу класу якості, повернення результату через REST-інтерфейс. Модуль співпрацює з моделлю, збереженою у файлі `drying_model.pkl`, що завантажується після запуску Flask-серверу. Інтеграція AI-модуля відбувається через HTTP-взаємодію. Алгоритм роботи наступний:

- Spring формує запит на ендпоїнт /validate у сервісному шарі додатку, використовуючи патерн Strategy для роботи з різними алгоритмами валідації (різними моделями);
- клас DryingAiValidator реалізує конкретну стратегію та відповідає за взаємодію з Python-модулем;
- сервіс відправляє POST-запит до Flask-API у форматі JSON, що містить значення параметрів сушіння для поточної партії;
- AI-модуль приймає запит, обробляє дані, викликає модель Random Forest та повертає прогноз;
- Spring отримує відповідь, формує текстове повідомлення з результатом перевірки і запропонованим значенням для стабілізації (у випадку, коли статус OVERHEAT або UNDERHEAT), і відправляє його на інтерфейс користувача.

Дані для аналізу та навчання моделі отримуються з бази даних MySQL лише на етапі тренування моделі, коли вони завантажуються через SQL-запити для формування навчальної вибірки. Детальний код реалізації наведено у Додатку Б.

4.2.2 Створення датасету у базі даних для навчання моделі

Для навчання моделі вимагався достатньо великий набір даних, що містив би взаємопов'язані параметри технологічного процесу: товщину покриття, температуру печі, фактичний час сушіння, а також підсумковий статус батареї (WORKING або DEFECTIVE_DRYING). Оскільки отримати фактичні виробничі дані було неможливо, було прийнято рішення про генерацію навчального датасету на основі реалістичного процесу сушіння.

Для досягнення даної цілі у базі даних MySQL було розроблено та використано спеціальну збережену процедуру seed_battery_data, яка автоматично формувала декілька десятків партій та сотні батарей з параметрами.

Збережена процедура приймає параметри: кількість партій, кількість батарей у кожній та стартовий номер партії. Всередині неї реалізовано послідовний алгоритм вставлення даних у взаємопов'язані таблиці `batch`, `battery`, `dosing_data`, `drying_data`.

Спочатку для кожної партії генерується номер, кількість батарей, параметри моделі акумулятора та технічної карти (зовнішній ключ). Процедура повертає ідентифікатор створеної партії, який потім використовується при створенні батарей.

Для кожного акумулятора партії формується товщина нанесеного покриття в діапазоні від 110 мкм до 125 мкм, вага, ємність, внутрішній опір та напруга, змодельовані випадковим чином [27]. Також створюється унікальний серійний номер. Приблизно 30% батарей маркуються як дефектні, логіка формується виходячи із фізичних закономірностей, описаних вище. Для батарей зі статусом `WORKING` параметри формуються наступним чином: температура трохи зменшується, а час сушіння зростає при збільшенні товщини. Також для варіативності додано незначні коливання даних.

При вставленні батареї у таблицю `battery` створюються записи у таблицях `dosing_data` (містить застосовану товщину покриття) та `drying_data` (містить температуру, час початку та завершення сушіння). При цьому записи прив'язуються до тієї ж батареї через спільний ID.

У рамках проєкту було створено 49 партій, 490 батарей та пов'язаних з ними параметрів. Код збереженої процедури наведено у Додатку Б.

4.3 Реалізація імітації надходження параметрів з обладнання

Враховуючи специфіку розробки підсистеми без безпосереднього доступу до фізичних виробничих ліній на етапі проєктування, було прийнято рішення про

розробку програмного модуля імітаційного моделювання. Цей модуль виконує роль цифрового двійника та забезпечує генерацію потоку даних, аналогічного тому, що надходить від реальних ПЛК.

Для забезпечення гнучкості системи та можливості подальшої інтеграції з реальним обладнанням без зміни бізнес-логіки, було використано принцип інверсії залежностей. Взаємодія з джерелами даних реалізована через шар абстракції (інтерфейси), що розділені за функціональними етапами виробництва:

- WeighingMixingPLC – для етапів підготовки сировини (зважування та змішування пасти/електроліту);
- CoatingPLC – для процесів нанесення пасти на колектори та сушіння;
- AssemblyPLC – для етапів складання (пресування та нарізання) та герметизації (вкладання елементів у корпус та заливки електроліту);
- TestingPLC – для проведення контрольних випробувань (циклу розряджання/заряджання, тестування на перегрів та коротке замикання).

Генерація значень враховує параметри, задані в технічних картах для конкретної партії продукції. Алгоритм генерації даних працює за наступним сценарієм:

- підсистема отримує номінальні значення та допустимі межі відхилень з поточної технічної карти;
- для імітації стабільного виробничого процесу генеруються значення, що знаходяться в межах допустимих відхилень;
- для перевірки надійності підсистеми управління якістю реалізовано механізм примусової генерації дефектів. Алгоритм з певною ймовірністю генерує значення, що виходять за межі допусків (наприклад, критичне підвищення температури під час тесту на перегрів).

4.4 Опис реалізації інтерфейсу підсистеми та тестування програмного засобу

Інтерфейс підсистеми було розроблено з використанням технологій Thymeleaf, HTML, CSS, JavaScript та Bootstrap 5. Це дозволило створити його адаптивним, інтуїтивно зрозумілим та орієнтованим на декілька ролей працівників.

Thymeleaf використовується як основний шаблонизатор. Він забезпечує генерацію HTML сторінок на стороні сервера і дозволяє безпосередньо працювати з моделями даних, що надходять з контролерів Spring. За допомогою виразів Thymeleaf реалізовано динамічне відображення даних про партії та окремі елементи, виведення параметрів технологічних операцій та валідаційні повідомлення і попередження.

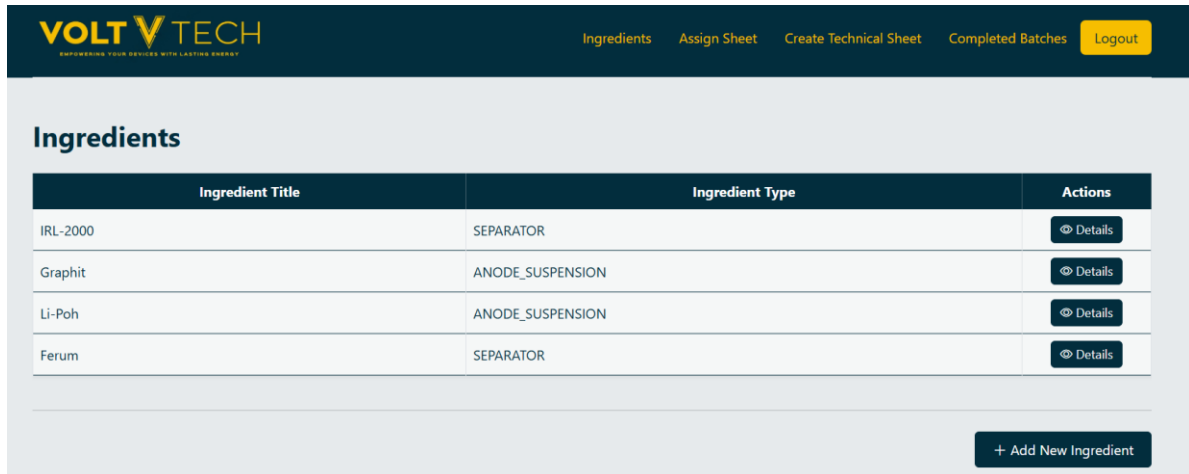
Основну структуру інтерфейсу оформлено з використанням класичного HTML5. Стилзація виконувалася як у вигляді власних CSS-файлів, і з використанням компонентів Bootstrap. Для окремих елементів були використані кастомні стилі, що дозволяють виділяти критичні параметри, створювати єдину кольорову гамму, покращувати навігацію між технологічними кроками.

JavaScript застосовувався для підвищення інтерактивності інтерфейсу. Були реалізовані: асинхронна взаємодія з сервером, імітація роботи технологічного обладнання в реальному часі, візуалізація аналітичних даних (використання бібліотеки Chart.js для відображення гістограми), управління динамічним контентом та формами.

Кожен з користувачів отримує доступ тільки до тих функцій, які відповідають його посаді: лаборант, працівник ремонтної бригади, а також диспетчери різних технологічних ділянок – станції зважування, замішування пасти, нанесення покриття на електроди, сушіння, пресування, нарізання, укладання елементів у корпус, заливання електроліту та тестування.

Лаборант працює з параметрами технологічних карт, створює нові, проводить перевірку коректності даних про інгредієнти та ініціює аналіз карти через AI-модуль. Його інтерфейс включає:

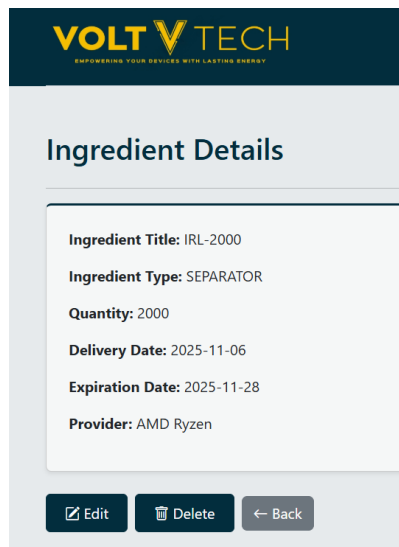
– форми для перегляду параметрів інгредієнтів / створення нового / редагування існуючого (рис. 4.3–4.6);



Ingredient Title	Ingredient Type	Actions
IRL-2000	SEPARATOR	Details
Graphit	ANODE_SUSPENSION	Details
Li-Poh	ANODE_SUSPENSION	Details
Ferum	SEPARATOR	Details

[+ Add New Ingredient](#)

Рисунок 4.3 – Форма для перегляду інгредієнтів



Ingredient Title: IRL-2000

Ingredient Type: SEPARATOR

Quantity: 2000

Delivery Date: 2025-11-06

Expiration Date: 2025-11-28

Provider: AMD Ryzen

[Edit](#) [Delete](#) [Back](#)

Рисунок 4.4 – Форма для детального перегляду характеристик

The screenshot shows a web form titled "Add New Ingredient" with a dark blue header. The form contains several input fields and a dropdown menu. At the bottom, there is a "Back" button and a "Save Ingredient" button.

Add New Ingredient

Title of ingredient:

Type of ingredient:
ANODE_SUSPENSION

Weight of Ingredient (g):

Quantity of Ingredient:

Delivery Date:

Expiration Date:

Provider:

← Back Save Ingredient

Рисунок 4.5 – Форма для додавання нового інгредієнта

The screenshot shows a web form titled "Edit Ingredient" with a dark blue header. The form contains several input fields and a dropdown menu, with some fields pre-filled with data. At the bottom, there is a "Back" button and a "Save Information" button.

Edit Ingredient

Title of ingredient:

Type of ingredient:
SEPARATOR

Quantity of Ingredient:

Delivery Date:

Expiration Date:

Provider:

← Back Save Information

Рисунок 4.6 – Форма для редагування параметрів існуючого інгредієнта

– форму для створення нової технічної карти (рис. 4.7). При введенні невалідних параметрів для сушіння і натисканні на кнопку «Validate with AI» модель AI сповістить про проблему та запропонує більш правильний параметр (рис. 4.8). При зміні даних на прийнятні і повторній перевірці проблема зникне і карту можна буде зберегти (без погодження модулем AI карту зберегти неможливо) (рис. 4.9);

New Technical Sheet

Sheet Title

Battery Model

1. Weighing & Mixing

Ingredients Composition

Ingredient	Type	Quantity (g/pcs)	Action
+ Add Ingredient			

Mixing Suspension Time (s)

Mixing Electrolyte Time (s)

2. Handling Electrodes

Substance Thickness (µm)

Cutting Accuracy (mm)

Drying Temperature (°C)

Drying Time (s)

Compression Ratio

3. Assembly

X-Ray Alignment Deviation (mm)

Empty Battery Weight (g)

Filled Battery Weight (g)

4. Testing

Stage 1: Initial Test

Voltage (V)

Capacity (mAh)

Resistance (mΩ)

Stage 2: Thermal & Cell Check

Case Temp (°C)

Cell Voltage (V)

Resistance (mΩ)

Рисунок 4.7 – Форма для створення нової технічної карти

⚠ Analysis Complete: Issues Found
Risk of defect: Cracking (Overheating) Drying time 3750s is critically long! Reduce it to 2700s.

Sheet Title: TS-34578 Battery Model: li-ion

1. Weighing & Mixing

Ingredients Composition

Ingredient	Type	Quantity (g/pcs)	Action
IRL-2000	SEPARATOR	300	
Li-Poh	ANODE_SUSPENSION	2700	

[+ Add Ingredient](#)

Mixing Suspension Time (s): Min 2000, Max 3000 Mixing Electrolyte Time (s): Min 2400, Max 2700

2. Handling Electrodes

Substance Thickness (µm): Min 115, Max 119 Cutting Accuracy (mm): Min 5, Max 7

Drying Temperature (°C): Min 115, Max 120 Drying Time (s): Min 3700, Max 3800

Рисунок 4.8 – Введення невалідних даних до карти (великий час сушіння)

✓ Analysis Complete: Valid
The technical data sheet is AI-verified. Parameters are in the optimal quality zone

Sheet Title: TS-34578 Battery Model: li-ion

1. Weighing & Mixing

Ingredients Composition

Ingredient	Type	Quantity (g/pcs)	Action
IRL-2000	SEPARATOR	300	
Li-Poh	ANODE_SUSPENSION	2700	

[+ Add Ingredient](#)

Mixing Suspension Time (s): Min 2000, Max 3000 Mixing Electrolyte Time (s): Min 2400, Max 2700

2. Handling Electrodes

Substance Thickness (µm): Min 115, Max 119 Cutting Accuracy (mm): Min 5, Max 7

Drying Temperature (°C): Min 115, Max 120 Drying Time (s): Min 2600, Max 2800

Рисунок 4.9 – Результат після введення коректних даних

– форми для призначення карти партії (рис. 4.10–4.11);

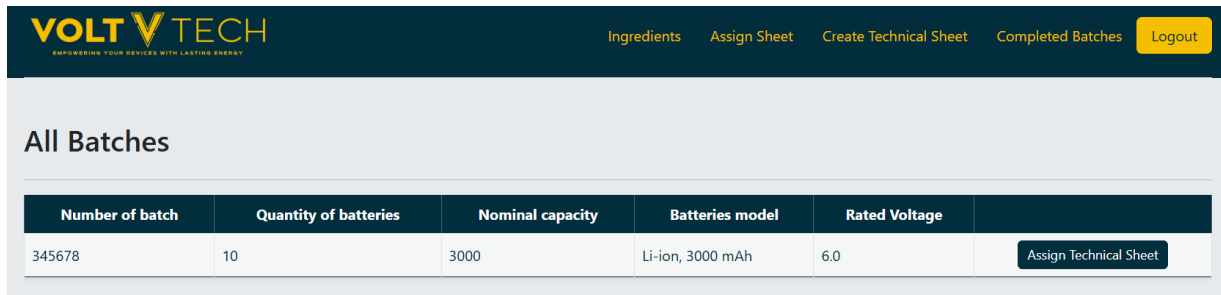


Рисунок 4.10 – Форма для призначення партії технічної карти

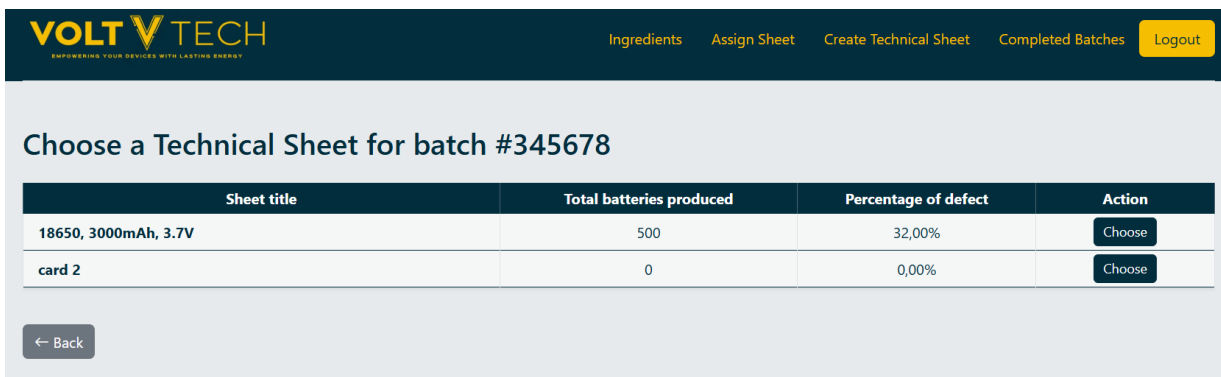


Рисунок 4.11 – Форма для вибору технічної карти з вказівкою на відсоток браку за нею

– форми для перегляду готових партій та їх детальної статистики (рис. 4.12–4.14);

Batch #	Date of Production	Total Produced	Valid %	Defect %	Actions
150	02.11.2025 21:16	10 pcs	70,00%	30,00%	Details
149	01.11.2025 21:16	10 pcs	70,00%	30,00%	Details
148	31.10.2025 21:16	10 pcs	50,00%	50,00%	Details
147	30.10.2025 21:16	10 pcs	60,00%	40,00%	Details

Рисунок 4.12 – Форма для перегляду готових батарей

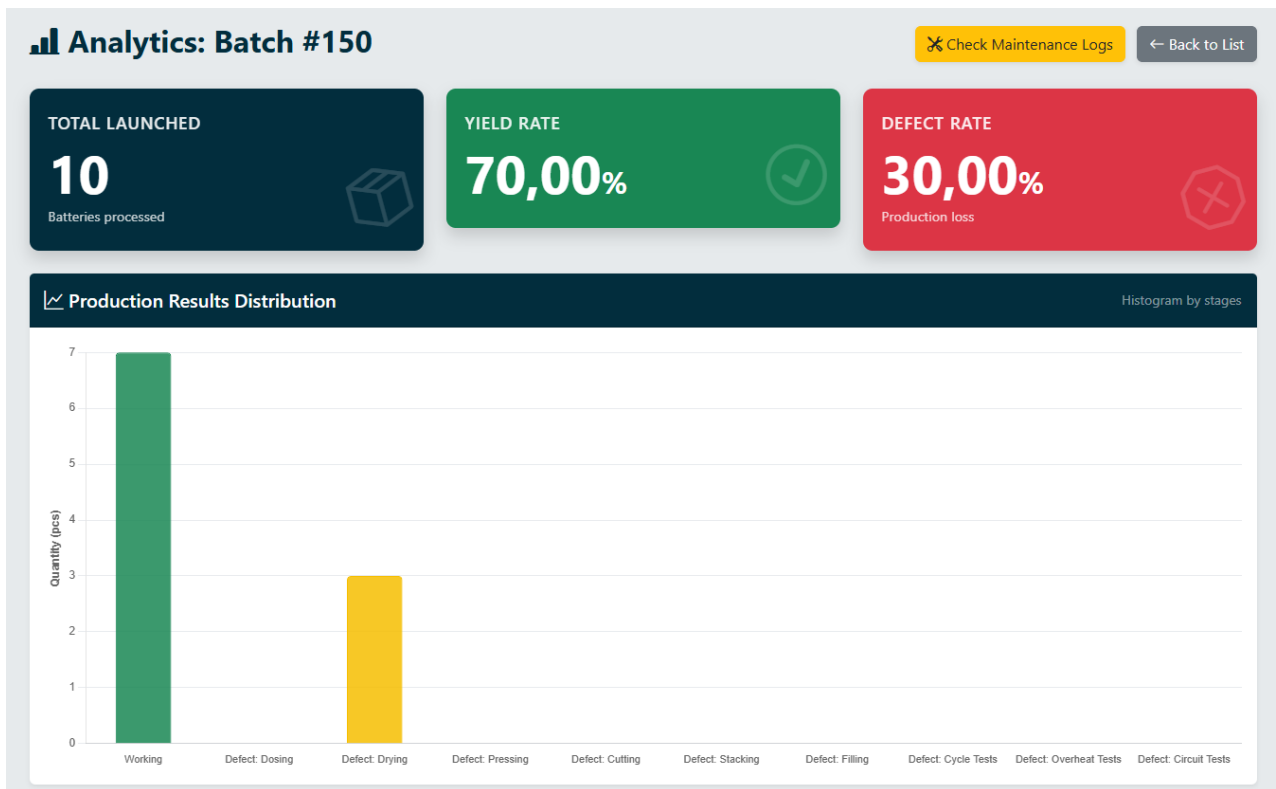


Рисунок 4.13 – Детальна статистика обраної батареї

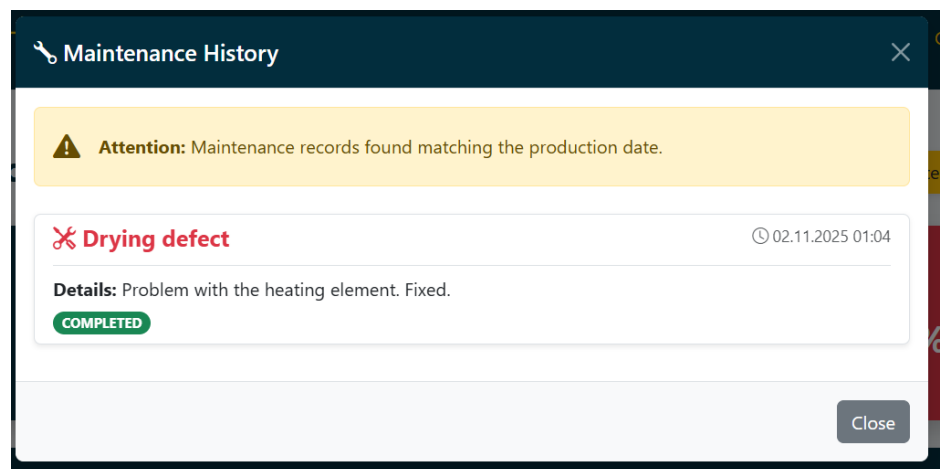
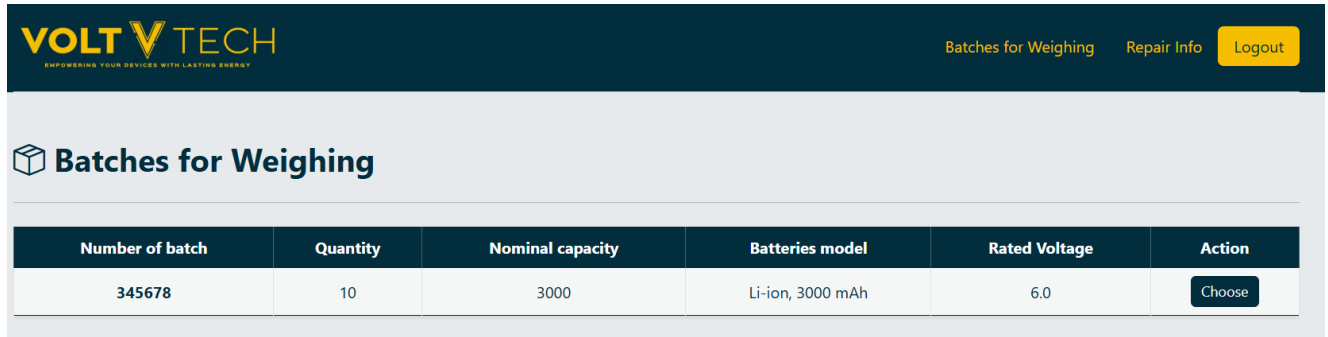


Рисунок 4.14 – Інформація про проведений у цей день ремонт

Кожен з диспетчерів працює на своїй станції, однак їх інтерфейси мають схожу структуру:

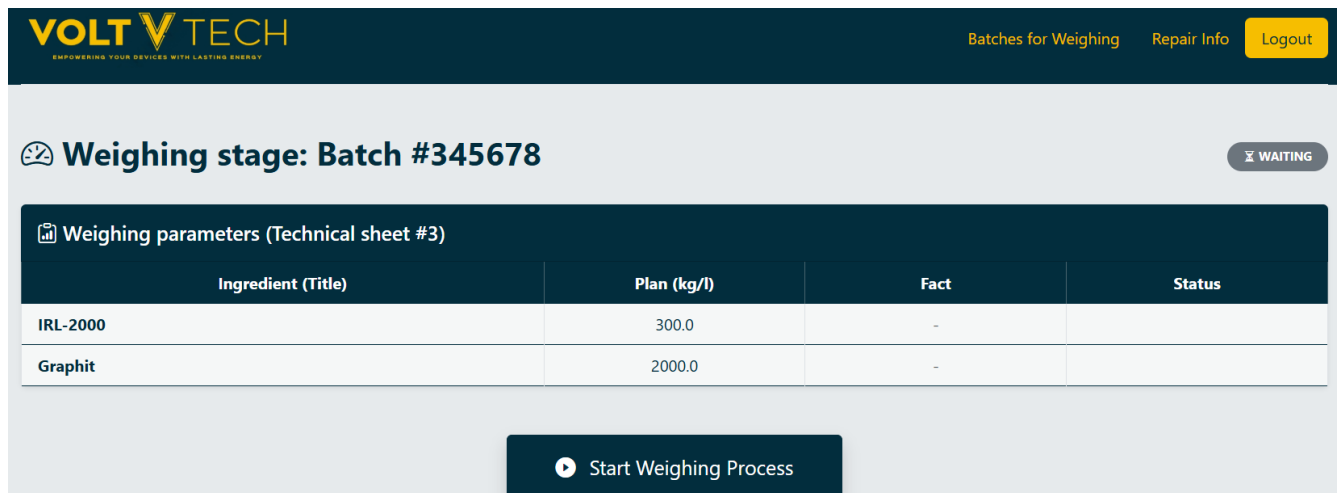
– форма для перегляду партій, які прийшли на станцію для обробки (рис. 4.15);



Number of batch	Quantity	Nominal capacity	Batteries model	Rated Voltage	Action
345678	10	3000	Li-ion, 3000 mAh	6.0	Choose

Рисунок 4.15 – Форма для перегляду партій, які прийшли на станцію зважування

– форма для запуску процесу для обраної партії (рис. 4.16). У разі появи невалідного значення процес зупиняється і виводиться повідомлення з варіантами дій (рис. 4.17). Якщо проблема вирішена – процес запускається знову. У разі валідних значень виводиться повідомлення про успішність виконання процесу (рис. 4.18);



Ingredient (Title)	Plan (kg/l)	Fact	Status
IRL-2000	300.0	-	
Graphit	2000.0	-	

Start Weighing Process

Рисунок 4.16 – Форма для запуску зважування інгредієнтів

VOLT V TECH EMPOWERING YOUR DEVICES WITH LASTING ENERGY

Batches for Weighing Repair Info Logout

Weighing stage: Batch #345678 ERROR

Weighing parameters (Technical sheet #3)

Ingredient (Title)	Plan (kg/l)	Fact	Status
IRL-2000	300.0	150,000	Reject
Graphit	2000.0	-	

⚠ Attention! Process Stopped!
A critical deviation of the ingredient weight from the technical sheet was detected.

[Make a Report](#) [Restart System](#)

Рисунок 4.17 – Результат появи невалідного значення

VOLT V TECH EMPOWERING YOUR DEVICES WITH LASTING ENERGY

Batches for Weighing Repair Info Logout

Weighing stage: Batch #345678 CHECKED

Weighing parameters (Technical sheet #3)

Ingredient (Title)	Plan (kg/l)	Fact	Status
IRL-2000	300.0	300,175	OK
Graphit	2000.0	1992,381	OK

✓ Weighing Successful!
All indicators are within normal range. The batch is ready for the next stage.

[Send to Next Station →](#)

Рисунок 4.18 – Результат успішного завершення зважування для партії

Форми для створення рапорту на ремонт, перегляду його статусу та інтерфейсу бригадира ремонтної бригади не відрізняються від форм, поданих для компоненту MES-системи «Управління виробничим процесом» [28].

4.5 Охорона праці

Впровадження та експлуатація підсистеми MES безпосередньо пов'язані з використанням широкого спектру електротехнічних приладів: від серверного обладнання та автоматизованих робочих місць операторів до промислових контролерів, інтегрованих у виробничі лінії. Основною загрозою є вірогідність ураження електричним струмом, яке може статися у результаті дотику до струмопровідних частин, які знаходяться під напругою, або до металічних корпусів обладнання, які опинилися під напругою через пошкодження ізоляції. Ігнорування правил безпеки має критичні наслідки для здоров'я людини. Проходження струму через організм викликає термічні опіки, судомні скорочення м'язів, фібриляцію шлуночків серця, і, у окремих випадках, летальні випадки.

Крім того, порушення у роботі електромережі, такі як короткі замикання або перевантаження, є одними з розповсюджених причин виникнення пожеж, що умовах хімічно активного виготовлення акумуляторів може призвести до катастрофічних наслідків, матеріальних збитків та зупинки технологічного процесу на невизначений період.

Безпечність використання обладнання в Україні регламентована державною нормативно-правовою базою. Основним документом, в якому описано порядок безпечної експлуатації електроустаткування, є НПАОП 40.1-1.21-98 [29]. Відповідно до нормативного акту, всі металічні частини електрообладнання та корпуси комп'ютерної техніки, які можуть опинитися під напругою при пошкодженні ізоляції, повинні бути заземлені. Документ також наголошує на використанні приладів запобіжного відключення, регулярному контролю опору ізоляції та наявності у персоналу відповідної групи допуску за електробезпекою (не нижче III групи для оперативно-ремонтного персоналу).

Одним з ключових технічних заходів, що забезпечує захист персоналу від ураження електричним струмом при непрямому дотику, є пристрій захисного заземлення. Ефективність заземлення напряму залежить від його опору, який має бути достатньо низьким для забезпечення перетікання струму в землю та спрацювання захисту. Для підтвердження відповідності нормам проводиться розрахунок опору розтіканню струму одиночного вертикального заземлювача, який часто використовується як елемент контуру заземлення для обчислювальних центрів та цехового обладнання. Розрахунок відбувається за формулою:

$$R = \frac{\rho}{2\pi l} \left(\ln \frac{2l}{d} + \frac{1}{2} \ln \frac{4t+l}{4t-l} \right), \quad (2.1)$$

де R – опір розтіканню струму, Ом;

ρ – питомий опір ґрунту, Ом·м;

l – довжина вертикального електрода, м;

d – діаметр електрода, м;

t – відстань від поверхні землі до середини вертикального електрода, м [30].

Величина t визначається як сума глибини траншеї та половини довжини стрижня. Використання даної залежності надає можливість спроектувати контур заземлення з урахуванням реальних умов монтажу, коли верхній кінець електрода знаходиться нижче рівня землі для захисту від механічних пошкоджень та кліматичних впливів

ВИСНОВКИ

В ході виконання кваліфікаційної роботи проведено аналіз предметної області, визначено основні вимоги до розроблюваної підсистеми MES на виробництві. Були виконані наступні завдання:

- проведено огляд та аналіз існуючих методів, засобів та автоматизованих систем управління якістю продукції;
- розроблено структурну схему та алгоритм роботи підсистеми управління якістю продукції;
- проведено вибір компонентів та створено алгоритм роботи підсистеми управління якістю продукції;
- спроектовано базу даних підсистеми автоматизації для управління якістю виготовлення телефонних акумуляторів;
- реалізовано підсистему автоматизації для управління якістю виготовлення телефонних акумуляторів у вигляді програмного засобу.

Запропонована підсистема автоматизації для управління якістю виготовлення телефонних акумуляторів, яку реалізовано у вигляді програмного засобу, слугує основою для подальшої інтеграції у виробничу інфраструктуру підприємства. Впровадження цього рішення сприятиме мінімізації відсотка браку та дозволить підвищити загальну ефективність та конкурентноспроможність виробництва телефонних акумуляторів на ринку.

Перспективами подальшого розвитку підсистеми є розширення функціональних можливостей за рахунок інтеграції з реальним виробничим обладнанням, удосконалення AI-модуля шляхом використання більших обсягів інформації, та створення додаткових модулів для перевірки отриманих даних для інших етапів виготовлення телефонних акумуляторів.

У результаті проведеного дослідження підготовлено та опубліковано наукову статтю у фаховому журналі, зміст якої узгоджується з Ціллю 9 ООН – «Індустріалізація, інновації та інфраструктура».

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Положення про академічну доброчесність [Електронний ресурс]: Наказ ХНУРЕ від 02 лютого 2021 р. №50. – Режим доступу: https://nure.ua/wp-content/uploads/Main_Docs_NURE/polozhennja-pro-akademichnu-dobrochesnist.pdf (дата звернення: 30.10.2025).

2. Стандарт вищої освіти за спеціальністю 151 «Автоматизація та комп'ютерно-інтегровані технології» галузі знань 15 «Автоматизація та приладобудування» для другого (магістерського) рівня вищої освіти, затверджений наказом МОН України № 1022 від 10.08.2020р. «Про затвердження стандарту вищої освіти за спеціальністю 151 Автоматизація та комп'ютерно-інтегровані технології для другого (магістерського) рівня вищої освіти» Режим доступу: <https://mon.gov.ua/static-objects/mon/sites/1/vishcha-osvita/zatverdzeni%20standarty/2020/08/10/151-avtomatizatsiya-ta-kit-magistr.pdf> (дата звернення: 30.10.2025).

3. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка, освітньо-професійних програм: «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2024. 57 с.

4. Основи наукових досліджень : підручник / І. Ш. Невлюдов, Ю. М. Олександров, А. О. Андрусевич, О. О. Чала ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Prague : OKTAN PRINT, 2024. – 468 с. DOI

<https://doi.org/10.46489/ONDNP> Режим доступу на ресурсі бібліотеки ХНУРЕ
<https://openarchive.nure.ua/handle/document/28574> (дата звернення: 30.10.2025).

5. Невлюдов І. Ш. Техніко-економічне обґрунтування інженерних рішень в інтелектуальному виробництві : підручник / І. Ш. Невлюдов. - Кривий Ріг : Чернявський Д. О., 2024. – 388 с. : іл. Режим доступу на ресурсі бібліотеки ХНУРЕ
<https://openarchive.nure.ua/handle/document/27408>

6. Невлюдов І.Ш. Автоматизована система керування технологічними процесами в SCADA системі TRACE MODE 6: Навчальний посібник / І.Ш.Невлюдов, А.О. Андрусевич, В.В. Євсєєв, С.С. Максимова, М.Г. Стародубцев, В.В.Невлюдова. – Кривий Ріг: Криворізький коледж НАУ, 2018. – 320 с.

7. ДСТУ 3008: 2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання / Нац. стандарт України. – Вид. офіц. – [Чинний від 2017 – 07 – 01]. – Київ: ДП «УкрНДНЦ», 2016. – 26 с.

8. Вінниченко С.О. Система автоматизації для забезпечення керування якістю продукції на всіх етапах виробництва // Автоматизація та приладобудування («Automation and Development of Electronic Devices» ADED-2024) : збірник студентських наукових статей / Харківський національний університет радіоелектроніки ; [редкол.: І.Ш. Невлюдов та ін.]. – Харків : ХНУРЕ, 2024. Вип. 2. С. 38–42

9. Intelligent manufacturing execution systems: A systematic review / A. Shojaeinasab та ін. Journal of Manufacturing Systems. 2022. Т. 62. С. 503–522. URL: <https://doi.org/10.1016/j.jmsy.2022.01.004> (дата звернення: 31.10.2025).

10. Antić S. The impact of MES digital technology on the digitalization of the quality management System (QMS): A case study. Tehnika. 2024. Т. 79, № 4. С. 497–504. URL: <https://doi.org/10.5937/tehnika2404497a> (дата звернення: 31.10.2025).

11. Heri Nurdiyanto. Critical Role of Manufacturing Execution Systems in Digital Transformation of Manufacturing Industry. Journal of Electrical Systems. 2024. Т. 20,

№ 7s. С. 2432–2436. URL: <https://doi.org/10.52783/jes.4038> (дата звернення: 31.10.2025).

12. Koval O. Хто такі SAP-спеціалісти, скільки вони заробляють і чим займаються в ІТ-компаніях і війську. DOU.ua. URL: <https://dou.ua/lenta/articles/sap-developer-career/> (дата звернення: 30.10.2025).

13. Quality Management (QM). SAP Help Portal | SAP Online Help. URL: https://help.sap.com/docs/SAP_ERP (дата звернення: 30.10.2025).

14. SAP Quality Management SAP QM. Sapinfocon. URL: <https://sapinfocon.blogspot.com/> (дата звернення: 30.10.2025).

15. SIMATIC IT | Siemens Software. Siemens Digital Industries Software. URL: <https://www.sw.siemens.com/technology/simatic-it/> (дата звернення: 30.10.2025).

16. Siemens MES Customer Reviews 2025 | Manufacturing Execution. SoftwareReviews. URL: <https://www.softwarereviews.com/products/siemens-mes> (дата звернення: 30.10.2025).

17. Rockwell Automation | UA. Rockwell Automation. URL: <https://www.rockwellautomation.com/en-ua/company/about-us/ukraine.html> (дата звернення: 30.10.2025).

18. Quality Management System (QMS) | Rockwell Automation. Plex. URL: <https://plex.rockwellautomation.com/en-us/products/quality-management-system.html> (дата звернення: 30.10.2025).

19. Plex Quality Management System | FactoryTalk | US. Rockwell Automation. URL: <https://www.rockwellautomation.com/en-us/products/software/factorytalk/operations/mes/plex-quality-management-system.html> (дата звернення: 30.10.2025).

20. What is DPMO: Defects Per Million Opportunities?. GembaAcademy. URL: <https://www.gembaacademy.com/gemba-insights/what-is-dpmo-defects-per-million-opportunities> (дата звернення: 30.10.2025).

21. What is Cp / Cpk?. GembaAcademy. URL: <https://www.gembaacademy.com/gemba-insights/what-is-cp-cpk> (дата звернення: 30.10.2025).
22. Ergashev S. B. o. Design of automated enterprise information systems using uml diagrams in the creation of applications. Design of automated enterprise information systems using uml diagrams in the creation of applications. 2023. Т. 3(12), 25–31. С. URL: <https://doi.org/10.17605/OSF.IO/ZEGBA> (дата звернення: 31.10.2025).
23. Хрустальова С. Розроблення структурної схеми модуля автоматизації на базі RFID – технологій / С. Хрустальова, С. Вишванюк // Виробництво & Мехатронні Системи 2023 : тези доповідей VII-ої Міжнар. конф., 19-20 жовтня 2023 р. – Харків, 2023. – С. 22–25.
24. Вінниченко С.О. Методика створення реляційної бази даних на основі програмного середовища MySQL / Вінниченко С.О., Яшков І.О., Олінкевич Я.В. // Modern research in world science. Proceedings of the 4th International scientific and practical conference. SPC «Sci-conf.com.ua», 10-12 липня 2022 р. – Львів, 2022. – С. 315–323.
25. Java Documentation - Get Started. Oracle Help Center. URL: <https://docs.oracle.com/en/java/> (дата звернення: 13.12.2025).
26. Свістільнік Р. Ф., Патлун Д. В. Дослідження впливу температури на електропровідні властивості полімерних композитних покриттів. Technologies and engineering. 2024. № 6. С. 101–109. URL: <https://doi.org/10.30857/2786-5371.2023.6.8> (дата звернення: 12.12.2025).
27. Розробка технології електропровідних гібридних композиційних покриттів / Р. Ф. Свістільнік та ін. Technologies and engineering. 2023. № 4. С. 60–70. URL: <https://doi.org/10.30857/2786-5371.2022.4.5> (дата звернення: 12.12.2025).
28. Вінниченко С. О. Розробка компоненту MES-системи для оптимізації виробничих процесів підприємства : пояснювальна записка до атестаційної роботи здобувача вищої освіти на першому (бакалаврському) рівні, спеціальність 151 –

Автоматизація та комп'ютерно-інтегровані технології / С. О. Вінниченко ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2024. – 114 с.

29. НПАОП 40.1-1.21-98. Правила безпечної експлуатації електроустановок споживачів (2029). ДНАОП - Нормативно-правова бібліотека інструкції документи. URL: <https://dnaop.com> (дата звернення: 13.12.2025).

30. Комплекс навчально-методичного забезпечення навчальної дисципліни "Організація керування умовами праці" підготовки першого (бакалаврського) рівня усіх спеціальностей та усіх напрямів університету [Електронний ресурс] / ХНУРЕ; розроб.: Т. Є. Стиценко, Г. В. Пронюк, О. В. Мамонтов. – Харків, 2024. – 164 с.