

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій

(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації

та робототехніки

(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА

### Пояснювальна записка

Перший (бакалаврський)

(рівень вищої освіти)

«Інтелектуальне керування складськими

операціями»

(тема)

Виконав:

студент 4 курсу, групи АКТАКІТ-20-1

Кривенко Д.О.

Спеціальності 151 Автоматизація та

комп'ютерно-інтегровані технології

Тип програми Освітньо-професійна

Освітня програма Автоматизація та

комп'ютерно-інтегровані технології

Керівник проф. Сезонова І.К

Допускається до захисту

Зав. кафедри КІТАР

(підпис)

Невлюдов І.Ш.

(прізвище, ініціали)

2024р.

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

05.06.2024 р.



---

(підпис)

Кривенко  
Д.О.

---

(ПІБ)

## ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Автоматики і комп'ютеризованих технологійКафедра Комп'ютерно-інтегрованих технологій, автоматизації та  
робототехнікиРівень вищої освіти перший другий (магістерський)Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технологіїТип програми Освітньо-професійнаОсвітня програма Комп'ютеризовані та робототехнічні системи  
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри КІТАР \_\_\_\_\_

(підпис)

«» \_\_\_\_\_ 2024 р.**ЗАВДАННЯ****НА КВАЛІФІКАЦІЙНУ РОБОТУ**студентові \_\_\_\_\_ Кривенку Денису Олексійовичу  
(прізвище, ім'я, по батькові)1. Тема роботи \_\_\_\_\_ Інтелектуальне керування складськими операціямиЗатверджена наказом по університету від 03.06.2024 р. № 544 СТ2. Термін подання студентом роботи до екзаменаційної комісії 25.06.20243. Вихідні дані до роботи Середовище програмування Visual Studio,  
середовище розробки AvaloniaUI, база даних Microsoft SQL Server,  
мова програмування C#

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

4.1 Вступ;

4.2 Аналіз технічного завдання;

4.3 Інтелектуальне керування складськими операціями;

4.4 Розробка програмного забезпечення;

4.5 Охорона праці;

4.6 Висновки;

4.7 Додатки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій  
демонстраційний матеріал представлений у форматі презентації PowerPoint  
(\* .pptx)

6. Консультанти розділів роботи

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

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання до кваліфікаційної роботи	10.04.2024	виконано
2	Вступ	11.04 – 12.04.24	виконано
3	Аналіз технічного завдання	12.04 – 13.04.24	виконано
4	Інтелектуальне керування складськими операціями	14.04 – 17.04.24	виконано
5	Розробка програмного забезпечення для системи інтелектуального керування складськими операціями	17.04 – 22.05.24	виконано
6	Охорона праці	22.05 – 27.05.24	виконано
7	Висновки	27.05 – 29.05.24	виконано
8	Оформлення пояснювальної записки	29.05 – 05.06.24	виконано
9	Подання роботи на перевірку Інтернет-сервісом StrikePlagiasm	05.06 – 06.06.24	виконано
10	Подання роботи на рецензію	07.06 – 08.06.24	виконано
11	Подання роботи на підпис зав. кафедри	09.06 – 10.06.24	виконано
12	Подання атестаційної роботи в ЕК	11.06.24	виконано

Дата видачі завдання 10.04.2024 р.

Студент

\_\_\_\_\_ (підпис)

Кривенко Д.О.

Керівник роботи

\_\_\_\_\_ (підпис)

\_\_\_\_\_ проф. Сезонова І.К.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: 93 с., 4 табл., 47 рис., 9 дод., 29 джерел.

СИСТЕМА, ІНТЕЛЕКТКАЛЬНЕ КЕРУВАННЯ, АНАЛІЗ, ВИТРАТИ, ПРОГРАМА, РІШЕННЯ, QR, СКАНЕР, АЛГОРИТМ, ЗНАЧЕННЯ, КОРИСТУВАЧ, БАЗА ДАНИХ, МЕТОД, ШТУЧНІЙ ІНТЕЛЕКТ, МАШИННЕ НАВЧАННЯ, СКЛАДСЬКІ ОПЕРАЦІЇ.

Мета роботи – вдосконалення керування складськими операціями за допомогою інтелектуальної автоматизованої системи.

Об’єкт розробки – інтелектуальна автоматизована система керування складськими операціями.

Предмет розробки – модель та програмна реалізація інтелектуального помічника для керування складськими операціями.

Результатом роботи є застосунок призначений для самостійної, автоматизованої роботи складського приміщення. Результатами досліджень з’ясовано, що впровадження інтелектуальної системи управління складськими операціями є ефективним та економічно вигідним рішенням, що дозволяє значно підвищити продуктивність складських процесів. Перспективи подальших досліджень у цьому напрямку включають розробку нових алгоритмів машинного навчання та інтеграцію інтелектуальних систем з іншими елементами ланцюга поставок.

## ABSTRACT

Explanatory note: 93 p., 4 tabl., 47 fig., 9 adj., 29 sources.

SYSTEM, INTELLIGENT CONTROL, ANALYSIS, COSTS, PROGRAM, SOLUTION, QR, SCANNER, ALGORITHM, VALUE, USER, DATABASE, METHOD, ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, WAREHOUSE OPERATIONS.

Purpose – to improve the management of warehouse operations using an intelligent automated system.

Object of development – an intelligent automated system for managing warehouse operations. The subject of development is a model and software implementation of an intelligent assistant for warehouse operations management.

The result of the work is an application designed for independent, automated operation of a warehouse. The research results have shown that the implementation of an intelligent warehouse operations management system is an effective and cost-efficient solution that can significantly increase the productivity of warehouse processes. Prospects for further research in this area include the development of new machine learning algorithms and the integration of intelligent systems with other elements of the supply chain.

## ЗМІСТ

Перелік умовних скорочень .....	9
Вступ.....	11
1 Аналіз технічного завдання.....	13
1.1 Інтелектуальне керування.....	13
1.2 Принципи автоматизації складських приміщень.....	14
1.3 Аналіз існуючих рішень .....	17
1.4 Підсумок аналізу технічного завдання.....	21
2 Інтелектуальне керування складськими операціями .....	23
2.1 Інтелектуальне керування складськими операціями .....	23
2.2 Економічна доцільність та ROI.....	25
2.2.1 Первинні витрати .....	25
2.2.2 Поточні витрати.....	26
2.2.3 Сукупні витрати.....	27
2.2.4 Сукупні вигоди .....	30
2.2.5 Розрахунок ROI.....	31
2.3 Приклади готового рішення .....	32
2.3.1 Система фулфілмент на “Новій пошті” .....	32
2.3.2 Система Fulfilment by Amazon .....	34
2.4 Вибір програмного середовища для реалізації проекту.....	36
3 Розробка програмного забезпечення для системи інтелектуального керування складськими операціями .....	38
3.1 Неперервна лінійна САУ .....	38
3.2 Опис алгоритмів програми.....	39

	8
3.2 Розробка структури БД .....	52
3.3 Розробка інтерфейсу користувача.....	57
4 Охорона праці .....	70
Висновки .....	72
Перелік джерел посилання .....	74
ДОДАТОК А Код авторизації користувачів.....	78
ДОДАТОК Б Код оновлення графіків складу.....	79
ДОДАТОК В Код керування зовнішніми замовленнями.....	81
ДОДАТОК Г Код замовлень потреб виробництва .....	83
ДОДАТОК Д Код сканеру для інвентаризації товару .....	85
ДОДАТОК Е Код для видалення товарів .....	87
ДОДАТОК Ж Код генерації QR-коду .....	89
ДОДАТОК И Код для генерації звітів та аналізу даних .....	91
ДОДАТОК К Демонстраційний матеріал у вигляді презентації.....	92
ДОДАТОК Л Відомість кваліфікаційної роботи .....	93

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

- БД – база даних;
- КІТАР – комп’ютерно-інтегрованих технологій, автоматизації та робототехніки;
- ПЗ – програмне забезпечення;
- САУ – система автоматизованого управління;
- СКБ – студентське конструкторсько-технологічне бюро;
- ТАУ – теорія автоматичного управління;
- ХНУРЕ – харківський національний університет радіоелектроніки;
- ШІ – штучний інтелект;
- AI (англ. Artificial Intelligence) – штучний інтелект;
- CRM (англ. Customer Relationship Management) – керування відносинами з клієнтами;
- CPS (англ. Cyber-Physical System) – кіберфізична система;
- ERP (англ. Enterprise Resource Planning) – планування ресурсів підприємств;
- FBA (англ. Fulfillment by Amazon) – фулфілмент Amazon;
- GUI (англ. graphical user interface) – графічний інтерфейс користувача;
- IDE (англ. Integrated development environment) – інтегроване середовище розробки;
- IMoWO (англ. Intelligent Management of Warehouse Operations) – інтелектуальне керування складськими операціями;
- IoT (англ. Internet of Things) – інтернет речей;
- IT (англ. Information Technology) – інформаційні технології;
- ML (англ. Machine Learning) – машинне навчання;
- MVVM (англ. Model-View-Viewmodel) – шаблон проектування архітектури додатків;
- QR (англ. Quick Response) – тип матричних штрихових кодів;

RDBMS (англ. Relational database management system) – система управління реляційними базами даних;

RFID (англ. Radio Frequency IDentification ) – радіочастотна ідентифікація;

ROI (англ. Return on Investment) – окупність інвестицій;

SQL ( англ. Structured Query Language) – мова структурованих запитів;

WMS (англ. Warehouse Management System) – система керування складом;

XAML (англ. Xtensible Application Markup Language) – розширювана мова розмітки для додатків.

## ВСТУП

Логістика та керування складськими операціями в сучасному світі зазнали значних змін за останні десятиліття. Ставши складнішими складські комплекси потребують більш нові підходи до керування, які зможуть забезпечити ефективність, гнучкість та оперативність процесів. Підвищення рівню обслуговування клієнтів, скорочення витрат та оптимізація логістичних ланцюжків являються сучасними умовами в глобальній конкуренції компаній.

Інтелектуальне керування складськими операціями, засноване на застосуванні штучного інтелекту, машинного навчання та аналітики даних, що являє собою перспективний напрямок для поліпшення керування запасами, планування та координації складських процесів. Автоматичний аналіз великих обсягів даних, виявлення прихованих закономірностей, ухвалення оптимальних рішень в реальному часі інтелектуальної системи призводить до зниження людського чинника і зменшення кількості помилок, тим самим значно збільшує ефективність роботи складського приміщення.

Значимість даної роботи полягає в результатах дослідження що можуть бути використані для розроблення та впровадження інноваційних рішень у галузі логістики. Запровадження систем інтелектуального керування складськими операціями дасть компаніям змогу для оперативного реагування на зміни попиту, оптимізацію використання ресурсів та забезпечення високої точності для виконання замовлень. Також результати даної роботи можуть суттєво вплинути на розвиток глобальної цілі №11 “Сталий розвиток міст і громад”.

Швидка урбанізація призводить до неефективного розподілу та використання ресурсів, понаднормово збільшуючи навантаження на навколишнє середовище. Підприємства, які працюють у містах та сприяють загальному економічному процвітанню, мають можливість підтримувати розвиток інфраструктури, долучаючись до розроблення стратегій розвитку

міст і громад, використовуючи свої можливості та знання для визначення інноваційних та економічно ефективних рішень складних, міждисциплінарних питань сталості міст [1].

Мета роботи – вдосконалення керування складськими операціями за допомогою інтелектуальної автоматизованої системи.

Об’єкт розробки – інтелектуальна автоматизована система керування складськими операціями.

Предмет розробки – модель та програмна реалізація інтелектуального помічника для керування складськими операціями.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- провести аналіз існуючих систем інтелектуального керування;
- провести аналіз прикладів автоматизації складських приміщень;
- провести аналіз технічного завдання та зробити підсумок;
- розрахувати економічну доцільність та ROI;
- розробити алгоритм програми, інтерфейс користувача та БД.

Кваліфікаційна робота виконана згідно ДСТУ 3008:2015 [2], а також за методичними вказівками з підготовки й оформлення кваліфікаційної роботи здобувачами першого (бакалаврського) рівня вищої освіти спеціальності 151 Автоматизація та комп’ютерно-інтегровані технології освітньої програми «Автоматизація та комп’ютерно-інтегровані технології» [3].

## 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

Для успішного аналізу технічного завдання потрібно зрозуміти що таке інтелектуальне керування, які існують принципи автоматизації складських приміщень та провести аналіз схожих існуючих рішень по керуванню складськими операціями, після чого зробити підсумок аналізу технічного завдання.

### 1.1 Інтелектуальне керування

Сучасний світ не стоїть на місці. З'являються нові пристрої, технології, ідеї та розробки. Для всього цього створюються нові підприємства та розширюються старі. При збільшенні виробництва з'являється більша потреба в зберіганні товару тому разом із підприємствами розширюються і їх склади.

Великі склади дають змогу зберігати неймовірну кількість сировини, деталей, готового товару і т.д. проте орієнтування в таких приміщеннях дуже складне. Для цього створено автоматизацію за допомогою роботів та штучного інтелекту.

Штучний інтелект – це здатність машини або системи робити те, що зазвичай вважається ознакою людини [4]. Насправді, саме за допомогою ШІ людству вдалося забезпечити та продовжувати забезпечувати усі процеси на виробництві швидкою та точною роботою.

В складських операціях ШІ може допомогти з рядом проблем таких як: прогнозування попиту, оптимізацію маршрутів, автоматизацію складських процесів, аналіз даних та прийняття рішень тощо.

При прогнозуванні попиту можна використовувати алгоритми машинного навчання для аналізу та прогнозування попиту на товари. ШІ може враховувати великий обсяг даних, таких як історичні продажі, погодні умови,

святкові дні тощо, що дозволяє більш точно прогнозувати майбутні потреби та оптимізувати рівень запасів.

ШІ може бути використаний для автоматизації різних аспектів складської діяльності, включаючи сортування, пакування, інвентаризацію та переміщення товарів. Роботизовані системи та механізми машинного навчання можуть допомогти в оптимізації цих процесів, зменшуючи витрати на трудові ресурси та скорочуючи час виконання операцій.

Використання алгоритмів оптимізації та машинного навчання може допомогти при оптимізації маршрутів доставки товарів, управління складським простором та планування завантаження складу. Це дозволяє скоротити витрати на транспорт та підвищити ефективність використання складських ресурсів.

ШІ може допомогти аналізувати великі обсяги даних, зібраних за допомогою датчиків на складі та інших джерел, для виявлення тенденцій, виявлення аномалій та прийняття обґрунтованих управлінських рішень. Наприклад, системи машинного навчання можуть автоматично визначати оптимальне розподілення товарів на складі для поліпшення доступності та прискорення обробки замовлень [5].

## 1.2 Принципи автоматизації складських приміщень

Автоматизація складських приміщень дуже важлива частина будь якого виробництва та підприємства. Вона дозволяє ефективно використовувати простір складського приміщення, оптимізувати процеси переміщення, автоматизувати інвентаризацію, мінімізувати людське втручання тим самим забезпечити безпеку, гнучкість та масштабованість підприємства.

Ефективне використання простору передбачає максимальне використання наявного простору на складі. Це може бути досягнуто шляхом впровадження різних систем складування (рис. 1.1).



Рисунок 1.1 – Складське приміщення із системами складування [6]

Як видно на рисунку системами складування можуть бути високорівневі стелажі, автоматизовані системи переміщення товарів та системи розподілення простору відповідно до характеристик товарів (наприклад, розмір, вага, швидкість обороту тощо).

Оптимізація процесів переміщення має на меті зменшення часу та витрат на переміщення товарів на складі (рисунок 1.2).

Для цього можна використовувати автоматизовані системи транспорту, такі як роботизовані підйомники, конвеєри, автоматичні транспортери тощо, що дозволяють ефективно та швидко переміщати товари в межах складу.

Автоматизація інвентаризації полягає у використанні сучасних технологій, таких як сканування штрих-кодів, RFID (англ. Radio Frequency Identification – радіочастотна ідентифікація) та автоматизовані системи обліку для швидкого та точного визначення кількості товарів на складі. Це допомагає уникнути помилок та забезпечити точний облік запасів.



Рисунок 1.2 – Роботизований навантажувач на підприємстві [7]

Мінімізація людського втручання передбачає використання автоматизованих систем управління складом. Це може включати автоматизовані системи вибірки товарів, автоматичне управління складом та роботизовані системи пакування (рисунок 1.3).

За допомогою роботизованих систем вдається зменшити: час виробництва, потребу в ручній праці та ризик виникнення помилок.

Забезпечення безпеки охоплює використання сучасних систем безпеки та контролю доступу для захисту товарів на складі від крадіжок та пошкоджень. Це може включати відеоспостереження, системи контролю доступу, сигналізацію та інші заходи безпеки.



Рисунок 1.3 – Робот-палетайзер на виробництві [8]

Гнучкість та масштабованість передбачає розробку систем, які можуть легко адаптуватися до змінних потреб бізнесу та масштабуватися з його зростанням. Це означає використання гнучких технологій, модульних систем та інших рішень, які дозволяють швидко впроваджувати зміни та розширювати функціональність складських систем.

### 1.3 Аналіз існуючих рішень

При автоматизації складських приміщень найбільш всього використовують “Систему керування складським приміщенням WMS”.

WMS (з англ. Warehouse Management System) – система управління складом, яка забезпечує комплексну автоматизацію всіх складських бізнес-процесів.

Це ПЗ є системою управління складським господарством, включаючи логістику, облік запасів, та планування. Використання цієї системи дозволяє знизити витрати та підвищити прозорість у складських операціях.

WMS система спеціально створена для автоматизації складських операцій, простіше кажучи інструмент, який допомагає логістичним провайдерам:

- поліпшити процеси управління персоналом та скоротити витрати на оплату праці;
- вести інформативну звітність про стан складських запасів;
- збільшити швидкість ухвалення рішень;
- мінімізувати виробничі ризики;
- підвищити якість обслуговування клієнтів.

Головна мета системи управління складом (WMS) полягає в автоматизації складу, прискоренні робочих процесів та зниженні можливих помилок, які можуть виникати через людський фактор.

Основні функції WMS виглядають так:

- оптимізація складського приміщення;
- оптимізація графіка роботи;
- контроль запасів;
- оптимізація зберігання;
- управління відвантаженням та завантаженням;
- документообіг;
- проведення інвентаризації;
- комплектація;
- збільшення кількості клієнтів;
- моніторинг та звітність.

Для оптимізації використання складського простору потрібно ввести інформацію про розмір складу та дані про запаси, такі як розміри піддонів, кількість існуючого обладнання та продуктів, у систему. Після цього

програмне забезпечення обробить ці дані та побудує діаграму, що допоможе оптимально розмістити власні ресурси та ефективно використовувати вільний простір.

Система аналізує майбутні робочі процеси компанії та, враховуючи кількість персоналу, може створювати оптимальний графік роботи. Наприклад, вона може вказувати, скільки працівників потрібно для вчасного виконання конкретного замовлення. Крім того, WMS система може підключатися до транспортних компаній та заздалегідь планувати час завантаження/відвантаження товарів у транспортний засіб.

У програмі чітко відображається, який товар потребує додаткового замовлення через дефіцит. Існують опції для поповнення одиниць товару, піддонів або коробок. Ця функція сприяє уникненню надмірного запасу на складі, оскільки товар замовляється лише у необхідній кількості, а не довільно або "на випадок".

Система класифікує товар за різними критеріями, такими як вага, потреба чи терміни зберігання. Таким чином, товари з високою активністю будуть розміщені ближче до зони відвантаження.

Система управління складом вміє автоматично зчитувати та ідентифікувати товар, що надходить на склад, створювати штрих-коди, перевіряти дані про вантаж та виявляти будь-які розбіжності в описі або характеристиках товару.

Після впровадження WMS склад та підприємство повністю переходять на електронний формат звітів та документації, відмовляючись від паперового варіанту. Це надзвичайно зручно та ефективно.

Проведення інвентаризації відбувається без перерви у робочих процесах. Це критично для забезпечення неперервності логістичних операцій.

Завдяки використанню готових пакувальних листів, де зазначені всі елементи замовлення та їх розташування, процес збору замовлень відбувається значно швидше.

Завдяки прискоренню складських процесів, підприємство може розширити свою базу клієнтів та встановити співпрацю з більшою кількістю партнерів.

Постійне відстеження та контроль над робочими процесами дозволяє виявити проблемні моменти на складі та ефективно їх усунути. Система надасть інформацію про потребу у персоналі та можливість оптимізації штату. Вона також надасть змогу оцінити продуктивність кожного працівника окремо. Більш того, це спростить контроль за ключовими показниками ефективності бізнесу, встановлення цілей для команди та моніторинг їх досягнень.

Принцип роботи системи простий (рисунок 1.4):

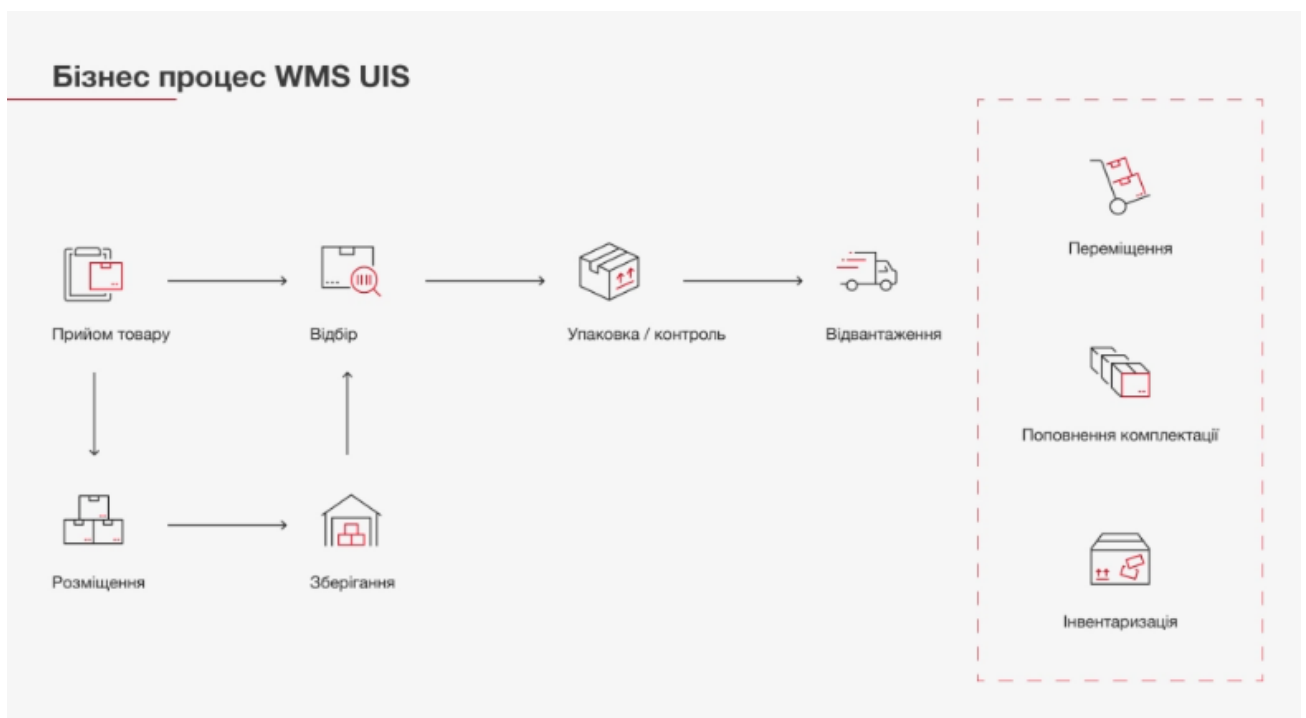


Рисунок 1.4 – Принцип роботи системи WMS

– при надходженні товару на склад, працівники одразу ідентифікують його, наприклад, за допомогою штрих-коду. Після отримання інформації про надходження товарів, система розпочинає автоматизований процес;

– працівники, які працюють з програмним забезпеченням, отримують вказівки від системи щодо розміщення товару, використання обладнання та

послідовності дій. Ці працівники можуть виконувати вказівки самостійно або делегувати їх колегам, які відповідають за відповідні процеси;

– за необхідності, система управління складом надає завдання спеціалістам, які відповідають за зберігання, пакування та відправлення товару.

Всі відомості про здійснені операції автоматично збираються у звіт, який потім доступний для перегляду компетентним спеціалістам, керівнику складу або власнику компанії [9].

#### 1.4 Підсумок аналізу технічного завдання

Проаналізувавши інформацію було виявлено що використання ІІІ в складських операціях може призвести до збільшення підприємства та зменшення часу на виробництво.

Керування складськими операціями за допомогою ІІІ призведе до зменшення помилок із-за людського фактору, зменшення часу на пошук та переміщення товару, збільшення безпеки про операціях, швидкий та точний аналіз великого обсягу даних підприємства.

За допомогою роботизованих систем вдається зменшити: час виробництва, потребу в ручній праці та ризик виникнення помилок, що теж впливає на збільшення підприємства та зменшення часу на виробництво.

Також було проаналізовані схожі технічні рішення, а саме системи управління складом (WMS).

Виходячи з цього основними функціями інтелектуального керування складськими операціями можна вважати:

- прогнозування попиту;
- оптимізація розміщення товарів на складі;
- автоматизовані планування та розподіл ресурсів;
- моніторинг та аналіз ефективності;
- оптимізація процесів відвантаження та приймання товарів;

- управління та оптимізація запасів;
- автоматичне планування маршрутів та переміщення товарів;
- використання аналітики та штучного інтелекту.

## 2 ІНТЕЛЕКТУАЛЬНЕ КЕРУВАННЯ СКЛАДСЬКИМИ ОПЕРАЦІЯМИ

### 2.1 Інтелектуальне керування складськими операціями

Склад – один із найважливіший компонентів оскільки логістична операційна ефективність визначається операційною ефективністю складу. Зазвичай вважалось що це приміщення в якому зберігаються запаси перед їхньою відправкою до покупця або торговця для продажу, однак із впровадженням технології “Індустрії 4.0” роль складу кардинально змінилася. З’явилася концепція “Розумного складу” що дозволила розширити сферу складської діяльності за допомоги підвищеної автоматизації складських функцій [10].

Індустрія 4.0, також відома як четверта промислова революція, належить до поточного періоду промислового розвитку, що характеризується впровадженням інтелектуальних технологій в усі аспекти виробничих процесів. Основними характеристиками Індустрії 4.0, що визначають сучасні виробничі процеси та технології являються: інтернет речей (IoT), RFID, цифрові (кіберфізичні) системи (CPS), великі дані (Big Data), штучний інтелект (AI) та машинне навчання (ML), роботизація і т.д. [11, 12].

Інтернет речей (IoT) передбачає підключення фізичних об’єктів до інтернету для збору та обміну даними. Це включає використання засобів ідентифікації, вимірювання, передавання та оброблення даних, а також виконавчих пристроїв для створення інтелектуальних систем [13].

RFID (Radio Frequency Identification) – технологія автоматичної ідентифікації та збору даних, яка використовує радіохвилі для зчитування і запису інформації на мікрочипах, які вбудовані в спеціальні мітки або картки. Ця технологія широко застосовується в різних галузях виробництва, логістиці, роздрібному продажі, бібліотеках а також охорона здоров'я [14].

Кіберфізичні системи (CPS) поєднують фізичні об'єкти з цифровими системами, створюючи інтерактивні середовища, де фізичні процеси контролюються і керуються через комп'ютерні системи [15].

Великі дані (Big Data) – це позначення сструктурованих та неструктурованих даних що охоплюють величезні обсяги та зростають експоненціально. Аналітика великих даних передбачає використання передових методів масово-паралельної обробки невизначено-структурованих даних, такі як СУБД NoSQL, алгоритми MapReduce і засоби проєкту Hadoop [16].

Штучний інтелект включає в себе створення систем, здатних правильно інтерпретувати зовнішні дані, аналізувати їх та використовувати для досягнення конкретних завдань за допомогою гнучкої адаптації. Машинне навчання є підмножиною AI, що дозволяє комп'ютерним системам покращувати свої здібності за допомогою проаналізованих даних та отриманого досвіду [17].

Роботизація передбачає використання роботів для виконання різних фізичних завдань в умовах спільної роботи з людьми, що матиме ключове значення для оптимізації індустрії та позбавлення працівників від виконання монотонних і небезпечних завдань [18].

Розумний склад – це склад, який створений максимально ефективним і включає в себе передовий досвід і складні технології, в результаті чого може працювати з максимальною ефективністю на постійно мінливому ринку. Комплектування, облік і доставка на такому складі мають бути автоматизовані, безпаперові та безпілотні. Склади також можуть бути перетворені на розумні за допомогою використання кіберфізичних систем (CPS), які дозволяють створювати віртуальні уявлення промислових операцій. Роботи, люди, пристрої CPS та інвентар – це чотири елементи інтелектуальних складів на основі CPS. Надійна локалізація предметів, ефективне планування зв'язку, ідентифікація людської діяльності та спільна робота декількох роботів вважаються основними труднощами додатків “Розумного складу” [10].

## 2.2 Економічна доцільність та ROI

Для будь якого підприємства впровадження інтелектуальної системи керування складськими операціями (WMS) є значною інвестицією. Вартість впровадження включає в себе сукупні витрати, що включають в себе первинні та поточні витрати, а також сукупні вигоди за допомогою яких потім буде розрахований ROI (Return on Investment, з англ. Окупність інвестицій).

### 2.2.1 Первинні витрати

Первинні витрати – це витрати які компанія несе на початкових етапах впровадження системи керування складом, це включає в себе придбання ліцензії модулів та додаткових функції на програмне забезпечення, обладнання, інтеграції з існуючими системами та навчання персоналу [19-22].

Вартість ліцензії на програмне забезпечення може варіюватися від 10,000 доларів до 100,000 доларів в залежності від функціоналу та масштабу системи [19], а додаткові модулі, наприклад модуль керування запасами чи інтеграція з поставниками коштують від 2,000 доларів до 10,000 доларів за модуль [20].

Перелічуючи первинні витрати під обладнанням вважалось наступне:

- сервери;
- сканери QR-кодів та RFID мітки;
- спеціальні прилади для збирання даних.

Вартість серверів для обробки даних коштують від 5,000 доларів до 20,000 доларів, сканери QR-кодів від 100 доларів до 500 доларів, RFID мітки від 0,10 центів до 1 долара за штуку та спеціальні прилади для персоналу від 300 доларів до 1,500 доларів [21].

Інтеграції з існуючими системами керування такими як ERP (Enterprise Resource Planning, з англ. Планування ресурсів підприємства) та CRM (Customer Relationship Management, з англ. керування відносинами з

клієнтами) може коштувати: ERP-системи від 5,000 доларів до 50,000 доларів, CRM-системи від 3,000 доларів до 20,000 доларів [22].

Навчання персоналу, а саме організація навчання користувачів та ІТ-персоналу, коштує від 2,000 доларів до 10,000 грн [19].

Виходячи з цього можна скласти таблицю первинних витрат із значеннями ( $\alpha$ ) всіх перелічених вимог та їхньою вартістю (таблиця 2.1)

Таблиця 2.1 – Первинні витрати

Значення ( $\alpha$ ) \ Вартість (\$)	Мінімальна	Максимальна
Ліцензія	10 000	100 000
Додаткові модулі	2 000	10 000
Сервери	5 000	10 000
Сканери QR-кодів	100	500
RFID мітки	0,10	1
Спеціальні прилади	300	1 500
ERP-системи	5 000	50 000
CRM-системи	3 000	20 000
Навчання персоналу	2 000	10 000

### 2.2.2 Поточні витрати

Поточні витрати – це витрати які компанія несе для постійного обслуговування та експлуатацію системи після її впровадження. Ці витрати включають в себе підтримку та обслуговування, заробітну плату персоналу, оновлення та модернізація [19-22].

Витрати на підтримку ПЗ включають оновлення програмного забезпечення, тех-підтримку та обслуговування обладнання, що коштуватиме 15-20% від початкової вартості ліцензії на рік. Включаючи вартість ліцензії, що вже була описана в первинних витратах, а саме від 10,000 доларів до

100,000 доларів, вартість підтримки ПЗ буде складати від 1,500-2,000 доларів до 15,000-20,000 доларів на рік [20].

Оновлення ПЗ яке включає в себе встановлення нових версій, виправлення помилок і додавання нових функцій коштуватиме від 2,000 доларів до 5,000 доларів на рік [20].

Заробітна плата ІТ-спеціалістам для обслуговування та підтримки системи може складати від 30,000 доларів до 60,000 доларів на рік. У залежності від розміру складу і складності системи може знадобитися від одного до кількох фахівців [21].

Оновлення та модернізація, яка включає в себе заміну застарілого обладнання, впровадження нових технологій і адаптацію до змін у процесах, коштуватиме від 5,000 доларів до 15,000 доларів на рік [21].

Виходячи з цього можна скласти таблицю поточних витрат із значеннями ( $\beta$ ) всіх перелічених вимог та їхньою вартістю (таблиця 2.2)

Таблиця 2.2 – Поточні витрати

Вартість (\$ на рік) Значення ( $\beta$ )	Мінімальна		Максимальна	
	Мінімальна	Максимальна	Мінімальна	Максимальна
Підтримка ПЗ	1 500	2 000	15 000	20 000
Оновлення ПЗ	2 000		5 000	
Заробітна плата	30 000		60 000	
Оновлення та модернізація	5 000		15 000	

### 2.2.3 Сукупні витрати

Дізнавшись про можливі первинні та поточні витрати можна скласти сукупні витрати для розрахунку ROI.

Сукупні витрати – це загальна сума всіх витрат, пов'язаних із впровадженням та експлуатацією WMS, тобто загальна сума первинних та поточних витрат [19-22]. Для розрахунку сукупних витрат необхідно скласти загальні первинні та поточні витрати за середній час використання системи (5 років).

Сукупні витрати загальних первинних витрат розраховується за допомогою формули (1):

$$\alpha_{\text{загальна}} = \alpha_1 + \alpha_2 + \dots + \alpha_{\text{кз.}}, \quad (1)$$

де  $\alpha_{\text{загальна}}$  – це значення загальних первинних витрат;

$\alpha_{\text{кз.}}$  – це кінцеве значення первинних витрат.

Використовуючи формулу (1) можна розрахувати значення загальних первинних мінімальних та максимальних витрат що будуть складати:

$$\alpha_{\text{загальна min}} = 10000 + 2000 + 5000 + 100 + 0,10 + 300 + 5000 + 3000 + 2000 = 27\,400,10 \text{ доларів,}$$

$$\alpha_{\text{загальна max}} = 100000 + 10000 + 10000 + 500 + 1 + 1500 + 50000 + 20000 + 10000 = 202\,001 \text{ доларів,}$$

де  $\alpha_{\text{загальна min}}$  – це значення загальних первинних мінімальних витрат;

$\alpha_{\text{загальна max}}$  – це значення загальних первинних максимальних витрат.

Сукупні витрати загальних поточних витрат розраховується за допомогою формули (2):

$$\beta_{\text{загальна}} = (\beta_1 + \beta_2 + \dots + \beta_{\text{кз.}}) \cdot t_{\text{вс.}}, \quad (2)$$

де  $\beta_{\text{загальна}}$  – це значення загальних поточних витрат;

$\beta_{\text{кз.}}$  – це кінцеве значення поточних витрат;

$t_{\text{вс.}}$  – це кількість років використання системи.

Використовуючи формулу (2) можна розрахувати значення загальних поточних мінімальних та максимальних витрат.

Якщо витрати на підтримку ПЗ складають 15%:

$$\beta_{\text{загальна min}} = (1500 + 2000 + 30000 + 5000) \cdot 5 = 192\,500 \text{ доларів на рік,}$$

$$\beta_{\text{загальна max}} = (15000 + 5000 + 60000 + 15000) \cdot 5 = 475\,000 \text{ доларів на рік,}$$

Якщо витрати на підтримку ПЗ складають 20%:

$$\beta_{\text{загальна min}} = (2000 + 2000 + 30000 + 5000) \cdot 5 = 195\,000 \text{ доларів на рік,}$$

$$\beta_{\text{загальна max}} = (20000 + 5000 + 60000 + 15000) \cdot 5 = 500\,000 \text{ доларів на рік.}$$

Зробивши розрахунки загальних первинних та поточних витрат можна розрахувати сукупні витрати ( $C_{\text{вит.}}$ ).

Якщо витрати на підтримку ПЗ складають 15%:

$$C_{\text{вит. min}} = \alpha_{\text{загальна min}} + \beta_{\text{загальна min}} = 27\,400,10 + 192\,500 = 219\,900,1 \text{ доларів,}$$

$$C_{\text{вит. max}} = \alpha_{\text{загальна max}} + \beta_{\text{загальна max}} = 202\,001 + 475\,000 = 677\,001 \text{ доларів.}$$

Якщо витрати на підтримку ПЗ складають 20%:

$$C_{\text{вит. min}} = \alpha_{\text{загальна min}} + \beta_{\text{загальна min}} = 27\,400,10 + 195\,000 = 222\,400,1 \text{ доларів,}$$

$$C_{\text{вит. max}} = \alpha_{\text{загальна max}} + \beta_{\text{загальна max}} = 202\,001 + 500\,000 = 702\,001 \text{ доларів.}$$

#### 2.2.4 Сукупні вигоди

Для подальшого розрахунку ROI, розрахувавши сукупні витрати, необхідно дізнатись про сукупні вигоди та провести їх розрахунок.

Сукупні вигоди – це загальна сума економії та додаткових доходів, які підприємство отримало в результаті впровадження WMS системи. Одними із основних речей сукупних виводів можна вважати економію витрат та підвищення доходів [19-22].

Економія витрат дозволяє: зменшити витрати на ручну працю за допомогою автоматизації процесів, що дозволяє зберегти 120 000 доларів на рік; оптимізувати використання складських площ, що дозволяє зменшити витрати на оренду або зберігання товарів та зберегти 40 000 доларів на рік, за рахунок кращого використання простору; зменшити надлишкові запаси за допомогою точного прогнозування попиту, що дозволяє зберегти 30 000 доларів на рік [19, 21, 22].

Підвищення доходів дозволяє: підвищити продаж за допомогою швидкого виконання замовлень та зниження кількості помилок, що підвищує задоволеність клієнтів та збільшує прибуток до 80 000 доларів на рік; адаптуватися до змін у попиті та впроваджувати нові товари, що дозволяє збільшити дохід на 50 000 доларів на рік [19, 21].

Дізнавшись про можливі сукупні вигоди можна провести розрахунок загальної економії витрат ( $E_{\text{заг.}}$ ) та загального підвищення доходів ( $P_{\text{заг.}}$ ) що складає:

$$E_{\text{заг.}} = 120\,000 + 40\,000 + 30\,000 = 190\,000 \text{ доларів на рік,}$$

$$P_{\text{заг.}} = 80\,000 + 50\,000 = 130\,000 \text{ доларів на рік.}$$

Виходячи з розрахунків сукупні вигоди ( $C_{\text{виг.}}$ ) становлять:

$$C_{\text{виг.}} = 190\,000 + 130\,000 = 320\,000 \text{ доларів на рік.}$$

### 2.2.5 Розрахунок ROI

Дізнавшись про сукупні витрати та вигода та розрахувавши їх можна приступити до розрахунку окупності інвестицій (ROI). Загальна формула окупності інвестицій виглядає наступним чином (формула (3) [19]):

$$\text{ROI} = \frac{C_{\text{виг.}} - C_{\text{вит.}}}{C_{\text{вит.}}} \cdot 100\%. \quad (3)$$

Перед розрахунком ROI розрахуємо  $C_{\text{виг.}}$  на 5 років, бо значення поточних витрат розраховувалось на середній час використання системи (5 років):

$$C_{\text{виг.}} = 320\,000 \cdot 5 = 1\,600\,000 \text{ доларів за 5 років.}$$

Тепер розрахувавши  $C_{\text{виг.}}$  на 5 років розрахуємо ROI.

Якщо витрати на підтримку ПЗ складають 15%:

$$\text{ROI min} = \frac{1\,600\,000 - 219\,900,1}{219\,900,1} \cdot 100\% \approx 627\%,$$

$$\text{ROI max} = \frac{1\,600\,000 - 677\,001}{677\,001} \cdot 100\% \approx 136\%.$$

Якщо витрати на підтримку ПЗ складають 20%:

$$\text{ROI min} = \frac{1\,600\,000 - 222\,400,1}{222\,400,1} \cdot 100\% \approx 619\%,$$

$$\text{ROI max} = \frac{1\,600\,000 - 702\,001}{702\,001} \cdot 100\% \approx 127\%.$$

Виходячи з розрахунків ROI, інвестиції для впровадження інтелектуальної системи керування складськими операціями (WMS) являється

прибутковою особливо для малих складів. Система WMS для малих складів дозволить повернути інвестиції швидше та багаторазово збільшити прибуток підприємства.

### 2.3 Приклади готового рішення

При аналізі готових рішень було обрано для прикладу дві компанії: українська приватна компанія з логістики “Нова пошта” та зарубіжна американська компанія електронної комерції “Amazon”. Обидві компанії використовують систему роботи складу під назвою фулфілмент.

#### 2.3.1 Система фулфілмент на “Новій пошті”

Система фулфілменту на “Новій пошті” включає комплекс послуг, спрямованих на обробку замовлень інтернет-магазинів та інших компаній, що займаються торгівлею. Система охоплює всі етапи логістичного ланцюжка – від прийому товарів на склад до їх доставки кінцевим покупцям.

Основні етапи роботи системи фулфілменту у Новій Пошті це:

- прийом та зберігання товарів;
- обробка замовлень;
- упаковка;
- доставка;
- обробка повернень.

Товари, які від інтернет-магазинів чи інших клієнтів, приймаються до складу Нової Пошти проходять перевірку на відповідність кількості та якості, зазначених у супровідних документах. Товари інвентаризуються та розміщуються на складі. Складські площі обладнані з урахуванням специфіки різних товарів, забезпечуючи їхнє безпечне та ефективне зберігання.

Як тільки клієнт інтернет-магазину оформляє замовлення, інформація про нього передається до системи фулфілменту Нової Пошти. З отриманого

замовлення співробітники складу підбирають потрібні товари, збирають з полиць та упаковують їх.

Товари упаковуються відповідно до вимог клієнта та типу товару. Забезпечені компанією різні варіанти упаковки дозволяють зберегти товар при транспортуванні. Запаковані замовлення маркуються та готуються до відправки перевіряючи правильність комплектації та пакування замовлення.

Готові замовлення надсилаються покупцям. Забезпечені компанією різні варіанти доставки включають: кур'єрську доставку до дверей, доставку в поштомати та пункти видачі. Клієнти можуть відстежувати статус своїх замовлень у режимі реального часу за допомогою системи відстеження на сайті або за допомогою мобільного додатку.

Якщо клієнт вирішує повернути товар, компанія приймає повернення і проводить їх обробку. Повернені товари перевіряються на відповідність умовам повернення та за необхідності повертаються на склад для повторного продажу.

Опираючись на основні етапи перевагами системи фулфілменту Нової Пошти можна вважати наступні висновки:

- “Нова Пошта” має широку мережу логістичних центрів та пунктів видачі по всій Україні, що забезпечує швидку та надійну доставку;

- використання фулфілмент-послуг дозволяє інтернет-магазинам та іншим клієнтам скоротити витрати на утримання власного складу та логістики;

- Нова Пошта пропонує різні варіанти доставки та упаковки, а також можливість відстеження замовлень, що робить процес зручним як для продавців, так і для покупців;

- система фулфілменту Нової Пошти легко інтегрується з різними платформами інтернет-торгівлі, що полегшує процес управління замовленнями та логістикою.

Таким чином, система фулфілменту Нової Пошти надає комплексне рішення для управління логістикою та доставкою товарів, що дозволяє бізнесам зосередитись на розвитку та обслуговуванні клієнтів [23].

### 2.3.2 Система Fulfilment by Amazon

Система фулфілменту на Amazon, відома як “Fulfillment by Amazon” (FBA), є комплексним набором послуг, який “Amazon” надає продавцям для управління їхніми товарами, складськими операціями та доставкою. Ця система охоплює всі етапи обробки замовлень, від прийому товарів складу до їх доставки кінцевим покупцям.

Основними етапами роботи системи фулфілменту на “Amazon” являються:

- прийом та зберігання товарів;
- обробка замовлень;
- упаковка;
- доставка;
- обробка повернень.

Продавці відправляють свої товари на склади “Amazon”. При відправці вони повинні дотримуватися інструкцій з упаковки та маркування товарів, щоб вони відповідали вимогам “Amazon”.

Після прибуття до складу товари проходять процес перевірки та інвентаризації. Це включає сканування штрих-кодів та перевірку відповідності товарів заявленим характеристикам.

Товари розміщуються на складі “Amazon” використовуючи просунуті системи управління складом (WMS), щоб оптимізувати розміщення товарів та забезпечити їх швидку доступність для обробки замовлень.

Як тільки клієнт робить замовлення на “Amazon”, система автоматично обробляє його та передає інформацію на відповідний склад. Працівники складу підбирають товари, вказані у замовленні, використовуючи системи автоматизованого пошуку та сортування.

Товари упаковуються у коробки з логотипом “Amazon” відповідно до вимог компанії, щоб забезпечити збереження товарів під час транспортування. Запаковані замовлення маркуються та готуються до відправлення. Це включає прикріплення етикеток з адресою доставки та перевірку правильності комплектації замовлення.

“Amazon” організує доставку замовлень кінцевим покупцям використовуючи різних логістичних партнерів та власні транспортні ресурси для забезпечення швидкої та надійної доставки.

Покупці можуть відстежувати статус своїх замовлень у режимі реального часу через веб-сайт “Amazon” або мобільний додаток. Якщо клієнт вирішує повернути товар, компанія приймає повернення і проводить їх обробку. Повернені товари перевіряються на відповідність умовам повернення та, якщо вони відповідають стандартам, повертаються на склад для повторного продажу або утилізуються, якщо це необхідно.

Виходячі з етапів роботи системи фулфілменту на “Amazon” можна зробити наступні висновки переваг цієї системи:

- “Amazon” має велику мережу складів по всьому світу, що дозволяє забезпечувати швидку доставку до будь-яких регіонів. Особливо це важливо для продавців, які хочуть охопити міжнародні ринки;

- продавці можуть зосередитися на розвитку свого бізнесу, не турбуючись про складування, пакування та доставку товарів;

- замовлення, виконані через FBA, часто отримують пріоритетне розміщення на “Amazon”, що підвищує їхню видимість та ймовірність покупки. Крім того, клієнти довіряють бренду “Amazon” та його стандартам обслуговування;

- товари, що зберігаються на складах “Amazon”, можуть бути доступні для учасників програми “Amazon Prime”, що гарантує швидшу доставку та може збільшити продаж.

Таким чином система фулфілменту “Amazon” (FBA) надає продавцям комплексне рішення для керування своїми товарами, від прийому та зберігання

до доставки та обробки повернень. Це дозволяє продавцям ефективно масштабувати свій бізнес і зосередитись на його розвитку, в той час як “Amazon” бере на себе всі аспекти логістики та обслуговування клієнтів [24].

#### 2.4 Вибір програмного середовища для реалізації проекту

При виборі програмного середовища для реалізації проекту були враховані кілька ключових факторів: кросплатформеність, можливість використання сучасних мов програмування, підтримка широкого спектру інтерфейсних елементів та активність спільноти розробників. Після детального аналізу усіх можливих варіантів було прийнято рішення використовувати кросплатформенний фреймворк “AvaloniaUI” в програмному середовищі “Visual Studio”.

AvaloniaUI – це кросплатформенний фреймворк для побудови графічних інтерфейсів користувача (GUI), який дозволяє розробникам створювати додатки, що працюють на різних операційних системах, таких як Windows, macOS, Linux і навіть на Android та IOS. AvaloniaUI базується на мовах C# та XAML, що робить його привабливим вибором для розробників знайомих з цими технологіями, особливо для тих хто працював з WPF [25].

Visual Studio – це інтегрована середа розробки (IDE), створена компанією Microsoft. Воно є потужним інструментом для розробки програмного забезпечення, підтримуючи широкий спектр мов програмування, включаючи C# та XAML, які використовуються в AvaloniaUI [26].

Перевагами даного фреймворку можна вважати наступні можливості:

- кросплатформеність;
- використання сучасних мов програмування;
- підтримка MVVM;
- активна спільнота та документація;
- гнучкість та простоту при використанні.

Рішення використовувати AvaloniaUI було прийнято на основі його відповідності ключовим вимогам проекту, а саме:

- “AvaloniaUI” дозволяє розроблювати додатки для різних платформ з використанням єдиної кодової бази, що значно знижує витрати на розробку і забезпечує послідовний користувацький досвід на різних пристроях;

- завдяки використанню C# та XAML, додатки на AvaloniaUI відрізняються високою продуктивністю та стабільністю;

- активна спільнота розробників і наявність великої кількості ресурсів та прикладів сприяють швидкому вирішенню проблем і впровадженню нових функцій.

На основі цих факторів було вирішено використовувати AvaloniaUI як основний фреймворк для реалізації проекту. Це рішення забезпечить високу якість кінцевого продукту, зручність в розробці та підтримці, а також широку доступність для користувачів різних платформ.

В якості БД було обрано Microsoft SQL Server.

Microsoft SQL Server – це реляційна система управління базами даних (RDBMS – Relational database management system), розроблена компанією Microsoft. Вона призначена для зберігання, управління та аналізу даних, і використовується в різних додатках, від малих до великих корпоративних рішень [27].

## **3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМИ ІНТЕЛЕКТУАЛЬНОГО КЕРУВАННЯ СКЛАДСЬКИМИ ОПЕРАЦІЯМИ**

### **3.1 Неперервна лінійна САУ**

Теорія автоматичного управління – це наука про принципи побудови та методи розрахунку систем автоматичного керування методи розрахунку систем автоматичного керування. Її висновки справедливі для різних систем незалежно від призначення та фізичної природи. Застосування систем, їх проектування та експлуатація неможливі без знань теорії автоматичного керування [28].

ТАУ при вивченні процесів управління абстрагується від фізичних і конструктивних особливостей систем і замість реальних систем розглядає їх математичні моделі. Основними методами дослідження в ТАУ є математичне моделювання, теорія звичайних диференціальних рівнянь, операційне обчислення і гармонійний аналіз [28].

Завдання теорії автоматичного управління:

а) синтез систем автоматичного управління. За допомогою методів синтезу можна здійснювати вибір схеми взаємодії елементів САУ, параметрів і характеристик цих елементів так, щоб система в цілому задовольняла заданим вимогам;

б) аналіз систем автоматичного управління. Методи аналізу дозволяють визначити якісні показники САУ, а у разі їх відмінності від заданих – вказати шляхи поліпшення статичних і динамічних властивостей системи;

в) корекція систем автоматичного управління. За допомогою корекції здійснюється зміна статичних і динамічних властивостей системи;

г) експериментальне дослідження і налагодження. Методи експериментального дослідження і налагодження САУ дають можливість

найраціональніше і оптимально дослідити систему в реальних умовах роботи [28].

У структурному аспекті будь-яку систему автоматичного управління можна представити взаємозв'язаною сукупністю об'єкту управління і пристрою управління (рисунок 3.1).

Функція стану об'єкту управління:

$$x(t) = X\{u(t), f(t), x_0(t)\}, \quad (3)$$

де  $u(t)$  – функція управління;

$f(t)$  – функція обурення;

$x_0(t)$  – початкові умови [28].



Рисунок 3.1 – Структурна схема системи автоматичного управління [28]

### 3.2 Опис алгоритмів програми

Алгоритми програми для інтелектуального керування складськими операціями орієнтовані на автоматизацію та оптимізацію процесів, пов'язаних з обробкою товарів на складі, зберіганням, а також здійсненням замовлень.

Проаналізувавши технічне завдання були розроблені такі алгоритми як: авторизація користувачів (ДОДАТОК А), оновлення графіків складу (ДОДАТОК Б), керування замовленнями, як для зовнішніх замовлень (ДОДАТОК В) так і для потреб виробництва (ДОДАТОК Г), сканер QR-коду для інвентаризації (ДОДАТОК Д) та видалення товарів (ДОДАТОК Е), генератор QR-кодів (ДОДАТОК Ж), для імітації QR-коду на товарі, генерація звітів та аналіз даних (ДОДАТОК И).

Алгоритм авторизації користувачів включає перевірку введених логіна і пароля користувача в БД (рисунок 3.2, рисунок 3.3).

```
//Autorization
_context = new AppDBContext();
LoginCommand = ReactiveCommand.CreateFromTask(LoginAsync);
```

Рисунок 3.2 – Ініціалізацію контексту БД і створення команди для асинхронної операції входу

```
//Autorization
Ссылка 1
private async Task LoginAsync()
{
    var user = await _context.UsersChart.FirstOrDefaultAsync(u => u.Login == Login && u.Password == Password);
```

Рисунок 3.3 – Строка для виконання асинхронного запиту до БД для пошуку користувача

У разі успішного входу програма змінює стан інтерфейсу, дозволяючи доступ до основних функцій. Якщо введені дані невірні, система відображає повідомлення про помилку.

Алгоритм оновлення даних на графіках складу передбачає завантаження даних із бази даних, фільтрацію їх за типом матеріалу та оновлення інтерфейсу користувача для відображення актуальної інформації про стан складу (рисунок 3.4, рисунок 3.5, рисунок 3.6).

```

// Chart
Ссылка: 12
public async void UpdateMainChart()
{
    await LoadDataFromDatabaseAsync();
    await FirstRackFilterData();
    await SecondRackFilterData();
    await ThirdRackFilterData();
    await FourthRackFilterData();
    UpdateOrderCollection();
    UpdateProductionCollection();
}

```

Рисунок 3.4 – Метод для оновлення таблиць програми

Строка коду “await LoadDataFromDatabaseAsync();” завантажує дані з бази даних і оновлює головну колекцію Main.

Строка коду “await FirstRackFilterData();” і аналогічні методи застосовує фільтри для завантажених даних за типом матеріалу.

Строки коду “UpdateOrderCollection();” та “UpdateProductionCollection();” оновлюють колекції замовлень і виробництва на основі завантажених і відфільтрованих даних.

```

Ссылка: 1
public async Task LoadDataFromDatabaseAsync()
{
    var main = await GetInfoFromDatabaseAsync();
    await Dispatcher.UIThread.InvokeAsync(() =>
    {
        Main.Clear();
        Main.AddRange(main);
    });
}

```

Рисунок 3.5 – Метод для завантаження даних з БД

Строка коду “var main = await GetInfoFromDatabaseAsync();” асинхронно отримує дані з бази даних.

Строки коду “await Dispatcher.UIThread.InvokeAsync(() => { Main.Clear(); Main.AddRange(main); });” оновлюють колекцію Main у користувацькому інтерфейсі, очищуючи старі дані та додаючи нові.

```

// First Rack Chart
Ссылка: 1
private async Task FirstRackFilterData()
{
    var filteredItems = Main.Where(item => item.MaterialType == "Plate").ToList();
    foreach (var item in filteredItems)
    {
        item.Rack = 1;
    }
    FirstFilteredMain = new ObservableCollection<MainChart>(filteredItems);
    await UpdateDatabaseRackValues(filteredItems);
}

```

Рисунок 3.6 – Метод для фільтрації даних за типом матеріалу

Метод “FirstRackFilterData” фільтрує дані з колекції “Main” на основі типу матеріалу, оновлює значення стійки для відфільтрованих елементів, зберігає їх у колекцію “FirstFilteredMain” і оновлює значення стійки в базі даних.

Алгоритм додавання зовнішніх замовлень передбачає перевірку наявності товару на складі (рисунок 3.7), зменшення кількості товару в разі підтвердження замовлення (рисунок 3.8), а також оновлення відповідних графіків (рисунок 3.9).

```

var mainChart = context.MainChart.FirstOrDefault(m => m.Name == OName);
if (mainChart == null)
{
    await Dispatcher.UIThread.InvokeAsync(() =>
    {
        StatusBar = "Name not found in main chart.";
    });
    return;
}

```

Рисунок 3.7 – Код для пошуку даних за назвою матеріалу

В строках цього коду виконується пошук першого запису в таблиці “MainChart”, де ім'я “Name” збігається з “OName”. Якщо запис не знайдено, оновлюється рядок стану “StatusBar” і метод завершує виконання.

```

string orderStatus;
if (mainChart.Quantity < OQuantity)
{
    orderStatus = "In process";
    await Dispatcher.UIThread.InvokeAsync(() =>
    {
        StatusBar = "Insufficient products in stock.";
    });
}
else
{
    mainChart.Quantity -= OQuantity;
    orderStatus = "Complete";
    await Dispatcher.UIThread.InvokeAsync(() =>
    {
        StatusBar = "Order successfully added.";
    });
}
}

```

Рисунок 3.8 – Код для перевірки кількості товару, та зменшення в разі підтвердження замовлення

В даному коді перевіряється, чи достатньо товарів у наявності. Якщо ні, встановлюється статус замовлення як “In process” і оновлюється рядок стану. Якщо товарів достатньо, кількість зменшується на “OQuantity”, статус замовлення встановлюється як “Complete”, і рядок стану оновлюється.

```

await Dispatcher.UIThread.InvokeAsync(() =>
{
    Order.Add(newOrder);
    CheckOrderLinesVisibility(newOrder);
});

var mainItem = Main.FirstOrDefault(m => m.Id == mainChart.Id);
if (mainItem != null)
{
    await Dispatcher.UIThread.InvokeAsync(() =>
    {
        mainItem.Quantity = mainChart.Quantity;
    });
}
}

```

Рисунок 3.9 – Код для оновлення відповідних графіків

Строки цього коду оновлюють інтерфейс користувача шляхом додавання нового замовлення в колекцію “Order” і перевірки видимості рядків замовлень. Також оновлюється кількість товарів в основному елементі, якщо він знайдений.

Алгоритм додавання виробничих замовлень схожий на алгоритм зовнішніх замовлень і передбачає перевірку наявності товару (рисунок 3.10), зменшення кількості товару на складі (рисунок 3.11), оновлення графіків виробництва і складу (рисунок 3.12).

```
var mainChart = context.MainChart.FirstOrDefault(m => m.Name == PName);
if (mainChart == null)
{
    await Dispatcher.UIThread.InvokeAsync(() =>
    {
        StatusBar = "Name not found in main chart.";
    });
    return;
}
```

Рисунок 3.10 – Код для пошуку даних за назвою матеріалу

В цих строках коду виконується пошук першого запису в таблиці “MainChart”, де ім'я “Name” збігається з “PName”. Якщо запис не знайдено, оновлюється рядок стану “StatusBar” і метод завершує виконання.

```
string productionStatus;
if (mainChart.Quantity < PQuantity)
{
    productionStatus = "In process";
    await Dispatcher.UIThread.InvokeAsync(() =>
    {
        StatusBar = "Insufficient products in stock.";
    });
}
else
{
    mainChart.Quantity -= PQuantity;
    productionStatus = "Complete";
    await Dispatcher.UIThread.InvokeAsync(() =>
    {
        StatusBar = "Production successfully added.";
    });
}
```

Рисунок 3.11 – Код для перевірки кількості товару, та зменшення в разі підтвердження замовлення

В даних строках коду перевіряється, чи достатньо товарів у наявності. Якщо ні, встановлюється статус виробництва як “In process” і оновлюється рядок стану. Якщо товарів достатньо, кількість зменшується на “PQuantity”, статус виробництва встановлюється як “Complete”, і рядок стану оновлюється.

```
await Dispatcher.UIThread.InvokeAsync(() =>
{
    Production.Add(newProduction);
    CheckProductionLinesVisibility(newProduction);
});

var mainItem = Main.FirstOrDefault(m => m.Id == mainChart.Id);
if (mainItem != null)
{
    await Dispatcher.UIThread.InvokeAsync(() =>
    {
        mainItem.Quantity = mainChart.Quantity;
    });
}
```

Рисунок 3.12 – Код для оновлення відповідних графіків

Строки цього коду оновлюють інтерфейс користувача шляхом додавання нового замовлення в колекцію “Production” і перевірки видимості рядків замовлень. Також оновлюється кількість товарів в основному елементі, якщо він знайдений.

Алгоритм роботи з камерами охоплює ініціалізацію камер (рисунок 3.13), розпізнавання QR-кодів (рисунок 3.14), оновлення кількості товарів на складі (рисунок 3.15) та відображення зображень з камер в інтерфейсі користувача (рисунок 3.16).

```

//Camera In
Ссылка: 1
private void InitializeINCamera()
{
    if (_captureIN != null)
    {
        _captureIN.Release();
        _captureIN.Dispose();
        _captureIN = null;
    }

    _captureIN = new VideoCapture(0);
    if (!_captureIN.IsOpened())
    {
        Dispatcher.UIThread.InvokeAsync(() =>
        {
            StatusBar = "Failed to initialize 'IN' camera.";
        }).Wait();
    }
}

```

Рисунок 3.13 – Код ініціалізації камери

Даний код виконує декілька дій, а саме: перевіряє, чи був об'єкт “\_captureIN” раніше ініціалізований (чи не дорівнює він null). Якщо об'єкт “\_captureIN” було ініціалізовано, виконується звільнення ресурсів:

- “Release()” звільняє камеру, якщо вона захоплена;
- “Dispose()” видаляє об'єкт, звільняючи керовані та некеровані ресурси.

Встановлює “\_captureIN” у “null” для запобігання повторному звільненню того самого ресурсу.

Створює новий об'єкт “VideoCapture”, який захоплює відео з камери.

Перевіряє, чи успішно відкрито камеру за допомогою методу “IsOpened()”.

```

Point2f[] points;
var result = _qrCodeINDetector.DetectAndDecode(mat, out points);

if (result != null)
{
    string decodedText = result;
    if (!string.IsNullOrEmpty(decodedText))
    {
        var parts = decodedText.Split(':');
        if (parts.Length == 2)
        {
            var productName = parts[0];
            var materialType = parts[1];
            IncrementProductQuantity(productName, materialType);
            Dispatcher.UIThread.InvokeAsync(() =>
            {
                StatusBar = $"Detected QR code: {decodedText}";
            }).Wait();
        }
        else
        {
            Dispatcher.UIThread.InvokeAsync(() =>
            {
                StatusBar = $"Invalid QR code format: {decodedText}";
            }).Wait();
        }
    }

    Thread.Sleep(2000);
}
}

```

Рисунок 3.14 – Код для розпізнавання QR-кодів

В цьому коді відбувається наступне:

- декларується масив точок і викликається метод для розпізнавання та декодування QR-коду, після чого йде перевірка, чи був розпізнаний QR-код;
- після перевірки йде отримання декодованого тексту і перевірка його вмісту;
- декодований текст розділяється на частини, після чого йде оновлення кількості товару і користувацького інтерфейсу;
- далі йде обробка некоректного формату QR-коду і пауза для запобігання багаторазовому зчитуванню одного і того ж QR-коду.

```

Ссылка 1
private void IncrementProductQuantity(string productName, string materialType)
{
    using var context = new AppDBContext();
    var item = context.MainChart.FirstOrDefault(x => x.Name == productName && x.MaterialType == materialType);

    if (item != null)
    {
        item.Quantity += 1;
        context.SaveChanges();
    }
}

```

Рисунок 3.15 – Початок коду для оновлення кількості товарів на складі

Метод “IncrementProductQuantity” виконує наступні дії.

Перевіряє, чи існує елемент у базі даних із зазначеними ім'ям продукту і типом матеріалу. Якщо елемент існує, збільшує його кількість на одиницю і оновлює користувацький інтерфейс. Якщо елемент не існує, створює новий елемент із заданими параметрами і встановлює кількість рівною одиниці.

В обох випадках оновлює рядок стану і видимість рядків стійок у користувацькому інтерфейсі. Викликає методи для оновлення замовлень і виробництва залежно від зміненої кількості товару. Зберігає зміни в базі даних і оновлює пов'язані колекції даних у користувацькому інтерфейсі.

```

Ссылка 1
public async Task StartINCameraAsync()
{
    await Task.Run(() => InitializeINCamera());

    if (!_captureIN.IsOpened())
    {
        await Dispatcher.UIThread.InvokeAsync(() =>
        {
            StatusBar = "Failed to open 'IN' camera.";
        });
        return;
    }

    _isCameraINRunning = true;
    await Dispatcher.UIThread.InvokeAsync(() =>
    {
        StatusBar = "'IN' camera started.";
    });
}

```

Рисунок 3.16 – Початок коду для відображення зображень з камер в інтерфейсі користувача

В даному коді відбувається наступне:

– ініціалізація камери в окремому потоці для запобігання блокування

користувачького інтерфейсу;

- перевірка, чи успішно відкрито камеру, і оновлення рядку стану в разі помилки;

- встановлення прапору, що вказує на успішний запуск камери, і оновлення рядку стану.

Алгоритм генерації QR-кодів охоплює перетворення введених текстових даних на QR-код (рисунок 3.17), створення зображення QR-коду (рисунок 3.18) та збереження його у файл (рисунок 3.19).

```
//QR-code
Ссылка: 1
public void GenerateQRCode()
{
    if (string.IsNullOrEmpty(Text1) || string.IsNullOrEmpty(Text2))
    {
        StatusBar = "One or both text fields are null or empty.";
        return;
    }

    var textToEncode = $"{Text1}:{Text2}";
    var writer = new BarcodeWriterPixelData
    {
        Format = BarcodeFormat.QR_CODE,
        Options = new EncodingOptions
        {
            Height = 200,
            Width = 200
        }
    };

    try
    {
        var pixelData = writer.Write(textToEncode);
    }
}
```

Рисунок 3.17 – Початок коду для перетворення введених текстових даних на QR-код

Метод “GenerateQRCode” генерує QR-код на основі рядка “textToEncode” і зберігає у змінну “pixelData”. Далі перетворює “pixelData” в об’єкт “Bitmap” за допомогою методу “CreateBitmapFromPixelData”. Якщо перетворення не вдалося (повернуто null), метод оновлює рядок стану з

повідомленням про помилку і завершує виконання. Якщо створення зображення успішне, воно зберігається у властивість “QRCode”.

QR-код зберігається у файл за допомогою методу “SaveQRCodeToFile”. Якщо в процесі генерації або збереження сталася помилка, вона перехоплюється і відображається в рядку стану.

```

Ссылка 1
public Bitmap CreateBitmapFromPixelData(PixelData pixelData)
{
    try
    {
        using (var writeableBitmap = new WriteableBitmap(
            new PixelSize(pixelData.Width, pixelData.Height),
            new Vector(96, 96), // DPI
            PixelFormat.Bgra8888,
            AlphaFormat.Premul))
        {
            using (var lockedBuffer = writeableBitmap.Lock())
            {
                System.Runtime.InteropServices.Marshal.Copy(pixelData.Pixels, 0, lockedBuffer.Address, pixelData.Pixels.Length);
            }
        }
        using (var stream = new MemoryStream())
    }
}

```

Рисунок 3.18 – Початок коду для створення зображення QR-коду

Метод “CreateBitmapFromPixelData” створює “WriteableBitmap” на основі даних пікселів (розміри, формат, DPI). Далі копіює дані пікселів у “WriteableBitmap” і зберігає вміст “WriteableBitmap” у потік пам'яті. Після збереження створює і повертає об'єкт “Bitmap” на основі даних із потоку пам'яті та обробляє можливі винятки, оновлюючи рядок стану і повертаючи null у разі помилки.

```

Ссылка 1
public void SaveQRCodeToFile(PixelData pixelData, string textToEncode)
{
    string folderPath = "D:\\all\\study\\VS\\Diplom directorys\\QRcode";
    if (!Directory.Exists(folderPath))
    {
        Directory.CreateDirectory(folderPath);
    }

    string sanitizedText = string.Concat(textToEncode.Split(Path.GetInvalidFileNameChars()));
    string fileName = $"{sanitizedText}.jpg";
    string filePath = Path.Combine(folderPath, fileName);

    try
    {
        using (var writeableBitmap = new WriteableBitmap(

```

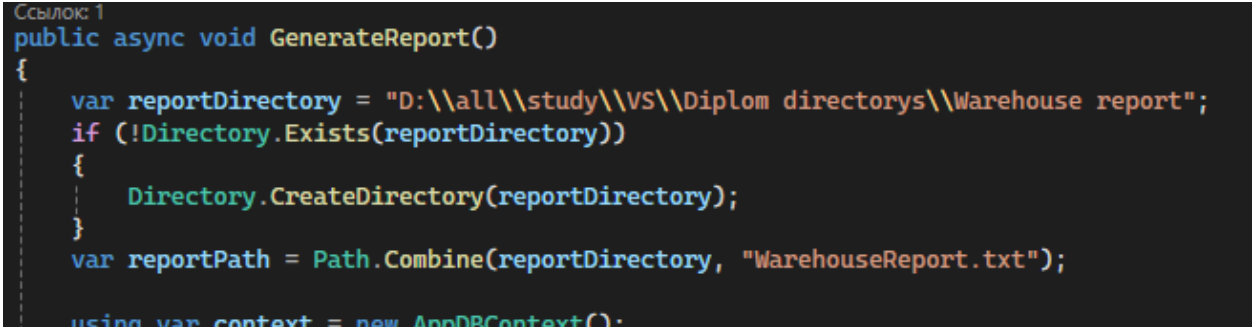
Рисунок 3.19 – Початок коду для збереження QR-коду у файл

Метод “SaveQRCodeToFile” визначає шлях до папки для збереження QR-кодів і створює її, якщо вона не існує. Далі формує коректне ім'я файлу на основі введеного тексту.

Створює об'єкт “WriteableBitmap” і копіює в нього дані пікселів, після чого зберігає “WriteableBitmap” у файл.

Оновлює рядок стану з повідомленням про успішне збереження або про помилку.

Алгоритм генерації звітів дає змогу створити текстовий звіт про стан складу (рисунок 3.20), збережений у файл, а також здійснювати аналіз даних для визначення загальної кількості товарів і різноманітності товарних найменувань (рисунок 3.21).



```

Ссылка 1
public async void GenerateReport()
{
    var reportDirectory = "D:\\all\\study\\VS\\Diplom directories\\Warehouse report";
    if (!Directory.Exists(reportDirectory))
    {
        Directory.CreateDirectory(reportDirectory);
    }
    var reportPath = Path.Combine(reportDirectory, "WarehouseReport.txt");
    using var context = new AppDBContext();

```

Рисунок 3.20 – Початок коду для створення текстового звіту про стан складу

Метод “GenerateReport” перевіряє наявність директорії для звіту і створює її, якщо вона не існує, після чого визначає повний шлях до файлу звіту.

Створює контекст бази даних, асинхронно отримує дані про стан складу з таблиці “MainChar” та формує рядок звіту з використанням “StringBuilder”, додаючи інформацію про кожен товар.

Асинхронно записує сформований звіт у текстовий файл та оновлює рядок стану з повідомленням про те, що звіт успішно згенерований і збережений.

```

Ссылка 1
public async void ShowAnalytics()
{
    using var context = new AppDBContext();
    var analyticsData = await context.MainChart.ToListAsync();

    var totalProducts = analyticsData.Sum(x => x.Quantity);
    var distinctProducts = analyticsData.Select(x => x.Name).Distinct().Count();

    StatusBar = $"Total products in warehouse: {totalProducts}, Various names: {distinctProducts}";
}

```

Рисунок 3.21 – Код для здійснення аналізу даних

Метод “ShowAnalytics” створює контекст бази даних і асинхронно отримує дані про стан складу з таблиці “MainChart”.

Обчислює загальну кількість товарів та кількість унікальних найменувань товарів на складі, після чого оновлює рядок стану з повідомленням про результати обчислень.

### 3.2 Розробка структури БД

Для функціонування програми була створена БД в програмі MySQL, яка складається з чотирьох таблиць: “MainChart”, “OrderChart”, “ProductionChart” та “UsersChart”.

Користувачі додаються в базу вручну в таблицю “UsersChart”. Під час введення та надсилання даних для авторизації виконується перевірка їхньої відповідності. Таким чином, користувач повинен мати кілька полів для порівняння. Також необхідно ідентифікувати кожного користувача.

Опис даних, які зберігаються в таблиці БД приведено в таблиці 3.1.

Таблиця 3.1 – Опис таблиці “UsersChart”

Назва	Тип даних	Первинний ключ	Додаткові опції
Id	Int	autoincrement primary key	Not Null
Login	VarChar	-	Not Null max-length = 50

Продовження таблиці 3.1

Назва	Тип даних	Первинний ключ	Додаткові опції
Password	VarChar	-	Not Null max-length = 50

За описом даних що зберігаються в таблиці БД був розроблений SQL запит для її створення:

```
CREATE TABLE UsersChart (
    Id INT PRIMARY KEY IDENTITY(1,1),
    Login NVARCHAR(50) NOT NULL,
    Password NVARCHAR(50) NOT NULL
);
```

В даному запиті разом із таблицею створюються колонка “Id” з автоінкрементом, колонка “Login” з кількістю символів 50 та колонка “Password” з кількістю символів 50.

Для заповнення таблиці “UsersChart” був створений запит:

```
INSERT INTO UsersChart (Login, Password)
VALUES
('admin', 'admin');
('Diplom', 'vgorle');
```

В даному запиті заповнюються колонки “Login” та “Password”.

Значення до таблиці “MainChart” додаються в БД за допомогою сканування QR-коду. При скануванні необхідно ідентифікувати кожен товар задля подальшої роботи разом із таблицями “OrderChart”, “ProductionChart”.

Опис даних, які зберігаються в таблиці БД приведено в таблиці 3.2.

Таблиця 3.2 – Опис таблиці “MainChart”

Назва	Тип даних	Первинний ключ	Додаткові опції
Id	Int	autoincrement primary key	Not Null

## Продовження таблиці 3.2

Назва	Тип даних	Первинний ключ	Додаткові опції
Data	Datetime	-	Not Null
Rack	Int	-	NOT NULL CHECK (Rack IN (1, 2, 3, 4))
Name	VarChar	-	Not Null max-length = 255
MaterialType	VarChar	-	Not Null max-length = 255
Quantity	Int	-	Not Null

За описом даних що зберігаються в таблиці БД був розроблений SQL запит для її створення:

```
CREATE TABLE MainChart (
    Id INT IDENTITY(1,1) PRIMARY KEY,
    Data DATETIME NOT NULL,
    Name NVARCHAR(255) NOT NULL,
    MaterialType NVARCHAR(255) NOT NULL,
    Quantity INT NOT NULL,
    Rack INT NOT NULL CHECK (Rack IN (1, 2, 3, 4))
);
```

В даному запиті разом із таблицею створюються колонки: “Id” з автоінкрементом, “Data”, “Name” з кількістю символів 255, “MaterialType” з кількістю символів 255, “Quantity” та “Rack” з перевіркою на числа 1,2,3 та 4.

Значення до таблиць “OrderChart”, “ProductionChart” додаються в БД вручну з текстових полів в програмі. При внесенні відбувається перевірка з даними з таблиці “MainChart”, тому потрібно ідентифікувати кожен запит задля подальшої роботи разом із таблицею.

Опис даних, які зберігаються в таблицях БД приведено в таблиці 3.3 та таблиці 3.4.

Таблиця 3.3 – Опис таблиці “OrderChart”

Назва	Тип даних	Первинний ключ	Додаткові опції
Id	Int	autoincrement primary key	Not Null
Status	VarChar	-	NOT NULL max-length = 50
Data	Datetime	-	Not Null
Name	VarChar	-	Not Null max-length = 100
MaterialType	VarChar	-	Not Null max-length = 50
Quantity	Int	-	Not Null

Таблиця 3.4 – Опис таблиці “ProductionChart”

Назва	Тип даних	Первинний ключ	Додаткові опції
Id	Int	autoincrement primary key	Not Null
Data	Datetime	-	Not Null
Name	VarChar	-	Not Null max-length = 100
MaterialType	VarChar	-	Not Null max-length = 50
Quantity	Int	-	Not Null
Status	VarChar	-	NOT NULL max-length = 50

За описом даних що зберігаються в таблицях БД були розроблені SQL запити для їх створення.

Запит для таблиці “OrderChart”:

```
CREATE TABLE OrderChart (  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    Data DATETIME NOT NULL,  
    Name NVARCHAR(100) NOT NULL,  
    MaterialType NVARCHAR(50) NOT NULL,  
    Quantity INT NOT NULL,  
    Status NVARCHAR(50) NOT NULL  
);
```

Запит для таблиці “ ProductionChart ”:

```
CREATE TABLE ProductionChart (  
    Id INT PRIMARY KEY IDENTITY(1,1),  
    Data DATETIME NOT NULL,  
    Name NVARCHAR(100) NOT NULL,  
    MaterialType NVARCHAR(50) NOT NULL,  
    Quantity INT NOT NULL,  
    Status NVARCHAR(50) NOT NULL  
);
```

Дані запити ідентичні та мають такі колонки: “Id” з автоінкрементом, “Data”, “Name” з кількістю символів 100, “ MaterialType ” з кількістю символів 50, “ Quantity ” та “ Status ” з кількістю символів 50.

### 3.3 Розробка інтерфейсу користувача

При запуску програми висвічується вікно авторизації (рисунок 3.22).

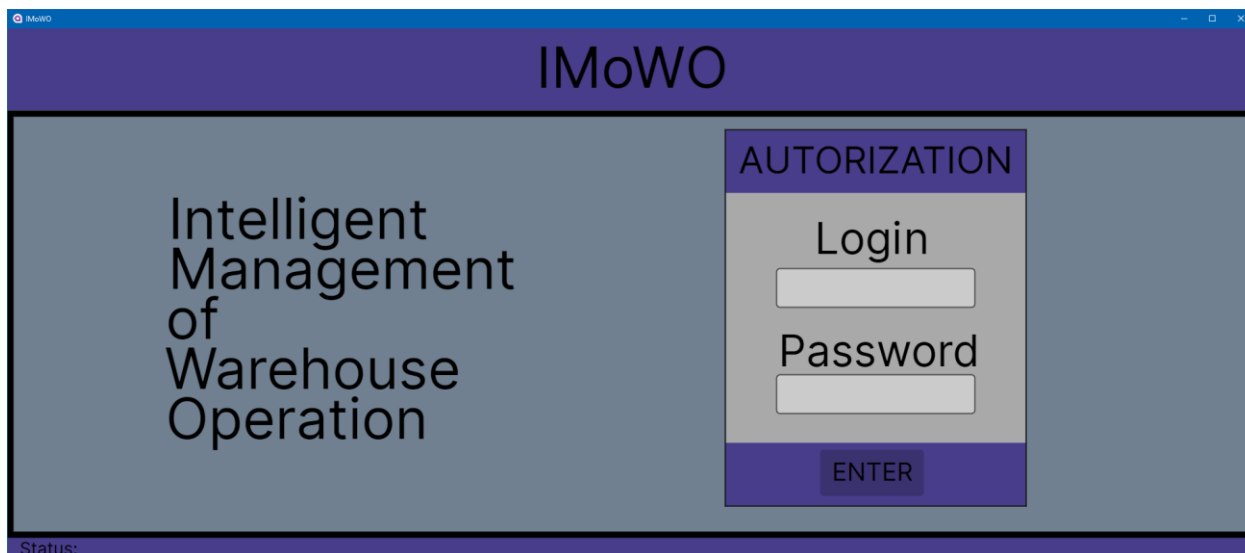


Рисунок 3.22 – Вікно авторизації

У вікні відображається назва додатку, поля для вводу логіну та паролю з назвами “Login” та “Password”, кнопка з назвою “ENTER” та текстова назва “Status” для виведення статусу в програмі. При введенні некоректних даних в поля “Login” та “Password” система виведе повідомлення до “Status” про некоректні дані (рисунок 3.23).

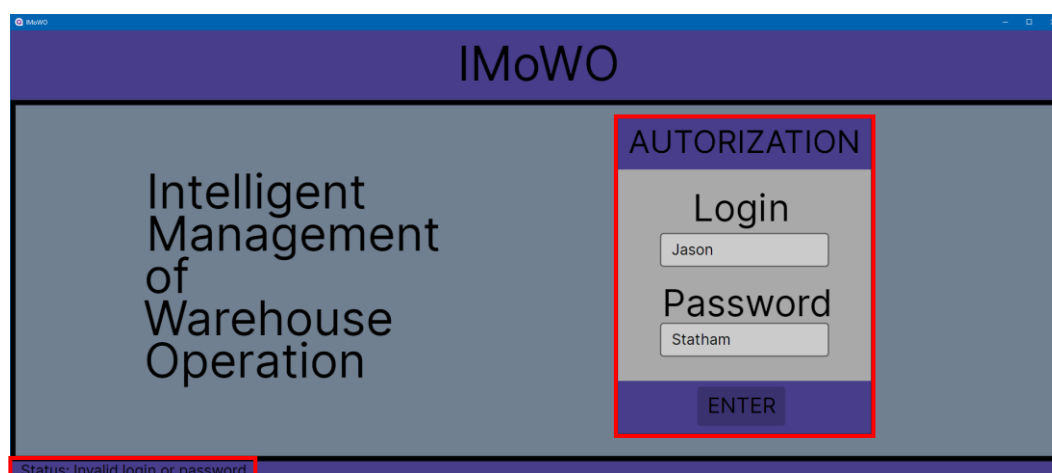


Рисунок 3.23 – Вікно авторизації з повідомленням в “Status” про некоректні дані

Як видно на рисунку в поля було введено дані однак система перевіривши дані користувача в БД вивела помилку про некоректні дані в “Status”. Якщо внести в поля дані користувача що є в БД, система перевірить дані на співпадіння. Якщо дані будуть вірні система виведе повідомлення про успіх та переадресує користувача до головного вікна (рисунок 3.24).

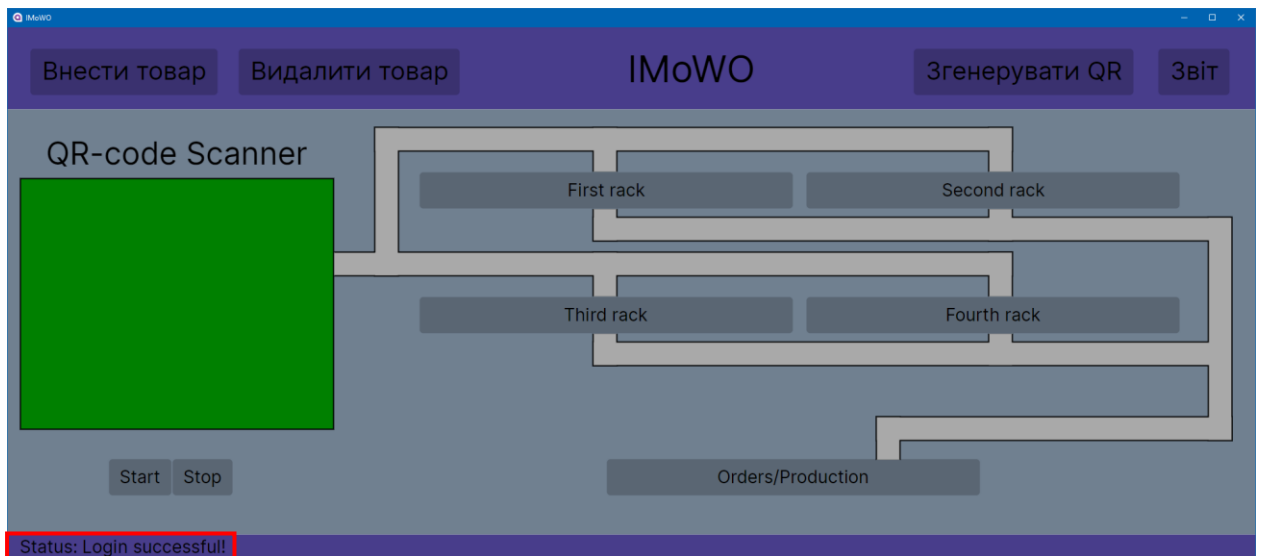


Рисунок 3.24 – Головне вікно програми після успішного входу

На головному вікні можна побачити основні елементи програми: текстову назву програми “IMoWO”, кнопки “Внести товар” та “Видалити товар” що відображають вікна “QR-code Scanner” із різним кольором вікна, зелений для внесення, а червоний для видалення, “Згенерувати QR” для генерування QR-коду, “Звіт” для відображення вікна з таблицею звітності, “First Rack” для відображення вікна з таблицею товару з типом матеріалу “Plate”, “Second Rack” для відображення вікна з таблицею товару з типом матеріалу “Wire”, “Third Rack” для відображення вікна з таблицею товару з типом матеріалу “Pipe”, “Fourth Rack” для відображення вікна з таблицею товару з типом матеріалу “Bag” та “Orders/Production” для відображення вікна з таблицями “Order” та “Production” для внесення замовлень.

Для того щоб імітувати QR-код товару, який прибув до складу, в програмі створено вікно із полями вводу даних з назвами “Name” та “MaterialType”, кнопками “ENTER” для генерування QR-коду та “Close” для зачинення вікна.

Для відкриття вікна потрібно натиснути кнопку “Згенерувати QR”. Вписавши дані в поля “Name” та “MaterialType” які повинні бути в QR-коді натискаємо кнопку “ENTER”. Система генерує QR-код та зберігає його по певній директорії висвічуючи повідомлення в “Status” (рисунок 3.25).

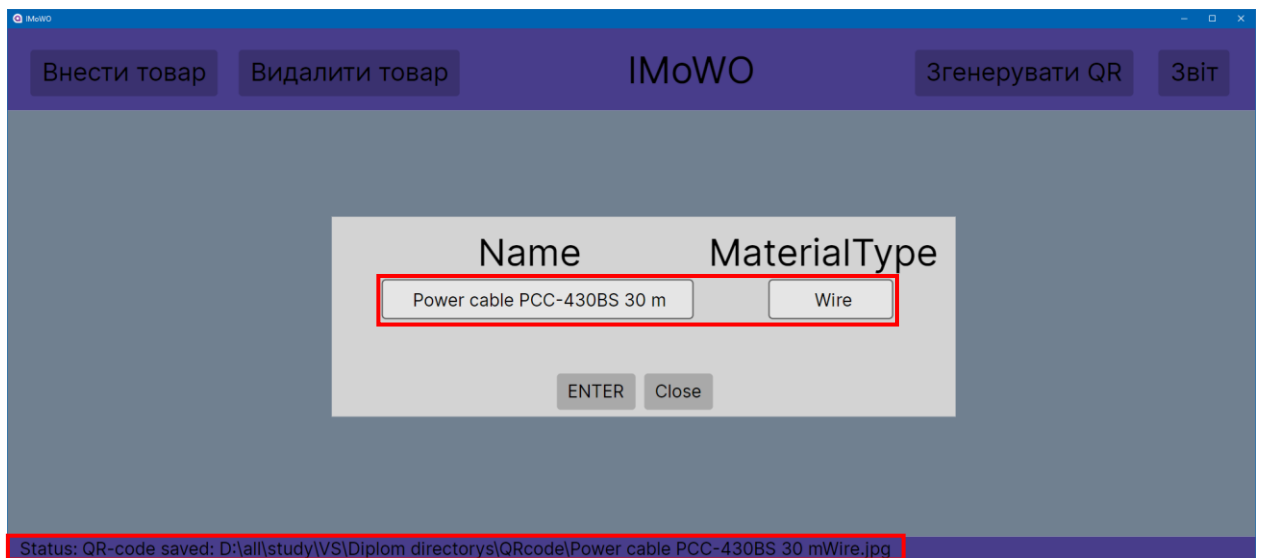


Рисунок 3.25 – Вікно генерації QR-коду із повідомленням про успішну генерацію та збереження

В полі “Status” бачимо зазначену директорію: “D:\all\study\VS\Diplom directories\QRcode”. Перейшовши по директорії бачимо створений QR-код, з назвою із полів “Name” та “MaterialType”. Разом із створеним кодом знаходяться коди товарів, що вже були створені та знаходяться в БД складського приміщення (рисунок 3.26).

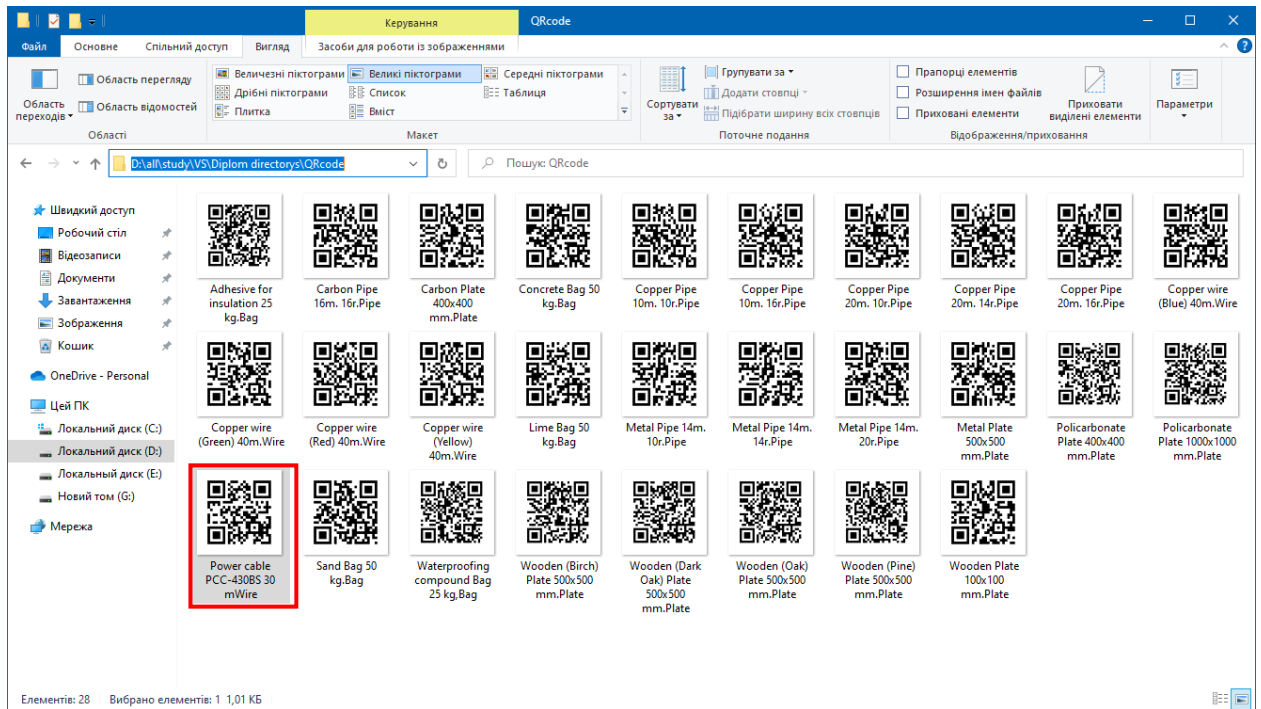


Рисунок 3.26 – Директорія із QR-кодами

Згенерувавши QR-код зачинаємо вікно та приступаємо до роботи програми.

Початково, при запуску програми зображено вікно для внесення товару до складу. У вікні знаходиться панель для відображення зображення зі сканеру та кнопки “Start” та “Stop” для його запуску та зупинки. Натискаємо на кнопку “Start” та починаємо роботу сканера разом із цим висвічується повідомлення про успішну ініціалізацію (рисунок 3.27).

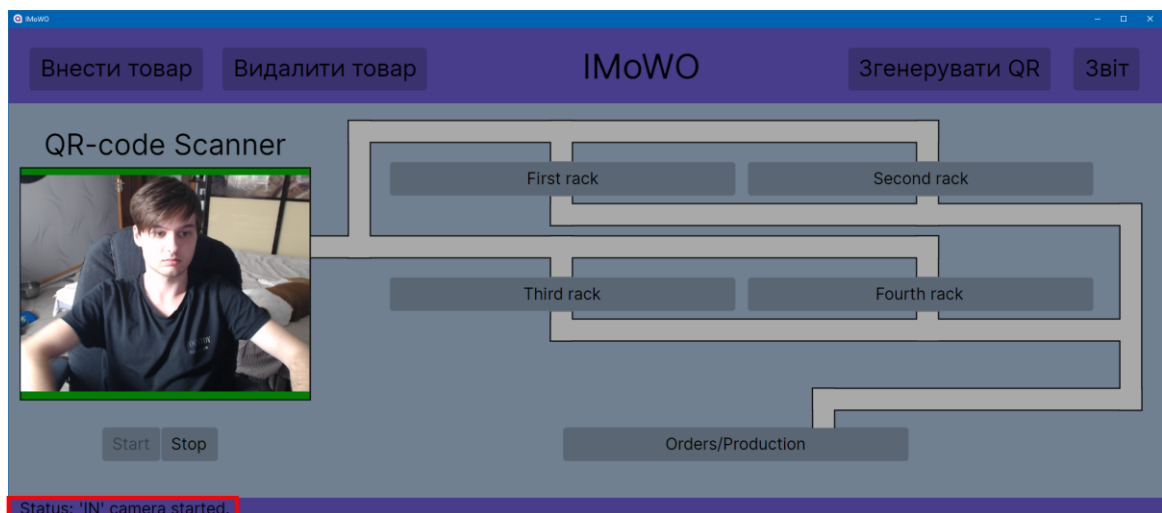


Рисунок 3.27 – Головне вікно із увімкненим сканером та повідомленням

Ініціалізувавши роботу сканеру в полі “Status” висвічується повідомлення про очікування QR-коду. Піднесши до сканеру QR-код система просканує його та проаналізує його дані.

Якщо такого значення немає на складі система автоматично додасть його до БД та направить його до потрібного стелажу висвітивши в полі “Status” про успіх (рисунок 3.28, рисунок 3.29).

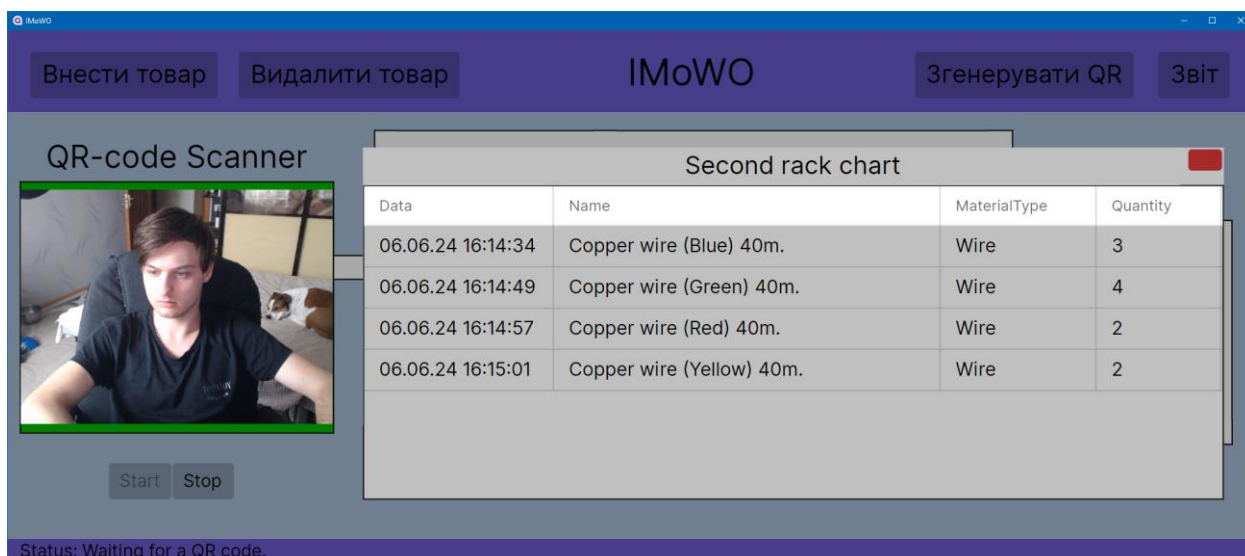


Рисунок 3.28 – Вікно з таблицею без змін

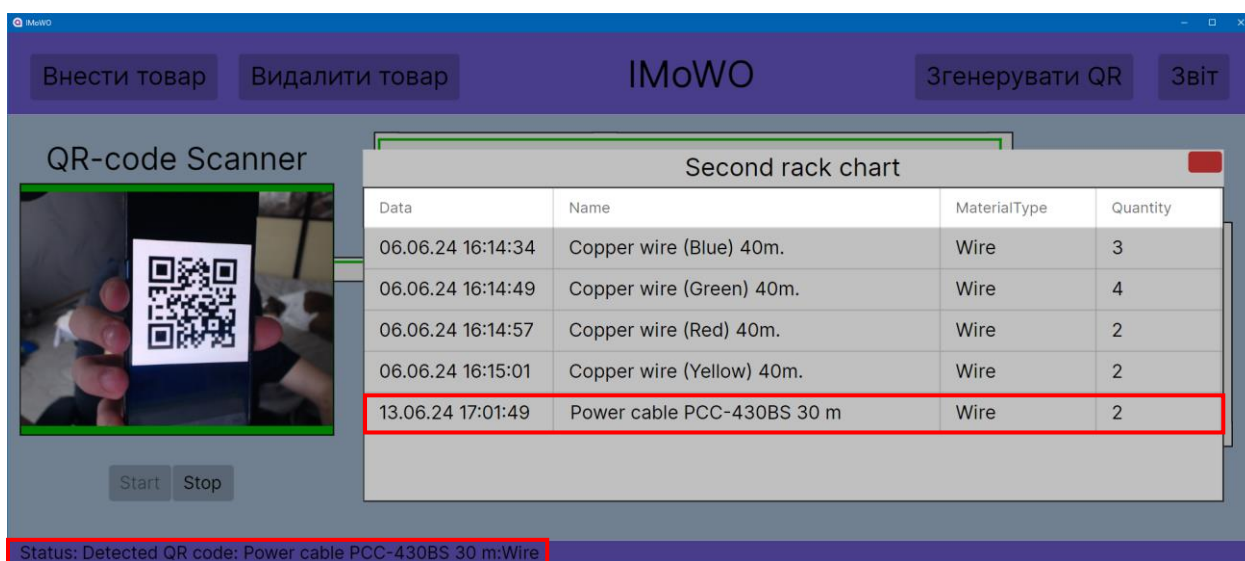


Рисунок 3.29 – Вікно з таблицею та внесеним значенням

Якщо таке значення вже є на складі система автоматично направить його на потрібний стелаж висвітливши в полі “Status” про успіх (рисунок 3.30).

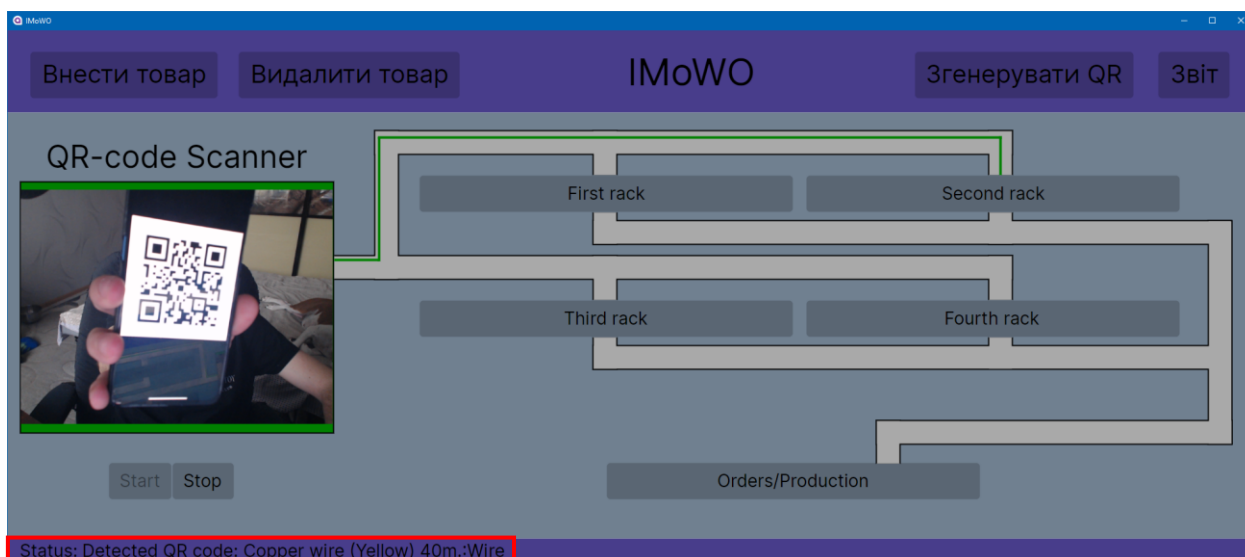


Рисунок 3.30 – Головне вікно з візуалізацією внесення значень

Внесши потрібні значення зупиняємо сканер натискаючи кнопку “Stop”. Після зупинки сканеру в полі “Status” висвічується повідомлення про успішну зупинку сканера (рисунок 3.31).

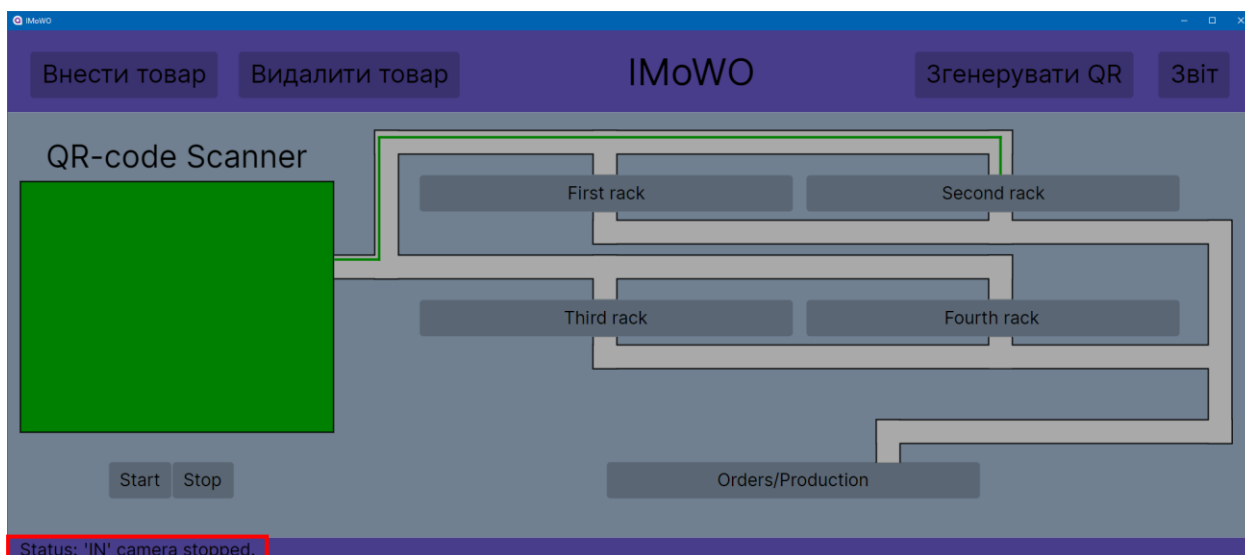


Рисунок 3.31 – Головне вікно з вимкненим сканером після внесення значень

Для імітації отримання замовлень або потреб підприємства було розроблено вікно яке розділене навпіл.

Ліва частина вікна має назву “Order” та містить в собі поля для вводу даних із назвами “Name” та “Quantity”, кнопку з назвою “Enter” для внесення та таблицю для відображення даних та статусу замовлення. Права частина вікна має назву “Production” та містить в собі такі ж самі елементи внесення та візуалізації але для потреб підприємства (рисунок 3.32).

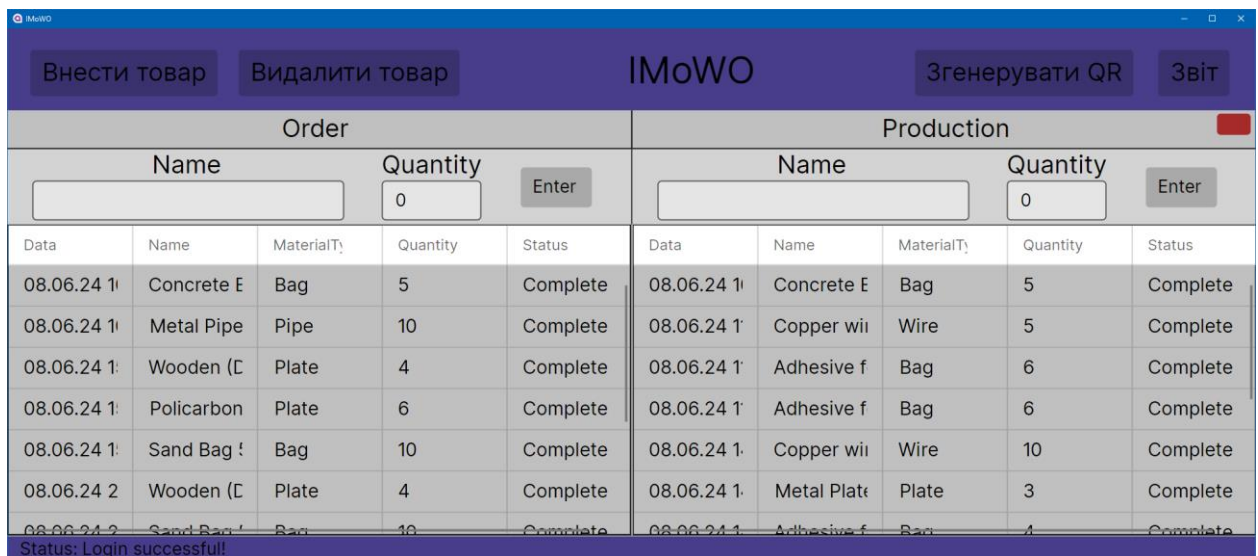


Рисунок 3.32 – Вікно замовлень та потреб підприємства

Для замовлення потрібно внести дані в поля “Name” і “Quantity” та натиснути кнопку “Enter”, після чого система проаналізує усі значення що є в БД складу та додасть замовлення в таблицю. Якщо кількість товару достатня на складі система самостійно забере товар зі стелажів, виконає замовлення та поставить статус замовлення в “Complete”, повідомивши про це в полі “Status” (рисунок 3.33).

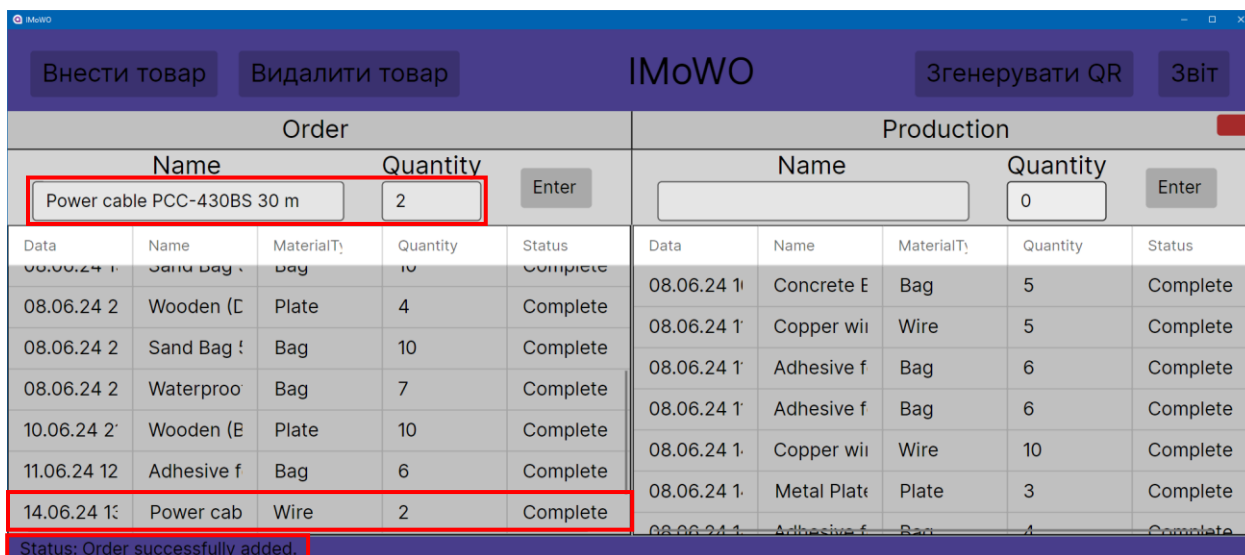


Рисунок 3.33 – Вікно замовлень та потреб підприємства із виконаним замовленням

Якщо кількість товару недостатня на складі система повідомить про недостатню кількість та поставить статус замовлення у “In process”, повідомивши про це в полі “Status” (рисунок 3.34).

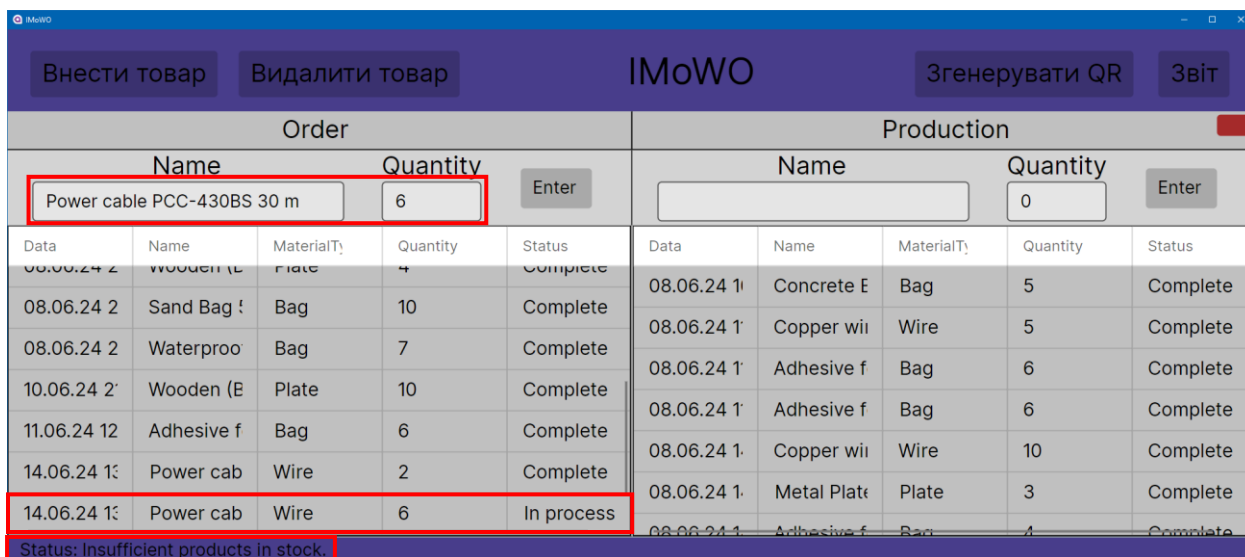


Рисунок 3.34 - Вікно замовлень та потреб підприємства із очікуючим замовленням

Як тільки на складі з'явиться достатня кількість товару система самостійно направить його до замовлення та змінить його статус на “Complete” (рисунок 3.35, рисунок 3.36).

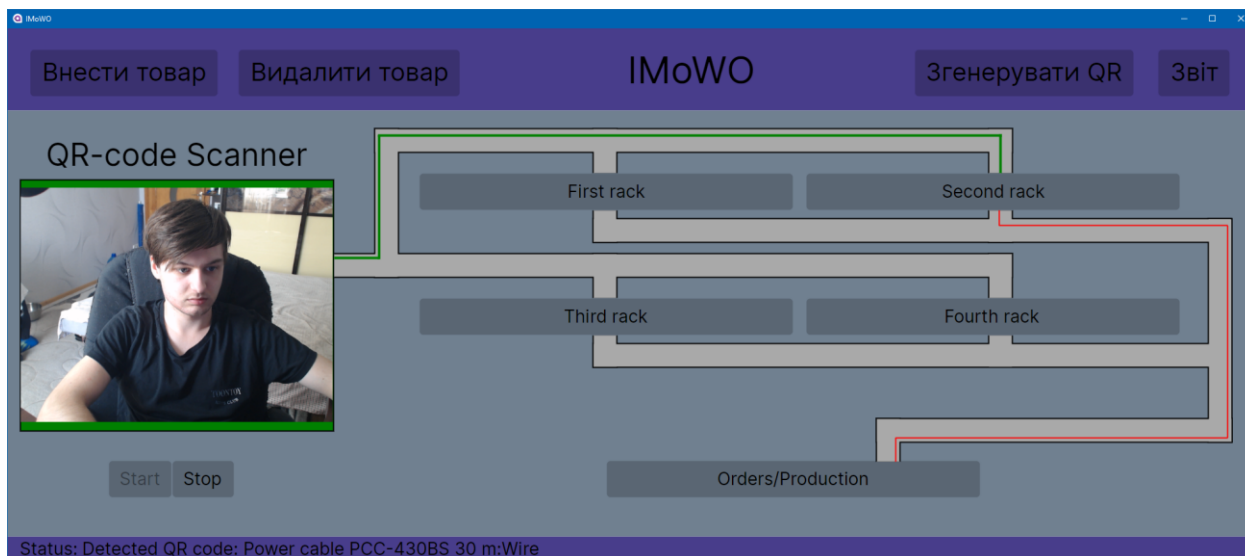


Рисунок 3.35 – Головне вікно із увімкненим сканером та візуалізацією переміщення товару до замовлення

Order					Production				
Name			Quantity	Enter	Name			Quantity	Enter
Power cable PCC-430BS 30 m			6	Enter				0	Enter
Data	Name	MaterialTy	Quantity	Status	Data	Name	MaterialTy	Quantity	Status
08.06.24 2	Wooden (E	Plate	4	Complete	08.06.24 1	Concrete E	Bag	5	Complete
08.06.24 2	Sand Bag f	Bag	10	Complete	08.06.24 1	Copper wi	Wire	5	Complete
08.06.24 2	Waterproof	Bag	7	Complete	08.06.24 1	Adhesive f	Bag	6	Complete
10.06.24 2	Wooden (E	Plate	10	Complete	08.06.24 1	Adhesive f	Bag	6	Complete
11.06.24 12	Adhesive f	Bag	6	Complete	08.06.24 1	Copper wi	Wire	10	Complete
14.06.24 13	Power cab	Wire	2	Complete	08.06.24 1	Metal Plat	Plate	3	Complete
14.06.24 13	Power cab	Wire	6	Complete	08.06.24 1	Adhesive f	Bag	4	Complete

Рисунок 3.36 - Вікно замовлень та потреб підприємства із виконаним замовленням після додавання товару

Для видалення товару зі складу необхідно змінити вікно “QR-code Scanner” зі внесення на видалення натиснувши кнопку “Видалити товар”.

Змінивши вікно натискаємо на кнопку “Start” та ініціалізуємо сканер. Після увімкнення сканеру система повідомить про це в полі “Status” (рисунок 3.37).

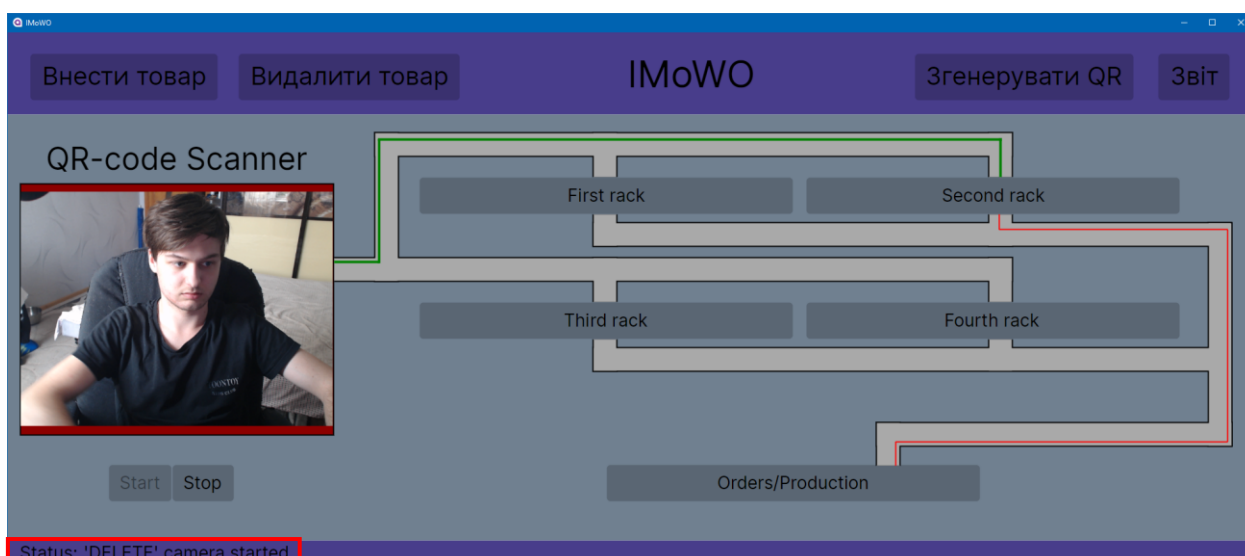


Рисунок 3.37 – Головне вікно із увімкненим сканером для видалення товару зі складу

Підносимо до сканеру QR-код товару який потрібно видалити, система сканує його та зчитує дані, після чого аналізує БД та видаляє товар повідомивши про це в полі “Status” (рисунок 3.38).

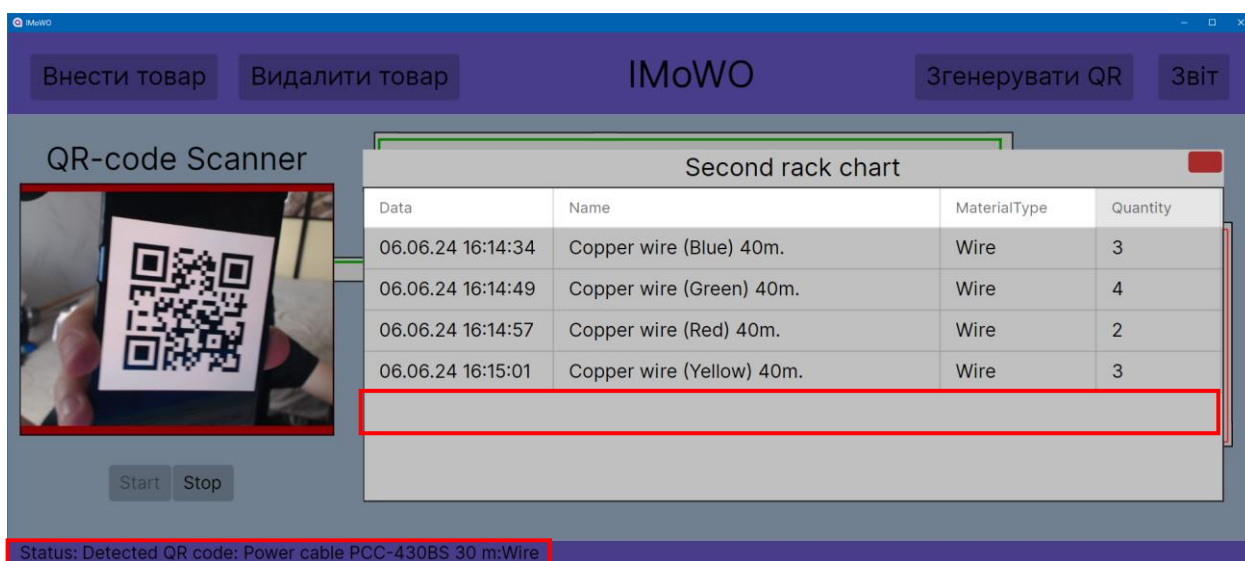


Рисунок 3.38 – Вікно з таблицею та видаленим значенням

Видаливши товар зупиняємо роботу сканера натиснувши на кнопку “Stop” після чого з’являється повідомлення в полі “Status” (рисунок 3.39).

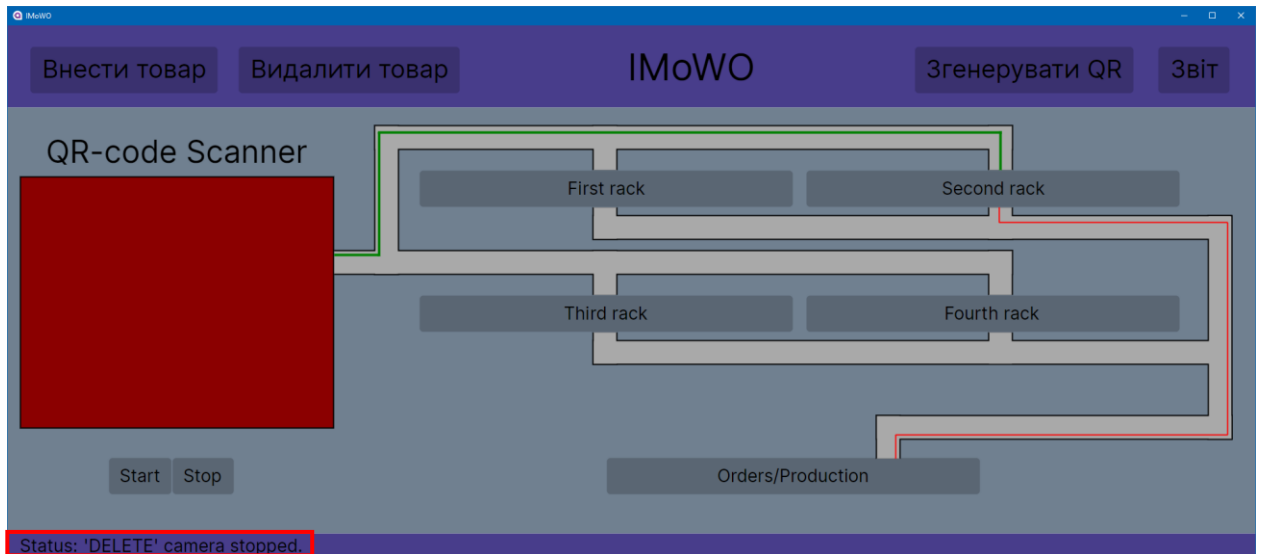


Рисунок 3.39 – Головне вікно із вимкненим сканером

Для перегляду таблиці із усім товаром на складі потрібно натиснути на кнопку “Звіт”, після чого відчиниться вікно із таблицею та кнопками “Згенерувати звіт”, “Аналіз”, “Сховати” (рисунок 3.40).

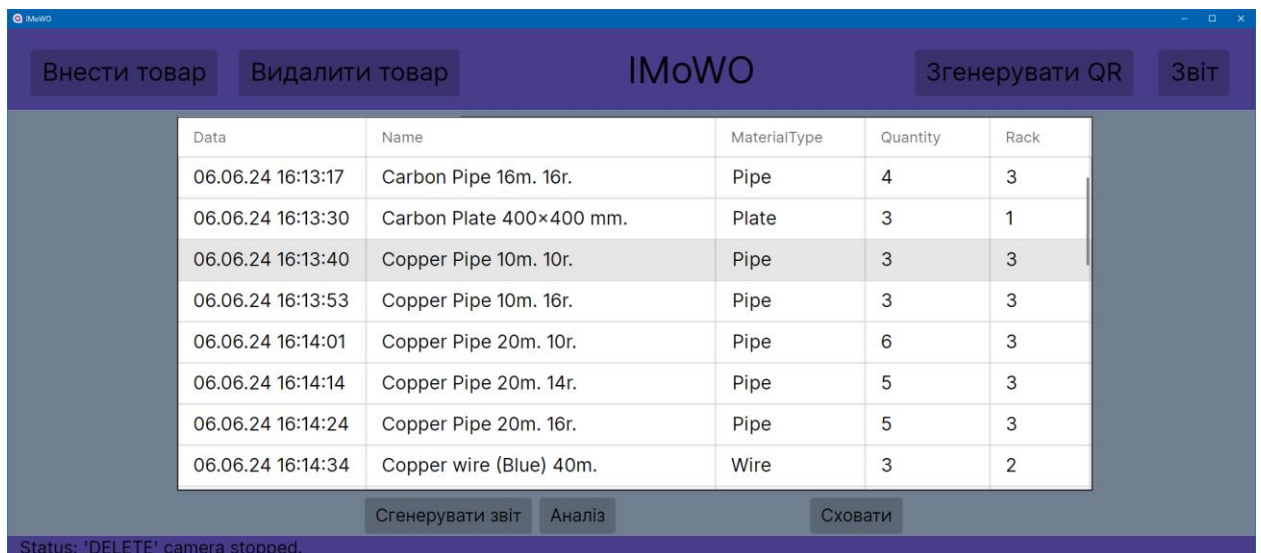
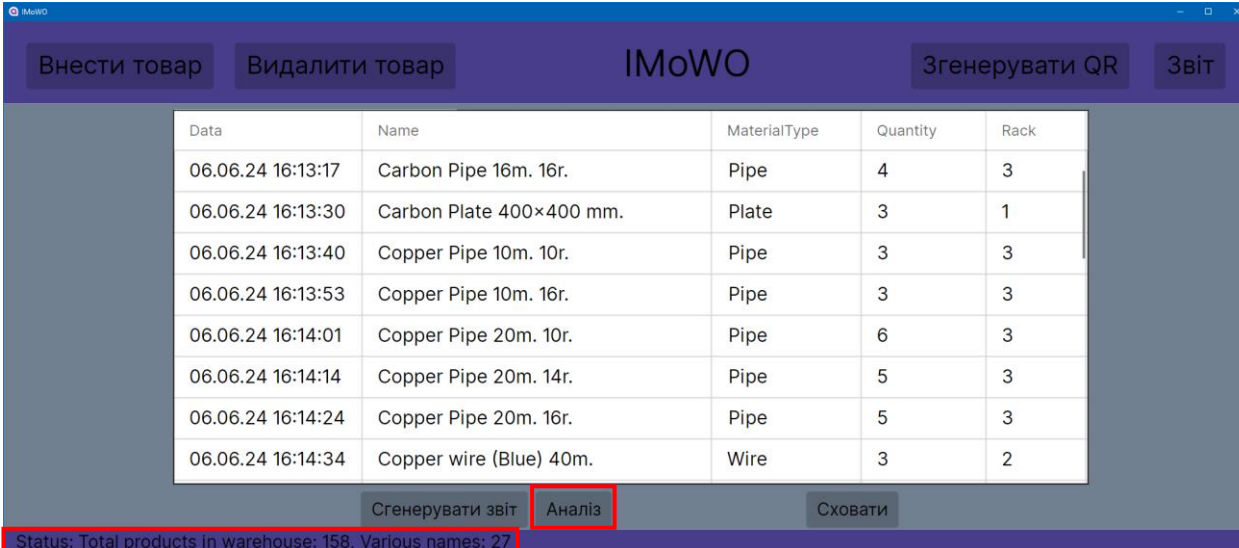


Рисунок 3.40 – Вікно із таблицею усіх товарів на складі

Натиснувши кнопку “Аналіз” система самостійно підрахує кількість товару на складі та різноманітність імен в полі “Status” (рисунок 3.41).

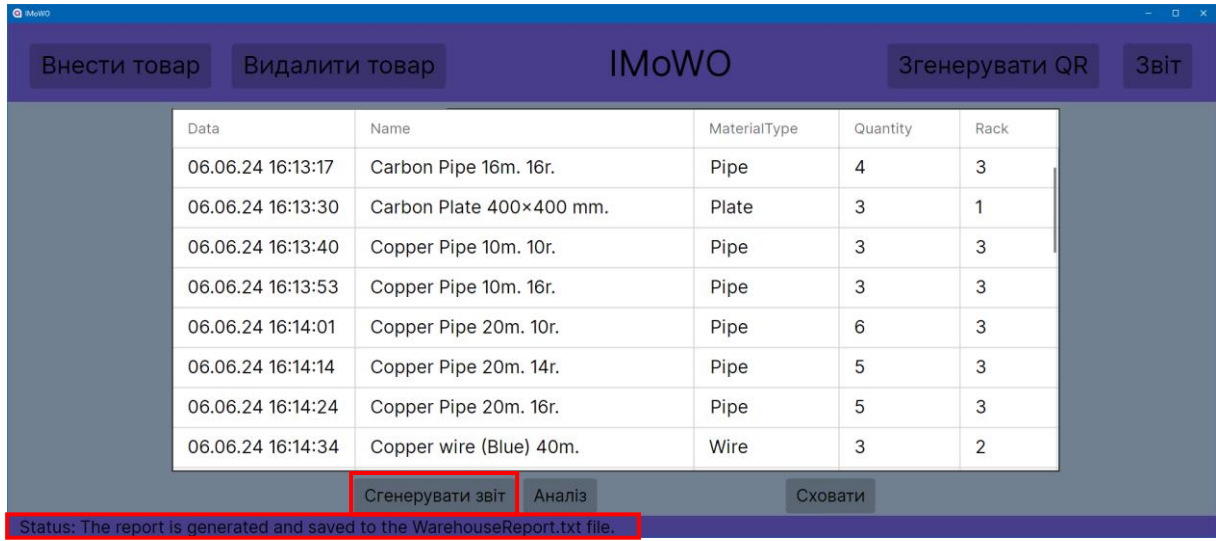


Data	Name	MaterialType	Quantity	Rack
06.06.24 16:13:17	Carbon Pipe 16m. 16r.	Pipe	4	3
06.06.24 16:13:30	Carbon Plate 400×400 mm.	Plate	3	1
06.06.24 16:13:40	Copper Pipe 10m. 10r.	Pipe	3	3
06.06.24 16:13:53	Copper Pipe 10m. 16r.	Pipe	3	3
06.06.24 16:14:01	Copper Pipe 20m. 10r.	Pipe	6	3
06.06.24 16:14:14	Copper Pipe 20m. 14r.	Pipe	5	3
06.06.24 16:14:24	Copper Pipe 20m. 16r.	Pipe	5	3
06.06.24 16:14:34	Copper wire (Blue) 40m.	Wire	3	2

Статус: Total products in warehouse: 158, Various names: 27

Рисунок 3.41 – Вікно із таблицею усіх товарів на складі та виконаним аналізом продукції

Натиснувши кнопку “Згенерувати звіт” система самостійно створить текстовий файл з найменуванням “WarehouseReport” який містить у собі усю інформацію про продукцію на складі та збереже по певній директорії повідомивши про це в полі “Status” (рисунок 3.42).



Data	Name	MaterialType	Quantity	Rack
06.06.24 16:13:17	Carbon Pipe 16m. 16r.	Pipe	4	3
06.06.24 16:13:30	Carbon Plate 400×400 mm.	Plate	3	1
06.06.24 16:13:40	Copper Pipe 10m. 10r.	Pipe	3	3
06.06.24 16:13:53	Copper Pipe 10m. 16r.	Pipe	3	3
06.06.24 16:14:01	Copper Pipe 20m. 10r.	Pipe	6	3
06.06.24 16:14:14	Copper Pipe 20m. 14r.	Pipe	5	3
06.06.24 16:14:24	Copper Pipe 20m. 16r.	Pipe	5	3
06.06.24 16:14:34	Copper wire (Blue) 40m.	Wire	3	2

Статус: The report is generated and saved to the WarehouseReport.txt file.

Рисунок 3.42 – Вікно із таблицею усіх товарів на складі та виконаною генерацією звіту

Перейшовши по даній директорії бачимо текстовий файл “WarehouseReport”. Відкривши його побачимо звіт по всій продукції зі складу (рисунок 3.43).

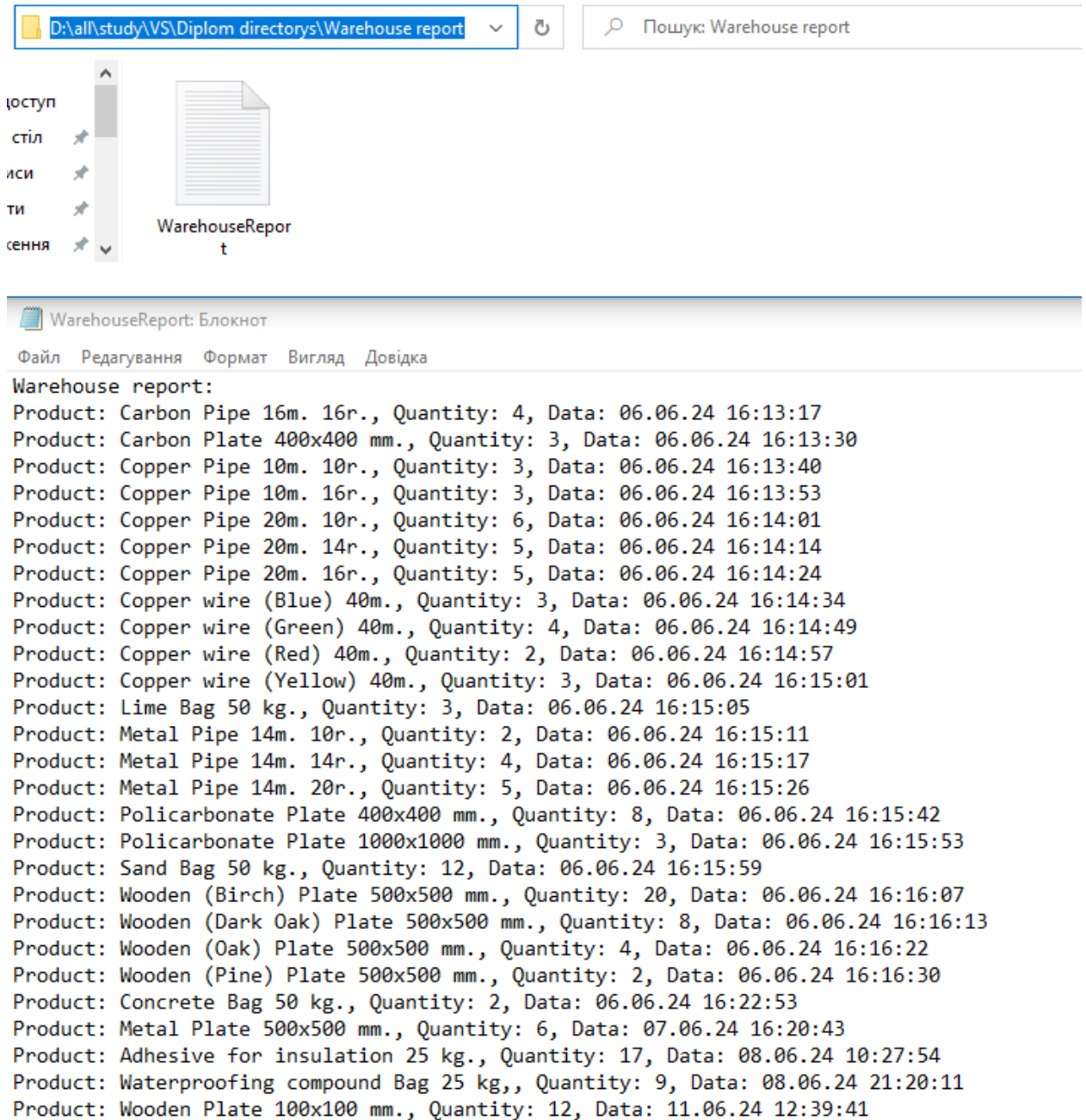


Рисунок 3.43 – Папка із текстовим файлом “WarehouseReport” та відчинений .txt файл

## 4 ОХОРОНА ПРАЦІ

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних, лікувально-профілактичних, реабілітаційних та інших заходів і засобів, спрямованих на збереження життя, здоров'я і працездатності людини в процесі трудової діяльності [29].

Коефіцієнт умов праці характеризує відповідність фактичних умов праці нормативним. Не всі матеріально-виробничі елементи умов праці визначаються кількісною оцінкою. Тому при розрахунку коефіцієнта умов праці необхідно брати ті з них, які можна виразити певною числовою величиною і для яких є нормативне значення. Такими елементами можуть бути освітленість, температура, вогкість, шум, вібрація тощо [30].

Коефіцієнт умов праці  $K_{y.n}$  визначається як середньо геометрична величина показників, які характеризують умови праці, за такою формулою

$$K_{y.n} = \sqrt[n]{a_1 a_2 \dots a_n}, \quad (4)$$

де  $a_1, a_2, \dots, a_n$  – індекс відхилення фактичних елементів умов праці від нормативних за відповідними показниками;

$n$  – кількість показників, які характеризують елементи умов праці, за якими проводилися виміри [30].

Охорона праці включає в себе такі аспекти як:

- законодавчі заходи;
- організаційно-технічні заходи;
- санітарно-гігієнічні заходи;
- лікувально-профілактичні заходи;
- навчання та інструктажі [29].

Охорона праці в інтелектуальному керуванні складськими операціями включає в себе використання автоматизованих систем і програмного

забезпечення для оптимізації процесів на складах. Охорона праці в цьому контексті має особливе значення, оскільки пов'язана з безпекою роботи людей і обладнання. Основними аспектами, які потрібно враховувати при роботі з даною системою, являються:

- використання сенсорів та аварійних вимикачів в складських операціях, забезпечує безпеку під час роботи системи та знижує фізичні навантаження на працівників, підвищуючи загальний рівень безпеки;

- забезпечення працівників переносними пристроями та іншими технологіями що дають змогу відстежувати фізичний стан у реальному часі. Це дозволить запобігти перевтоми та інших ризиків для здоров'я працівників, а також провести аналіз даних, за допомогою яких можливо вчасно виявляти потенційні проблеми;

- використання роботизованих підйомників, транспортних засобів, конвеєрних стрічок та інших автоматизованих рішень задля оптимізації процесів на складах, що дозволяє зменшити фізичне навантаження на працівників і підвищує ефективність роботи;

- використання онлайн-платформ, інтерактивних інструментів, таких як віртуальна реальність (VR), задля навчання працівників безпечної роботи з новими технологіями, дозволяє зменшити ризики травмування працівників та отримувати знання про безпечні практики та процедури в будь-який зручний час при навчанні;

- використання інтелектуальних систем керування будівлями що автоматично регулюють умови праці, такі як освітлення, температура та вентиляція, дозволяють забезпечити комфортні умови для працівників та запобігти небезпечним умовам праці;

- використання великих даних та аналітики задля аналізу інцидентів, трендів і впровадження нових технологій, що дає змогу прогнозувати можливі ризики та розробляти превентивні заходи для їх запобігання задля поліпшення умов праці.

## ВИСНОВКИ

В ході виконання були розглянуті, проаналізовані та вирішені такі питання та задачі як:

- аналіз технічного завдання;
- дослідження за темою інтелектуальне керування складськими операціями;
- розробка програмного забезпечення для системи інтелектуального керування складськими операціями;
- розробка основних аспектів для безпеки при роботі із системою інтелектуального керування складськими операціями.

Під час аналізу технічного завдання було розглянуто про сучасні системи автоматизації складських приміщень (WMS), сучасні роботизовані системи транспортування та схожі варіанти рішень за темою. Проаналізувавши отриману інформацію були зроблені підсумки аналізу технічного завдання, задля отримання наступних кроків для роботи над кваліфікаційною роботою бакалавру.

В ході дослідження було з'ясовано що таке інтелектуальне керування складськими операціями, розраховано економічну доцільність та ROI такої системи та приведені приклади готових рішень систем автоматизації складських приміщень. Також було обрано програмне середовище для реалізації проекту.

Розроблені алгоритми в програмному середовищі Visual Studio на мові C#, структура БД в реляційній системі управління базами даних Microsoft SQL Server та розроблений сам інтерфейс користувача на кросплатформенному фреймворку AvaloniaUI.

В роботі розглянуті питання охорони праці та розроблені основні аспекти для забезпечення безпеки які потрібно враховувати при роботі з даною системою.

Підсумовуючи висновки для виконання кваліфікаційної роботи були виконанні наступні задачі:

- проаналізовано сучасні системи автоматизації складських приміщень (WMS) ;
- проаналізовано сучасні роботизовані системи транспортування;
- проаналізовано схожі варіанти рішень за темою інтелектуальне керування складськими операціями;
- зроблені підсумки аналізу технічного завдання;
- досліджено тему інтелектуальне керування складськими операціями;
- розраховано економічну доцільність та ROI;
- приведені приклади готових рішень систем автоматизації складських приміщень;
- розроблені алгоритми програми, структура БД та інтерфейс користувача;
- розглянуті питання охорони праці та розроблені основні аспекти для забезпечення безпеки які потрібно враховувати при роботі з системою.

Отримані результати дослідження показують, що впровадження інтелектуальної системи керування складськими операціями є ефективним і економічно вигідним рішенням, яке дозволяє значно підвищити продуктивність складських процесів. Перспективи подальших досліджень у цьому напрямку включають розробку нових алгоритмів машинного навчання для подальшої оптимізації складських операцій, а також інтеграцію інтелектуальних систем з іншими елементами ланцюга поставок.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ціль 11. Сталий розвиток міст і громад [Електронний ресурс] URL: <https://business.diia.gov.ua/en/handbook/sustainable-development-goals/cil-11-stalij-rozvitok-mist-i-gromad> дата звернення - [15.06.2024].
2. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. – 29 с. дата звернення - [29.04.2024].
3. Методичні вказівки з підготовки кваліфікаційної роботи бакалавра для студентів усіх форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» / Упоряд.: І.Ш. Невлюдов, А.О. Андрусевич, О.В. Токарєва, С.П. Новоселов, О.В Сичова. Харків: ХНУРЕ, 2022. – 55 с. дата звернення - [29.04.2024].
4. TheTransmitted Що таке штучний інтелект? [Електронний ресурс] URL: <https://thetransmitted.com/adlucem/shho-take-shtuchnij-intelekt-shi/> дата звернення - [29.04.2024].
5. «ШТУЧНИЙ ІНТЕЛЕКТ» ТА ЙОГО ВИКОРИСТАННЯ В ЛОГІСТИЧНИХ ПРОЦЕСАХ Наталія Ю. Кирлик 2021 [Електронний ресурс] URL: [https://eco-science.net/wp-content/uploads/2021/12/9-10.21.\\_topik\\_Kyrlyk-N.Yu\\_.59-66.pdf](https://eco-science.net/wp-content/uploads/2021/12/9-10.21._topik_Kyrlyk-N.Yu_.59-66.pdf) дата звернення - [29.04.2024].
6. Robotpalletiseermachine: een praktijkvoorbeeld [Електронний ресурс] URL: <https://www.arco-solutions.nl/automatisch-palletiseren-een-praktisch-voorbeeld/> дата звернення - [30.04.2024].
7. Роботи навантажувачі [Електронний ресурс] URL: <https://ua.lanxinagv.com/agv/amr-forklift/robot-forklift-trucks.html> дата звернення - [30.04.2024].
8. Робот палетайзер, палетоукладач мішків, коробок, тюків [Електронний ресурс] URL:

<https://www.itpgroup.com.ua/partners/paglierani/robot-paletizer-paletoukladach>  
дата звернення - [30.04.2024].

9. Що таке WMS система? [Електронний ресурс] URL:  
<https://uislab.com/uk/chto-takoe-wms-sistema/> дата звернення - [03.05.2024].

10. Smart warehouse: A bibliometric analysis and future research direction  
[електронний ресурс] URL:  
<https://www.sciencedirect.com/science/article/pii/S2667344423000063> дата  
звернення - [25.05.2024].

11. ЧОМУ ТАК ВАЖЛИВА ЧЕТВЕРТА ПРОМИСЛОВА  
РЕВОЛЮЦІЯ? РОЗБИРАЄМОСЯ В ТЕХНОЛОГІЯХ ІНДУСТРІЇ 4.0  
[електронний ресурс] URL: <https://idcard.com.ua/ua/blog/why-is-the-fourth-industrial-revolution-so-important/> дата звернення - [25.05.2024].

12. Цифрова трансформація бізнесу Industry 4.0 [електронний ресурс]  
URL: <https://www.it.ua/knowledge-base/technology-innovation/industry-4> дата  
звернення - [25.05.2024].

13. Цифрова трансформація бізнесу Internet of Things, IoT  
[електронний ресурс] URL: <https://www.it.ua/knowledge-base/technology-innovation/internet-veschej-internet-of-things-iot> дата звернення - [25.05.2024].

14. ПРИНЦИП РОБОТИ RFID [електронний ресурс] URL:  
<https://idcard.com.ua/ua/blog/princzip-raboty-rfid/> дата звернення - [25.05.2024].

15. Кіберфізична система [електронний ресурс] URL:  
[https://www.wikidata.uk-ua.nina.az/Кіберфізична\\_система.html](https://www.wikidata.uk-ua.nina.az/Кіберфізична_система.html) дата звернення -  
[25.05.2024].

16. Цифрова трансформація бізнесу Big Data [електронний ресурс]  
URL: <https://www.it.ua/knowledge-base/technology-innovation/big-data-bolshie-dannye> дата звернення - [25.05.2024].

17. Цифрова трансформація бізнесу Artificial intelligence, AI  
[електронний ресурс] URL: <https://www.it.ua/knowledge-base/technology-innovation/artificial-intelligence> дата звернення - [25.05.2024].

18. Цифрова трансформація бізнесу Machine Learning, ML

[електронний ресурс] URL: <https://www.it.ua/knowledge-base/technology-innovation/machine-learning> дата звернення - [25.05.2024].

19. A Step-by-Step Guide to Calculating the ROI of Your WMS [електронний ресурс] URL: <https://www.generixgroup.com/en/blog/step-step-guide-calculating-roi-your-wms> дата звернення - [12.05.2024].

20. WMS ROI: A guide to measuring and calculating the ROI of your WMS [електронний ресурс] URL: <https://www.zoho.com/inventory/guides/maximizing-wms-roi-tips-and-strategies.html> дата звернення - [12.05.2024].

21. WMS ROI: How to Measure & Calculate ROI [електронний ресурс] URL: <https://www.logiwa.com/blog/warehouse-management-software-roi> дата звернення - [12.05.2024].

22. Top Five Ways Warehouse Management Systems Deliver ROI [електронний ресурс] URL: <https://www.inboundlogistics.com/articles/top-five-ways-warehouse-management-systems-deliver-roi/> дата звернення - [12.05.2024].

23. Фулфілмент для вашого бізнесу [електронний ресурс] URL: [https://novaposhta.ua/biznes\\_klientam/warehouse\\_logistics/en/?howWork](https://novaposhta.ua/biznes_klientam/warehouse_logistics/en/?howWork) дата звернення - [24.05.2024].

24. ЯК ПРАЦЮЄ AMAZON FBA: СЕКРЕТ УСПИХУ НАЙБІЛЬШОГО МАРКЕТПЛЕЙСА [електронний ресурс] URL: <https://4b.ua/blog/amazon-fba-how-it-works/> дата звернення - [24.05.2024]

25. AvaloniaUI [електронний ресурс] URL: <https://avaloniaui.net/> дата звернення - [24.05.2024].

26. Visual Studio 2022 [електронний ресурс] URL: <https://learn.microsoft.com/en-us/visualstudio/ide/whats-new-visual-studio-2022?view=vs-2022> дата звернення - [24.05.2024].

27. What is SQL Server? [електронний ресурс] URL: <https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver16> дата звернення - [24.05.2024].

28. ТЕОРІЯ АВТОМАТИЧНОГО УПРАВЛІННЯ (ЗБІРНИК ЗАДАЧ)

Навчальний посібник для студентів усіх форм навчання спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології/І.Ш. Невлюдов, О.В. Токарева; Харків. нац. ун-т радіоелектроніки. - Харків: Панов А.М., 2020. – 240 с.

29. ЗАКОН УКРАЇНИ Про охорону праці (Відомості Верховної Ради України (ВВР), 1992, № 49, ст.668), Розділ I, ЗАГАЛЬНІ ПОЛОЖЕННЯ, Стаття 1. [Електронний ресурс] URL: <https://zakon.rada.gov.ua/laws/show/2694-12#Text> дата звернення - [15.06.2024].

30. Комплекс навчально-методичного забезпечення навчальної дисципліни «Організація керування умовами праці» підготовки освітнього рівня бакалавр усіх спеціальностей та усіх напрямів університету / ХНУРЕ; розроб.: Т. Є. Стищенко, Г. В. Пронюк, Н. М. Сердюк. – Харків, 2017. – 108 с. [Електронний ресурс]. Режим доступу: <https://catalogue.nure.ua/document=218933> дата звернення - [15.06.2024].