

ДОДАТОК А

Апробація наукових результатів дослідження

Міністерство освіти і науки України



NURE

Харківський національний університет
радіоелектроніки

ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

ADED-2024

(Випуск 2)

[електронне видання]



<http://nure.ua/department/kafedra-komp-yuterno-integrovanih-tehnologiy-avtomatizatsiyi-ta-mehatroniki-kitam>



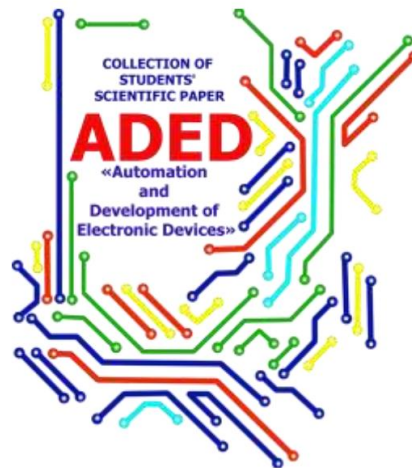
<http://itez.zntu.edu.ua/>



<http://kafea.kdu.edu.ua>

Харків 2024

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки
(КІТАР)



ЗБІРНИК
студентських наукових статей
«Автоматизація та приладобудування»
«Automation and Development of Electronic Devices»
ADED-2024
(Випуск 2)
[електронне видання]

Харків 2024

Автоматизація та Приладобудування («Automation and Development of Electronic Devices» ADED-2024) [Електронний ресурс] : збірник студентських наукових статей / Харківський національний університет радіоелектроніки ; [редкол.: І.Ш. Невлюдов та ін.]. – Харків : ХНУРЕ, 2024. – Вип. 2. – 290с.

Collection of Students' Scientific Paper «Automation and Development Of Electronic Devices» ADED-2024 Part 2 (Key infrastructure 2024) - Kharkiv/ The Editorial.: Nevlyudov I.Sh. (head), that all. Kharkiv: Kind of Kharkiv National University of Radio Electronics [electronic edition], 2024. – 290p with.

Рекомендовано рішенням
Науково-технічної ради
Харківського національного
університету радіоелектроніки
протокол №6 від 29.11.2018

Рекомендовано рішенням Вченої ради
факультету Автоматики і комп'ютеризованих технологій
Харківського національного
університету радіоелектроніки
протокол № 4 від 26.12.2024

Збірник містить наукові статті здобувачів першого (бакалаврського), другого (магістерського) рівнів вищої освіти кафедри комп'ютерно-інтегрованих технологій, автоматизації та робототехніки (КІТАР) Харківського національного університету радіоелектроніки, кафедри Інформаційних технологій електронних засобів (ІТЕД) Запорізького національного технічного університету та кафедри Електронних апаратів (ЕА) Кременчуцького національного університету ім. М. Остроградського які навчаються за спеціальностями: 151 Автоматизація та комп'ютерно-інтегровані технології, 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка, 172 Телекомунікації та радіотехніка, 171 Електроніка та 163 Біомедична інженерія. Статті надані в авторській редакції.

©ХНУРЕ, 2024 рік

ЗМІСТ

<i>Гребенков Д.В.</i> Дослідження використання повітряних безпілотних систем та їх класифікація	8
<i>Івашенко К.В.</i> Розробка багатоканальної системи подачі філаменту для багатокольорового 3D друку	15
<i>Кальченко А.С.</i> Розробка полярного 3D принтеру з можливістю друку без технологічних підтримок ...	20
<i>Піхтерьов А.Д.</i> Корекція системи координат полярного 3D принтеру для підвищення якісних показників друку	29
<i>Вінниченко С.О.</i> Система автоматизації для забезпечення керування якістю продукції на всіх етапах виробництва	38
<i>Івашенко К.В.</i> Системи мультиматеріального 3D-друку	43
<i>Лашин З.В.</i> Аналіз методів та принципів використання автоматизованих керованих транспортних засобів у виробничому процесі	53
<i>Єчевський А. Д.</i> Розумний світлофор: технологія майбутнього для сучасних міст	64
<i>Маруніч Р.В.</i> Особливості застосування IoT у сфері безпеки	71
<i>Твердохліб А.О.</i> Роль штучного інтелекту в оптимізації інформаційно-пошукових систем	77
<i>Shchokolov I.S.</i> The role of automation and cals systems in changing human factor in production	82
<i>Поліканов К.А.</i> Ключові функції та можливості інтелектуальних систем для модульного житла	87
<i>Сухомлінова Д.А.</i> Огляд концепцій дистанційного керування та моніторингу дронів	92
<i>Артюх В.С., Кащеев В.А.</i> Аналіз та моделювання Shuttle-систем	97
<i>Обривко Є.В.</i> Аналіз методів і функцій захисту даних для ресурсів дистанційного навчання	107
<i>Сверчков М.О.</i> Системи автоматизації для модульних роботизованих систем виробничного призначення	113
<i>Панков А.А.</i> Дослідження методів розробки програмного модуля автоматизованого управління замкненою виробничою ділянкою	118
<i>Петров Е.С.</i> Аналіз методів підвищення ефективності складального виробництва за принципами Lean Production	126
<i>Сагула О.О.</i> Аналіз програмного нейромережевого модуля для виявлення дронів на основі Yolov5..	130

УДК 338.4

СИСТЕМА АВТОМАТИЗАЦІЇ ДЛЯ ЗАБЕЗПЕЧЕННЯ КЕРУВАННЯ ЯКІСТЮ ПРОДУКЦІЇ НА ВСІХ ЕТАПАХ ВИРОБНИЦТВА

С.О. Вінниченко

Харківський національний університет радіоелектроніки

Україна, 61166, Харків, пр. Науки 14

E-mail: sofia.vinnychenko@nure.ua

Анотація: У даній статті розглядається роль MES в управлінні якістю продукції на підприємстві. Описано функціональні можливості системи, які забезпечують зниження кількості браку, часу простоїв та витрат на перероблення. Особливу увагу приділено основним процесам виготовлення продукції, таким як контроль сировини, процесу виготовлення, проведення проміжних інспекцій, фінальна перевірка готових виробів. Розглянуто подальші перспективи використання MES для оптимізації виробництва та зниження витрат.

Ключові слова: MES, якість, виробництво, продукція, дефекти.

AUTOMATION SYSTEM TO ENSURE PRODUCT QUALITY MANAGEMENT AT ALL STAGES OF PRODUCTION

S. Vinnychenko

Kharkiv Kharkiv National University of Radio Electronics

Ukraine, 61166, Kharkiv, Nauky av, 14

E-mail: sofia.vinnychenko@nure.ua

Annotation: This article examines the role of the MES in managing product quality within an enterprise. It describes the functional capabilities of the system that provide for reducing the amount of defective products, reducing downtime, and reducing rework costs. Particular attention is paid to the key manufacturing processes, such as raw material control, manufacturing process control, in-process inspections, and final inspection of finished products. The article also considers the further prospects for using MES to optimize production and reduce costs.

Key words: MES, quality, production, products, defects.

Сучасні підприємства прагнуть забезпечити високу якість своєї продукції, щоб відповідати змінним вимогам ринку, мінімізувати витрати та підвищувати конкурентноздатність. Одним з ефективних інструментів, які сприяють досягання вищевказаних цілей, є система управління виробництвом MES (Manufacturing Execution System). MES-система забезпечує автоматизацію та управління всіма виробничими процесами на рівні цеху, з'єднуючи в єдиний цифровий ланцюг всі етапи – від отримання сировини до випуску готової продукції. Впровадження системи на підприємстві дозволяє не лише оптимізувати виробничі процеси, а й здійснювати постійний контроль якості продукції, що є важливим для складних та багатокомпонентних підприємств.

Даний матеріал присвячено дослідженню ролі MES-системи в забезпеченні та управлінні якістю продукції на виробничому підприємстві, охоплюючи всі етапи її виготовлення. Розглядається вплив системи на підвищення таких показників ефективності, як зниження кількості браку, зниження часу простоїв та витрат на перероблення, а також підвищення рівня відповідності стандартам та рівня задоволеності клієнтів отриманою продукцією.

MES-система керує якістю продукції на підприємстві через деякі ключові функції та процеси, забезпечуючи відповідність виробів нормативним вимогам:

– вхідний контроль сировини та комплектуючих;

- контроль якості у процесі виготовлення;
- управління проміжними інспекціями;
- перевірка готових виробів [1].

Розглянемо кожен процес більш детально.

1) Вхідний контроль сировини та комплектуючих

Даний процес є першим кроком у ланцюгу забезпечення якості продукції, націлений на перевірку відповідності вхідних матеріалів встановленим стандартам та вимогам. В MES-системі даний процес синхронізується з іншими етапами виробництва, що дає можливість оперативно виявляти та усувати сировину з дефектами.

При надходженні на склад матеріалів та комплектуючих MES реєструє їх, фіксує дані постачальника, дату поставки, номер партії, найменування та кількість одиниць. В систему також заноситься супутна документація, така як сертифікати якості, паспорти та специфікації. Грунтуючись на отриманих даних, система перевіряє відповідність надходжень встановленим по їхнім документам вимогам. На даному етапі співробітники можуть порівнювати фізичні дані матеріалів з записами у системі, щоб додатково впевнитися у їх відповідності специфікаціям. Якщо компанія-виробник продукції використовує галузеві або внутрішні стандарти, то MES-система може посылатися на них для додаткової перевірки параметрів сировини. При виявленні відхилень система фіксує і класифікує їх: серйозні дефекти, які виключають використання даного матеріалу, або менш критичні, які можливо усунути. MES може запропонувати подальші дії в залежності від класифікації дефектів – повернення постачальнику, доопрацювання на місці або дозвіл на використання в залежності від встановлених критеріїв відбору сировини. За результатами перевірок матеріали отримують статус у системі: «схвалено до використання», «умовно схвалено», «відхилено». Сировина, яка не пройшла перевірку, блокується для подальшого використання. Затверджені матеріали переміщуються у зону готовності до виробництва, а відхилені – в зону доопрацювання або повернення. Повідомлення про статус партії можуть бути відправлені системою на робочі термінали персоналу, електронні пошти або через інтегровану систему оповіщень компанії. MES також фіксує всі дії з матеріалами та зберігає історію перевірок у базі даних.

2) Контроль якості у процесі виготовлення

Даний процес дає можливість виявляти дефекти та запобігати їм на кожному етапі виготовлення, таким чином забезпечуючи випуск продукції високої якості. MES відіграє важливу роль у автоматизації даного процесу, відслідковуючи критичні параметри та вчасно повідомляючи про всі відхилення.

На етапі підготовки до виготовлення система завантажує параметри та стандарти для кожного виду продукції. Ці параметри включають такі показники, як температура, тиск, час обробки, швидкість обладнання та інше. Також можливе завантаження робочих інструкцій та протоколів, яким повинні слідувати оператори. Після запуску процесу система отримує дані з виробничого обладнання та датчиків в режимі реального часу. Вся інформація автоматично збирається та відображається на моніторах працівників. Якщо значення параметрів відхиляються від встановлених норм, система одразу сповіщує персонал, а в критичних випадках зупиняє обладнання, запобігаючи виготовленню продукції з дефектами. Для особливо важливих процесів виготовлення (наприклад, у фармацевтичній або харчовій промисловості) контроль може бути доповнений жорсткими нормативними вимогами, такими як GMP (Good Manufacturing Practice), HACCP (Hazard Analysis and Critical Control Points), які також враховуються у MES [2]. Якщо в процесі виготовлення продукції було виявлено дефекти, то MES автоматично зафіксує дані про них, вносячи інформацію про їх природу, час виникнення та етап виробництва, на якому це відбулося. Також система здатна організовувати коригувальні та попереджувальні дії (CAPA – Corrective Action / Preventive

Action) при виявленні проблем [3]. Коли реєструється дефект, вона здатна запропонувати процедури для усунення причини, як, наприклад, зміна налаштувань обладнання, додаткова перевірка та інше. Система повідомляє відповідальних осіб про можливі заходи для вирішення проблеми та вимагає підтвердження для їх виконання, якщо це доцільно.

3) Управління проміжними інспекціями

Даний процес дає можливість контролювати якість продукції на ключових етапах виробничого процесу, що допомагає попередити передавання дефектів на наступні стадії розробки та мінімізувати ризик випуску бракованої продукції.

На етапі планування виробництва визначаються критично важливі точки контролю якості, в яких необхідно буде проводити проміжні інспекції. Ці точки встановлюються в залежності від характеристик продукції та етапів, де найбільш вірогідна поява відхилень. Для кожної інспекційної точки MES задає конкретні критерії перевірки (наприклад, точність розмірів, зовнішній вигляд) які відповідають стандартам якості та вимогам технічних умов. Також система автоматично планує проміжні інспекції для кожної партії продукції. Вони призначаються з визначеною заздалегідь періодичністю або після досягнення конкретного етапу (наприклад, після обробки на станку, збирання деталей або фарбування). Якщо необхідна перевірка з боку людини, то MES направляє повідомлення конкретним бригадам або спеціалістам. Також можливе блокування подальшого руху продукції по виробничій лінії з боку системи, поки інспекція не буде завершена. Результати перевірок з боку персоналу заносяться у систему і якщо в її процесі було виявлено відхилення, то вони фіксуються та класифікуються за типом, рівнем критичності та можливим причинам виникнення. Також реєструються всі дані про проблему: час виявлення, етап виготовлення та вид відхилення. При виявленні дефектів система може пропонувати коригуючі дії CAPA (описані в п. 2).

Після проведення інспекції та виконання всіх перевірок MES реєструє статус партії. Якщо продукція відповідає вимогам, то її просування на наступний етап виробництва буде дозволено. У випадку виявлення критичних дефектів, які неможливо усунути на місці, система блокує подальшу обробку даної партії та ініціює додаткові інспекції, повернення продукції на попередні етапи для доопрацювання або утилізацію партії.

4) Перевірка готових виробів

Даний процес є завершальним етапом контролю якості, який забезпечує відповідність продукції встановленим стандартам та вимогам перед відправкою замовнику або на склад. Етап є критично важливим для мінімізації ризиків повернення продукції та збереження репутації компанії.

Для кожної партії продукції формується план перевірок в залежності від її типу, вимог замовника та галузевих стандартів. План включає в себе перелік тестів, параметрів, методів перевірки та критеріїв відповідності. Перевірка може бути вибірковою (для частини виробів з партії) або повною (для кожного виробу) в залежності від рівня критичності продукції та специфіки замовлення. Загальний план перевірки якості включає в себе:

- візуальний огляд: перевіряється зовнішній вигляд виробу, відповідність фарбування, маркування стандартам;
- вимірювання параметрів: використовуються прилади для перевірки точності розмірів, маси, геометрії та інших фізичних характеристик;
- функціональні тести: перевіряються ключові робочі характеристики виробу (наприклад, електричні параметри, герметичність). Для автоматизованих процесів використовуються спеціальні стенди та обладнання.
- випробування надійності: в деяких випадках проводиться тестування на зносостійкість, ударні навантаження, температуру та вологість, з метою гарантування довговічності та якості роботи виробу.

Результати усіх тестів та перевірок заносяться у відповідні розділи в MES. Якщо вони зібрані з використанням автоматизованого обладнання, то передаються в систему безпосередньо. Вона також реєструє, скільки виробів в партії виявилось бракованими, щоб визначити рівень відповідності всієї партії стандартам. Після перевірок MES обробляє результати та автоматично присвоює продукції статуси:

- «схвалено»: продукція відповідає всім вимогам та може бути відвантажена;
- «умовно схвалено»: продукція потребує незначних доопрацювань перед відвантаженням;
- «відхилено»: продукція не відповідає вимогам та відправляється в зону доопрацювання або на утилізацію.

За результатами перевірки готових виробів MES формує звіти, які включають в себе кількість перевірених одиниць, процент відповідності, типи дефектів та частоту їх появи. Кожен продукт або партія зі схвалених системою для подальшого відвантаження замовнику маркується унікальним ідентифікатором (наприклад, штрих-кодом або QR-кодом) (рис. 1.1)



Рисунок 1.1 – Зображення QR-коду на батареї для смартфона [4]

Після завершення всіх процесів MES контролює відвантаження продукції замовнику, оновлюючи статус партії та реєструючи її переміщення.

ВИСНОВКИ. Підсумовуючи, можна зазначити, що MES забезпечує комплексне управління якістю продукції на всіх етапах виробництва: від вхідного контролю сировини до перевірки готової продукції. Система автоматизує ключові процеси, включаючи інспекції, документування даних, управління коригувальними діями та аналіз відхилень. Це дає можливість скоротити вплив людського фактору, прискорити виконання перевірок та забезпечити відповідність міжнародним стандартам. Подальший розвиток включає в себе інтеграцію з іншими компонентами MES, з системами типу ERP, SCADA, використання штучного інтелекту для предиктивного аналізу відхилень, тощо.

ЛІТЕРАТУРА

1. MES Quality Control in Manufacturing. *SkyPlanner*. URL: <https://skyplanner.ai/resources/mes-quality-control-in-manufacturing/> (дата звернення: 15.11.2024).
2. Manufacturing Execution Systems – MES / ред. J. Kletti. Berlin, Heidelberg : Springer Berlin Heidelberg, 2017. 271 с.
3. Production operation management using Manufacturing Execution Systems (MES) / M. I. Mahmoud та ін. *2015 11th international computer engineering conference (ICENCO)*,

м. Cairo, Египт, 29–30 груд. 2015 р. 2015. URL: <https://doi.org/10.1109/icenco.2015.7416334> (дата звернення: 15.11.2024).

4. LiFePO₄ cells QR codes decoder : gobelpower.com. *GobelPower*. URL: https://www.gobelpower.com/lifepo4_decoder.html (дата звернення: 15.11.2024).

Науковий керівник: Хрустальова Софія Володимирівна, доцент кафедри КІТАР Харківського національного університету радіоелектроніки

ДОДАТОК Б
Лістинг програми

Код навчання моделі:

```

import pandas as pd
import numpy as np
from sqlalchemy import create_engine
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
import joblib

FACTORY_CONFIG = {
    "BASELINE_THICKNESS": 117.5,
    "BASELINE_TEMP": 115.0,
    "BASELINE_TIME": 2700,

    # Коефіцієнти залежності
    "TEMP_COEFFICIENT": 0.5,
    "TIME_COEFFICIENT": 100,

    "TEMP_TOLERANCE": 15.0,
    "TIME_TOLERANCE": 600
}

DATABASE_URI = "mysql+pymysql://root:root@localhost/manufacture"
engine = create_engine(DATABASE_URI)

query = """
SELECT
    d.applied_thickness,
    dr.furnace_temp,
    TIME_TO_SEC(TIMEDIFF(dr.end_time, dr.start_time)) AS drying_time
FROM battery b
JOIN dosing_data d ON d.battery_id = b.id
JOIN drying_data dr ON dr.battery_id = b.id;
"""

data = pd.read_sql(query, engine)
print(f"Завантажено реальних записів з БД: {len(data)}")

def classify_quality_strict(row):

    t_real = row['applied_thickness']
    temp_real = row['furnace_temp']
    time_real = row['drying_time']

```

```

cfg = FACTORY_CONFIG

delta_thickness = t_real - cfg["BASELINE_THICKNESS"]

ideal_temp = cfg["BASELINE_TEMP"] - (delta_thickness * cfg["TEMP_COEFFICIENT"])
ideal_time = cfg["BASELINE_TIME"] + (delta_thickness * cfg["TIME_COEFFICIENT"])

if time_real > (ideal_time + cfg["TIME_TOLERANCE"]):
    return 1

if time_real < (ideal_time - cfg["TIME_TOLERANCE"]):
    return 2

if temp_real > (ideal_temp + cfg["TEMP_TOLERANCE"]):
    return 1

if temp_real < (ideal_temp - cfg["TEMP_TOLERANCE"]):
    return 2

if time_real > 5000: return 1
if temp_real > 160: return 1
if temp_real < 60: return 2

return 0

synthetic_rows = []

for _ in range(200):
    t = np.random.uniform(116, 119)
    temp =

    time_bad = np.random.uniform(3350, 3800)
    synthetic_rows.append({"applied_thickness": t, "furnace_temp": temp, "drying_time": time_bad,
"forced_label": 1})

    time_good = np.random.uniform(2500, 3100)
    synthetic_rows.append({"applied_thickness": t, "furnace_temp": temp, "drying_time": time_good,
"forced_label": 0})

for _ in range(100):
    t = np.random.uniform(110, 125)

```

```

synthetic_rows.append(
    {"applied_thickness": t, "furnace_temp": 115, "drying_time": 32000, "forced_label": 1})
synthetic_rows.append(
    {"applied_thickness": t, "furnace_temp": 200, "drying_time": 2700, "forced_label": 1})

synthetic_df = pd.DataFrame(synthetic_rows)
full_data = pd.concat([data, synthetic_df], ignore_index=True)

print(f"Даних після додавання синтетики: {len(full_data)}")

def get_final_label(row):
    if pd.notna(row.get('forced_label')):
        return int(row['forced_label'])
    return classify_quality_strict(row)

full_data['status_code'] = full_data.apply(get_final_label, axis=1)

X = full_data[["applied_thickness", "furnace_temp", "drying_time"]]
y = full_data["status_code"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = RandomForestClassifier(n_estimators=200, max_depth=10, random_state=42)
model.fit(X_train, y_train)

print("\nЗвіт по якості моделі:")
print(classification_report(y_test, model.predict(X_test)))

joblib.dump(model, "drying_quality_model.pkl")
print("Модель збережено у drying_quality_model.pkl")

```

Код запуску сервера Flask:

```

from flask import Flask, request, jsonify
import joblib
import numpy as np

app = Flask(__name__)

model = joblib.load("drying_quality_model.pkl")

STATUS_MAP = {
    0: "OK",
    1: "OVERHEAT_CRACKING",
    2: "UNDERHEAT_HUMIDITY"
}

```

```

def find_recommendation(thickness, temp, time, original_prediction):

    if time > 3600:
        suggested_time = 2700
        return "drying_time", suggested_time, f"Drying time {int(time)}s is critically long! Reduce
it to {suggested_time}s."

    step_temp = 5
    step_time = 300

    best_temp = temp
    for _ in range(10):
        if original_prediction == 1:
            best_temp -= step_temp
        elif original_prediction == 2:
            best_temp += step_temp

        if model.predict([[thickness, best_temp, time]])[0] == 0:
            return "furnace_temp", best_temp, f"Change the temperature to {best_temp}°C"
    best_time = time
    for _ in range(10):
        if original_prediction == 1:
            best_time -= step_time
        elif original_prediction == 2:
            best_time += step_time

        if model.predict([[thickness, temp, best_time]])[0] == 0:
            return "drying_time", best_time, f"The time does not match the thickness. Set
{int(best_time)} s."
    return None, None, "The settings are critical. Manually change both the time and temperature."
@app.route("/validate/drying", methods=["POST"])
def validate_drying():
    data = request.get_json()

    thickness = float(data["applied_thickness"])
    temp = float(data["furnace_temp"])
    time = float(data["drying_time"])
    X_new = np.array([[thickness, temp, time]])

    prediction_code = model.predict(X_new)[0]
    status_text = STATUS_MAP.get(prediction_code, "UNKNOWN")

    response = {"status": status_text}

    if prediction_code != 0:
        param, value, msg = find_recommendation(thickness, temp, time, prediction_code)

        if param:
            response["recommendation"] = {
                "parameter": param,
                "suggested_value": value,
                "message": msg
            }
        else:
            response["recommendation"] = {"message": "The settings are critical. Manually reduce
both the time and temperature."}

    return jsonify(response)
if __name__ == "__main__":
    app.run(port=5001)

```

Код інтерфейсу AiValidatorStrategy:

```

package com.example.Manufac.ai;

import com.example.Manufac.dto.TechnicalSheetDTO;

import java.util.Map;

public interface AiValidatorStrategy {
    boolean supports(String processType);
    Map<String, Object> validate(TechnicalSheetDTO sheetDTO);
}

```

Код класу DryingAiValidator:

```

package com.example.Manufac.ai;

import com.example.Manufac.dto.TechnicalSheetDTO;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Component;
import org.springframework.web.client.RestTemplate;

import java.util.HashMap;
import java.util.Map;
@Component
public class DryingAiValidator implements AiValidatorStrategy{

    private final RestTemplate restTemplate = new RestTemplate();
    private final String FLASK_URL = "http://127.0.0.1:5001/validate/drying";

    @Override
    public boolean supports(String processType) {
        return "DRYING".equalsIgnoreCase(processType);
    }

    @Override
    public Map<String, Object> validate(TechnicalSheetDTO sheetDTO) {
        Map<String, Object> uiResponse = new HashMap<>();

        try {
            double finalThickness = (sheetDTO.getThicknessOfSubstanceMax() +
sheetDTO.getThicknessOfSubstanceMin()) / 2.0;
            int finalDryingTemp = (sheetDTO.getDryingTemperatureMax() +
sheetDTO.getDryingTemperatureMin()) / 2;
            Long timeAvg = (sheetDTO.getTimeOfDryingMax() + sheetDTO.getTimeOfDryingMin()) / 2;

            Map<String, Object> requestData = new HashMap<>();
            requestData.put("applied_thickness", finalThickness);
            requestData.put("furnace_temp", finalDryingTemp);
            requestData.put("drying_time", timeAvg);

```

```

        ResponseEntity<Map> responseEntity = restTemplate.postForEntity(FLASK_URL, requestData,
Map.class);
        Map<String, Object> flaskBody = responseEntity.getBody();

        String status = (String) flaskBody.get("status");

        if ("OK".equals(status)) {
            uiResponse.put("valid", true);
            uiResponse.put("message", "The technical data sheet is AI-verified. Parameters are
in the optimal quality zone");
        } else {
            uiResponse.put("valid", false);
            String defectMsg = getDefectMessage(status);

            if (flaskBody.containsKey("recommendation")) {
                Map<String, Object> rec = (Map<String, Object>) flaskBody.get("recommendation");
                uiResponse.put("message", defectMsg + " " + rec.get("message"));
                uiResponse.put("suggested_value", rec.get("suggested_value"));
            } else {
                uiResponse.put("message", defectMsg);
            }
        }

        } catch (Exception e) {
            uiResponse.put("valid", false);
            uiResponse.put("message", "AI module connection error: " + e.getMessage());
        }

        return uiResponse;
    }

    private String getDefectMessage(String status) {
        if ("OVERHEAT_CRACKING".equals(status)) return "Risk of defect: Cracking (Overheating)";
        if ("UNDERHEAT_HUMIDITY".equals(status)) return "Risk of defect: High humidity
(Underheating)";
        return "Unknown defect risk";
    }
}

```

Код сервісу AiValidationService:

```

@Service
public class AiValidationService {
    private final List<AiValidatorStrategy> validators;

    public AiValidationService(List<AiValidatorStrategy> validators) {
        this.validators = validators;
    }

    public Map<String, Object> validateProcess(String processType, TechnicalSheetDTO dto) {
        return validators.stream()
            .filter(v -> v.supports(processType))
            .findFirst()
            .map(v -> v.validate(dto))
            .orElseThrow(() -> new IllegalArgumentException("No AI model found for process: " +
processType));
    }
}

```

Метод, що перехоплює ендпоінт разом з параметром DRYING зі сторінки HTML, щоб запустити процес валідації:

```

@PostMapping("/validate-sheet")
@ResponseBody
public Map<String, Object> validateSheet(@RequestBody TechnicalSheetDTO sheetDTO,
    @RequestParam(defaultValue = "DRYING") String processType)
{
    return aiValidationService.validateProcess(processType, sheetDTO);
}

```

Код збереженої процедури:

```

SET @OLD_SQL_MODE = @@sql_mode;
SET sql_mode = '';

DROP PROCEDURE IF EXISTS seed_battery_data;
DELIMITER $$

CREATE PROCEDURE seed_battery_data(IN batches INT, IN per_batch INT, IN start_batch_number INT)
BEGIN
    DECLARE b INT DEFAULT 0;
    DECLARE i INT;
    DECLARE batch_num INT DEFAULT start_batch_number;
    DECLARE per INT DEFAULT per_batch;
    DECLARE avg_thickness DOUBLE DEFAULT 117.5;
    DECLARE avg_temp INT DEFAULT 115;
    DECLARE avg_time_ms INT DEFAULT 2700000;
    DECLARE min_thickness DOUBLE DEFAULT 110;
    DECLARE max_thickness DOUBLE DEFAULT 125;
    DECLARE min_time_ms INT DEFAULT 1800000;
    DECLARE max_time_ms INT DEFAULT 3600000;
    DECLARE min_temp INT DEFAULT 80;
    DECLARE max_temp INT DEFAULT 150;
    DECLARE r DOUBLE;
    DECLARE t DOUBLE;
    DECLARE temp INT;
    DECLARE time_ms INT;
    DECLARE status VARCHAR(20);
    DECLARE serial VARCHAR(50);
    DECLARE weight DOUBLE;
    DECLARE capacity INT;
    DECLARE res INT;
    DECLARE vol DOUBLE;
    DECLARE batch_id BIGINT;
    DECLARE inserted_bat_id BIGINT;

    WHILE b < batches DO
        INSERT INTO batch (number_of_batch, quantity_of_batteries, nominal_capacity, batteries_model,
rated_voltage, technicalSheet_id)
        VALUES (batch_num, per, 3000, '18650', 3.7, 3);
        SET batch_id = LAST_INSERT_ID();

        SET i = 0;
    
```

```

WHILE i < per DO
  SET t = ROUND(min_thickness + RAND() * (max_thickness - min_thickness), 1);

  IF RAND() < 0.30 THEN
    IF t > avg_thickness THEN
      SET temp = GREATEST(min_temp, FLOOR(avg_temp - (10 + RAND()*25)));
      SET time_ms = LEAST(max_time_ms, avg_time_ms + FLOOR(200000 + RAND()*1000000));
    ELSE
      SET temp = LEAST(max_temp, FLOOR(avg_temp + (10 + RAND()*25)));
      SET time_ms = GREATEST(min_time_ms, avg_time_ms - FLOOR(200000 + RAND()*1000000));
    END IF;
    SET status = 'DEFECTIVE';
  ELSE
    SET temp = FLOOR(avg_temp - ((t - avg_thickness) * 0.5) + (RAND()*6 - 3));
    IF temp < min_temp THEN SET temp = min_temp; END IF;
    IF temp > max_temp THEN SET temp = max_temp; END IF;

    SET time_ms = ROUND(avg_time_ms + ((t - avg_thickness) * 50000) + (RAND()*200000 - 100000));
    IF time_ms < min_time_ms THEN SET time_ms = min_time_ms; END IF;
    IF time_ms > max_time_ms THEN SET time_ms = max_time_ms; END IF;

    SET status = 'WORKING';
  END IF;

  SET serial = CONCAT('B', batch_num, '-', LPAD(i+1,3,'0'));
  SET weight = ROUND(46.5 + RAND()*1.5, 2);
  SET capacity = 2980 + FLOOR(RAND()*140);
  SET res = 18 + FLOOR(RAND()*15);
  SET vol = ROUND(3.65 + RAND()*0.12, 2);

  INSERT INTO battery (serial_number, weight, actual_capacity, actual_internal_resistance,
voltage, status, batch_id)
  VALUES (serial, weight, capacity, res, vol, status, batch_id);
  SET inserted_bat_id = LAST_INSERT_ID();

  INSERT INTO dosing_data (applied_thickness, battery_id)
  VALUES (t, inserted_bat_id);

  INSERT INTO drying_data (id, start_time, end_time, furnace_temp, battery_id)
  VALUES (inserted_bat_id, '08:00:00', ADDTIME('08:00:00', SEC_TO_TIME(ROUND(time_ms/1000))),
temp, inserted_bat_id);

```

```

        SET i = i + 1;
    END WHILE;

    SET b = b + 1;
    SET batch_num = batch_num + 1;
    END WHILE;
END$$

DELIMITER ;

CALL seed_battery_data(49, 10, 102);

DROP PROCEDURE IF EXISTS seed_battery_data;

SET sql_mode = @OLD_SQL_MODE;

```

Код PLCService:

```

package com.example.Manufac.service.plc;

import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;

@Service
@RequiredArgsConstructor
public class PLCService {

    private final WeighingMixingPLC weighingMixingPLC;
    private final CoatingPLC coatingPLC;
    private final AssemblyPLC assemblyPLC;
    private final TestingPLC testingPLC;

    public void simulateWeighingProcess(Integer batchNumber, boolean isRetry) {
        weighingMixingPLC.simulateWeighingProcess(batchNumber, isRetry);
    }

    public void simulateMixingProcess(Integer batchNumber) {
        weighingMixingPLC.simulateMixingProcess(batchNumber);
    }

    public void simulateDosingProcess(Integer batchNumber) {
        coatingPLC.simulateDosingProcess(batchNumber);
    }

    public void simulateDryingProcess(Integer batchNumber) {
        coatingPLC.simulateDryingProcess(batchNumber);
    }

    public void simulatePressingProcess(Integer batchNumber) {
        coatingPLC.simulatePressingProcess(batchNumber);
    }
}

```

```

public void simulateCuttingProcess(Integer batchNumber) {
    assemblyPLC.simulateCuttingProcess(batchNumber);
}

public void simulateStackingProcess(Integer batchNumber) {
    assemblyPLC.simulateStackingProcess(batchNumber);
}

public void simulateCellElectrolyteProcess(Integer batchNumber) {
    assemblyPLC.simulateCellElectrolyteProcess(batchNumber);
}

public void simulateFirstTestProcess(Integer batchNumber) {
    testingPLC.simulateFirstTestProcess(batchNumber);
}

public void simulateSecondTestProcess(Integer batchNumber) {
    testingPLC.simulateSecondTestProcess(batchNumber);
}

public void simulateThirdTestProcess(Integer batchNumber) {
    testingPLC.simulateThirdTestProcess(batchNumber);
}
}

```

Код WeighingMixingPLC:

```

package com.example.Manufac.service.plc;

public interface WeighingMixingPLC {

    void simulateWeighingProcess(Integer batchNumber, boolean isRetry);
    void simulateMixingProcess(Integer batchNumber);
}

```

Код WeighingMixingPLCImpl:

```

package com.example.Manufac.service.plc;

import com.example.Manufac.constants.BatchStatus;
import com.example.Manufac.dto.IngredientItemDTO;
import com.example.Manufac.dto.TechnicalSheetViewDTO;
import com.example.Manufac.exception.NotFoundException;
import com.example.Manufac.models.Batch;
import com.example.Manufac.models.Ingredient;
import com.example.Manufac.models.TechnicalDataSheet;
import com.example.Manufac.models.WeighingItem;
import com.example.Manufac.models.measured.MixingData;
import com.example.Manufac.models.measured.WeighingData;
import com.example.Manufac.repo.*;
import com.example.Manufac.service.TechnicalSheetsService;
import jakarta.transaction.Transactional;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;
import java.time.LocalDateTime;
import java.util.List;

```

```

import static com.example.Manufac.utils.SimulationUtils.generateRandomValidValue;

@Service
@RequiredArgsConstructor
public class WeighingMixingPLCImpl implements WeighingMixingPLC{

    private final BatchRepository batchRepository;
    private final IngredientRepository ingredientRepository;
    private final TechnicalSheetsService technicalSheetsService;
    private final WeighingDataRepository weighingDataRepository;
    private final WeighingItemRepository weighingItemRepository;
    private final MixingDataRepository mixingDataRepository;

    @Override
    @Transactional
    public void simulateWeighingProcess(Integer batchNumber, boolean isRetry) {
        Batch batch = batchRepository.findBatchByNumberOfBatch(batchNumber)
            .orElseThrow(() -> new NotFoundException("Batch not found"));

        TechnicalSheetViewDTO techSheet =
technicalSheetsService.findById(batch.getTechnicalSheet().getId());

        WeighingData weighingData =
weighingDataRepository.findByBatchForWeighting_NumberOfBatch(batch.getNumberOfBatch())
            .orElseGet(() -> {
                WeighingData newData = new WeighingData();
                newData.setBatchForWeighting(batch);
                return weighingDataRepository.save(newData);
            });

        if (weighingData.getItems() != null && !weighingData.getItems().isEmpty()) {
            weighingItemRepository.deleteAll(weighingData.getItems());
            weighingData.getItems().clear();
            weighingDataRepository.saveAndFlush(weighingData);
        }

        List<IngredientItemDTO> requiredIngredients = techSheet.getIngredientItems();

        for (int i = 0; i < requiredIngredients.size(); i++) {
            IngredientItemDTO targetItem = requiredIngredients.get(i);

            double targetQuantity = targetItem.getQuantity();
            double generatedRealQuantity;

            if (i == 0 && !isRetry) {
                generatedRealQuantity = targetQuantity * 0.5;
            } else {
                generatedRealQuantity = generateRandomValidValue(targetQuantity);
            }

            Ingredient ingredient =
ingredientRepository.findById(targetItem.getIngredientId()).orElseThrow(() -> new
NotFoundException("Ingredient not found"));

            WeighingItem weighingItem = new WeighingItem();
            weighingItem.setWeighingData(weighingData);
            weighingItem.setIngredient(ingredient);
            weighingItem.setRealQuantity(generatedRealQuantity);
            weighingItemRepository.save(weighingItem);

            boolean isValid = checkTolerance(targetQuantity, generatedRealQuantity);

```

```

        if (!isValid) {
            batch.setBatchStatus(BatchStatus.ERROR);
            batch.setStartOfProduction(LocalDateTime.now());
            batchRepository.save(batch);
            return;
        }
    }

    batch.setBatchStatus(BatchStatus.WEIGHING_CHECKED);
    batch.setStartOfProduction(LocalDateTime.now());
    batchRepository.save(batch);
}

private boolean checkTolerance(double target, double real) {
    double tolerancePercent = 0.05;
    double diff = Math.abs(target - real);
    return diff <= (target * tolerancePercent);
}

@Override
@Transactional
public void simulateMixingProcess(Integer batchNumber) {
    Batch batch = batchRepository.findBatchByNumberOfBatch(batchNumber)
        .orElseThrow(() -> new NotFoundException("Batch not found"));

    TechnicalDataSheet techSheet = batch.getTechnicalSheet();

    long minSusp = techSheet.getTimeOfMixingSuspensionMin();
    long maxSusp = techSheet.getTimeOfMixingSuspensionMax();
    long factSusp = generateRandomLong(minSusp, maxSusp);

    long minElec = techSheet.getTimeOfMixingElectrolyteMin();
    long maxElec = techSheet.getTimeOfMixingElectrolyteMax();
    long factElec = generateRandomLong(minElec, maxElec);

    MixingData mixingData =
        mixingDataRepository.findMixingDataByMixingData_NumberOfBatch(batchNumber)
            .orElse(new MixingData());

    mixingData.setMixingData(batch);
    mixingData.setFactTimeSuspension(factSusp);
    mixingData.setFactTimeElectrolyte(factElec);

    mixingDataRepository.save(mixingData);

    batch.setBatchStatus(BatchStatus.MIXING_CHECKED);
    batchRepository.save(batch);
}

private long generateRandomLong(long min, long max) {
    return min + (long) (Math.random() * (max - min));
}
}

```

ДОДАТОК В
Демонстраційний матеріал

