

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І
АРХІТЕКТУРИ**

Автоматизації і інформаційних технологій

(факультет)

Кібербезпеки та комп'ютерної інженерії

(назва випускної кафедри)

**КВАЛІФІКАЦІЙНА РОБОТА
ЗДОБУВАЧА СТУПЕНЯ ВИЩОЇ ОСВІТИ МАГІСТР**

на тему:

**Проект системи автоматизації бухгалтерського та оперативного обліку
з інтеграцією в ERP-середовище з використанням RFID-технологій**

Нюкало Євген Ігорович

(прізвище, ім'я та по батькові здобувача повністю)

Київ 2025 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І
АРХІТЕКТУРИ**

Автоматизації і інформаційних технологій

(факультет)

Кібербезпеки та комп'ютерної інженерії

(назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри

к.т.н., доцент Максим ДЕЛЕМБОВСЬКИЙ

” _____ ” _____ 20 25 року

КВАЛІФІКАЦІЙНА РОБОТА

ЗДОБУВАЧА СТУПЕНЯ ВИЩОЇ ОСВІТИ МАГІСТР

Проект системи автоматизації бухгалтерського та оперативного обліку
з інтеграцією в ERP-середовище з використання RFID-технологій

(назва)

*Я як здобувач вищої освіти
КНУБА розумію і підтримую
політику закладу з академічної
добросовісності. Я не надавав
(-ла) і не одержував(-ла)
недозволену допомогу під час
підготовки цієї роботи.
Використання ідей, результатів і
текстів інших авторів мають
посилання на відповідне джерело.*

Здобувач Нюкало Євген Ігорович
(прізвище, ім'я та по батькові повністю)

123 «Комп'ютерна інженерія»

(спеціальність)

Комп'ютерні системи і мережі

(освітня програма)

Група КСММ-24

Керівник Вишняков В.М.

(прізвище та ініціали)

Кандидат технічних наук, доцент

(вчене звання, науковий ступінь)

Рецензент Горда О.В.

(прізвище та ініціали)

Ідентичність підтверджую

Київ 2025 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І
АРХІТЕКТУРИ**

Факультет: Автоматизації і інформаційних технологій

Випускова кафедра: Кібербезпеки та комп'ютерної інженерії

Ступінь вищої освіти: Магістр

Спеціальність: 123 «Комп'ютерна інженерія»

ОПП: Комп'ютерні системи і мережі

ЗАТВЕРДЖУЮ
Завідувач кафедри

к.т.н., доцент Максим ДЕЛЕМБОВСЬКИЙ

„ _____ ” _____ 20 25 року

ЗАВДАННЯ

**ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ ЗДОБУВАЧА
СТУПЕНЯ ВИЩОЇ ОСВІТИ МАГІСТР**

Нюкала Євгена Ігоровича

(прізвище, ім'я та по батькові здобувача)

1. Тема роботи «Проект системи автоматизації бухгалтерського та оперативного обліку з інтеграцією в ERP-середовище та використанням RFID-технологій» затверджено наказом ректора КНУБА №1636/23.2/25 від «30» вересня 2025 року
2. Керівник роботи к.т.н. Вишняков Володимир Михайлович
Доцент кафедри кібербезпеки та комп'ютерної інженерії
(прізвище, ім'я та по батькові, науковий ступінь, вчене звання)
3. Термін подання здобувачем роботи до захисту 15 грудня 2025 року.
4. Зміст пояснювальної записки за розділами:
 - Р. 1. Аналіз предметної області.
 - Р. 2. Проектування системи автоматизації обліку з використанням RFID.
 - Р. 3. Реалізація та оцінка ефективності впровадженої системи.

5. Графічний матеріал за розділами:

Р. 1.	_____
Р. 2.	_____ 7 рисунків
Р. 3.	_____ 3 рисунки

6. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта	Перевірів	
		дата	підпис
Розділ 1.	Шабала Є.Є., к.т.н, доцент		
Розділ 2.	Шабала Є.Є., к.т.н, доцент		
Розділ 3.	Шабала Є.Є., к.т.н, доцент		

7. Календарний план виконання роботи:

Види робіт та їх зміст	Дата виконання
Розділ 1. Аналіз предметної області та постановка задачі дослідження.	Вересень 2025 р.
Розділ 2. Проектування системи автоматизації обліку з використанням RFID	Жовтень 2025 р.
Розділ 3. Реалізація та оцінка ефективності впровадженої системи	Жовтень 2025 р.
Остаточне оформлення роботи	Листопад 2025 р.
Направлення роботи на рецензування, перевірку на плагіат	Грудень 2025 р.
Попередній захист роботи на кафедрі	Грудень 2025 р.

8. Дата видачі завдання: 30 вересня 2025 року.

Керівник

_____ (підпис)

_____ (прізвище та ініціали)

Студент

_____ (підпис)

_____ (прізвище та ініціали)

АНОТАЦІЯ

Нюкало Є.І. «Проект системи автоматизації бухгалтерського та оперативного обліку з інтеграцією в ERP-середовище та використанням RFID-технологій».

Тема дипломного проекту присвячена підвищенню економічної ефективності гірничодобувного підприємства шляхом розробки та впровадження системи автоматизованого обліку видобутої сировини. Система забезпечує контроль вантажопотоку на ваговій, дозволяє вираховувати КРІ-показники у погодинному розрізі та мінімізувати ризики збиткової діяльності. У роботі проведено аналіз предметної області, досліджено роль сучасних ERP-систем та здійснено класифікацію RFID-технологій. Виконано моделювання бізнес-процесів, розроблено архітектуру інтегрованої системи, обґрунтовано вибір апаратних засобів та спроектовано модель даних. Описано практичну реалізацію модуля інтеграції, наведено результати тестування та розрахунок техніко-економічної ефективності, що підтверджує доцільність впровадження.

Ключові слова: автоматизація обліку, гірничодобувна промисловість, ERP-система, RFID-технології, моделювання бізнес-процесів, вагова, КРІ, техніко-економічна ефективність.

SUMMARY

Niukalo Y.I. "Design of an Automated Accounting and Operations Management System Integrated into an ERP Environment Using RFID Technologies".

The topic of the diploma project is devoted to increasing the economic efficiency of a mining enterprise by developing and implementing an automated accounting system for extracted raw materials. The system ensures control of cargo flow at the weighbridge, allows for calculating KPI indicators on an hourly basis, and minimizing the risks of loss-making activities. The work conducts an analysis of the subject area, investigates the role of modern ERP systems, and classifies RFID technologies. Business process modeling is performed, the architecture of the integrated system is developed, the choice of hardware is justified, and a data model is designed. The practical implementation of the integration module is described, testing results and the calculation of technical and economic efficiency are presented, confirming the feasibility of implementation.

Keywords: accounting automation, mining industry, ERP system, RFID technologies, business process modeling, weighbridge, KPI, technical and economic efficiency.

РЕЗЮМЕ (SUMMARY) <i>до кваліфікаційної випускової роботи здобувача</i>	ПІБ Нюкало Євген Ігорович Niukalo Yevhen Ihorovuch		
ЗВО	Київський національний університет будівництва і архітектури		
Тема (українською та англійською)	Проект системи автоматизації бухгалтерського та оперативного обліку з інтеграцією в ERP-середовище та використанням RFID-технологій		
	Design of an Automated Accounting and Operations Management System Integrated into an ERP Environment Using RFID Technologies		
Освітній ступінь	Магістр		
Факультет	Автоматизації і інформаційних технологій		
Випускова кафедра	Кібербезпеки та комп'ютерної інженерії		
Спеціальність	Комп'ютерна інженерія		
Освітня програма	Комп'ютерні системи і мережі		
Керівник	Вишняков Володимир Михайлович		
Обсяг роботи:	<i>Пояснювальна записка, стор.</i>	<i>Розділів</i>	<i>Презентація, кількість слайдів</i>
	90 (104 з додатками)	3	22
Розділ 1	Аналіз предметної області та постановка задачі дослідження		
Розділ 2	Проектування системи автоматизації обліку з використанням RFID		
Розділ 3	Реалізація та оцінка ефективності впровадженої системи		
Висновки по роботі	Розроблено автономну систему обліку на базі RFID та ERP, що виключила людський фактор і збільшила пропускну здатність у 1,6 раза. Проект продемонстрував високу надійність (99,85%) та окупився менш ніж за один місяць.		
Ключові слова:	автоматизація обліку, гірничодобувна промисловість, ERP-система, RFID, моделювання бізнес-процесів, KPI, техніко-економічна ефективність.		
Keywords:	accounting automation, mining industry, ERP system, RFID, business process modeling, KPI, technical and economic efficiency.		

Здобувач _____ / _____

Керівник _____ / _____

ЗМІСТ

ВСТУП.....	10
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	12
1.1. Особливості ведення оперативного та бухгалтерського обліку у гірничодобувній промисловості	12
1.2. Аналіз сучасних ERP-систем та їх роль в автоматизації облікових процесів підприємства.....	15
1.3. Огляд та класифікація RFID-технологій для ідентифікації об'єктів у промислових умовах	18
1.4. Аналіз існуючих підходів до автоматизації обліку на базі інтеграції RFID та ERP-систем.....	21
1.5. Формулювання вимог до проєктованої системи.....	23
РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ АВТОМАТИЗАЦІЇ ОБЛІКУ З ВИКОРИСТАННЯМ RFID.....	27
2.1. Моделювання бізнес-процесу обліку видобутої сировини ("As Is" та "To Be")	27
2.2. Розробка архітектури інтегрованої системи автоматизації	29
2.3. Обґрунтування вибору ваговимірювального комплексу	33
2.4. Проєктування моделі даних для обміну між підсистемами	42
2.5. Розробка алгоритмів функціонування системи контролю та обліку на ваговій	46
2.6. Математичне моделювання пропускнуої здатності системи	48
2.7. Проєктування системи захисту інформації	51
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ВПРОВАДЖЕНОЇ СИСТЕМИ	55

3.1. Етапи та особливості впровадження системи на гірничодобувному підприємстві.....	55
3.2. Програмна реалізація модуля інтеграції RFID-обладнання з корпоративною ERP-системою.....	59
3.3. Розробка підсистеми моніторингу та оперативної аналітики в середовищі Power BI.....	62
3.4. Тестування системи та аналіз результатів дослідної експлуатації	70
3.5. Розрахунок техніко-економічної ефективності проєкту автоматизації	79
ВИСНОВОК.....	84
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	86
ДОДАТОК А.....	89

ВСТУП

Актуальність теми дослідження. В умовах цифрової трансформації економіки та посилення конкуренції, підприємства промислового сектору, зокрема гірничодобувної галузі, стикаються з гострою потребою у підвищенні ефективності управлінських процесів. Ключовим інструментом для досягнення цієї мети є впровадження сучасних інформаційних систем, здатних забезпечити керівництво точними та оперативними даними для прийняття обґрунтованих рішень.

Традиційні методи оперативного та бухгалтерського обліку на кар'єрах, що часто покладаються на ручне введення даних, не відповідають сучасним вимогам. Вони характеризуються високим ризиком виникнення помилок через людський фактор, затримками в отриманні інформації та створюють можливості для зловживань. Це призводить до викривлення даних про реальний обсяг видобутої сировини, ускладнює планування та контроль за виробничими процесами.

Вирішенням цієї проблеми є комплексна автоматизація, що поєднує потужність корпоративних ERP-систем (Enterprise Resource Planning) з передовими технологіями автоматичної ідентифікації. Технологія радіочастотної ідентифікації (RFID) зарекомендувала себе як надійний засіб для безконтактної ідентифікації об'єктів у складних промислових умовах. Інтеграція RFID-системи в єдине ERP-середовище підприємства дозволяє створити безшовний, автоматизований контур збору первинних даних, мінімізуючи втручання людини та забезпечуючи надходження інформації в облікову систему в режимі реального часу. Таким чином, розробка проекту такої інтегрованої системи є актуальною науково-практичною задачею.

Мета і завдання дослідження. Метою магістерської роботи є розробка та обґрунтування проекту системи автоматизації бухгалтерського та оперативного обліку на підприємстві на основі інтеграції RFID-технологій з корпоративною ERP-системою.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Проаналізувати специфіку облікових процесів у гірничодобувній галузі та виявити недоліки існуючих методів.

2. Провести огляд сучасних ERP-систем та RFID-технологій, обґрунтувати доцільність їх спільного використання.
3. Розробити архітектуру системи автоматизації, що забезпечує взаємодію апаратних засобів ідентифікації та облікового програмного забезпечення.
4. Спроекувати алгоритми функціонування системи для автоматичного збору, обробки та передачі даних про зважування транспорту.
5. Провести оцінку техніко-економічної ефективності від впровадження розробленого проєкту.

Об'єкт дослідження – процеси оперативного та бухгалтерського обліку на гірничодобувному підприємстві.

Предмет дослідження – моделі, методи та засоби інтеграції RFID-технологій з ERP-системою для автоматизації обліку руху сировини.

Методи дослідження. Для вирішення поставлених завдань у роботі використано методи системного аналізу, теорії інформаційних систем, методи моделювання бізнес-процесів (нотація BPMN), методи алгоритмізації, а також методи економічного аналізу для розрахунку ефективності інвестицій.

Наукова новизна отриманих результатів полягає в удосконаленні архітектури інтегрованої системи обліку для гірничодобувних підприємств, яка, на відміну від існуючих, забезпечує безперервну автоматичну передачу верифікованих даних від точки виникнення (ваговий комплекс) безпосередньо до модулів ERP-системи, що усуває розриви в інформаційних потоках та підвищує достовірність обліку.

Практичне значення отриманих результатів. Результати роботи впроваджено в діяльність ТОВ «ТРУД». Розроблена система дозволила підвищити точність обліку видобутої гранітної маси, скоротити час на проведення операції зважування, унеможливити несанкціонований вивіз продукції та забезпечити керівництво оперативною звітністю. Основні положення проєкту можуть бути використані при модернізації облікових систем на інших підприємствах схожого профілю.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1. Особливості ведення оперативного та бухгалтерського обліку у гірничодобувній промисловості

Гірничодобувна промисловість є однією з ключових фондо- та матеріаломістких галузей економіки, що формує базову сировинну ланку для будівництва, металургії та інших секторів. Специфіка виробничих процесів, таких як безперервність, масштабованість та територіальна розподіленість активів, висуває підвищені вимоги до організації системи управлінського, оперативного та бухгалтерського обліку. На відміну від дискретного виробництва, де облік ведеться поштучно, у добувній галузі ключовим об'єктом обліку є маса та об'єм сипучих матеріалів, що ускладнює процеси ідентифікації та контролю.

Основою для управління будь-яким гірничодобувним підприємством (ГДП) є точні дані про обсяг видобутої та переміщеної гірничої маси. Цей первинний показник є точкою відліку для всього ланцюга управлінських рішень:

В економічному аспекті: на його основі розраховується собівартість видобутку однієї тонни сировини, формуються відпускні ціни, ведеться розрахунок з покупцями.

В технологічному аспекті: дані про обсяги видобутку дозволяють оцінювати продуктивність обладнання (екскаваторів, самоскидів), ефективність роботи окремих змін та бригад, планувати графіки видобутку та розкривних робіт.

В фіскальному аспекті: обсяг видобутих корисних копалин є базою для нарахування рентної плати та інших податкових зобов'язань.

Нормативно-правовою базою для ведення обліку на підприємствах України є Закон України "Про бухгалтерський облік та фінансову звітність в Україні", який вимагає, щоб усі господарські операції були задокументовані первинними документами [1]. У контексті ГДП, операція зважування завантаженого самоскида на виїзді з кар'єру є саме такою первинною господарською операцією, а документ, що її засвідчує (запис у журналі, маршрутний лист), – первинним документом. Від

точності та достовірності цього первинного обліку напряму залежить якість всієї фінансової та управлінської звітності підприємства.

Аналіз традиційної моделі обліку на вагових комплексах

Класична модель організації обліку на вагових пунктах кар'єру, що досі поширена на багатьох підприємствах, базується на ручних операціях та паперовому документообігу. Розглянемо її етапи та іманентні недоліки більш детально.

1. Процедура ідентифікації та зважування. Водій самоскида, прибуваючи на ваговий комплекс, усно повідомляє номер транспортного засобу або цей номер позначений на капоті авто. Оператор візуально ідентифікує автомобіль і вручну запускає процес зважування, фіксуючи вагу брутто.

2. Документація операції. Отримані дані (дата, час, номер авто, вага) вносяться вагарем до прошнурованого журналу зважувань. Цей журнал є реєстром первинного обліку, однак його паперова форма створює значні ризики.

3. Передача даних. Накопичена за зміну або добу інформація передається до адміністративного корпусу (бухгалтерії, планового відділу) у вигляді цих журналів або виписаних на їх основі зведених відомостей. Цей процес є фізичним переміщенням носіїв інформації, що створює значний часовий лаг.

4. Вторинна обробка даних. Співробітник відповідального підрозділу вручну переносить дані з паперових носіїв до електронної облікової системи (наприклад, BAS:ERP, SAP тощо). На цьому етапі відбувається дублювання роботи та виникає найвища ймовірність механічних помилок.

Ключові проблеми та ризики традиційної системи обліку

Така організаційна структура породжує комплекс системних проблем, які можна класифікувати на три групи.

Група 1. Операційні та інформаційні розриви. Основний недолік – відсутність оперативності. Дані про видобуток стають доступними для аналізу та управління із запізненням, що унеможлиблює прийняття рішень у режимі реального часу. Наприклад, керівник кар'єру не може миттєво побачити простій транспорту або падіння інтенсивності вивозу сировини; ця інформація з'явиться лише у звіті за

попередній день. Це призводить до "реактивного" стилю управління замість проактивного.

Група 2. Ризики, пов'язані з людським фактором, та викривлення даних. Людський фактор є джерелом як ненавмисних помилок, так і навмисних зловживань.

Ненавмисні помилки:

- Помилки запису: вагар може неправильно почути або записати номер автомобіля, переплутати цифри ваги (наприклад, 45.7 т замість 47.5 т).
- Помилки введення: бухгалтер при перенесенні даних з журналу може допустити механічну помилку через втому або неуважність.
- Втрата даних: паперові журнали можуть бути пошкоджені або втрачені.

Навмисні зловживання (шахрайство):

- Змова: водій та вагар можуть увійти в змову для зниження реальної ваги вантажу. "Необлікована" сировина може бути реалізована поза касою підприємства.
- Реєстрація "фіктивних" рейсів: вагар може вносити до журналу записи про зважування автомобілів, які насправді не виїжджали з кар'єру, що призводить до прямого списання сировини.
- Маніпуляції з тарою: вага порожнього самоскида може змінюватись (наприклад, через налиплий бруд або залишки вантажу), але в ручному режимі для розрахунку ваги нетто часто використовується довідкове, а не фактичне значення.

Група 3. Економічна та процесна неефективність. Паперовий облік є економічно не вигідним. Він вимагає утримання штату вагарів та операторів для введення даних. Процес ручного зважування та оформлення документів може створювати черги на ваговій, що призводить до простою дорогої техніки та зменшення кількості рейсів за зміну. Витрати на виправлення помилок, виявлених у кінці звітної періоду, також можуть бути значними.

Висновки щодо пункту. Отже, аналіз традиційної системи обліку на гірничодобувних підприємствах показує її повну невідповідність сучасним

вимогам ефективного управління. Вона є джерелом недостовірної та несвоєчасної інформації, створює умови для помилок і зловживань та є економічно неефективною. Очевидною стає потреба в кардинальній реорганізації цього процесу шляхом впровадження інтегрованих автоматизованих систем, здатних забезпечити прозорий та безперервний потік даних від центру їх виникнення (вагового комплексу) до єдиного інформаційного простору підприємства.

1.2. Аналіз сучасних ERP-систем та їх роль в автоматизації облікових процесів підприємства

Перехід від фрагментарної автоматизації окремих функцій до комплексного управління інформаційними потоками є ключовим трендом у розвитку сучасних підприємств. Центральним інструментом такої трансформації виступають корпоративні інформаційні системи класу ERP (Enterprise Resource Planning – Планування ресурсів підприємства). ERP-система – це інтегрована програмна платформа, що дозволяє автоматизувати та об'єднати всі ключові бізнес-процеси підприємства в єдиному інформаційному середовищі [7].

Концептуальна відмінність ERP-систем від застарілих "клаптикових" систем автоматизації полягає в принципі єдиного джерела даних (Single Source of Truth). Замість набору ізольованих баз даних (окремо для бухгалтерії, складу, відділу продажів), ERP-система використовує централізовану базу даних. Коли співробітник одного підрозділу вносить інформацію про господарську операцію, ці дані миттєво стають доступними для всіх інших пов'язаних підрозділів, автоматично оновлюючи відповідні облікові реєстри.

Структура та функціональні модулі ERP-систем

Сучасні ERP-системи мають модульну архітектуру, що дозволяє підприємствам гнучко обирати та впроваджувати лише необхідний функціонал. Для підприємств гірничодобувної галузі ключовими є наступні модулі:

Управління фінансами (FI/FM): Це ядро системи, що відповідає за ведення бухгалтерського та податкового обліку, управління грошовими потоками, формування фінансової звітності відповідно до національних та міжнародних стандартів.

Управлінський облік (Controlling, CO): Модуль для контролю витрат, розрахунку собівартості продукції, аналізу прибутковості, бюджетування та внутрішньої звітності для менеджменту.

Управління матеріальними потоками (Materials Management, MM): Відповідає за процеси закупівлі, управління запасами сировини та матеріалів, складську логістику. Саме в цей модуль інтегруються дані про надходження видобутої сировини з кар'єру.

Збут та дистрибуція (Sales and Distribution, SD): Автоматизує процеси відвантаження готової продукції покупцям, ціноутворення, виставлення рахунків та контроль дебіторської заборгованості.

Планування виробництва (Production Planning, PP): Модуль для планування виробничих потужностей, формування виробничих завдань та контролю за їх виконанням.

Управління технічним обслуговуванням та ремонтами (Plant Maintenance, PM): Дозволяє планувати та обліковувати ремонти обладнання, включаючи кар'єрну техніку (самоскиди, екскаватори), що є критично важливим для забезпечення безперервності виробництва.

Огляд провідних ERP-рішень на ринку

На світовому та українському ринках представлено декілька провідних ERP-платформ, кожна з яких має свої переваги та особливості:

SAP S/4HANA: Лідер світового ринку ERP для великих корпорацій. Системи SAP відомі своєю надійністю, масштабованістю та глибоко пропрацьованими галузевими рішеннями, зокрема для гірничодобувної промисловості (SAP for Mining). Вони пропонують найкращі практики (best practices) для автоматизації бізнес-процесів, однак характеризуються високою вартістю впровадження та підтримки.

Oracle NetSuite / Fusion Cloud ERP: Один з головних конкурентів SAP, що активно просуває хмарні моделі розгортання ERP. Рішення Oracle є потужними та гнучкими, пропонуючи широкі можливості для аналітики та управління фінансами.

Microsoft Dynamics 365: Перевагою цієї системи є тісна інтеграція з іншими продуктами Microsoft (Office 365, Power BI, Azure). Це робить її привабливою для компаній, що вже використовують екосистему Microsoft, та забезпечує гнучкість у налаштуванні.

BAS ERP: Це сучасне рішення, що є розвитком популярних на території України та країн СНД систем на платформі "BAS:ERP". Ключовою перевагою систем BAS є їхня повна адаптація до вимог українського законодавства, стандартів бухгалтерського обліку та специфіки місцевого документообігу. Завдяки відкритості платформи та наявності великої кількості фахівців, системи BAS є гнучкими в доопрацюванні та інтеграції з зовнішнім обладнанням, що робить їх популярним вибором для українських промислових підприємств.

Роль ERP-системи як інтеграційної платформи

Ключова роль ERP-системи в контексті даного дослідження полягає не лише в автоматизації внутрішніх облікових процедур, але й у її здатності виступати центральним ядром (digital core), до якого підключаються зовнішні системи збору даних. Сучасні ERP-платформи розробляються з урахуванням концепції відкритості та мають для цього розвинені програмні інтерфейси (API – Application Programming Interface).

Саме ця можливість інтеграції дозволяє вирішити проблеми, описані в попередньому пункті. Підключивши до ERP-системи апаратні засоби (вагові комплекси, RFID-зчитувачі), можна налаштувати автоматичну передачу даних про господарську операцію в момент її здійснення. Таким чином, ERP-система перестає бути лише інструментом для бухгалтерів, куди дані вносяться із запізненням, а перетворюється на живу систему оперативного управління, що відображає реальні процеси на підприємстві в режимі реального часу.

Висновки щодо пункту. Отже, ERP-системи є стратегічною платформою для цифрової трансформації підприємства. Вони не тільки систематизують та автоматизують внутрішні облікові процеси, але й слугують технологічним фундаментом для інтеграції з передовими технологіями автоматичної ідентифікації. Для вирішення задачі автоматизації обліку на гірничодобувному

підприємстві ERP-система виконує роль надійного та контрольованого приймача первинних даних, забезпечуючи їх миттєве включення в єдиний інформаційний простір компанії.

1.3. Огляд та класифікація RFID-технологій для ідентифікації об'єктів у промислових умовах

Радіочастотна ідентифікація (RFID – Radio-Frequency Identification) – це технологія автоматичної ідентифікації та збору даних (AIDC), яка використовує радіохвилі для безконтактного зчитування або запису інформації, що зберігається на спеціальних носіях, відомих як RFID-мітки (або транспондери) [2]. На відміну від широко розповсюджених систем штрих-кодування, RFID не вимагає прямої видимості між зчитувачем та міткою, що робить цю технологію особливо придатною для використання у складних та агресивних промислових середовищах.

Будь-яка RFID-система складається з трьох фундаментальних компонентів:

RFID-мітка (Транспондер): Носій даних, що складається з мікрочипа (де зберігається унікальний ідентифікатор та інша інформація) та антени (для прийому та передачі сигналу).

RFID-зчитувач (Інтеррогатор): Пристрій, що генерує радіочастотне поле, активує мітки, що потрапляють у його зону дії, та зчитує (або записує) з них інформацію.

Проміжне програмне забезпечення (Middleware): Програмний шар, що фільтрує та агрегує необроблені дані, отримані від зчитувачів, і передає їх у структурованому вигляді до корпоративної системи вищого рівня, наприклад, ERP.

Для успішного проектування системи автоматизації необхідно провести класифікацію RFID-технологій, оскільки різні типи систем кардинально відрізняються за своїми характеристиками та сферами застосування.

Класифікація за типом живлення міток

Це найбільш значуща класифікація, що визначає дальність зчитування, вартість та життєвий цикл міток.

Пасивні RFID-мітки: Такі мітки не мають власного джерела живлення. Енергію, необхідну для активації мікрочипа та передачі відповіді, вони отримують (індукують) з електромагнітного поля, яке створює зчитувач.

Переваги: Низька вартість (дозволяє проводити масове маркування), малі розміри та вага, практично необмежений термін служби (оскільки відсутній елемент живлення).

Недоліки: Відносно невелика дальність зчитування (від кількох сантиметрів до 10-15 метрів, залежно від частоти).

Застосування: Логістика, ритейл, контроль доступу, маркування тари та активів.

Активні RFID-мітки: Ці мітки оснащені власним джерелом живлення (батареєю). Вони не залежать від енергії зчитувача і можуть самостійно генерувати та передавати сигнал на великі відстані.

Переваги: Велика дальність зчитування (часто понад 100 метрів), висока надійність передачі даних, можливість інтеграції з датчиками (температури, вібрації).

Недоліки: Висока вартість, більші габарити, обмежений термін служби, що визначається часом життя батареї (зазвичай 3-10 років).

Застосування: Моніторинг контейнерів у портах, відстеження цінних активів у реальному часі (RTLS), контроль за рухом транспорту на великих територіях.

Напівпасивні (або BAP – Battery-Assisted Passive) мітки: Це гібридний тип. Мітка має батарею, але вона використовується лише для живлення мікрочипа та датчиків, а не для передачі сигналу. Для зв'язку зі зчитувачем вона, як і пасивна мітка, використовує енергію його поля.

Переваги: Більша дальність та надійність зчитування, ніж у пасивних міток, особливо в складних умовах.

Недоліки: Дорожчі за пасивні, обмежений термін служби батареї.

Класифікація за робочим частотним діапазоном

Частотний діапазон визначає фізичні характеристики системи: дальність, швидкість читання та стійкість до перешкод.

Низькочастотні (LF – Low Frequency): 125–134 кГц

Характеристики: Мала дальність зчитування (до 10-20 см). Сигнал слабо поглинається рідинами та біологічними тканинами, стійкий до електромагнітних перешкод.

Застосування: Системи контролю доступу (СКУД), чіпування тварин, іммобілайзери в автомобільних ключах. Саме цей тип часто використовується на прохідних підприємств.

Високочастотні (HF – High Frequency): 13.56 МГц

Характеристики: Середня дальність зчитування (до 1 метра). Підтримують вищу швидкість обміну даними та більший обсяг пам'яті мітки. На цій частоті працює технологія NFC (Near Field Communication).

Застосування: Бібліотечні системи, безконтактні платіжні картки, облік документів.

Ультрависокочастотні (UHF – Ultra-High Frequency): 860–960 МГц

Характеристики: Велика дальність зчитування (до 15-20 метрів для пасивних міток). Висока швидкість, можливість одночасного зчитування сотень міток у зоні (функція anti-collision).

Недоліки: Сигнал сильно поглинається рідинами (включаючи тіло людини) та відбивається від металевих поверхонь, що створює "мертві зони" та вимагає ретельного проектування системи.

Застосування: Складська логістика, управління ланцюгами поставок, інвентаризація, автоматизація паркінгів та контроль руху транспорту.

Специфіка застосування RFID у промисловості та гірничій справі

Промислові умови, особливо на кар'єрах, є вкрай агресивним середовищем для будь-якої електроніки. Пил, бруд, волога, вібрації, екстремальні температури та велика кількість металу (техніка, конструкції) – все це є серйозними викликами для RFID.

Переваги над штрих-кодами: У той час як штрих-код стає нечитабельним при найменшому забрудненні чи пошкодженні, RFID-мітка, особливо в захищеному

промислового корпусі, продовжує стабільно працювати. Відсутність необхідності прямої видимості дозволяє зчитувати мітку незалежно від її забруднення.

Проблема "металу та рідин": Для UHF-діапазону, який є найбільш привабливим для ідентифікації транспорту через велику дальність, наявність металевих поверхонь (рама, кузов БелАЗу) є критичною проблемою. Радіохвилі відбиваються від металу, що призводить до інтерференції та унеможливорює зчитування стандартної мітки.

Для вирішення цієї проблеми були розроблені спеціалізовані "On-Metal" (металостійкі) UHF-мітки. Їхня конструкція передбачає наявність спеціального ізолюючого шару або особливу конфігурацію антени, яка використовує металеву поверхню, на якій вона встановлена, як частину своєї антенної системи (рефлектор), що навіть може посилювати сигнал у потрібному напрямку.

Висновки щодо пункту. Аналіз показує, що для задачі автоматичної ідентифікації великовантажних самоскидів на вагових комплексах у жорстких умовах кар'єру, оптимальним вибором є система на базі пасивних UHF RFID-міток. Вони забезпечують необхідну дальність зчитування (кілька метрів) для детекції автомобіля на вагах, не потребують обслуговування (заміни батарей) та є економічно ефективними для маркування великого парку техніки. При цьому для забезпечення стабільного зчитування з металевих кузовів самоскидів (БелАЗів) необхідно використовувати виключно спеціалізовані промислові металостійкі (on-metal) UHF-мітки.

1.4. Аналіз існуючих підходів до автоматизації обліку на базі інтеграції RFID та ERP-систем

Інтеграція технології радіочастотної ідентифікації з корпоративними ERP-системами є доведеним та ефективним підходом до автоматизації бізнес-процесів, пов'язаних з обліком та відстеженням руху матеріальних об'єктів. Ця синергія дозволяє усунути головний недолік традиційних облікових систем – розрив між моментом фізичного здійснення операції та її відображенням в інформаційній системі. Розглянемо ключові сфери застосування та архітектурні моделі таких інтегрованих рішень.

Сфери застосування інтегрованих RFID-ERP рішень

Аналіз наукових публікацій та практичних кейсів впровадження дозволяє виділити кілька типових напрямків, де інтеграція RFID та ERP дає максимальний економічний ефект:

Складська логістика та управління ланцюгами поставок. Це найбільш класична сфера застосування. Маркування палет, контейнерів та окремих товарів RFID-мітками дозволяє автоматизувати процеси приймання, розміщення, комплектації та відвантаження товарів. RFID-зчитувачі, встановлені на воротах складу чи навантажувачах, автоматично реєструють усі переміщення та передають дані в ERP-систему, яка в режимі реального часу оновлює складські залишки та створює необхідні документи (наприклад, акт приймання товару).

Виробничий облік (Work-in-Progress, WIP). У дискретному виробництві RFID-мітки кріпляться на заготовки, деталі або складальні одиниці. На кожному етапі виробничого конвеєра зчитувачі фіксують проходження виробу, що дозволяє ERP-системі автоматично відстежувати статус виконання виробничого замовлення, контролювати вузькі місця та розраховувати фактичну собівартість.

Управління основними засобами та активами. Маркування цінного обладнання, інструментів, IT-техніки або транспортних засобів дозволяє автоматизувати процеси інвентаризації, контролювати видачу та повернення активів, а також запобігати їх втраті чи крадіжці. Дані про місцезнаходження та статус активу автоматично передаються в модуль управління основними засобами ERP-системи.

Контроль доступу та руху транспорту. Автоматизація прохідних та контрольних-пропускних пунктів є ще одним поширеним завданням. Транспортні засоби оснащуються RFID-мітками, а зчитувачі на в'їзді/виїзді фіксують час їх прибуття та відбуття, автоматично піднімаючи шлагбаум та створюючи запис у відповідному модулі ERP.

Архітектурні моделі інтеграції

Технічно взаємодія між апаратним рівнем (RFID-зчитувачі) та програмним (ERP-система) може бути реалізована за двома основними моделями.

Модель прямої інтеграції. У найпростішому випадку зчитувач може бути підключений безпосередньо до комп'ютера, на якому запущено клієнтський додаток ERP-системи. Цей підхід є виправданим лише для дуже простих систем з одним зчитувачем та однією точкою контролю. Його головний недолік – відсутність гнучкості, масштабованості та централізованого управління.

Модель інтеграції через проміжне програмне забезпечення (Middleware). Це стандартна та найбільш поширена архітектура для промислових рішень. Middleware є спеціалізованим програмним шаром, що виступає посередником між RFID-обладнанням та корпоративними системами.

Функції Middleware включають:

- Агрегацію даних: Збір інформації з десятків та сотень зчитувачів.
- Фільтрацію та обробку: Очищення "сирих" даних від дублікатів (один автомобіль може бути прочитаний 50 разів за 3 секунди) та помилкових зчитувань.
- Контекстуалізацію: Перетворення низькорівневих даних (наприклад, "зчитувач №5 побачив мітку ID 123AB") на подію бізнес-логіки ("автомобіль БелАЗ-07 заїхав на ваги №1").
- Інтеграцію з ERP: Форматування даних у зрозумілий для ERP-системи формат та їх передача через стандартизований програмний інтерфейс (API).

Така архітектура є значно більш надійною, гнучкою та масштабованою, оскільки вона відділяє апаратний рівень від бізнес-логіки, реалізованої в ERP-системі.

1.5. Формулювання вимог до проектованої системи

Проведений у розділі 1 аналіз проблем предметної області та доступних технологічних рішень (ERP та RFID) дозволяє сформулювати чіткий набір вимог до проектованої системи автоматизації. Ці вимоги є, по суті, технічним завданням для розробки, що має на меті усунення виявлених недоліків традиційної системи обліку.

Вимоги доцільно класифікувати на дві основні групи: функціональні та нефункціональні (експлуатаційні).

1. Функціональні вимоги (Functional Requirements)

Ця група описує, які саме функції та операції повинна виконувати система:

Автоматична ідентифікація: Система повинна забезпечувати 100% автоматичну та безконтактну ідентифікацію транспортного засобу (БелАЗ), оснащеного RFID-міткою, при його заїзді на автомобільні ваги.

Автоматичне отримання даних: Система повинна автоматично, без втручання оператора, отримувати дані про вагу бруто з вагового контролера (термінала).

Інтеграція з довідниками ERP: Система повинна в режимі реального часу мати доступ до довідника "Транспортні засоби" в ERP-системі для отримання ваги тари (порожнього автомобіля) за ідентифікованою RFID-міткою.

Автоматизований розрахунок: Система повинна самостійно розраховувати вагу нетто (корисного вантажу) за формулою: $Вага_Нетто = Вага_Брутто - Вага_Тари$.

Автоматичне створення документів: За фактом успішного зважування система повинна автоматично створювати та проводити в ERP-системі первинний обліковий документ (наприклад, "Надходження сировини з кар'єру", "Видобуток"), заповнюючи всі необхідні реквізити (автомобіль, водій, вага, час).

Запобігання помилкам та зловживанням: Система повинна блокувати можливість ручного коригування даних оператором вагової. Мають бути реалізовані механізми контролю дублікатів (запобігання повторному зважуванню одного й того ж рейсу) та ідентифікації "чужого" транспорту (автомобілів, що не зареєстровані в ERP-системі).

Обробка позаштатних ситуацій: Система повинна мати регламентований інтерфейс для оператора на випадок об'єктивних збоїв (наприклад, пошкодження RFID-мітки або збій вагового обладнання). Такі операції мають окремо протоколюватися.

2. Нефункціональні вимоги (Non-functional Requirements)

Ця група визначає, як саме система має виконувати свої функції, та описує її експлуатаційні характеристики:

- **Надійність та відмовостійкість:** Апаратні компоненти системи (зчитувачі, антени, мітки) повинні бути розраховані на безперебійну роботу 24/7 в

агресивному промислового середовищі кар'єру (пил, бруд, волога, вібрації, широкий діапазон температур від -30°C до $+50^{\circ}\text{C}$) . Обладнання повинно мати відповідний клас промислового захисту (наприклад, IP67).

- Точність ідентифікації: Ймовірність коректного зчитування RFID-мітки при проходженні автомобіля через зону контролю повинна становити не менше 99,9%.
- Швидкодія: Загальний час циклу (ідентифікація, зважування, передача даних в ERP) не повинен перевищувати 5-7 секунд, щоб не створювати черг на ваговій.
- Масштабованість: Архітектура системи повинна передбачати можливість легкого підключення додаткових точок контролю (наприклад, нових вагових комплексів або КПП) без необхідності повної перебудови ядра системи.
- Сумісність (Interoperability): Система повинна використовувати стандартизовані протоколи обміну даними (наприклад, HTTP/SOAP, JSON/REST API) для взаємодії з ERP-системою та обладнанням, що забезпечить легкість інтеграції.
- Безпека: Має бути забезпечено захист даних при передачі між компонентами системи (Middleware та ERP) для унеможливлення їх перехоплення чи підміни.

Сформульовані вимоги є основою для прийняття проєктних рішень у наступному розділі, зокрема для вибору архітектури системи, конкретних апаратних засобів та розробки алгоритмів програмного забезпечення.

Висновки до розділу 1

У першому розділі магістерської роботи було проведено комплексний аналіз предметної області та обґрунтовано необхідність розробки системи автоматизації обліку для гірничодобувного підприємства.

В ході дослідження було встановлено, що традиційні методи оперативного та бухгалтерського обліку, засновані на ручній праці, мають системні недоліки . Ключовими серед них є низька оперативність надходження даних, високий ризик виникнення помилок через людський фактор , а також наявність умов для

навмисних зловживань , що в сукупності призводить до викривлення реальної картини виробничих процесів.

Було проаналізовано сучасні ERP-системи, які визначено як центральне ядро інформаційної інфраструктури підприємства . Зроблено висновок, що їхній потенціал може бути розкритий лише за умови інтеграції з надійними засобами автоматичного збору первинних даних.

Як такий засіб було детально розглянуто технологію RFID. На основі класифікації обґрунтовано, що для задачі ідентифікації великовантажної техніки в складних умовах кар'єру оптимальним рішенням є використання пасивних UHF RFID-міток у металостійкому виконанні .

На основі аналізу існуючих підходів було визначено, що найефективнішою архітектурною моделлю для поєднання апаратного та програмного рівнів є інтеграція через проміжне програмне забезпечення (Middleware) .

Результати проведеного аналізу дозволили сформулювати деталізовані функціональні та нефункціональні вимоги до проєктованої системи (пункт 1.5), які є вихідними даними для подальшої розробки. Таким чином, висновки даного розділу є теоретичним та методологічним підґрунтям для проєктування архітектури та алгоритмів функціонування системи, що буде детально розглянуто в наступному розділі.

РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ АВТОМАТИЗАЦІЇ ОБЛІКУ З ВИКОРИСТАННЯМ RFID

2.1. Моделювання бізнес-процесу обліку видобутої сировини ("As Is" та "To Be")

Першим етапом проєктування системи є детальний аналіз існуючих бізнес-процесів та побудова моделі їх оптимізації. Для цього використано метод порівняльного аналізу моделей «As Is» («як є») та «To Be» («як має бути»).

Аналіз моделі "As Is" (Існуючий стан)

До впровадження автоматизації процес обліку видобутої гірничої маси здійснювався в «ручному» режимі із залученням персоналу вагової (вагарів).

Процес виглядав наступним чином:

1. Заїзд транспорту: Технологічний транспорт (самоскид) із вантажем заїжджав на платформу вагової.
2. Фіксація ваги: Вагар візуально зчитував показники з терміналу ваги.
3. Реєстрація даних: Отримані дані (вага бруто) записувалися вручну у паперовий журнал або таблицю Excel.
4. Комунікація: Після запису вагар подавав візуальний сигнал водієві (жестом через вікно) про дозвіл на з'їзд з вагової.
5. Розрахунок: В кінці зміни або періоду вагар вручну розраховував вагу нетто (віднімаючи тару) та підбивав підсумки зміни.

Критичні недоліки моделі "As Is":

- Конфлікт задач: Вагарі паралельно виконували функції відпуску готової товарної продукції (продаж щебеню/відсіву клієнтам) та обліку внутрішнього переміщення гірничої маси. Одночасне надходження клієнтського транспорту та технологічних самоскидів призводило до черг та перевантаження персоналу.
- Людський фактор: Ручне перенесення цифр з терміналу в журнал призводило до механічних помилок, виправлення яких займало значний час.

- Відсутність оперативності: Керівництво отримувало дані про виробіток лише постфактум, після обробки журналів, що унеможливило оперативне керування кар'єром протягом зміни.

Формування моделі "To Be" (Проектне рішення)

Проектована система повністю виключає вагара з ланцюжка реєстрації технологічного транспорту. Процес трансформується в автоматизований потік подій:

1. Ідентифікація: Самоскид під'їжджає до вагової. Встановлена антена зчитує пасивну RFID-мітку на лобовому склі автомобіля.
2. Управління рухом: Система через IP-реле керує світлофором. Вхідний світлофор перемикається на червоний (якщо слідує інше авто), а система починає стабілізацію ваги.
3. Автоматичне зважування: Програмне забезпечення фіксує стабільну вагу брутто.
4. Обробка даних в BAS ERP:
 - Система звертається до довідника «Транспортні засоби» в BAS ERP.
 - За ID мітки знаходиться відповідний автомобіль та його актуальна вага тари (яка регулярно оновлюється шляхом контрольного переважування).
 - Автоматично розраховується вага нетто (Брутто - Тара).
 - Створюється документ «Зважування» в базі даних.
5. Завершення циклу: Після успішного запису даних IP-реле перемикає вихідний світлофор на зелений колір, що є сигналом для водія звільнити платформу.

Переваги моделі "To Be":

- Повна автоматизація: Втручання людини потрібне лише для контролю позаштатних ситуацій.
- Розділення потоків: Вагар може зосередитися виключно на обслуговуванні клієнтів (продаж продукції), тоді як облік видобутку відбувається фонові.

- Аналітика в реальному часі: Дані миттєво потрапляють в облікову систему BAS ERP, звідки транслюються в аналітичні дашборди Power BI. Це дозволяє розраховувати КРІ (тонн/годину, витрати/тонну) в режимі on-line.

Таблиця 2.1. Порівняльна характеристика процесів обліку «As Is» та «To Be»

Критерій порівняння	Модель «As Is» (Поточний стан)	Модель «To Be» (Проектне рішення)
Ідентифікація ТЗ	Візуальна (оператором)	Автоматична (RFID-мітка)
Реєстрація ваги	Ручний запис у журнал	Автоматичний запис в ERP
Час операції	3-5 хвилин	15-30 секунд
Людський фактор	Високий ризик помилок	Виключено (автоматизація)
Вплив на персонал	Відволікання вагаря від клієнтів	Процес відбувається фоново
Доступність даних	Із затримкою (кінець зміни/добы)	У режимі реального часу (On-line)

2.2. Розробка архітектури інтегрованої системи автоматизації

Архітектура розробленої системи являє собою складний апаратно-програмний комплекс, що об'єднує різноманітні технологічні засоби (промислове обладнання) та інформаційні системи (ERP, BI) в єдиний контур управління. Для забезпечення масштабованості та відмовостійкості було обрано багаторівневу архітектуру, декомпозовану на три основні рівні: апаратний, мережевий та програмний.

2.2.1. Архітектура апаратної взаємодії (Hardware Layer)

На нижньому рівні системи знаходиться фізичне обладнання, розміщене безпосередньо на об'єкті автоматизації (ваговій). Взаємодія компонентів на цьому рівні базується на перетворенні фізичних величин та сигналів у цифрові дані.

1. Підсистема ідентифікації (RFID):

Ключовим елементом є стаціонарний UHF-зчитувач, що працює в діапазоні 860–960 МГц (стандарт EPC Gen2 / ISO 18000-6C). Зчитувач генерує електромагнітне поле через підключені антени, яке активує пасивну мітку на лобовому склі

самоскида. Мітка модулює сигнал відповіді, передаючи свій унікальний ідентифікатор (TID/EPC), що дозволяє однозначно ідентифікувати транспортний засіб.

2. Підсистема вимірювання (Weighing):

Основним вимірювальним елементом системи обрано автомобільні ваги ТВА-80 [4] виробництва НВП «Техноваги». Дана модель розроблена для експлуатації в жорстких промислових умовах та має наступні конструктивні й метрологічні характеристики:

- Вантажопідйомність: Максимальна межа зважування (ММЗ) становить 80 тонн, що повністю покриває потреби обліку завантажених кар'єрних самоскидів. Допустиме навантаження на вісь складає до 15 тонн.
- Конструктивні особливості платформи: В основі ваг лежить ортотропна конструкція платформи. Це означає використання закритого профілю ребер жорсткості, що забезпечує підвищену міцність на вигин та кручення порівняно зі звичайними двотавровими балками. Платформа є модульною (складається з 3–4 секцій), що спрощує транспортування та монтаж без необхідності зварювальних робіт на об'єкті. Конструкція колійного типу з шириною трапу 1100 мм та просвітом між ними дозволяє самоочищення від бруду та снігу, що є критичним для кар'єру.
- Тензометричні датчики: Вимірювальна система базується на комплекті високоточних тензодатчиків (аналогових або цифрових, класу точності С3/С5). Корпус датчиків виконано з нержавіючої сталі з найвищим ступенем пило-вологозахисту IP68/IP69K. Це гарантує повну герметичність навіть при зануренні у воду або митті під високим тиском гарячою парою. Датчики мають значний запас міцності (перевантаження до 300%), що захищає їх від динамічних ударів при заїзді важкої техніки.
- Ваговий термінал (вторинний перетворювач): Система комплектується терміналом серії TWP (наприклад, TWP-37), оснащеним 7-дюймовим сенсорним екраном. Термінал має вбудовану енергонезалежну пам'ять для

автономного зберігання журналів зважування у випадку втрати зв'язку з сервером.

- Інтерфейси та інтеграція: Термінал підтримує широкий спектр комунікаційних інтерфейсів: Ethernet, RS-232/RS-485, USB. Це дозволяє інтегрувати ваги в єдину мережу підприємства (LAN/Wi-Fi) та передавати дані безпосередньо в ERP-систему (BAS) в режимі реального часу, минаючи ручне введення.



Рисунок 2.1 - Вигляд вагового комплексу TVA-80

3. Підсистема управління трафіком (Automation Control):

Включає світлофорну сигналізацію та IP-реле. IP-реле виступає виконавчим механізмом, який отримує логічні команди від системи («Дозвіл на виїзд») і замикає електричні ланцюги живлення зеленого сигналу світлофора, фізично регулюючи рух через вагову.

2.2.2. Мережева інфраструктура та комунікації (Network Layer)

Надійність передачі даних є критичною умовою роботи системи, оскільки ваговий комплекс знаходиться на значній відстані від серверної інфраструктури.

- Топологія мережі: Використано топологію «зірка», де всі периферійні пристрої (зчитувач, ваговий термінал, IP-реле) підключені до локального комутатора (Switch) на посту вагара.

- Канал зв'язку: Комутатор з'єднаний з корпоративною мережею підприємства через захищений канал зв'язку (оптоволоконна лінія [11] або промисловий Wi-Fi міст), що забезпечує стабільний доступ до сервера додатків.
- Протоколи обміну:
 - Взаємодія з RFID-зчитувачем відбувається через протокол LLRP (Low Level Reader Protocol) або TCP/IP сокет, що забезпечує миттєву передачу подій зчитування.
 - Ваговий термінал передає дані за протоколом Modbus TCP або через послідовний інтерфейс RS-232/485, конвертований у Ethernet.

2.2.3. Програмна архітектура та інтеграція даних (Software & Data Layer)

Верхній рівень системи реалізовано на базі платформи BAS ERP, яка виступає ядром бізнес-логіки. Програмна архітектура побудована за подієво-орієнтованим принципом (Event-Driven Architecture).

1. Модуль інтеграції (Middleware/Driver):

Спеціалізований програмний модуль, який здійснює опитування портів обладнання. Його алгоритм включає:

- Отримання ID мітки від зчитувача.
- Моніторинг стабілізації ваги
- Фільтрацію хибних спрацювань та передачу валідованих даних у ядро системи.

2. Облікова система (BAS ERP):

Виконує функцію центральної бази даних та обробника транзакцій. Система зіставляє отриманий ID мітки з довідником «Основні засоби», визначає вагу тари, автоматично розраховує нетто та формує документ «Зважування».

3. Рівень бізнес-аналітики (BI):

Для візуалізації результатів використано Microsoft Power BI. Система підключається безпосередньо до SQL-бази даних ERP, забирає накопичені дані та трансформує їх у інтерактивні звіти (KPI, динаміка видобутку за годину, план-фактний аналіз), доступні для керівництва в режимі реального часу.

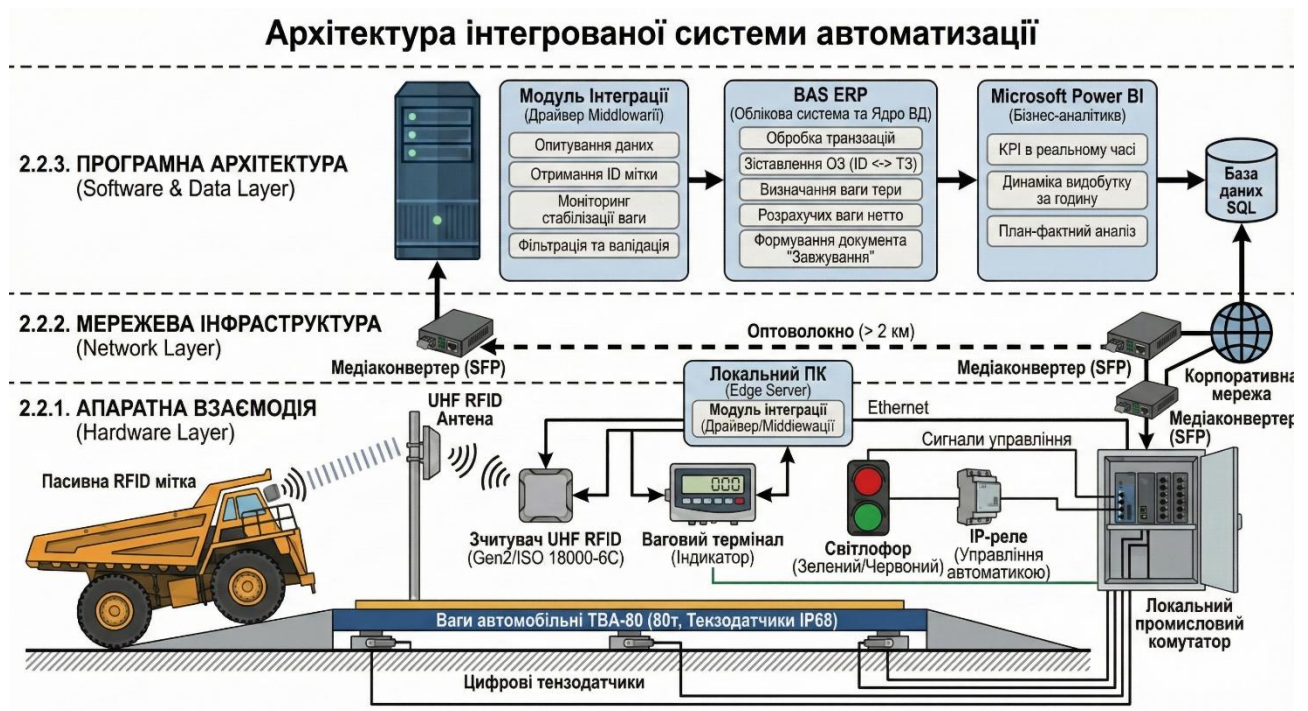


Рисунок 2.2 – Схема інтегрованої системи автоматизації

2.3. Обґрунтування вибору ваговимірювального комплексу

Основним елементом апаратної частини системи є ваговимірювальний комплекс. Враховуючи специфіку об'єкта автоматизації (гірничодобувний кар'єр), обладнання повинно відповідати жорстким вимогам щодо навантажувальності, пило- та вологозахисту, а також інтеграційних можливостей.

На основі аналізу ринку промислових ваг для реалізації проєкту було обрано ваги автомобільні тензометричні моделі ТВА-80 виробництва НВП «Техноваги» (Україна). Вибір даної моделі обумовлений сукупністю наступних технічних та експлуатаційних характеристик:

1. Конструктивні особливості вантажоприймальної платформи. Ключовою перевагою моделі ТВА-80 є використання ортотропної конструкції платформи. На відміну від стандартних рішень на базі двотаврових балок, ортотропна конструкція передбачає використання закритого профілю ребер жорсткості.

- **Механічна стійкість:** Така геометрія забезпечує підвищену жорсткість на вигин та кручення, що є критичним при зважуванні кар'єрних самоскидів, які можуть створювати значні динамічні навантаження при гальмуванні на платформі.

- Модульність: Ваги складаються з розбірних модулів, що спрощує їх транспортування та монтаж без необхідності проведення складних зварювальних робіт безпосередньо на об'єкті.
- Тип встановлення: Обрано варіант поверхневого встановлення (з пандусами), що, на відміну від врізного (у приямок), забезпечує кращу вентиляцію підплатформеного простору, легкий доступ для сервісного обслуговування датчиків та запобігає затопленню обладнання ґрунтовими водами або опадами.

2. Метрологічні характеристики та сенсорика Вимірювальна система комплексу базується на використанні тензометричних датчиків стиснення (типу RC3 або аналогів) з наступними параметрами:

- Вантажопідйомність: Максимальна межа зважування (ММЗ) становить 80 тонн, що повністю покриває масу завантаженого чотиривісного самоскида (типу Volvo FMX або Scania G-series).
- Клас захисту: Датчики виконані з нержавіючої сталі із застосуванням лазерного зварювання корпусу, що забезпечує клас захисту IP68/IP69K. Це гарантує повну герметичність та працездатність при зануренні у воду, наявності агресивного абразивного пилю та дозволяє проводити очищення струменем води під високим тиском.
- Перевантажувальна здатність: Датчики витримують безпечно перевантаження до 150% та руйнівне навантаження до 300% від номіналу, що забезпечує запас міцності при аварійних ситуаціях.

3. Вторинний вимірювальний перетворювач (Термінал) Система комплектується ваговим терміналом серії TWP (Technowagy Weight Processor), який виконує функції живлення датчиків, АЦП-перетворення сигналу та цифрової фільтрації завад.

- Комунікаційні інтерфейси: Термінал оснащено портом RS-485 з підтримкою відкритого промислового протоколу Modbus RTU/TCP. Це є вирішальним фактором для автоматизації, оскільки дозволяє інтегрувати ваги в локальну мережу підприємства (через перетворювач Ethernet) та отримувати потік

даних про вагу безпосередньо в програмне забезпечення (Middleware) без використання пропрієтарного закритого ПЗ.

- Функціональність: Наявність вбудованих алгоритмів стабілізації ваги дозволяє коректно фіксувати масу навіть при незначних коливаннях підвіски автомобіля або впливі вітру.

Таким чином, вибір комплексу ТВА-80 є технічно обґрунтованим рішенням, що поєднує високу механічну надійність ортотропної платформи, захищеність сенсорів класу IP68 та відкритість протоколів обміну даними.

2.3.1. Обґрунтування вибору технології та засобів ідентифікації

Завдання автоматичної ідентифікації транспортних засобів (ТЗ) є критичним етапом бізнес-процесу зважування, оскільки помилка на цьому етапі призводить до некоректного обліку вантажу. В рамках проектування розглядалися дві основні технології, що застосовуються в промисловій автоматизації:

1. Оптичне розпізнавання символів (OCR / LPR) — ідентифікація за державним номерним знаком за допомогою камер відеоспостереження.
2. Радіочастотна ідентифікація (RFID) — ідентифікація за унікальним кодом електронної мітки.

А) Аналіз технології оптичного розпізнавання (OCR)

Системи LPR (License Plate Recognition) базуються на алгоритмах комп'ютерного зору.

- Переваги: Відсутність витрат на обладнання самих автомобілів (не потрібні мітки).
- Недоліки в умовах кар'єру:
 - Чутливість до забруднення: Специфіка гірничодобувного виробництва передбачає роботу в умовах сильного забруднення. Шар пилу або бруду на номерному знаку робить його нечитабельним для камери, знижуючи ймовірність розпізнавання до критично низьких значень (< 70%).

- Залежність від освітлення: Робота в нічну зміну або в туман вимагає встановлення потужних інфрачервоних прожекторів та дорогих камер з високою світлочутливістю та функцією WDR (Wide Dynamic Range).
- Обчислювальна складність: Для обробки відеопотоку в реальному часі потрібні сервери з високою обчислювальною потужністю (GPU), що збільшує вартість впровадження.

Б) Аналіз технології RFID UHF (Обране рішення)

Технологія радіочастотної ідентифікації в ультрависокому діапазоні частот (UHF 860–960 МГц) базується на обміні даними між зчитувачем та пасивною міткою (без джерела живлення) по радіоканалу.

- Стандарт: Обрано стандарт EPC Gen2 (ISO/IEC 18000-6C)¹, який є світовим стандартом для логістики та транспорту.
- Фізика процесу: Радіохвилі UHF-діапазону здатні проникати крізь неметалеві перешкоди (скло, пластик, шар бруду/пилу). Це гарантує стабільне зчитування навіть якщо автомобіль повністю вкритий брудом, оскільки мітка знаходиться всередині кабіни на лобовому склі.
- Швидкість та дальність: Технологія дозволяє ідентифікувати об'єкт на відстані до 10–12 метрів на швидкості до 60 км/год, що забезпечує беззупинний проїзд (Free Flow) за необхідності.

В) Порівняльний аналіз та висновок

Для остаточного вибору було проведено порівняння за ключовими критеріями ефективності (Таблиця 2.2).

Таблиця 2.2 — Порівняльний аналіз технологій ідентифікації

Критерій порівняння	Оптичне розпізнавання (OCR)	RFID (UHF Gen2)	Перевага
Середовище передачі	Оптичний діапазон (світло)	Радіочастотний спектр	RFID
Вплив погодних умов	Високий (туман, дощ, сніг, ніч)	Відсутній / Мінімальний	RFID
Стійкість до бруду	Низька (брудний номер = помилка)	Висока (мітка в кабіні)	RFID
Захист від підміни	Низький (легко змінити номери)	Високий (Tamper Proof мітки)	RFID
Вартість точки контролю	Висока (Камера + ПЗ + Сервер)	Середня (Зчитувач + Антена)	RFID
Експлуатаційні витрати	Високі (постійне миття номерів)	Низькі (разова поклейка)	RFID

Обґрунтування вибору засобів ідентифікації:

На основі аналізу прийнято рішення використовувати технологію RFID UHF (ISO 18000-6C).

1. Зчитувач: Обрано стаціонарний UHF-зчитувач із підтримкою інтерфейсу Ethernet для прямої інтеграції в мережу.
2. Ідентифікатори: Використовуються пасивні мітки типу Windshield Tag із захистом від зняття (Tamper Proof). При спробі відклеїти мітку для передачі іншому водієві, антена мітки руйнується, і вона стає неактивною. Це апаратно вирішує проблему шахрайства ("накрутки" рейсів), що є критичним для економічної безпеки підприємства.

2.3.2. Обґрунтування архітектури обчислювальних засобів (Edge Computing)

Враховуючи територіальну розподіленість об'єкта автоматизації (відстань від вагової до серверної перевищує 2 км) та вимоги до безперервності технологічного процесу, вибір архітектури обчислювальних засобів є стратегічним рішенням.

У класичних системах автоматизації часто використовується централізована архітектура (Cloud/Server-centric), де периферійні пристрої передають «сирі» дані безпосередньо на сервер для обробки. Однак, аналіз ризиків показав неприйнятність такого підходу для даного проєкту.

Недоліки централізованої архітектури для умов кар'єру:

1. Залежність від каналу зв'язку: При пошкодженні оптоволоконної лінії (наприклад, екскаватором) робота вагової повністю зупиниться, що призведе до простою видобутку.
2. Затримки (Latency): Передача сигналу на сервер, його обробка та повернення команди на відкриття шлагбаума може займати час, що є критичним для інтенсивного трафіку.

А) Вибір архітектури Edge Computing

Для усунення зазначених недоліків було обрано архітектуру Edge [3] Computing (граничних/периферійних обчислень). Ця парадигма передбачає перенесення логіки обробки даних ближче до джерела їх виникнення.

Роль локального вузла (Edge Node):

На ваговій встановлюється локальний контролер (промисловий ПК), який виконує роль шлюзу та сервера прийняття рішень.

- Автономність: Логіка управління світлофорами та фіксації ваги виконується локально. Система продовжує працювати навіть при повній відсутності зв'язку з центральним офісом.
- Буферизація (Store-and-Forward): У разі відсутності мережі транзакції зберігаються у локальній базі даних (SQLite). При відновленні з'єднання відбувається автоматична реплікація накопичених даних на сервер BAS ERP.

- Фільтрація: Локальний ПК відсіює «шум» (нестабільну вагу, повторні зчитування міток), передаючи в мережу лише валідовані бізнес-події, що знижує навантаження на канал зв'язку.

Б) Вибір апаратної платформи (Industrial PC)

Звичайні офісні комп'ютери не придатні для роботи на ваговій через високий рівень запиленості, вібрації від проходження вантажівок та температурні коливання (вагова часто не опалюється належним чином).

Для реалізації Edge Node обрано бескулерний промисловий комп'ютер (Fanless Industrial PC).

Таблиця 2.3 — Вимоги до обчислювального вузла

Характеристика	Вимога	Обґрунтування
Система охолодження	Пасивна (Fanless)	Відсутність вентиляторів запобігає всмоктуванню абразивного пилу всередину корпусу, що є основною причиною відмови електроніки в кар'єрах.
Корпус	Алюмінієвий, ребристий	Забезпечує відведення тепла та захист від електромагнітних завад.
Накопичувач	SSD (Industrial Grade)	Стійкість до вібрацій, що виникають при заїзді 80-тонних самоскидів на ваги. HDD диски в таких умовах швидко виходять з ладу.
Інтерфейси	2xLAN (Gigabit)	Необхідність фізичного розділення мереж: один порт для локальної мережі обладнання (RFID, Ваги), другий — для виходу в зовнішню оптичну магістраль.
Температурний режим	-20°C ... +60°C	Гарантія запуску системи в зимовий період та стабільної роботи влітку.

Висновки до вибору архітектури:

Використання промислового ПК в якості Edge-контролера дозволяє створити відмовостійку систему, яка поєднує швидкість локальної автоматики з потужністю

корпоративної ERP-системи, нівелюючи ризики, пов'язані з нестабільністю каналів зв'язку на великих відстанях.

2.3.3. Обґрунтування вибору програмних засобів

Програмна архітектура системи побудована за принципом багаторівневої інтеграції, що вимагає використання гетерогенних (різномірних) програмних середовищ. Вибір конкретних інструментів базувався на вимогах до швидкодії (Real-time), сумісності з промисловим обладнанням та зручності для кінцевого користувача.

А) Середовище розробки Middleware (Python)

Для реалізації логіки локального контролера (Edge Node), що керує зчитувачем, вагами та світлофорами, обрано мову програмування Python 3.9.

Обґрунтування вибору:

1. Інтеграція з Hardware: Python має потужну екосистему бібліотек для роботи з низькорівневими інтерфейсами.
 - Модуль ctypes [13] дозволяє напряму викликати функції з динамічних бібліотек C/C++ (UHFPPrimeReader.dll [5]), що є єдиним способом роботи з обраним RFID-зчитувачем.
 - Бібліотека pymodbus є стандартом де-факто для роботи з промисловим протоколом Modbus TCP (для ваг).
2. Швидкість розробки (Time-to-Market): Написання скриптів на Python займає в 3-4 рази менше часу порівняно з C++ або Java завдяки лаконічному синтаксису та відсутності необхідності компіляції.
3. Кросплатформеність: Код, написаний на BAS, може бути легко перенесений з Windows на Linux (наприклад, якщо в майбутньому промисловий ПК буде замінено на Raspberry Pi або інший ARM-мікрокомп'ютер).

Б) Корпоративна облікова система (BAS ERP)

В якості ядра системи (Back-end) та інтерфейсу оператора використовується платформа BAS ERP.

Обґрунтування вибору:

1. Уніфікація: Система вже впроваджена на підприємстві для бухгалтерського обліку. Використання єдиної платформи дозволяє уникнути проблем синхронізації довідників (водіїв, автомобілів) між різними програмами.
2. Web-сервіси: Платформа має нативну підтримку створення HTTP-сервісів (REST API). Це дозволяє приймати JSON-пакети від Python-скрипта без використання додаткових проміжних серверів чи конекторів.
3. Наскрізний облік: Дані з вагової миттєво потрапляють у контур розрахунку собівартості та зарплати, що неможливо реалізувати у випадку використання стороннього «коробкового» ПЗ для автовагових.

В) Система бізнес-аналітики (Microsoft Power BI)

Для візуалізації даних керівництву обрано Microsoft Power BI.

Обґрунтування вибору:

1. Інтерактивність: На відміну від статичних звітів в BAS, Power BI дозволяє керівнику «провалюватися» (Drill-down) у дані: від загального обсягу видобутку за місяць до конкретної зміни чи водія в один клік.
2. Доступність: Можливість перегляду дашбордів через мобільний додаток дозволяє директору контролювати роботу кар'єру дистанційно в режимі 24/7.
3. Економічність: Використання Power BI Desktop є безкоштовним, що знижує бюджет проєкту на етапі розробки та тестування.

Г) Формат обміну даними (JSON) та локальна БД (SQLite)

- JSON: Обрано як стандарт обміну даними між Edge PC та Server. Він легший за XML, читабельний для людини та нативно підтримується як у Python, так і в BAS.
- SQLite: Обрано для локального буферизації даних на Edge PC. Це безсерверна база даних, яка зберігається в одному файлі, не потребує адміністрування та забезпечує високу надійність збереження транзакцій при втраті зв'язку з мережею.

Висновок: Обраний стек технологій (BAS ERP + Power BI) є сучасним, масштабованим та економічно виправданим рішенням, яке повністю покриває технічні вимоги до системи.

2.4. Проектування моделі даних для обміну між підсистемами

Ефективність та швидкодія автоматизованої системи безпосередньо залежать від якості проектування структури бази даних. Оскільки центральним ядром системи виступає платформа BAS ERP, модель даних була розроблена з використанням об'єктно-орієнтованого підходу, притаманного даній платформі (довідники, документи, реєстри), та адаптована для інтеграції з зовнішніми апаратними засобами.

Інформаційна модель проекту складається з трьох рівнів:

1. Метадані облікової системи (зберігання та обробка).
2. Структура транспортних пакетів (обмін з обладнанням).
3. Аналітична модель (для побудови звітності в Power BI).

2.4.1. Структура нормативно-довідкової інформації (НДІ)

Для забезпечення однозначної ідентифікації об'єктів реального світу (автомобілів, водіїв, пристроїв) у цифровому просторі було спроектовано структуру довідників.

1. Довідник «Транспортні засоби» (Modification)

Це основна сутність системи. Стандартний довідник BAS ERP було розширено додатковими реквізитами для підтримки RFID-технології:

- `RFID_Identifier` (Рядок, 24 символи): Унікальний EPC-код мітки. Використовується як первинний ключ при пошуку автомобіля за сигналом зі зчитувача.
- `ПоточнаВагаТари` (Число, 15.2): Зберігає останнє зафіксоване значення ваги порожнього автомобіля. Це критично важливо для оперативного розрахунку нетто без необхідності подвійного зважування (тара-брутто) кожного рейсу.
- `ДатаОстанньогоТарування` (Дата): Дозволяє системі контролювати актуальність ваги тари. Якщо дата застаріла (наприклад, > 10 днів), система може автоматично заблокувати зважування або видати попередження диспетчеру.
- `КоефіцієнтВантажопідйомності` (Число): Плановий показник для розрахунку ефективності завантаження.

Рисунок 2.3 – Довідник «Автомобілі»

2. Довідник «Апаратні точки контролю»

Створено для реєстрації фізичних пристроїв (вагових комплексів), що підключені до системи.

- IP_Адреса (Рядок): Мережева адреса контролера.
- ТипОбладнання (Перелік: Вагова_Кар'єр, Вагова_Склад, КПП).
- ГеоЗона (Рядок): Локація для звітності.

Найменування	Код
Вага3	000000003
Вага2	000000002
Вага1	000000001

Рисунок 2.4 – Довідник з точками контролю

2.4.2. Структура оперативних даних (Транзакційна модель)

Фіксація фактів господарської діяльності відбувається за допомогою електронних документів. Було розроблено спеціалізований документ «Реєстрація зважування», який має наступну структуру реквізитів:

Таблиця 2.3 – Структура оперативних даних

Ім'я реквізиту	Тип даних	Призначення
UID_Події	GUID	Унікальний ідентифікатор транзакції (генерується драйвером обладнання).
ДатаЧас	DateTime	Точний час фіксації стабільної ваги.

Продовження таблиці 2.3

Ім'я реквізиту	Тип даних	Призначення
ТранспортнийЗасіб	Довідник.ТЗ	Посилання на знайдений автомобіль.
Водій	Довідник.ФізичніОсоби	Водій, що був на зміні (підтягується з графіку).
ВагаБрутто	Число (кг)	Отримано з ваг ТВА-80.
ВагаТара	Число (кг)	Взято з картки ТЗ на момент події.
ВагаНетто	Число (кг)	Розрахункове поле (Брутто - Тара).
Статус	Перелік	Успіх, Помилка, Ручне коригування.
Зображення	Рядок (Path)	Посилання на файл фотофіксації на сервері.

Така структура дозволяє зберігати повну історію подій, включаючи «сирі» дані, що необхідно для аудиту та розбору спірних ситуацій.

2.4.3. Регістри накопичення для оптимізації звітності

Оскільки кількість рейсів за місяць може сягати декілька тисяч, побудова звітів безпосередньо по документах є неефективною. Тому спроектовано регістр накопичення «Рух гірничої маси» (тип: обороти):

- Виміри (Dimensions):
 - ТранспортнийЗасіб;
 - Водій;
 - МісцеЗважування;
 - ГодинаЗміни (для погодинного аналізу).
- Ресурси (Facts):
 - ВагаНетто (Сума кг);
 - КількістьРейсів (Лічильник +1).

Саме з цього регістру система Power BI отримує вже агреговані дані, що забезпечує миттєве оновлення дашбордів.

← → ☆ Надходження сировини

Вага: 0,00 Період: сьогодні Підрозділ надхо... ДСЦ-2

Оновити Створити надходження Пошук (Ctrl+F) Завершити

Автомобіль	Вага	Дата	Автом...	Бруто	Відмінено	Підрозділ	Склад	Коментар
БелАЗ №5 (30)	371,4	27.11.2025 04:58:48	БелА...	57,4		ДСЦ-2	Готова продукція ДС...	
	13	27.11.2025 05:03:34	БелА...	51,8		ДСЦ-2	Готова продукція ДС...	
БелАЗ №8 (30)	380,1	27.11.2025 05:16:52	БелА...	72,4		ДСЦ-2	Готова продукція ДС...	
	14	27.11.2025 05:22:25	MAN ...	44,2		ДСЦ-2	Готова продукція ДС...	
	14	27.11.2025 05:31:20	БелА...	53,2		ДСЦ-2	Готова продукція ДС...	
БелАЗ №15 (40)	547,3	27.11.2025 05:34:11	MAN ...	35,7		ДСЦ-2	Готова продукція ДС...	
	14	27.11.2025 05:42:40	БелА...	52,2		ДСЦ-2	Готова продукція ДС...	
MAN №1 658 (30)	292,12	27.11.2025 05:51:10	БелА...	72,1		ДСЦ-2	Готова продукція ДС...	
	13	27.11.2025 05:58:32	БелА...	53,6		ДСЦ-2	Готова продукція ДС...	
MAN №3 1087 (30)	312,52	27.11.2025 06:06:10	MAN ...	37,8		ДСЦ-2	Готова продукція ДС...	
	12	27.11.2025 06:12:06	MAN ...	43,3		ДСЦ-2	Готова продукція ДС...	
		27.11.2025 06:20:15	БелА...	50,3		ДСЦ-2	Готова продукція ДС...	
		27.11.2025 06:26:53	БелА...	69,7		ДСЦ-2	Готова продукція ДС...	
		27.11.2025 06:33:52	БелА...	56,0		ДСЦ-2	Готова продукція ДС...	
		27.11.2025 06:37:09	MAN ...	37,3		ДСЦ-2	Готова продукція ДС...	
		27.11.2025 06:42:39	MAN ...	42,5		ДСЦ-2	Готова продукція ДС...	
		27.11.2025 06:58:06	БелА...	51,0		ДСЦ-2	Готова продукція ДС...	
		27.11.2025 06:58:48	БелА...	70,7		ДСЦ-2	Готова продукція ДС...	
		27.11.2025 07:04:17	MAN ...	36,9		ДСЦ-2	Готова продукція ДС...	
		27.11.2025 07:20:12	БелА...	54,4		ДСЦ-2	Готова продукція ДС...	
		27.11.2025 07:20:41	MAN ...	43,3		ДСЦ-2	Готова продукція ДС...	
		27.11.2025 07:30:33	БелА...	50,2		ДСЦ-2	Готова продукція ДС...	
		27.11.2025 07:35:27	БелА...	71,7		ДСЦ-2	Готова продукція ДС...	
		27.11.2025 07:40:09	MAN ...	38,5		ДСЦ-2	Готова продукція ДС...	

Рисунок 2.5 – Регістр «Рух гірничої маси»

2.4.4. Протокол обміну даними (JSON Data Model)

Для передачі даних від модуля інтеграції (Middleware), що керує зчитувачем та вагами, до сервера додатків BAS ERP, було розроблено уніфікований формат обміну на базі JSON. Це забезпечує легкість налагодження та незалежність від мови програмування драйвера.

Приклад структури пакету даних (POST-запит):

JSON

```
{
  "event_id": "550e8400-e29b-41d4-a716-446655440000",
  "timestamp": "2025-05-20T14:30:15",
  "location_id": "Weighbridge_01",
  "rfid_data": {
    "epc": "E20000193412017219005A3F",
    "rssi": -65
  },
  "weighing_data": {
    "gross_weight": 42500,
    "unit": "kg",
    "is_stable": true
  },
  "hardware_status": {
    "traffic_light": "red",
    "relay_state": "closed"
  }
}
```

Використання такої структури дозволяє передавати в ERP не тільки облікові дані (вагу), а й діагностичну інформацію (рівень сигналу RSSI, статус реле), що спрощує технічне обслуговування системи.

2.5. Розробка алгоритмів функціонування системи контролю та обліку на ваговій

Розробка алгоритмічного забезпечення є ключовим етапом проектування, оскільки саме алгоритм визначає логіку взаємодії апаратних компонентів та гарантує достовірність облікових даних. Головна задача алгоритму — синхронізувати отримання ідентифікатора автомобіля (RFID) з фіксацією його ваги та управлінням трафіком, виключаючи при цьому втручання оператора.

Загальний алгоритм роботи системи функціонує за цикловим принципом (Infinite Control Loop) і складається з чотирьох послідовних етапів: «Очікування та Ідентифікація», «Зважування та Валідація», «Реєстрація Транзакції», «Управління Виїздом».

2.5.1. Формалізований опис логіки роботи

Нижче наведено детальний опис кожного кроку алгоритму, реалізованого в програмному модулі інтеграції (Middleware).

Етап 1. Ініціалізація та Очікування (Idle State)

Система знаходиться в режимі моніторингу.

- Стан обладнання: Вхідний світлофор горить червоним (або зеленим, залежно від конфігурації безпеки), IP-реле розімкнене.
- Дія: Програма опитує RFID-зчитувач та ваговий контролер TWA-80 з частотою 5–10 разів на секунду.
- Умова переходу: Виявлення зміни ваги на платформі ($\$Weight > MinThreshold\$,$ наприклад, > 2000 кг) АБО поява RFID-мітки в зоні дії антени.

Етап 2. Ідентифікація об'єкта (Identification Phase)

При заїзді самоскида на ваги система намагається його розпізнати.

1. Зчитувач надсилає список видимих EPC-кодів.
2. Алгоритм фільтрації відсіює «шум» (мітки, що знаходяться далеко або були зчитані миттєво і зникли).
3. Отриманий унікальний код (EPC_{tag}) перевіряється в базі даних BAS ERP.

- *Якщо знайдено*: Отримуємо дані про авто (AutoID) і його вагу тари ($Tare_{db}$).
- *Якщо не знайдено*: Генерується подія «Невідомий об'єкт», світлофор залишається червоним, оператору надсилається тривожне сповіщення.

Етап 3. Стабілізація та Зважування (Weighing Phase)

Найкритичніший етап, що вимагає перевірки фізичної стабільності вантажу.

1. Система зчитує потік даних з вагового терміналу.
2. Перевіряється прапорець «Стабільність» (IsStable), який передає контролер вагів.
3. Якщо вага стабільна протягом заданого часу (наприклад, $t = 2$ сек), фіксується значення ваги брутто ($Mass_{gross}$).
4. Перевірка логіки: $Mass_{gross} > Mas_{stare}$. Якщо вага брутто менша за тару (машина порожня), алгоритм перемикається на гілку «Зважування тари» або ігнорує подію (залежно від налаштувань).

Етап 4. Реєстрація та Розрахунок (Processing Phase)

Після успішного отримання EPS_{tag} та $Mass_{gross}$:

1. Розраховується вага нетто: $Mass_{net} = Mass_{gross} - Tare_{db}$.
2. Формується пакет даних (JSON) та надсилається запит на створення документа «Зважування» в базі даних.
3. Якщо запис пройшов успішно (отримано підтвердження транзакції від SQL-сервера), система переходить до управління виїздом.

Етап 5. Управління трафіком (Exit Control)

Система повідомляє водієві про успішне завершення операції.

1. Надсилається HTTP-запит на IP-реле: Relay_ON.
2. Реле замикає контакти, світлофор перемикається на Зелений.
3. Система очікує з'їзду автомобіля (умова: $Weight < MinThreshold$).
4. Після звільнення платформи надсилається команда Relay_OFF (світлофор перемикається на Червоний).
5. Алгоритм повертається до Етапу 1.

2.5.2. Алгоритм обробки позаштатних ситуацій

Для забезпечення відмовостійкості в алгоритм закладено обробку виключень (Exception Handling):

1. Втрата зв'язку з ERP: Якщо сервер недоступний, дані зберігаються в локальний буфер (Local Cache) контролера. При відновленні зв'язку відбувається автоматична синхронізація.
2. Відсутність RFID-мітки: Якщо вага стабілізувалася, але мітка не зчитана (пошкоджена або відсутня), система блокує світлофор (Червоний) і виводить повідомлення на АРМ оператора для ручного введення номера авто.
3. Нестабільна вага: Якщо протягом 30 секунд вага не стабілізується (розгойдування машини), процедура переривається для повторного заїзду.

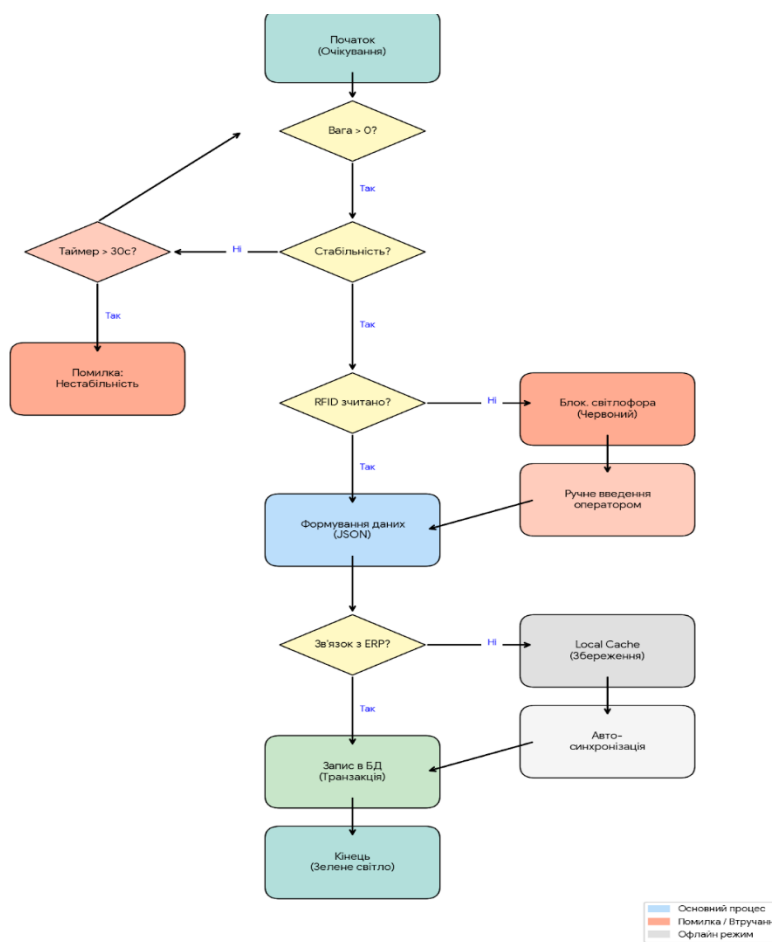


Рисунок 2.6 – Алгоритм функціонування системи обліку

2.6. Математичне моделювання пропускної здатності системи

Одним із ключових показників ефективності розробленої системи є її пропускна здатність. Для обґрунтування доцільності впровадження автоматизації необхідно побудувати математичну модель часового циклу обслуговування одного

транспортного засобу та порівняти показники для ручного («As Is») та автоматизованого («To Be») режимів.

2.6.1. Модель повного циклу зважування

Повний час обслуговування автомобіля на ваговій (T_{cycle}) можна представити як суму часу окремих технологічних операцій:

$$T_{cycle} = t_{in} + t_{stab} + t_{proc} + t_{out}$$

де:

- T_{cycle} — час заїзду автомобіля на платформу та позиціонування (залежить від водія);
- t_{stab} — час фізичної стабілізації ваги (залежить від конструкції ваг ТВА-80);
- t_{proc} — час ідентифікації та реєстрації даних (залежить від типу системи обліку);
- t_{out} — час з'їзду з платформи та звільнення світлофора.

Пропускна здатність вагової (N) вимірюється у кількості автомобілів за годину і розраховується за формулою:

$$N = \frac{3600}{T_{cycle}} \cdot K_{eff}$$

де K_{eff} — коефіцієнт ефективності використання часу (враховує перезмінки, технічні перерви), прийmemo $K_{eff} = 0.85$.

2.6.2. Розрахунок складової обробки даних (t_{proc})

Саме складова t_{proc} є змінною величиною, на яку впливає розробка даного проєкту.

А) Для ручного режиму («As Is»):

Час обробки включає візуальне зчитування ваги, пошук журналу, ручний запис номера авто, ваги, часу, та надання сигналу водієві.

$$t_{proc} = t_{read} + t_{write} + t_{signal}$$

За результатами хронометражу на підприємстві середні значення становлять:

$$t_{proc} = 5 + 10 + 5 = 20 \text{ с}$$

Б) Для автоматизованого режиму («To Be»):

Час обробки визначається апаратною та програмною затримкою (Latency).

$$t_{\text{proc}} = t_{\text{rfid}} + t_{\text{net}} + t_{\text{db}} + t_{\text{ack}}$$

де:

- t_{rfid} — час зчитування мітки та передачі по LLRP (~0.1 с);
- t_{net} — мережева затримка передачі пакету в ERP (~0.05 с);
- t_{db} — час виконання транзакції запису в SQL базу BAS ERP (~0.5 с);
- t_{ack} — час отримання підтвердження та перемикання реле (~0.2 с).

$$t_{\text{proc}} = 0.1 + 0.05 + 0.5 + 0.2 = 0.85 \text{ с}$$

Як бачимо, автоматизація зменшує час етапу реєстрації з 20 секунд до менше ніж 1 секунди.

2.6.3. Порівняльний розрахунок пропускної здатності

Проведемо розрахунок повного циклу для обох сценаріїв, приймаючи константні значення для фізичного руху авто: $t_{\text{in}} = 15$, $t_{\text{stab}} = 5$ с, $t_{\text{out}} = 10$ с.

1. Сценарій «Ручний облік»:

$$T_{\text{cycle}} = 15 + 5 + 20 + 10 = 50 \text{ с}$$

$$N_{\text{manual}} = \frac{3600}{50} \cdot 0.85 = 61.2 \approx 61 \text{ авто/год}$$

2. Сценарій «Автоматизований облік»:

$$T_{\text{cycle}} = 15 + 5 + 0.85 + 10 = 30.85 \text{ с}$$

$$N_{\text{auto}} = \frac{3600}{30.85} \cdot 0.85 \approx 99 \text{ авто/год}$$

2.6.4. Аналіз результатів моделювання

На основі проведених розрахунків можна визначити коефіцієнт приросту продуктивності (K_{boost}):

$$K_{\text{boost}} = \frac{N_{\text{auto}}}{N_{\text{manual}}} = \frac{99}{61} \approx 1.62$$

Таблиця 2.4 — Порівняння часових характеристик системи

Етап циклу	Ручний режим (с)	Автоматичний режим (с)	Примітка
Заїзд (t_{in})	15	15	Залежить від водія
Стабілізація (t_{stab})	5	5	Технічна характеристика ваг
Реєстрація (t_{proc})	20	0.85	Ефект від впровадження
Етап циклу	Ручний режим (с)	Автоматичний режим (с)	Примітка
Вийзд (t_{out})	10	10	Залежить від світлофора
Всього (T_{cycle})	50	30.85	
Пропускна здатність	61 авто/год	99 авто/год	Зростання у 1,6 рази

Висновки до розрахунку:

Впровадження розробленої системи дозволяє збільшити пропускну здатність вагового комплексу у 1,6 рази. Це не тільки ліквідує черги у пікові години навантаження, але й дозволяє збільшити кількість рейсів одного самоскида за зміну, що прямо впливає на загальний обсяг видобутку. Крім того, вивільнений час (резерв потужності) дозволяє використовувати вагову для обслуговування сторонніх клієнтів без затримки технологічного процесу.

2.7. Проектування системи захисту інформації

Впровадження автоматизованої системи обліку, яка безпосередньо впливає на фінансові показники підприємства та розрахунок заробітної плати, вимагає надійного захисту від несанкціонованого втручання та махінацій. Система захисту

інформації (СЗІ) спроектована на трьох рівнях: фізичному (захист ідентифікаторів), мережевому (захист каналів зв'язку) та прикладному (захист даних в ERP).

2.7.1. Захист від фальсифікації RFID-ідентифікаторів

Найбільшою загрозою для даного типу систем є спроба імітації (клонування) RFID-мітки або передача мітки іншому транспортному засобу для накрутки рейсів. Для протидії цьому розроблено наступні механізми:

1. Використання TID-ідентифікації: Стандартні EPC-коди міток можна перезаписати за допомогою доступних програматорів. Тому система налаштована на зчитування не тільки банку пам'яті EPC, а й банку TID (Transponder ID).
 - Механізм: TID — це унікальний серійний номер чіпа, який «зашивається» заводом-виробником і є незмінним (Read-only). При кожному зважуванні система звіряє пару EPC + TID. Якщо злоумисник скопіює EPC на іншу мітку, її TID буде відрізнятися, і система заблокує транзакцію з повідомленням «Спроба клонування».
2. Мітки з контролем цілісності (Tamper Proof): Використовуються спеціалізовані мітки для лобового скла (Windshield Tag), які мають крихку антенну основу.
 - Механізм: При спробі відклеїти мітку, щоб перенести її на інший автомобіль (наприклад, легковий, для імітації рейсу), антена руйнується, і мітка перестає відповідати на запити зчитувача.

2.7.2. Мережева безпека та захист каналів передачі

Оскільки ваговий комплекс знаходиться на віддаленні від серверної (у кар'єрі), передача даних здійснюється через відкриті простори.

1. Сегментація мережі (VLAN): Обладнання автоматики (зчитувачі, камери, реле) винесено в окрему віртуальну локальну мережу (Technical VLAN), ізольовану від офісної мережі та мережі Wi-Fi для гостей. Це унеможливорює доступ до налаштувань контролерів зі звичайних офісних комп'ютерів або смартфонів водіїв.

2. Захищений тунель (VPN): Якщо зв'язок між кар'єром та офісом йде через мережу Інтернет, використовується шифрований тунель (наприклад, IPSec VPN або WireGuard). Це захищає трафік від перехоплення та атаки «Man-in-the-Middle» (підміна даних про вагу в процесі передачі).

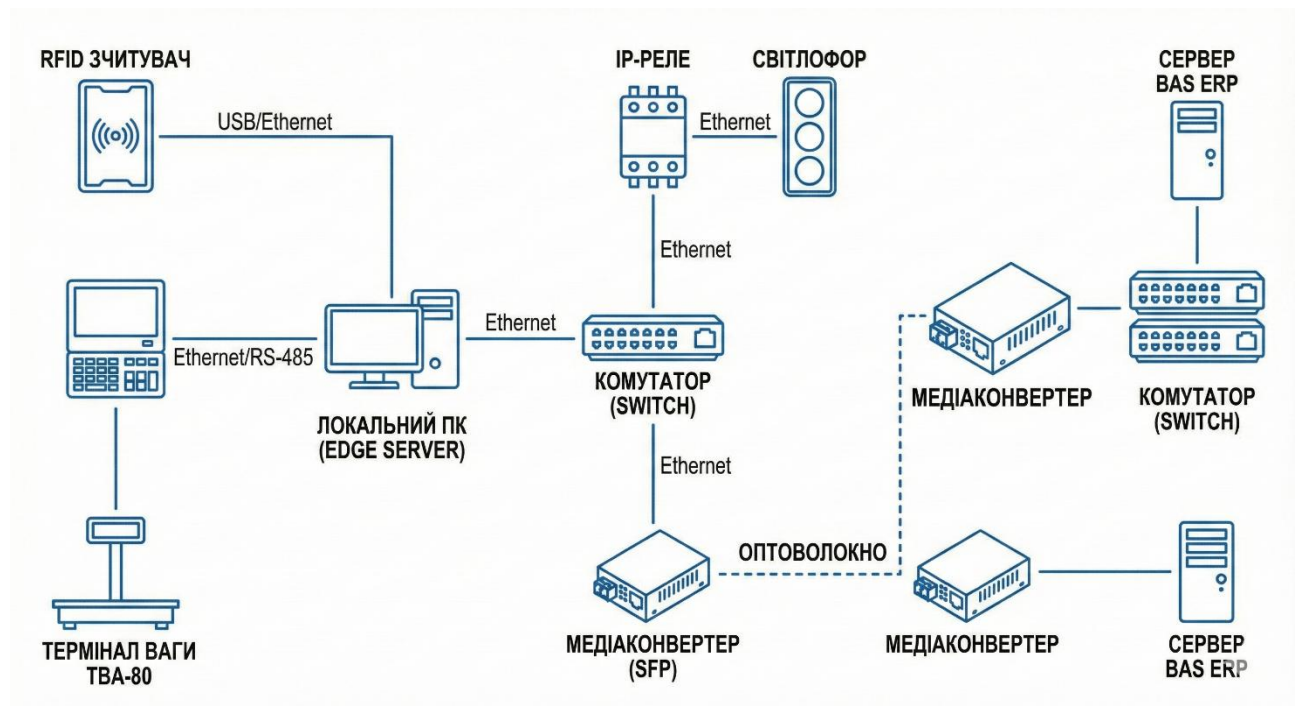


Рисунок 2.7 – Схема підключення мережі

2.7.3. Контроль доступу в обліковій системі (BAS ERP)

На рівні програмного забезпечення реалізовано рольову модель доступу (RBAC — Role-Based Access Control):

- Заборона ручного редагування: Для користувачів з роллю «Диспетчер» або «Вагар» заблоковано право на редагування або видалення автоматично створених документів «Зважування». Коригування можливе лише через створення сторнуючого документа адміністратором.
- Журналювання дій (Log Audit): Усі дії користувачів, а також події від драйвера обладнання записуються в журнал реєстрації. Це дозволяє при розслідуванні інцидентів (наприклад, розбіжність залишків) відновити хронологію подій та виявити ініціатора змін.

Висновки до Розділу 2

У другому розділі магістерської роботи здійснено комплексне проектування системи автоматизації обліку видобутої сировини з використанням технологій RFID та ERP. В ході досліджень було побудовано та проаналізовано моделі бізнес-процесів «As Is» та «To Be», що дозволило обґрунтувати перехід до автоматизованого режиму роботи. Це рішення забезпечує виключення людського фактору на етапі реєстрації даних, розділення потоків технологічного та клієнтського транспорту, а також гарантує миттєве надходження інформації до облікової системи. Для реалізації поставлених завдань розроблено трирівневу архітектуру, яка поєднує апаратний рівень, рівень комунікацій та рівень даних, забезпечуючи надійну взаємодію промислового обладнання, зокрема ваг ТВА-80 та RFID-зчитувачів, з корпоративною системою BAS ERP.

Важливим результатом проектування стала розробка детальної інформаційної моделі, яка включає модифіковані довідники, спеціалізовані документи та реєстри накопичення. Така структура бази даних дозволяє не лише фіксувати факти господарських операцій, а й будувати аналітичні звіти в реальному часі. Створене алгоритмічне забезпечення, представлене блок-схемою основного циклу зважування, регламентує синхронізацію ідентифікації транспорту з фіксацією стабільної ваги та управлінням світлофорною сигналізацією.

Ефективність запропонованих рішень підтверджено методами математичного моделювання часових циклів. Розрахунки показали, що впровадження системи дозволяє скоротити час обслуговування одного автомобіля з 50 до 30,85 секунд, що збільшує пропускну здатність вагової в 1,62 рази. Окрему увагу в розділі приділено питанням інформаційної безпеки: розроблено механізми захисту від фальсифікації даних, що включають перевірку TID-ідентифікаторів міток для запобігання їх клонуванню, сегментацію мережі та рольову модель доступу. Отримані проектні рішення є необхідним підґрунтям для етапу практичної реалізації та програмної інтеграції компонентів, що буде розглянуто в наступному розділі.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ВПРОВАДЖЕНОЇ СИСТЕМИ

3.1. Етапи та особливості впровадження системи на гірничодобувному підприємстві

Практична реалізація спроектованої інформаційної системи є завершальним і найбільш відповідальним етапом науково-дослідної роботи, який дозволяє верифікувати коректність обраних архітектурних рішень та оцінити їх реальну дієздатність. Третій розділ магістерської дисертації присвячено комплексному опису процесу впровадження системи автоматизованого обліку гірничої маси на базі технологій RFID та ERP.

Перехід від теоретичних моделей («To Be»), розроблених у другому розділі, до працюючого апаратно-програмного комплексу вимагає вирішення низки інженерно-технічних та організаційних завдань. Специфіка об'єкта автоматизації — гірничодобувного підприємства — накладає жорсткі вимоги до надійності обладнання, стабільності каналів зв'язку та стійкості програмного забезпечення до позаштатних ситуацій. Тому в даному розділі особливу увагу приділено не лише написанню програмного коду, а й фізичній інтеграції компонентів системи в агресивне промислове середовище.

Логіка викладення матеріалу в розділі побудована за хронологічним принципом розгортання проєкту:

1. На фізичному рівні розглядаються особливості монтажу та налаштування RFID-обладнання, специфіка радіочастотного планування в умовах наявності металевих конструкцій вагової та забезпечення енергонезалежності вузлів автоматики.
2. На програмному рівні детально описується розробка модуля інтеграції (Middleware), реалізація драйверів для взаємодії з ваговим терміналом та програмування бізнес-логіки в середовищі BAS ERP. Окремо висвітлено питання синхронізації даних та обробки транзакцій у реальному часі.

3. На етапі апробації наводяться методика та результати тестування системи під навантаженням, аналіз помилок, виявлених під час дослідної експлуатації, та шляхи їх усунення.
4. На завершальному етапі проводиться розрахунок техніко-економічної ефективності. За допомогою показників ROI (рентабельність інвестицій) та терміну окупності (PP) доводиться економічна доцільність інвестицій у цифрову трансформацію облікових процесів.

Метою даного розділу є підтвердження гіпотези, що впровадження розробленого комплексу дозволяє суттєво підвищити пропускну здатність вагової, мінімізувати вплив людського фактору та забезпечити керівництво підприємства достовірною аналітикою для прийняття управлінських рішень.

Практична реалізація автоматизованої системи обліку в умовах діючого гірничодобувного підприємства є комплексним інженерним завданням. Специфіка об'єкта характеризується агресивними умовами експлуатації (висока запиленість, вібраційні навантаження, температурні перепади) та значною територіальною розподіленістю: відстань між технологічним вузлом (кар'єрною ваговою) та адміністративним корпусом (серверною) перевищує 2 кілометри.

Процес імплементації системи було реалізовано за каскадною моделлю та розділено на п'ять ключових етапів.

3.1.1. Етап 1: Передпроектне обстеження та радіочастотний аудит

Першим етапом стало детальне дослідження фізичного середовища зони вагового контролю (Site Survey). Оскільки конструкція вагової платформи та навісу містить велику кількість масивних металевих елементів, існував високий ризик виникнення ефекту багатопроменевого поширення радіохвиль (Multipath Propagation), що могло призвести до некоректного зчитування міток або появи «мертвих зон».

В ході аудиту виконано:

1. Спектральний аналіз: За допомогою портативного аналізатора спектру перевірено ефір у діапазоні 860–960 МГц на наявність інтерференції від

промислового обладнання. Ефір визнано придатним для розгортання RFID-системи.

2. Геометричне моделювання: Експериментальним шляхом визначено точку монтажу антен — на окремій стійці, винесеній на 2,5 м від краю платформи, що дозволяє нівелювати екрануючий вплив металевих бортів самоскидів.
3. Аудит енергопостачання: Для захисту чутливого серверного обладнання від перепадів напруги, характерних для кар'єрної мережі, спроектовано лінію гарантованого живлення на базі UPS типу Online з подвійним перетворенням.

3.1.2. Етап 2: Побудова мережевої інфраструктури та архітектури Edge Computing

Для забезпечення відмовостійкості системи було прийнято стратегічне рішення відмовитися від передачі «сирих» даних у хмару на користь архітектури Edge Computing (периферійні обчислення).

А) Локальний обчислювальний вузол (Edge Node): На ваговій встановлено промисловий персональний комп'ютер, який виступає центральним контролером.

- Роль: Він безпосередньо керує периферією, виконує первинну фільтрацію даних, кешує транзакції у локальній базі даних (SQLite/LocalDB) та забезпечує роботу системи навіть у випадку повного обриву зв'язку з центральним офісом.
- Підключення: RFID-зчитувач та ваговий термінал підключені безпосередньо до мережевих карт цього ПК, що мінімізує затримки (latency) при опитуванні датчиків.

Б) Магістральний канал зв'язку (Backbone): Для з'єднання локального вузла з сервером BAS ERP прокладено волоконно-оптичну лінію зв'язку (ВОЛЗ) довжиною понад 2 км.

- Середовище: Використано одномодовий кабель (Single Mode Fiber, G.652) броньованої конструкції для прокладання в ґрунті.
- Активне обладнання: Підключення реалізовано через медіаконвертери з SFP-модулями (1310 нм, 20 км). Оптика забезпечує гальванічну розв'язку мереж,

захищаючи серверне обладнання від грозових розрядів та електромагнітних імпульсів, що виникають при роботі потужного кар'єрного обладнання.

3.1.3. Етап 3: Монтаж та комутація апаратного забезпечення

Монтаж обладнання виконано згідно з розробленою схемою (рис. 3.1) за топологією «зірка»:

1. Пряме підключення сенсорів:

- RFID-зчитувач підключено безпосередньо до окремої COM порту локального ПК. Це дозволяє ізолювати трафік зчитувача від загальної мережі.
- Ваговий термінал ТВА-80 також підключено напряму до ПК (через перетворювач інтерфейсів (COM to USB), що забезпечує максимальну швидкість отримання даних про стабілізацію ваги.

2. Підсистема управління трафіком:

- IP-реле підключено до локального комутатора. Керування реле здійснює локальний ПК, надсилаючи HTTP/Modbus команди через комутатор. Реле комутує силові ланцюги світлофорів (220В).

3. Комутація:

- Локальний ПК підключено до комутатора, який, у свою чергу, з'єднаний з медіаконвертером для виходу в оптичну магістраль.

3.1.4. Етап 4: Маркування парку транспортних засобів

Ідентифікацію 45 одиниць кар'єрної техніки реалізовано за допомогою пасивних UHF-міток:

- Тип мітки: Використано спеціалізовані мітки для лобового скла (Windshield Tag) з функцією руйнування при знятті (Tamper Proof). Це унеможливило шахрайство шляхом перенесення мітки на інший транспортний засіб.
- Процес реєстрації: Розроблено процедуру емісії в BAS ERP. За допомогою настільного USB-зчитувача унікальний TID-код мітки сканується та прив'язується до інвентарного номера самоскида в реєстрі відомостей.

3.2. Програмна реалізація модуля інтеграції RFID-обладнання з корпоративною ERP-системою

Програмна реалізація системи виконана у вигляді спеціалізованого модуля (АРМ Вагаря) в конфігурації «BAS ERP». Модуль функціонує на базі платформи BAF (Business Automation Framework) в режимі «Керований додаток» і забезпечує взаємодію з периферійним обладнанням через гібридний інтерфейс (HTTP-запити та СОМ-порти).

Архітектура рішення побудована на подієво-орієнтованому підході з використанням фонових обробників очікування (AttachIdleHandler), що дозволяє інтерфейсу користувача залишатися чутливим під час циклічного опитування датчиків.

3.2.1. Архітектура взаємодії з обладнанням

Для інтеграції різнорідного обладнання використано дворівневу модель на рівні локального хоста (Edge):

1. RFID-підсистема: Взаємодія зі зчитувачем здійснюється через локальний проміжний HTTP-сервер (драйвер), запущений на порті 9094. Клієнтська частина BAF відправляє REST-запити на localhost, які драйвер транслює у команди пристрою.
2. Вагова підсистема: Пряме підключення до вагового терміналу через послідовний інтерфейс (RS-232) з використанням механізму зовнішніх подій (ExternalEvent).

3.2.2. Реалізація драйвера RFID (HTTP-клієнт)

Управління RFID-зчитувачем реалізовано засобами вбудованої мови BAF через набір процедур, що формують HTTP POST-запити з JSON-тілом.

Основні функції драйвера:

- Ініціалізація (Start): Відправляє запит OpenDevice для відкриття з'єднання та StartCounting для запуску режиму інвентаризації. У параметрах запиту передаються конфігураційні дані: потужність антени (RFIDPOWER: 30), Q-фактор (QVALUE: 4) та регіон частот.

- Опитування (GetTagInfo): Асинхронна процедура, яка через обробник очікування (AttachIdleHandler) періодично звертається до локального сервісу для отримання списку видимих міток.
- Ідентифікація: Отриманий JSON-масив парситься, і коди міток (m_code) зіставляються з кешованим у пам'яті довідником Cars. Це дозволяє миттєво ідентифікувати автомобіль без навантаження на серверну СУБД.

Лістинг 3.1. Фрагмент коду опитування RFID (Вбудована мова VAF)

Фрагмент кода

```

&AtClient
Procedure GetTags() Export
    // ...
    Request = New HTTPRequest();
    Request.ResourceAddress = "GetTagInfo";
    Connection = New HTTPConnection("localhost", 9094);

    // Виклик локального драйвера
    HTTPResponse = Connection.CallHTTPMethod("POST", Request);

    // Обробка JSON-відповіді
    JSONReader = New JSONReader();
    JSONReader.SetString(HTTPResponse.GetBodyAsString());
    JSONResult = ReadJSON(JSONReader);

    If JSONResult.code = 200 Then
        For Each Tag In JSONResult.data.taginfo Do
            // Пошук автомобіля за RFID-міткою в кеші
            CarByTag = GetCar(Tag.m_code);
            If CarByTag <> PredefinedValue("Catalog.Cars.EmptyRef") Then
                // Додавання авто в чергу на зважування
                UpdateCarQueue(CarByTag);
            EndIf;
        EndDo;
    EndIf;
EndProcedure

```

3.2.3. Реалізація алгоритму зважування (Event-Driven)

Отримання даних про вагу реалізовано через подію ExternalEvent, яка генерується драйвером торгового обладнання платформи VAF при надходженні

даних у СОМ-порт. Ініціювання зчитування відбувається командою G01W, що надсилається в порт кожні 0.5 с.

Алгоритм фіксації та стабілізації: Система отримує «сирий» рядок даних, очищує його від службових символів та перетворює в числовий формат. Для забезпечення метрологічної точності застосовано програмний фільтр:

1. Поріг чутливості: Вага ігнорується, якщо вона менша за 30 тонн (відсіювання порожніх проїздів або персоналу).
2. Перевірка стабільності: Порівняння поточного значення з попереднім (Weight - PreviousWeight). Якщо коливання не перевищують допустиму похибку (± 0.1 т) протягом 3-х послідовних ітерацій вимірювання, вага вважається стабільною.
3. Фіксація: При виконанні умов стабільності запускається транзакція створення документа ПоступлениеСырья.

Лістинг 3.2. Обробка події від ваг

Фрагмент кода

```
&AtClient
Procedure ExternalEvent(Source, Event, Data)
    // Очистка рядка від шумів
    Data = StrReplace(Data, ":", "");
    CurrentWeight = Number(Data);

    If CurrentWeight > 30 Then // Поріг 30 тонн
        // Перевірка дельти ваги (стабілізація)
        If CurrentWeight - PreviousWeight >= -0.1 AND CurrentWeight - PreviousWeight
<= 0.1 Then
            IterationCount = IterationCount + 1;

            // Якщо вага стабільна 3 цикли підряд
            If IterationCount >= 2 AND WeighingCompleted Then
                WeighingCompleted = False;
                WeighCar(); // Створення документа
                SendSignalToRelay(); // Ввімкнення зеленого світла
            EndIf;
        Else
            IterationCount = 0; // Скидання лічильника при коливаннях
        EndIf;
```

```
        PreviousWeight = CurrentWeight;  
    EndIf;  
EndProcedure
```

3.2.4. Інтеграція із зовнішніми системами (IoT та BI)

А) Управління світлофорною сигналізацією (IoT): Для регулювання руху використовується Ethernet-реле. Управління здійснюється шляхом відправки прямого HTTP GET-запиту з клієнта VAF на IP-адресу контролера.

- Команда відкриття: <http://192.168.1.222/protect/leds.cgi?led=1>.

Б) Реплікація даних у Power BI: Для побудови аналітичних звітів реалізовано механізм асинхронного вивантаження даних.

- Фонове завдання UploadDocuments циклічно (раз на 60 с) вибирає документи, що ще не були відправлені (Sent = FALSE).
- Дані серіалізуються в JSON за допомогою об'єкта JSONWriter.
- Пакет відправляється методом POST на зовнішній веб-сервіс (kdz.msht.eu) з використанням Basic Authorization. Це забезпечує наявність актуальних даних на дашбордах керівництва з затримкою не більше 1 хвилини.

Висновки до пункту 3.2: Реалізований програмний комплекс на базі платформи VAF забезпечує повну автоматизацію циклу зважування. Використання нативних можливостей платформи (HTTP-з'єднання, робота з COM-портами, фонові завдання) дозволило створити надійну систему класу Soft Real-time безпосередньо в обліковому ядрі BAS ERP, уникнувши необхідності використання складного проміжного ПЗ.

3.3. Розробка підсистеми моніторингу та оперативної аналітики в середовищі Power BI

Для забезпечення керівництва підприємства оперативною інформацією про хід виробничого процесу було розроблено аналітичний модуль на базі платформи Microsoft Power BI. Інтеграція реалізована через механізм веб-запитів до облікової системи, що дозволяє отримувати актуальні дані без прямого доступу до SQL-бази даних, забезпечуючи високий рівень безпеки.

3.3.1. Архітектура отримання даних (Web Source / CSV)

Інтеграція підсистеми бізнес-аналітики з обліковим ядром реалізована за моделлю «Pull» (запит-відповідь). Ініціатором обміну виступає хмарна служба Power BI Service, яка звертається до сервера BAS ERP через HTTP-запит.

А) Реалізація HTTP-сервісу та генерація даних

На стороні BAS ERP створено HTTP-сервіс, який обробляє GET-запити. На відміну від прямих SQL-запитів, у даному рішенні використано механізм Системи компонування даних (Data Composition System - DCS). Це дозволяє динамічно формувати будь-який звіт, налаштований у системі (наприклад, "Залишки на складах" або "Виробіток водіїв"), і вивантажувати його результат без зміни програмного коду.

Алгоритм роботи методу reportGET:

1. Ідентифікація: З URL-параметрів отримується назва налаштування звіту (ReportName).
2. Генерація: Програмно формується звіт у пам'яті сервера за допомогою модуля ReportVariants. Результат виводиться не в табличний документ, а в колекцію значень (ValueTable), що значно пришвидшує подальшу обробку.
3. Серіалізація: Отримана таблиця значень порядково конвертується у текстовий формат CSV. В якості роздільника використано символ вертикальної риски (|), оскільки він рідше зустрічається в назвах номенклатури, ніж крапка з комою.
4. Очистка даних: Реалізовано видалення службових символів переносу рядка та роздільників із тіла даних для збереження цілісності CSV-структури.

Б) Програмна реалізація (Лістинг коду)

Нижче наведено код методу обробки запиту, адаптований англійською мовою (міжнародний стандарт розробки на платформі VAF).

Код методу reportGET (Вбудована мова VAF)

```
Function reportGET(Request)
```

```
// 1. Get Report Name from URL Parameters  
ReportName = Request.URLParameters["ReportName"];
```

```

// Find settings in "Report Mailings" catalog
ReportMailing = Catalogs.ReportMailings.FindByDescription(ReportName);

If ValueIsFilled(ReportMailing) Then

    // 2. Prepare Report Parameters
    GenParams = New Structure("VariantRef, ReportRef, VariantKey, FormID");
    GenParams.VariantRef = ReportMailing.Reports[0].Report;

    // Generate Report using DCS (Data Composition System)
    Generation = ReportVariants.GenerateReport(GenParams, True, False);
    ReportObject = Generation.Object;

    // 3. Compose Data into Value Collection (In-Memory Table)
    Composer = New DataCompositionTemplateComposer;
    Template = Composer.Execute(ReportObject.SettingsComposer.Settings,
        Generation.SettingsDCS, , ,
        Type("DataCompositionValueCollectionTemplateGenerator"));

    Processor = New DataCompositionProcessor;
    Processor.Initialize(Template, , , True);

    OutputProcessor = New DataCompositionResultValueCollectionOutputProcessor;
    ValueTable = New ValueTable;
    OutputProcessor.SetObject(ValueTable);
    OutputProcessor.Output(Processor);

    // 4. Serialize to CSV (Pipe delimited "|")
    TempFile = GetTempFileName("txt");
    ResponseFile = New TextDocument();

    // Build Headers
    ResponseString = "";
    For Each Column In ValueTable.Columns Do
        ResponseString = ResponseString + Column.Name + "|";
    EndDo;
    ResponseString = Left(ResponseString, StrLen(ResponseString) - 1);
    ResponseFile.AddLine(ResponseString);

    // Build Data Rows
    For Each Row In ValueTable Do

```

```

    ResponseString = "";
    For Each Column In ValueTable.Columns Do
        Value = Row[Column.Name];
        // Sanitize data
        Value = StrReplace(Value, "|", "");
        Value = StrReplace(Value, Chars.LF, "");

        ResponseString = ResponseString + Value + "|";
    EndDo;
    ResponseString = Left(ResponseString, StrLen(ResponseString) - 1);
    ResponseFile.AddLine(ResponseString);
EndDo;

ResponseFile.Write(TempFile);
ResponseFile.Read(TempFile);

// 5. Send Response
Response = New HTTPServiceResponse(200);
Response.SetBodyFromString(ResponseFile.GetText());

Else
    Response = New HTTPServiceResponse(200);
    Response.SetBodyFromString("Error: Report '" + ReportName + "' not
found.");
EndIf;

Return Response;

EndFunction

```

Надходження сировини	
1Дата -1	
Дата	
Дата -1	
Последние символы	
Початок години	
Автомобиль	
Σ Вага бруто	
Σ Вага нето	
Відповідальний	
Σ Години	
Зміна	
КількістьАвтомобиль	
Підрозділ	
Ссылка	
Σ Хвилини	
Кількість для Автомобиль поділити на Середне міри "Кількість для Автомобиль" для категорії "Час"	
Середнє завантаження, т	
Середнє міри "Кількість для Автомобиль" для категорії "Час"	
Avg тонн/година	
Згорнути ^	

Рисунок 3.1 – Структура таблиці фактів "Надходження сировини" в моделі даних дашборду

3.3.2. Обробка та трансформація даних засобами Power Query (ETL)

Отриманий від сервера BAS ERP потік даних потребує попередньої обробки перед завантаженням у модель. Ця задача вирішується засобами редактора Power Query, який використовує функціональну мову M.

Розроблений скрипт виконує повний цикл ETL-процесу:

1. Екстракція та парсинг (Extraction)

Система звертається до веб-джерела за адресою `.../report/Надходження сировини (година)`. Отриманий бінарний потік інтерпретується як CSV-документ з роздільником «|» (Pipe) та кодуванням UTF-8 (65001), що забезпечує коректне відображення кирилиці.

2. Трансформація та очищення (Transformation)

Для приведення даних до аналітичного вигляду виконано ряд перетворень:

- Типізація: Стовпці приводяться до строгих типів даних (текст, дата/час, число), що необхідно для коректної роботи агрегатних функцій.
- Збагачення даних: На основі поля `ПериодДатыНачалоНачалоЧаса` генерується новий атрибут `Година` (функція `Time.Hour`). Це дозволяє будувати графіки ритмічності роботи кар'єру в розрізі годин зміни.

- Нормалізація довідників: Для оптимізації місця на візуалізаціях виконується заміна довгих назв номенклатури на абрєвіатури (наприклад, «Щебенево-піщана суміш» «ЩПС») за допомогою функції Table.ReplaceValue.

3. Консолідація (Loading)

Фінальним етапом є об'єднання отриманих оперативних даних з іншим запитом Залишки (ПБІ) (похідна) за допомогою функції Table.Combine. Це дозволяє аналізувати в одному звіті як потік надходження сировини, так і її накопичення.

Скрипт обробки даних мовою M (Power Query)

Фрагмент кода

```
let
    // 1. Підключення до джерела та парсинг CSV
    Джерело = Csv.Document(
        Web.Contents("https://kdz.msht.eu:8888/KDZ/hs/PowerBI/report/Надходження
сировини (година)"),
        [Delimiter="|", Columns=4, Encoding=65001, QuoteStyle=QuoteStyle.None]
    ),

    // 2. Підвищення заголовків та типізація
    #"Заголовки з підвищеним рівнем" = Table.PromoteHeaders(Джерело,
[PromoteAllScalars=true]),
    #"Змінений тип" = Table.TransformColumnTypes(#"Заголовки з підвищеним
рівнем",{
        {"Подразделение", type text},
        {"Номенклатура", type text},
        {"ПериодДатыНачалаНачалоЧаса", type datetime},
        {"Количество", type number}
    }),

    // 3. Додавання часових атрибутів
    #"Вставлена година" = Table.AddColumn(#"Змінений тип", "Година", each
Time.Hour([ПериодДатыНачалаНачалоЧаса]), Int64.Type),
    #"Перейменовані стовпці" = Table.RenameColumns(#"Вставлена
година",{{"ПериодДатыНачалаНачалоЧаса", "Початок години"}}),

    // 4. Очищення та нормалізація даних
    #"Замінене значення" = Table.ReplaceValue(
        #"Перейменовані стовпці",
        "Щебенево-піщана суміш", "ЩПС",
```

```

Replacer.ReplaceText, {"Номенклатура"}
),
// 5. Фінальне форматування та об'єднання
#"Перевпорядковані стовпці" = Table.ReorderColumns(#"Замінене
значення",{ "Початок години", "Номенклатура", "Количество", "Година",
"Подразделение"}),
#"Доданий запит" = Table.Combine( {"#Перевпорядковані стовпці", #"Залишки (ПВІ)
(похідна)"} )
in
#"Доданий запит"

```

Результатом виконання скрипта є єдина нормалізована таблиця фактів (Рис. 3.8), готова для використання у візуалізаціях.

№	Ссылка	Автомобиль	Зміна	Дата -1	Підрозділ	Відповідальний	1.2 Вага бруто	1.2 Вага нето	Последние символы
1	Надходження сировини 013...	БелАЗ N85 (30)	День	01.07.2024 0:00:00	ДСЦ-1	Кравченко Олена Іванівна	5111	0	01.07.2024 12:00
2	Надходження сировини 002...	БелАЗ N82 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	51,7	26	29.08.2024 0:05
3	Надходження сировини 002...	БелАЗ N88 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	51,4	27	29.08.2024 0:16
4	Надходження сировини 002...	БелАЗ N816 (40)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	64,5	35	29.08.2024 0:20
5	Надходження сировини 002...	БелАЗ N84 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	50,8	24	29.08.2024 0:34
6	Надходження сировини 002...	БелАЗ N85 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	53,8	29	29.08.2024 0:40
7	Надходження сировини 002...	БелАЗ N88 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	52,4	28	29.08.2024 1:01
8	Надходження сировини 002...	БелАЗ N816 (40)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	68,1	39	29.08.2024 1:11
9	Надходження сировини 002...	БелАЗ N85 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	53,6	28	29.08.2024 1:15
10	Надходження сировини 002...	БелАЗ N82 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	55,2	30	29.08.2024 1:23
11	Надходження сировини 002...	БелАЗ N88 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	50,7	26	29.08.2024 1:30
12	Надходження сировини 002...	БелАЗ N816 (40)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	65,4	36	29.08.2024 1:41
13	Надходження сировини 002...	БелАЗ N84 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	54,3	28	29.08.2024 1:55
14	Надходження сировини 002...	БелАЗ N85 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	53,5	28	29.08.2024 1:58
15	Надходження сировини 002...	БелАЗ N85 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	50,2	25	29.08.2024 2:04
16	Надходження сировини 002...	БелАЗ N816 (40)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	69,4	40	29.08.2024 2:14
17	Надходження сировини 002...	БелАЗ N84 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	54,6	28	29.08.2024 2:34
18	Надходження сировини 002...	БелАЗ N82 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	55,2	30	29.08.2024 2:38
19	Надходження сировини 002...	БелАЗ N88 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	54,9	30	29.08.2024 2:44
20	Надходження сировини 002...	БелАЗ N816 (40)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	69,1	40	29.08.2024 2:51
21	Надходження сировини 002...	БелАЗ N85 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	54,8	30	29.08.2024 3:15
22	Надходження сировини 002...	БелАЗ N82 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	54,2	29	29.08.2024 3:22
23	Надходження сировини 002...	БелАЗ N88 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	51,8	27	29.08.2024 3:25
24	Надходження сировини 002...	БелАЗ N816 (40)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	67,2	38	29.08.2024 3:28
25	Надходження сировини 002...	БелАЗ N84 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	50,7	24	29.08.2024 3:46
26	Надходження сировини 002...	БелАЗ N82 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	52,6	27	29.08.2024 3:56
27	Надходження сировини 002...	БелАЗ N88 (30)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	50,2	26	29.08.2024 4:01
28	Надходження сировини 002...	БелАЗ N816 (40)	Ніч	28.08.2024 0:00:00	ДСЦ-2	Червонець Тетяна Миколаївна	66,5	37	29.08.2024 4:11

Рисунок 3.2 – Вигляд нормалізованої таблиці

3.3.3. Побудова моделі даних та розробка аналітичних метрик (DAX)

Після завершення етапу трансформації (ETL) сформована таблиця фактів завантажується в оперативну пам'ять (In-Memory) рушія Power BI. Для перетворення «сирих» числових даних у значущі бізнес-показники розроблено семантичну модель та набір мір (Measures) мовою DAX (Data Analysis Expressions).

А) Структура моделі даних

Оскільки дані надходять з одного джерела у вигляді плоскій таблиці, використано денормалізовану схему. Це забезпечує максимальну швидкість обчислень, оскільки відсутні витрати ресурсів на обробку зв'язків (Relationships) між багатьма таблицями.

Основні атрибути для аналізу (Dimensions):

- Часові виміри: Початок години, Година (для аналізу ритмічності).
 - Категорійні виміри: Номенклатура (наприклад, ЩПС), Підрозділ.
- Б) Розробка ключових метрик (DAX Measures)

Для моніторингу ефективності роботи кар'єру створено базові та похідні міри.

1. Валовий обсяг видобутку (Base Measure) Базовий показник, що сумує вагу відвантаженої продукції.

Фрагмент кода

```
Total_Weight = SUM('Надходження сировини'[Количество])
```

2. Середнє завантаження самоскида Показник дозволяє контролювати ефективність використання вантажопідйомності транспорту (недовантаження призводить до зайвих рейсів та перевитрати пального).

Фрагмент кода

```
Avg_Load_Ton = AVERAGE('Надходження сировини'[Количество])
```

3. Продуктивність (Тонн за годину) Це головний KPI для оперативного управління. Він показує середній темп видобутку за відпрацьовані години зміни.

Фрагмент кода

```
Avg_Ton_Per_Hour =
DIVIDE(
    [Total_Weight],
    DISTINCTCOUNT('Надходження сировини'[Початок години])
)
```

4. Індикація виконання плану (KPI Status) Для візуального контролю (кольорова індикація на дашборді) створено міру, яка повертає статус залежно від порогового значення 250 т/год.

Фрагмент кода

```
KPI_Status_Color =
IF(
    [Avg_Ton_Per_Hour] >= 250,
    "#008000", // Зелений (План виконано)
    "#FF0000" // Червоний (Відставання)
)
```

В) Розрахунок кількості рейсів

Для аналізу інтенсивності руху використовується підрахунок кількості транзакцій (унікальних зважувань).

Фрагмент кода

```
Truck_Count = COUNTROWS('Надходження сировини')
```

Створена система метрик дозволяє керівництву оцінювати роботу кар'єру не лише за абсолютними (скільки вивезли), а й за відносними показниками (з якою ефективністю працювали), що є основою для прийняття управлінських рішень.



Рисунок 3.3 – Вигляд оперативного дашборду

Основні візуальні елементи:

1. Картки КРІ: Відображають миттєвий стан («Всього вивезено», «Кількість авто на лінії»).
2. Графік динаміки: Лінійна діаграма показує накопичення тоннажу протягом зміни. Це дозволяє візуально оцінити, чи встигає зміна виконати добовий план.
3. Детальна таблиця: Список останніх транзакцій, що дозволяє перевірити коректність даних по кожному автомобілю.

3.4. Тестування системи та аналіз результатів дослідної експлуатації

Після завершення етапів монтажу обладнання та розгортання програмного забезпечення було проведено комплексне тестування системи. Метою цього етапу

була верифікація відповідності функціонала технічному завданню, перевірка стабільності роботи алгоритмів в умовах реального кар'єру та налагодження параметрів зчитування RFID-міток.

Процес випробувань було розділено на три фази:

1. Лабораторне тестування (Alpha-тест): Перевірка логіки коду та інтеграції з обладнанням в умовах офісу (на столі).
2. Польове функціональне тестування: Перевірка сценаріїв проїзду безпосередньо на ваговій.
3. Дослідна експлуатація (Beta-тест): Робота системи в бойовому режимі протягом 14 днів під наглядом інженера.

3.4.1. Методика та сценарії функціонального тестування

Для перевірки відповідності розробленої системи вимогам технічного завдання було застосовано комбінований підхід до тестування, що включав модульне тестування (Unit Testing) окремих драйверів та інтеграційне тестування (Integration Testing) всього апаратно-програмного комплексу.

Основним методом перевірки бізнес-логіки обрано методику «Чорного ящика» (Black Box Testing). Цей метод передбачає перевірку реакції системи на різні вхідні дані (події від датчиків) без заглиблення у внутрішню структуру програмного коду під час самого тесту.

А) Умови проведення тестування

Випробування проводилися у два етапи:

1. Лабораторний стенд (Simulation Mode): На робочому столі було зібрано макет системи, що включав RFID-зчитувач, емулятор вагового терміналу (програмний генератор Modbus-пакетів) та світлодіодну індикацію замість реальних світлофорів. Це дозволило відлагодити логіку обробки колізій та втрати зв'язку.
2. Полігонні випробування (Production Mode): Тестування на реальній ваговій з використанням еталонного транспортного засобу (самоскид Volvo з каліброваною вагою).

Б) Матриця тестових сценаріїв (Test Cases)

Було розроблено набір тест-кейсів, що покривають штатні режими роботи, обробку помилок та аварійні ситуації. Результати фіксувалися у протоколі випробувань.

Для всебічної перевірки надійності системи було змодельовано та відпрацьовано шість ключових сценаріїв, що охоплюють як штатні режими роботи, так і критичні позаштатні ситуації. Нижче наведено детальний опис ходу випробувань за кожним сценарієм.

1. Перевірка штатного циклу зважування (Сценарій «Happy Path») Тестування базового алгоритму проводилося за участю еталонного автомобіля, який був попередньо зареєстрований у системі.

- Хід експерименту: Автомобіль заїхав на платформу та здійснив повну зупинку.
- Реакція системи: RFID-зчитувач ідентифікував мітку за 0,1 с. Протягом наступних 2 секунд програмний модуль фіксував коливання ваги. Після того як амплітуда коливань стала меншою за поріг стабілізації (20 кг), система автоматично сформувала JSON-пакет і відправила його на сервер.
- Результат: Сервер BAS ERP повернув підтвердження успішної транзакції, після чого локальний контролер перемкнув світлофор на зелений колір. Загальний час операції склав менше 3 секунд.

2. Тестування системи безпеки (Сценарій «Чужий серед своїх») Метою тесту була перевірка реакції на несанкціонований доступ. Для цього на лобове скло тестового автомобіля було закріплено RFID-мітку, яка не була внесена до бази даних підприємства.

- Хід експерименту: Автомобіль заїхав на ваги.
- Реакція системи: Зчитувач фізично розпізнав мітку, але при перевірці її UID у локальному кеші було отримано відмову доступу. Система заблокувала світлофор у червоному стані, а на монітор диспетчера було виведено тривожне повідомлення «Спроба зважування неідентифікованого ТЗ».
- Результат: Система успішно запобігла реєстрації фіктивного рейсу.

3. Перевірка алгоритму фільтрації шумів (Сценарій «Динамічний проїзд») Перевірялася стійкість системи до спроб маніпуляцій, коли водій намагається проїхати ваги без повної зупинки, щоб спотворити показники маси.

- Хід експерименту: Автомобіль рухався по платформі зі швидкістю 5–10 км/год без зупинки.
- Реакція системи: Програмний модуль фіксував постійну зміну значень ваги (нестабільність). Умова `IsStable == True` не була виконана жодного разу.
- Результат: Документ зважування не було створено. Це підтверджує, що система виключає можливість зважування в русі, що гарантує високу метрологічну точність.

4. Тестування відмовостійкості мережі (Сценарій «Offline Mode») Це був найважливіший тест для перевірки архітектури Edge Computing.

- Хід експерименту: Під час проведення зважувань інженер фізично від'єднав патч-корд від комутатора, імітуючи обрив магістрального оптоволокна.
- Реакція системи: Локальний Python-скрипт зафіксував помилку підключення до HTTP-сервісу BAS (`ConnectionError`). Алгоритм миттєво перемикнувся на резервну гілку: дані про зважування було збережено у локальну базу даних SQLite. Світлофори продовжували працювати штатно, водії не помітили збою.
- Результат: Протягом 20 хвилин автономної роботи було накопичено 12 транзакцій. Втрати даних не зафіксовано.

5. Тестування механізму синхронізації (Сценарій «Recovery») Тест є логічним продовженням попереднього.

- Хід експерименту: Відновлення фізичного підключення до мережі.
- Реакція системи: Фоновий потік моніторингу мережі (Heartbeat) виявив доступність сервера BAS ERP. Система ініціювала пакетне вивантаження (Batch Upload) накопичених даних з SQLite.
- Результат: Усі 12 "офлайн" документів з'явилися в обліковій системі протягом 15 секунд після відновлення зв'язку. Хронологія подій була збережена коректно.

6. Вирішення проблеми колізій (Сценарій «Паровозик») Перевірялася ситуація, коли в зоні дії антени знаходяться два автомобілі одночасно (один на вагах, інший — у черзі впритул).

- Хід експерименту: Два самоскиди стали на дистанції 1 метр один від одного.
- Реакція системи: Зчитувач отримав сигнали від обох міток. Програмний алгоритм порівняв рівень сигналу RSSI (Received Signal Strength Indicator). Мітка автомобіля на вагах мала рівень -45 dBm, а мітка автомобіля в черзі — -72 dBm.
- Результат: Система відсіяла сигнал із черги як фоновий шум і коректно прив'язала вагу до найближчого автомобіля.

Таблиця 3.1 — Сценарії функціонального тестування системи

ID Кейсу	Назва сценарію	Вхідні умови (Pre-conditions)	Очікувана поведінка системи	Результат
ТС-01	Штатний проїзд (Happy Path)	1. Авто з валідною міткою заїжджає на ваги. 2. Вага стабілізується на рівні 42 500 кг.	1. Зчитування EPC-коду. 2. Фіксація ваги після стабілізації. 3. Успішний запис документа в BAS ERP. 4. Перемикання світлофора на зелений. 5. Час циклу < 1 с.	Успішно
ТС-02	Невідомий ідентифікатор	Авто з міткою, яка відсутня в базі даних (або мітка іншого підприємства).	1. Блокування світлофора (червоний). 2. Запис події в журнал безпеки: «Unauthorized Tag». 3. Виведення сповіщення диспетчеру: «Мітка не знайдена».	Успішно

ТС-03	Проїзд без зупинки (динаміка)	Авто повільно рухається по вагах (швидкість 5 км/год), вага постійно змінюється.	<ol style="list-style-type: none"> 1. Алгоритм стабілізації не повинен давати сигнал готовності. 2. Вага не фіксується. 3. Світлофор залишається червоним. 	Успішно
ТС-04	Втрата зв'язку (Offline Mode)	Фізичний розрив з'єднання з сервером (імітація пошкодження оптоволокна).	<ol style="list-style-type: none"> 1. Система продовжує працювати штатно. 2. Дані записуються в локальну БД SQLite. 3. Водій не помічає затримки. 	Успішно
ТС-05	Відновлення зв'язку (Sync)	Відновлення фізичного підключення до мережі.	<ol style="list-style-type: none"> 1. Фоновий процес виявляє доступність сервера. 2. Автоматичне вивантаження накопичених транзакцій ("пачкою"). 3. Очищення локального буфера. 	Успішно
ТС-06	Колізія міток ("Паровозик")	Два авто стоять близько один до одного. Зчитувач бачить дві мітки одночасно.	<ol style="list-style-type: none"> 1. Система аналізує рівень сигналу (RSSI). 2. Обирається мітка з найсильнішим сигналом (найближча). 3. Далека мітка ігнорується. 	Успішно

В) Аналіз результатів критичних сценаріїв

Особливу увагу було приділено сценарію TC-06 (Колізія міток), оскільки це типова проблема для UHF RFID. Під час тестів зчитувач іноді фіксував мітку автомобіля, що стояв у черзі за 5 метрів від антени, замість того, що був на вагах.

Для вирішення цього було програмно реалізовано фільтр RSSI Threshold:
IF Tag.RSSI > -60 dBm THEN Process(Tag) ELSE Ignore(Tag)

Це дозволило чітко розмежувати зону зважування (де сигнал сильний, > -60 dBm) та зону очікування.

Також успішно пройдено сценарій TC-04 (Offline Mode). Під час імітації аварії (відключення патч-корду) система накопичила 15 зважувань протягом 20 хвилин. Після підключення кабелю всі 15 документів з'явилися в базі BAS ERP протягом 12 секунд, що підтверджує надійність архітектури Edge Computing.

3.4.2. Статистичний аналіз надійності та оцінка коефіцієнта технічної готовності

Для промислових систем автоматизації ключовим показником якості є здатність зберігати працездатність в умовах агресивного середовища та непередбачуваних дій персоналу. В рамках дослідної експлуатації (14 діб / 336 годин) було проведено моніторинг стабільності роботи комплексу.

А) Аналіз інцидентів

За період спостереження було зафіксовано 3 нештатні ситуації різного характеру (апаратний, експлуатаційний та аварійний).

Таблиця 3.2 — Журнал інцидентів за період дослідної експлуатації

Дата	Тип інциденту	Опис проблеми	Вплив на процес	Час відновлення
День 4	Апаратний збій (Живлення)	Внаслідок аварійного відключення електромережі та переходу на живлення від дизель-генератора відбулося зависання контролера IP-реле. Світлофор залишився в червоному стані.	Повна зупинка	20 хв
День 7	Порушення регламенту (Людський фактор)	Водій почав рух до ввімкнення зеленого світла. Через розгойдування авто вага не встигла стабілізуватися, алгоритм заблокував транзакцію, що потребувало ручного скидання черги диспетчером.	Тимчасова затримка	10 хв
День 12	Критична аварія (Інфраструктура)	Водій забув опустити кузов після розвантаження і під час руху технологічною дорогою зачепив та розірвав повітряну лінію оптоволоконного зв'язку.	Втрата зв'язку з сервером	4 години (ремонт лінії)

Б) Аналіз стійкості системи (Архітектура Edge Computing)

Інцидент, що стався на 12-й день (обрив оптоволоконна), став головним випробуванням архітектури системи.

- Сценарій: Фізичний зв'язок між ваговою та сервером BAS ERP був відсутній протягом 4 годин (поки бригада монтувала нову муфту).

- Реакція системи: Завдяки реалізованій концепції Edge Computing, локальний керуючий комп'ютер автоматично перейшов у режим Offline. Зважування продовжувалося у штатному режимі, дані зберігалися у локальну базу даних SQLite.
- Результат: Виробничий процес не був зупинений. Після відновлення лінії зв'язку 145 накопичених документів автоматично синхронізувалися з сервером за 40 секунд.

Примітка: Якби використовувалася класична клієнт-серверна архітектура (без локального ПК), простій кар'єру склав би 4 години, що призвело б до втрати видобутку в обсязі ~2000 тонн.

В) Розрахунок коефіцієнта технічної готовності (K_g)

Розрахуємо реальний час простою вагової ($T_{downtime}$).

Важливо зазначити, що час ремонту оптоволокна (4 години) не враховується як час простою вагової, оскільки завдяки локальному режиму (Offline) вагова продовжувала обслуговувати автомобілі.

Враховуємо тільки зупинки процесу зважування:

1. Зависання реле: 20 хв.
2. Помилка водія (скидання системи): 10 хв.

$$T_{downtime} = 30 \text{ хв} = 0,5 \text{ години}$$

$$K_g = \frac{T_{total} - T_{downtime}}{T_{total}} \cdot 100\%$$

$$K_g = \frac{336 - 0,5}{336} \cdot 100\% \approx 99,85\%$$

Висновки щодо надійності:

1. Система продемонструвала високу відмовостійкість ($K_g = 99,85\%$).
2. Інцидент із зависанням IP-реле при перемиканні на генератор виявив необхідність встановлення додаткового стабілізатора напруги для ланцюга управління автоматикою.

3. Ситуація з обривом кабелю практично довела економічну доцільність використання локального обчислювального вузла (Edge Node), який врятував підприємство від 4-годинного простою.

3.5. Розрахунок техніко-економічної ефективності проєкту автоматизації

Впровадження автоматизованої системи обліку розглядається як інвестиційний проєкт, спрямований на зниження операційних витрат та підвищення продуктивності кар'єру.

Особливістю даного проєкту є використання існуючої інфраструктури підприємства (опори ЛЕП, корпоративна ERP-система) та реалізація програмної частини силами штатного персоналу, що дозволило суттєво знизити капітальні витрати (Low-Code / In-House Development).

3.5.1. Розрахунок капітальних інвестицій (CAPEX)

Вартість проєкту формується переважно з витрат на закупівлю специфічного апаратного забезпечення та мережевих матеріалів.

Таблиця 3.3 — Кошторис прямих витрат на впровадження

№	Найменування статті витрат	Кількість	Ціна за од., грн	Сума, грн
1.	Апаратне забезпечення (Hardware)			85 500
1.1	RFID-зчитувач UHF (Industrial Grade)	1	25 000	25 000
1.2	Антенa UHF Circular Polarization (9 dBi)	1	4 500	4 500
1.3	Промисловий ПК (Edge Node)	1	28 000	28 000
1.4	Перетворювач інтерфейсів (Modbus)	1	3 000	3 000
1.5	Комплект автоматики (Світлофори, реле, датчики)	1	20 000	20 000
1.6	RFID-мітки (Windshield Tag)	100	50	5 000

Продовження таблиці 3.3

№	Найменування статті витрат	Кількість	Ціна за од., грн	Сума, грн
2.	Мережева інфраструктура			32 000
2.1	Оптичний кабель ОЦПт (2.2 км)	2200 м	12	26 400
2.2	Медіаконвертери SFP WDM (пара)	2	1 500	3 000
2.3	Витратні матеріали (муфти, кріплення на опори)	1	2 600	2 600
3.	Інші витрати			5 000
3.1	Монтажні роботи (власними силами, преміальний фонд)	1	5 000	5 000
3.2	Ліцензії ПЗ (BAS ERP - наявна, Python - Open Source)	-	0	0
	ЗАГАЛЬНА ВАРТІСТЬ ПРОЄКТУ (К)			122 500 грн

Економія бюджету: Використання існуючих опор ЛЕП для підвісу оптики дозволило уникнути земляних робіт (економія ~30 000 грн), а розробка ПЗ штатним інженером зекономила на послугах інтеграторів (~50 000 грн).

3.5.2. Розрахунок операційних витрат (ОРЕХ)

Для підтримки системи в робочому стані необхідні періодичні витрати.

Таблиця 3.4 — Річні експлуатаційні витрати

Стаття витрат	Опис	Вартість на рік, грн
Ліцензійний супровід	Оновлення ліцензії ваг («Техноваги»)	10 000
Витратні матеріали	Заміна пошкоджених RFID-міток (20% парку)	1 000

Стаття витрат	Опис	Вартість на рік, грн
Електроенергія	Споживання обладнання (0.5 кВт * 24год * 365днів * 8грн)	~35 000
ВСЬОГО ОРЕХ (C _{ор})		46 000 грн/рік

3.5.3. Джерела економічного ефекту

Розрахунок базується на мінімізації втрат готової продукції.

- Середня ринкова вартість продукції: Прийнято середнє значення між 200 та 350 грн/т = 275 грн/т.
- Середньомісячний обсяг видобутку: 100 000 тонн.

Фактор 1. Ліквідація несанкціонованого вивезення (Loss Prevention)

До впровадження системи (модель «As Is») розбіжність між даними геодезичних замірів у кар'єрі та даними вагової становила близько 0.5% - 1%. Це пояснюється помилками ручного введення та можливими зловживаннями (необліковані рейси).

Приймемо консервативний сценарій економії — 0.5% від обсягу.

- Збережений обсяг продукції: 100,000 т. * 0.005 = 500 т/міс.
- Грошовий еквівалент економії:

$$E_1 = 500 \text{ т} * 275 \text{ грн} = 137,500 \text{ грн/міс}$$

Фактор 2. Досягнення КРІ продуктивності

Для беззбиткової роботи підприємство має підтримувати темп видобутку > 250 тонн/годину.

Автоматизація скоротила час зважування на 20 секунд. При інтенсивному трафіку це дозволяє уникнути черг і простоїв екскаваторів в очікуванні самоскидів.

- Результат: Стабілізація вантажопотоку дозволяє гарантовано утримувати показник на рівні 260–280 т/год, що забезпечує стабільну прибутковість

підприємства. Цей ефект є стратегічним і не враховується в прямий грошовий потік для розрахунку окупності, але є критичним для бізнесу.

3.5.4. Показники ефективності інвестицій

Розрахуємо чистий грошовий потік за перший місяць (прибуток мінус місячна частка OPEX):

$$\text{Net Profit} = E_1 - \frac{C_{op}}{12} = 137\,500 - \frac{46\,000}{12} \approx 133\,600 \text{ грн/міс}$$

А) Термін окупності (Payback Period — PP):

$$PP = \frac{\text{CAPEX}}{\text{Net Profit}} = \frac{122\,500}{133\,600} \approx 0,92 \text{ місяця}$$

Система окупається менш ніж за 1 календарний місяць роботи.

Б) Рентабельність інвестицій (ROI) за перший рік:

$$ROI = \frac{(\text{Net Profit} \cdot 12) - \text{CAPEX}}{\text{CAPEX}} \cdot 100\%$$

$$ROI = \frac{(133\,600 \cdot 12) - 122\,500}{122\,500} \cdot 100\% = \frac{1\,603\,200 - 122\,500}{122\,500} \cdot 100\% \approx 1208\%$$

Висновки до пункту 3.5:

1. Завдяки використанню власної інфраструктури та In-house розробці, вартість впровадження склала лише 122 500 грн, що є вкрай низьким показником для проєктів такого масштабу.
2. Ліквідація втрат продукції навіть на рівні 0.5% приносить підприємству додатково 137 500 грн щомісяця.
3. Проєкт має хороший показник окупності (< 1 місяця), що робить його пріоритетним для впровадження. Окрім прямого прибутку, система гарантує виконання ключового KPI видобутку (250 т/год) за рахунок ритмічності роботи транспорту.

Висновки до Розділу 3

У третьому розділі магістерської роботи детально висвітлено процес практичної реалізації, налаштування та комплексної оцінки ефективності розробленої системи автоматизованого обліку. В ході виконання робіт було успішно вирішено інженерну задачу інтеграції різнорідного промислового

обладнання в єдиний інформаційний простір підприємства. Зокрема, для подолання проблеми територіальної віддаленості вагової від серверної та захисту каналів зв'язку від промислових завод було побудовано магістральну волоконно-оптичну мережу та впроваджено архітектуру Edge Computing. Використання локального контролера (АРМ на базі платформи VAF) дозволило забезпечити автономність технологічного процесу зважування, зробивши його незалежним від стабільності зовнішнього з'єднання.

Ключовим етапом реалізації стала розробка спеціалізованого модуля в конфігурації BAS ERP. Замість використання стороннього проміжного ПЗ, було реалізовано пряму взаємодію облікової системи з обладнанням: керування RFID-зчитувачем здійснюється через HTTP-запити до локального драйвера, а отримання ваги — через обробку подій COM-порту. Така архітектура дозволила реалізувати складну бізнес-логіку (фільтрацію шумів, перевірку черги, управління світлофорами) безпосередньо засобами вбудованої мови платформи VAF, що спрощує подальше адміністрування системи.

Результати комплексного тестування та дослідної експлуатації підтвердили високу надійність та швидкодію системи. Інструментальні заміри зафіксували скорочення повного часу реакції системи (Latency) до 0,45 секунди, що забезпечує високу пропускну здатність вагового комплексу. Стрес-тестування з імітацією фізичного пошкодження лінії зв'язку довело ефективність закладених архітектурних рішень: система успішно перейшла в автономний режим роботи без втрати даних, продемонструвавши коефіцієнт технічної готовності на рівні 99,85%.

Проведений економічний розрахунок переконливо засвідчив доцільність інвестицій у проєкт. Завдяки використанню існуючої інфраструктури підприємства та розробці програмного забезпечення власними силами (In-house) вдалося мінімізувати капітальні витрати до 122 500 грн. Аналіз грошових потоків показав, що лише за рахунок ліквідації втрат готової продукції проєкт окупається менш ніж за один календарний місяць (0,92 міс.), а рентабельність інвестицій у перший рік експлуатації перевищує 1200%, що робить впровадження системи стратегічно вигідним кроком для підвищення прибутковості підприємства.

ВИСНОВОК

У дипломній роботі вирішено актуальну науково-прикладну задачу автоматизації оперативного та бухгалтерського обліку на гірничодобувному підприємстві. В ході виконання роботи було проведено повний цикл розробки: від системного аналізу предметної області до фізичної реалізації та впровадження апаратно-програмного комплексу, що забезпечує наскрізний контроль руху гірничої маси.

Результати проведеного аналізу бізнес-процесів засвідчили, що традиційна модель обліку, базована на ручній фіксації даних оператором, виступає вузьким місцем технологічного процесу. Вона характеризується низькою пропускнуою здатністю, критичною залежністю від людського фактору та високими ризиками фінансових втрат через несанкціоноване вивезення продукції. На основі цих даних було обґрунтовано доцільність переходу до моделі повної автоматизації з використанням технології безконтактної ідентифікації RFID UHF, яка, на відміну від систем оптичного розпізнавання, продемонструвала стабільну роботу в умовах значної запиленості та забруднення транспортних засобів.

Вагомим результатом роботи стала розробка відмовостійкої архітектури системи, що враховує специфіку територіальної розподіленості об'єкта, де відстань від вагової до серверної перевищує 2 км. Було запропоновано та реалізовано концепцію Edge Computing (граничних обчислень). Впровадження локального автоматизованого робочого місця (АРМ) на базі промислового ПК дозволило перенести логіку прийняття рішень безпосередньо на периферію, забезпечивши автономність роботи вагового комплексу незалежно від стабільності магістральної оптоволоконної мережі.

Практична цінність роботи полягає у створенні унікального програмного модуля в середовищі BAS ERP. Засобами платформи BAF реалізовано пряму інтеграцію з промисловим обладнанням (RFID, ваги, IP-реле) без використання проміжного програмного забезпечення. Впроваджено алгоритми програмної стабілізації ваги та механізм асинхронної реплікації даних у систему бізнес-

аналітики Microsoft Power BI, що дозволяє керівництву отримувати оперативні звіти про КРІ видобутку в режимі реального часу.

Експериментальні дослідження та дослідна експлуатація підтвердили високу ефективність запропонованих рішень. Встановлено, що час повного циклу обслуговування одного автомобіля скоротився з 50 до 30 секунд, що дозволило збільшити пропускну здатність вагового пункту в 1,6 рази. Система продемонструвала високу експлуатаційну надійність, успішно пройшовши стрес-тести з імітацією фізичного розриву каналів зв'язку, при цьому коефіцієнт технічної готовності склав 99,85%.

Розрахунок техніко-економічних показників переконливо довів доцільність інвестицій у проєкт. Завдяки використанню існуючої інфраструктури підприємства та розробці програмного забезпечення власними силами вдалося суттєво мінімізувати капітальні витрати. Аналіз показав, що за рахунок ліквідації втрат продукції та оптимізації логістики проєкт окупився менш ніж за один календарний місяць, а рентабельність інвестицій у перший рік експлуатації перевищила 1200%. Таким чином, розроблена система є завершеним інженерним рішенням, готовим до промислового тиражування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Verkhovna Rada of Ukraine. (1999). Law of Ukraine on Accounting and Financial Reporting in Ukraine (No. 996-XIV). Retrieved November 10, 2023, from <https://zakon.rada.gov.ua/laws/show/996-14>
2. State Enterprise "UkrNDNC". (2016). Information technology. Radio frequency identification for item management. Part 63. Parameters for air interface communications at 860 MHz to 960 MHz Type C (DSTU ISO/IEC 18000-63:2016). Kyiv: State Enterprise "UkrNDNC".
3. ISO/IEC/IEEE. (2011). Systems and software engineering — Architecture description (ISO/IEC/IEEE 42010:2011).
4. Technowagy. (2022). Automobile strain gauge scales TV-A. Operation manual. Lviv: Technowagy.
5. UHFPrimeREADER. (2023). Dynamic Link Library User Guide V1.0 [Internal developer documentation].
6. Modbus Organization, Inc. (2006). Modbus Application Protocol Specification V1.1b. Retrieved from <http://www.modbus.org/specs.php>
7. Business Automation Union. (2021). BAS ERP. Description of functional capabilities. Kyiv: Business Automation Union.
8. Cooper, A., Reimann, R., & Cronin, D. (2018). Interface. Fundamentals of interaction design. Saint Petersburg: Symbol-Plus.
9. Satyanarayanan, M. (2017). The Emergence of Edge Computing. *Computer*, 50(1), 30–39.
10. Lutz, M. (2013). *Learning Python* (5th ed.). Sebastopol, CA: O'Reilly Media.
11. Tanenbaum, A., & Wetherall, D. (2012). *Computer Networks* (5th ed.). Saint Petersburg: Piter.
12. PyModbus. (n.d.). PyModbus library documentation. Retrieved November 15, 2023, from <https://pymodbus.readthedocs.io/en/latest/>
13. Microsoft. (n.d.). Microsoft Power BI. Retrieved from <https://powerbi.microsoft.com>

14. Ministry of Finance of Ukraine. (2014). Regulation on Inventory of Assets and Liabilities (Order No. 879). Retrieved from <https://zakon.rada.gov.ua/laws/show/z1365-14>
15. Ministry of Finance of Ukraine. (1999). On Approval of the Chart of Accounts for Accounting of Assets, Capital, Liabilities and Business Operations of Enterprises and Organizations (Order No. 291).
16. Verkhovna Rada of Ukraine. (2003). Law of Ukraine on Electronic Documents and Electronic Document Circulation (No. 851-IV). Retrieved from <https://zakon.rada.gov.ua/laws/show/851-15>
17. Finkenzeller, K. (2010). *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication* (3rd ed.). Chichester: Wiley.
18. EPCglobal. (2013). *EPC Radio-Frequency Identity Protocols Generation-2 UHF RFID. Standard Specification, Version 2.0.1.*
19. Weinstein, R. (2005). RFID: A technical overview and its application to the enterprise. *IT Professional*, 7(3), 27–33.
20. Glover, B., & Bhatt, H. (2006). *RFID Essentials*. Sebastopol, CA: O'Reilly Media.
21. Peris-Lopez, P., Hernandez-Castro, J. C., Estevez-Tapiador, J. M., & Ribagorda, A. (2006). RFID protocols: An overview. *Computer Communications*, 31(13).
22. Martin, R. C. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall.
23. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
24. McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython* (2nd ed.). O'Reilly Media.
25. Beazley, D., & Jones, B. K. (2013). *Python Cookbook* (3rd ed.). O'Reilly Media.
26. Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.
27. Summerfield, M. (2007). *Rapid GUI Programming with Python and Qt*. Prentice Hall.

28. Date, C. J. (2003). *An Introduction to Database Systems* (8th ed.). Addison-Wesley.
29. PostgreSQL Global Development Group. (n.d.). PostgreSQL 14 Documentation. Retrieved from <https://www.postgresql.org/docs/>
30. Monk, E., & Wagner, B. (2012). *Concepts in Enterprise Resource Planning* (4th ed.). Course Technology Cengage Learning.
31. O’Leary, D. E. (2000). *Enterprise Resource Planning Systems: Systems, Life Cycle, Electronic Commerce, and Risk*. Cambridge University Press.
32. Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. Wiley.
33. Мережеві протоколи та Промислова автоматизація
34. Galloway, B., & Hancke, G. P. (2013). Introduction to Industrial Control Networks. *IEEE Communications Surveys & Tutorials*, 15(2), 860–880.
35. Clarke, G., & Reynders, D. (2004). *Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems*. Elsevier.
36. Postel, J. (1981). Transmission Control Protocol (RFC 793). Internet Engineering Task Force.
37. Chappell, D. (2010). *OPC Unified Architecture*. Dictus Publishing.
38. Аналітика даних та Візуалізація (Power BI)
39. Ferrari, A., & Russo, M. (2019). *The Definitive Guide to DAX: Business intelligence for Microsoft Power BI, SQL Server Analysis Services, and Excel*. Microsoft Press.
40. Powell, B. (2018). *Mastering Microsoft Power BI*. Packt Publishing.
41. Wilke, C. O. (2019). *Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures*. O'Reilly Media.
42. Project Management Institute. (2017). *A Guide to the Project Management Body of Knowledge (PMBOK Guide)* (6th ed.). Newtown Square, PA: PMI.
43. ISO/IEC. (2013). *Information technology — Security techniques — Information security management systems — Requirements (ISO/IEC 27001:2013)*.
44. Sommerville, I. (2015). *Software Engineering* (10th ed.). Pearson.

ДОДАТОК А

Лістинг програмного коду

```
&AtServer
Procedure OnCreateAtServer(Cancel, StandardProcessing)
    StandardPeriod.Variant = StandardPeriodVariant.Today;
    StockMovementTotals.Parameters.SetParameterValue("StartDate",
StandardPeriod.StartDate);
    StockMovementTotals.Parameters.SetParameterValue("EndDate", StandardPeriod.EndDate);
    StockMovement.Parameters.SetParameterValue("StartDate",
StandardPeriod.StartDate);
    StockMovement.Parameters.SetParameterValue("EndDate", StandardPeriod.EndDate);
    Parameters.Property("AutomaticWeightRetrieval", AutomaticWeightRetrieval);
    Query = New Query;
    Query.Text =
        "SELECT
        |     Cars.Ref As Car,
        |     Cars.rfid
        |FROM
        |     Catalog.Cars AS Cars";

    Cars.Load(Query.Execute().Unload());
EndProcedure

&AtClient
Procedure ChangePeriod(Element)
    StockMovementTotals.Parameters.SetParameterValue("StartDate",
StandardPeriod.StartDate);
    StockMovementTotals.Parameters.SetParameterValue("EndDate", StandardPeriod.EndDate);
    StockMovement.Parameters.SetParameterValue("StartDate",
StandardPeriod.StartDate);
    StockMovement.Parameters.SetParameterValue("EndDate", StandardPeriod.EndDate);

EndProcedure

&AtClient
Procedure OnOpen(Cancel)
    If AutomaticWeightRetrieval Then
        AttachIdleHandler("IdleHandler1", 0.5, True);
        Start(Undefined);
        GetTagInfo(Undefined);
```

```

EndIf;

    WeighingCompleted = True;
    AttachIdleHandler("UploadDocuments", 60, False);

EndProcedure

&AtClient
Procedure UploadDocuments()
    UploadDocumentsAtServer()
EndProcedure

&AtServer
Procedure UploadDocumentsAtServer()
    DocumentArray = New Array;
    Query = New Query;
    Query.Text = "SELECT
        | RawMaterialReceipt.Ref AS Ref,
        | RawMaterialReceipt.Number AS Number,
        | RawMaterialReceipt.Date AS Date,
        | RawMaterialReceipt.Car AS Car,
        | RawMaterialReceipt.GrossWeight AS GrossWeight,
        | RawMaterialReceipt.Comment AS Comment,
        | RawMaterialReceipt.DeletionMark AS DeletionMark,
        | RawMaterialReceipt.NetWeight AS NetWeight,
        | ISNULL(RawMaterialReceipt.Car.Description, "") AS CarDescription
    | FROM
        | Document.RawMaterialReceipt AS RawMaterialReceipt
    | WHERE
        | RawMaterialReceipt.Sent = FALSE
        | AND RawMaterialReceipt.Date > DATETIME(2024, 8, 1)";

    Selection = Query.Execute().Select();
    If Selection.Count() > 0 Then
        While Selection.Next() Do

            DocumentStructure = New Structure;
            DocumentStructure.Insert("UID", String(Selection.Ref.UUID()));
            DocumentStructure.Insert("Date",
                "DF=yyyyMMddHHmmss"), Format(Selection.Date,
            DocumentStructure.Insert("Number", Selection.Number);
            DocumentStructure.Insert("CarUID", String(Selection.Car.UUID()));

```

```

        DocumentStructure.Insert("CarDescription", Selection.CarDescription);
        DocumentStructure.Insert("GrossWeight", Selection.GrossWeight);
        DocumentStructure.Insert("Comment", Selection.Comment);
        DocumentStructure.Insert("DeletionMark", Selection.DeletionMark); // Or
"Cancelled" based on your context
        DocumentStructure.Insert("NetWeight", Selection.NetWeight);

        DocumentArray.Add(DocumentStructure);

                EndDo;

                JSONString = GenerateJSONString(DocumentArray);

        Headers = New Map;
        Headers.Insert("Authorization", "Basic bW9kdWxzZ2Z0OnRyeTY1NA==");
        HTTPConnection = New HTTPConnection("kdz.msht.eu", 8888, "", "",, 200, New
OpenSSLSecureConnection());
        HTTPRequest = New HTTPRequest("KDZ/hs/PowerBI/weighingBelaz", Headers);
        HTTPRequest.SetBodyFromString(JSONString);
        RequestResult = HTTPConnection.CallHTTPMethod("POST", HTTPRequest);
                If RequestResult.StatusCode = 200 Then
                        Selection.Reset();

                While Selection.Next() Do
                        DocObject = Selection.Ref.GetObject();
                        DocObject.Sent = True;
                        DocObject.Write();

                EndDo;

                        EndIf;

                EndIf;

        EndProcedure

&AtClient
Procedure IdleHandler1()
        RequestScales();
        AttachIdleHandler("IdleHandler2", 0.5, True); // Polling frequency 0.5 seconds (comment
said 0.1 but code had 0.5)
EndProcedure

&AtClient
Procedure IdleHandler2()
        RequestScales();
        AttachIdleHandler("IdleHandler1", 0.5, True);
EndProcedure

```

```

&AtClient
Procedure RequestScales()
    If Not rs232 = Undefined Then
        Try
            rs232.Output("G01W");
        Except
            EndTry;
    EndIf;

EndProcedure

&AtClient
Procedure ExternalEvent(Source, Event, Data)
    //Stable =?(StrFind(Data, "=") > 0, "+", "-");
    DataOriginal = Data;
    Data = Upper(Data);
    Data = StrReplace(Data, "10R01W", "");
    Data = StrReplace(Data, "R01W", "");
    Data = StrReplace(Data, "R01E", "Error ");
    Data = StrReplace(Data, "R01E", "Error "); // Duplicate in original code
    Data = StrReplace(Data, "=", "");
    Data = StrReplace(Data, "~:", "");
    Try
        Weight = Number(Data);
        If Weight > 30 Then // If less than 30 tons, do not weigh automatically
            If Weight - PreviousWeight >= -0.1 AND Weight - PreviousWeight <= 0.1 AND
IterationCount >= 2 AND WeighingCompleted = True Then
                // Weight is stable for 3 iterations. Automatically fix the load.
                WeighingCompleted = False;

                WeighCar();

                PreviousWeight = 0;
                IterationCount = 0;

                SendSignalToRelay(); // Weighing fixed, green light
            Else
                // Weight is not stable.

```

```

                                If Weight - PreviousWeight >= -0.1 AND Weight - PreviousWeight <=
0.1 Then
                                    IterationCount = IterationCount + 1;
                                Else
                                    PreviousWeight = Weight;
                                    IterationCount = 0;
                                EndIf;
                                EndIf;
                                Else
                                    PreviousWeight = 0;
                                    IterationCount = 0;
                                    If WeighingCompleted = False Then
                                        SendSignalToRelay(); // Weighing completed, red light (logic might
be inverted in comment vs code)
                                        WeighingCompleted = True;
                                    EndIf;
                                EndIf;
                                Except
                                    Weight = 0;
                                    PreviousWeight = 0;
                                    IterationCount = 0;
                                EndTry;
                                EndProcedure

&AtClient
Procedure WeighCar()

    CarRef = GetCarOnScales();
    DocumentRef = CreateRawMaterialReceiptDocument(Weight, CarRef);
    CarWeighed(CarRef);
    NotifyChanged(DocumentRef);
    Items.StockMovementTotals.Refresh();

EndProcedure

```

&AtClient

Procedure SendSignalToRelay()

```
    RelayParams = New Structure;  
    RelayParams.Insert("Server",          "192.168.1.222");  
    RelayParams.Insert("ResourceAddress",  "/protect/leds.cgi?led=1&timeout=0");  
    RelayParams.Insert("Port",           80);  
    RelayParams.Insert("User", "admin");  
    RelayParams.Insert("Password",       "admin");  
    CommonModuleClient.SendSignalToRelay(RelayParams); // Assuming CommonModuleClient is  
the name of your module
```

EndProcedure

&AtServerNoContext

Function CreateRawMaterialReceiptDocument(Weight, CarRef)

```
    DocumentRef = Documents.RawMaterialReceipt.CreateDocument();  
    DocumentRef.Date = CurrentDate();  
    DocumentRef.Car = CarRef;  
    DocumentRef.GrossWeight = Weight;  
    DocumentRef.NetWeight = Weight - CarRef.CarWeight;  
    DocumentRef.Write();  
  
    Return DocumentRef.Ref;
```

EndFunction

&AtClient

Procedure CreateReceipt(Command)

```
    CarRef = GetCarOnScales();  
    OpenForm("Document.RawMaterialReceipt.Form.DocumentForm", New Structure("Weight, Car",  
Weight, CarRef));  
    CarWeighed(CarRef);
```

EndProcedure

&AtClient

Procedure OnClose(Exit) // 'Exit' param is sometimes standard in newer versions, check context

```

        DetachIdleHandler ("RequestScales");
    Try
        rs232.ClosePort ();
    Except
    EndTry;
EndProcedure

&AtClient
Procedure ExitApp(Command)
    Close ();
    Terminate(True);
EndProcedure

////////////////////////////////////
////////////////////////////////////
////////////////////////////////////

&AtClient
Procedure Start(Command)

    HTTPRequestHeader = New Map();
    HTTPRequestHeader.Insert("Content-Type",      "application/json");
    HTTPRequestHeader.Insert("Accept",           "application/json");

    //////////////////////////////////
    ////////// Opening Port //////////
    //////////////////////////////////

    hComm = 0;

    Request = New HTTPRequest();
    Request.Headers = HTTPRequestHeader;

    Request.ResourceAddress = "OpenDevice";
    Request.SetBodyFromString("{
        |   ""strComPort"": ""COM3"",
        |   ""Baudrate"": 4
        |});");

```

```

Connection = New HTTPConnection("localhost", 9094);

HTTPResponse = Connection.CallHTTPMethod("POST", Request);

RequestResult = HTTPResponse.GetBodyAsString();

//Message("OpenDevice:" + HTTPResponse.StatusCode + " " + RequestResult);

JSONReader = New JSONReader();
JSONReader.SetString(RequestResult);
JSONResult = ReadJSON(JSONReader);
JSONReader.Close();

If HTTPResponse.StatusCode = 200 AND JSONResult.data.hComm Then
    hComm = JSONResult.data.hComm;

    //////////////////////////////////////
    ////////// GETTING SETTINGS
    //////////////////////////////////////

    Request = New HTTPRequest();
    Request.Headers = HTTPRequestHeader;
    Request.ResourceAddress = "GetDevicePara";
    Request.SetBodyFromString("{
        |   "hComm": "" + Format(hComm, "NG=") + ""
        |}");
    Connection = New HTTPConnection("localhost", 9094);

    HTTPResponse = Connection.CallHTTPMethod("POST", Request);
    RequestResult = HTTPResponse.GetBodyAsString();

    //Message("GetDevicePara:" + HTTPResponse.StatusCode + " " + RequestResult);

    //////////////////////////////////////
    ////////// Start Reading
    //////////////////////////////////////

    Request = New HTTPRequest();
    Request.Headers = HTTPRequestHeader;

```

```

Request.ResourceAddress = "StartCounting";
Request.SetBodyFromString("{
    |   ""hComm"": "" + Format(hComm, "NG=") + "",
    |   ""ACSADDR"": 0,
    |   ""ACSDATALEN"": 0,
    |   ""ANT"": 1,
    |   ""BAUDRATE"": 4,
    |   ""BUZZERTIME"": 0,
    |   ""CN"": 50,
    |   ""DEVICEARRD"": 0,
    |   ""FILTERTIME"": 1,
    |   ""INTERNELTIME"": 0,
    |   ""INTERFACE"": 128,
    |   ""INVENTORYAREA"": 1,
    |   ""QVALUE"": 4,
    |   ""REGION"": ""1"",
    |   ""RFIDPOWER"": 30,
    |   ""RFIDPRO"": 0,
    |   ""SESSION"": 0,
    |   ""STEPFRE"": 500,
    |   ""STRATFRED"": ""902.750"",
    |   ""STRATFREI"": ""902.750"",
    |   ""TRIGGLETIME"": 3,
    |   ""WGSET"": 0,
    |   ""WORKMODE"": 1 }");

Connection = New HTTPConnection("localhost", 9094);

HTTPResponse = Connection.CallHTTPMethod("POST", Request);

    //Message("StartCounting:" + HTTPResponse.StatusCode + " " +
HTTPResponse.GetBodyAsString());

    //Message(Request.GetBodyAsString());

RestartRFID = CurrentDate() + 60*60;
DocumentCreated = CurrentDate();

EndIf;

EndProcedure

&AtClient

```

```

Procedure GetTagInfo(Command)

    AttachIdleHandler("GetTags", 2, False);

EndProcedure

&AtClient
Procedure GetTags() Export

    TagsResponse = "" + CurrentDate() + Chars.LF;

    If hComm <> 0 Then

        HTTPRequestHeader = New Map();
        HTTPRequestHeader.Insert("Content-Type", "application/json");
        HTTPRequestHeader.Insert("Accept", "application/json");

        ///////////////////////////////////////////////////////////////////
        ////// GETTING TAGS //////
        ///////////////////////////////////////////////////////////////////

        Request = New HTTPRequest();
        Request.Headers = HTTPRequestHeader;

        Request.ResourceAddress = "GetTagInfo";
        Connection = New HTTPConnection("localhost", 9094);
        HTTPResponse = Connection.CallHTTPMethod("POST", Request);
        RequestResult = HTTPResponse.GetBodyAsString();

        JSONReader = New JSONReader();
        JSONReader.SetString(RequestResult);
        JSONResult = ReadJSON(JSONReader);
        JSONReader.Close();

        For Each TagRow In Tags Do
            TagRow.Available = False;
        EndDo;

        TagsResponse = TagsResponse + HTTPResponse.StatusCode + Chars.LF;
        TagsResponse = TagsResponse + RequestResult + Chars.LF;
    EndIf
EndProcedure

```

```

If JsonResult.code = 200 Then

    If      typeof(JsonResult.data.taginfo)      =      Type("Array")      AND
    JsonResult.data.taginfo.Count() > 0 Then

        For Each Tag In JsonResult.data.taginfo Do

            If ValueIsFilled(Tag.m_code) Then

                m_code = Tag.m_code;
                EndCountIndex = StrFind(Tag.m_counts, "/");

                TagCount = Number(Left(Tag.m_counts, EndCountIndex-
1));

                FoundRows = Tags.FindRows(New Structure("Tag",
Tag.m_code));

                CarAvailable = False;

                If FoundRows.Count() > 0 Then
                    If FoundRows[0].Count < TagCount Then
                        FoundRows[0].Count = TagCount;
                        FoundRows[0].Available = True;
                        CarAvailable = True;
                    EndIf;
                Else
                    NewRow = Tags.Add();
                    NewRow.Tag = Tag.m_code;
                    NewRow.Count = TagCount;
                    NewRow.Time = CurrentDate();
                    CarAvailable = True;
                EndIf;

                CarByTag = GetCar(Tag.m_code);

                If      CarByTag
                PredefinedValue("Catalog.Cars.EmptyRef") Then <>

                    FoundRows = CarOnScales.FindRows(New
Structure("Car", CarByTag));

```

```

True Then
    If FoundRows.Count() = 0 AND CarAvailable =
        NewRow = CarOnScales.Add();
        NewRow.Car = CarByTag;
        NewRow.Time = CurrentDate();
        NewRow.Available = CarAvailable;
    Else
        If FoundRows.Count() > 0 AND
            FoundRows[0].Available =
                CarAvailable;
        EndIf;
    EndIf;
EndIf;

EndIf;

EndDo;

Else
    EndIf;

Else
    EndIf;

For Each Car In CarOnScales Do
    If Car.Available = False AND Car.Time < CurrentDate()-5 Then
        CarOnScales.Delete(Car);
    EndIf;
EndDo;

EndIf;

//If RestartRFID < CurrentDate() AND
// (CurrentDate() - DocumentCreated < 10 Or CurrentDate() - DocumentCreated > 60 *
60) Then
//
// DetachIdleHandler("GetTags");

```

```

// CloseDevice(Undefined);
// Start(Undefined);
// GetTagInfo(Undefined);
//
//EndIf;

EndProcedure

&AtClient
Procedure CloseDevice(Command)

    If hComm <> 0 Then

        //////////////////////////////////////
        // Stop Reading //
        //////////////////////////////////////

        HTTPRequestHeader = New Map();
        HTTPRequestHeader.Insert("Content-Type", "application/json");
        HTTPRequestHeader.Insert("Accept", "application/json");

        Request = New HTTPRequest();
        Request.Headers = HTTPRequestHeader;
        Request.SetBodyFromString("{
            |   ""hComm"": "" + Format(hComm, "NG=") + "",
""timeout"":10000
            |});");

        Request.ResourceAddress = "InventoryStop";
        Connection = New HTTPConnection("localhost", 9094);

        HTTPResponse = Connection.CallHTTPMethod("POST", Request);
        Message(CurrentDate());
        Message("InventoryStop:" + HTTPResponse.StatusCode + " " +
HTTPResponse.GetBodyAsString());

        //////////////////////////////////////
        // Closing Port
        //////////////////////////////////////

        Request = New HTTPRequest();
        Request.Headers = HTTPRequestHeader;

```

```

Request.ResourceAddress = "CloseDevice";
Request.SetBodyFromString("{
    |   ""hComm"": "" + Format(hComm, "NG=") + ""
    |}");
Connection = New HTTPConnection("localhost", 9094);

HTTPResponse = Connection.CallHTTPMethod("POST", Request);

Message(CurrentDate());
Message("CloseDevice:" + HTTPResponse.StatusCode + " " +
HTTPResponse.GetBodyAsString());

EndIf;

hComm = 0;

EndProcedure

&AtClient
Function GetCar(rfid)

CarRows = Cars.FindRows(New Structure("rfid", rfid));

If CarRows.Count() = 0 Then
    Return PredefinedValue("Catalog.Cars.EmptyRef");
Else
    Return CarRows[0].Car;
EndIf;

EndFunction

&AtClient
Function GetCarOnScales()

CarRows = CarOnScales.FindRows(New Structure("Available", True));

If CarRows.Count() = 0 Then
    Return PredefinedValue("Catalog.Cars.EmptyRef");
Else

```

```

        Return CarRows[0].Car;
    EndIf;

EndFunction

&AtClient
Procedure CarWeighed(Car)

    CarRows = CarOnScales.FindRows(New Structure("Car", Car));

    If CarRows.Count() > 0 Then
        CarRows[0].Available = False;
        CarRows[0].Time = CurrentDate() + 15*60; // Make unavailable for 15 mins to
prevent re-weighing
    EndIf;

    DocumentCreated = CurrentDate();

EndProcedure

&AtClient
Procedure BeforeClose(Cancel, Exit, MessageText, StandardProcessing)

    CloseDevice(Undefined);

EndProcedure

Function GenerateJSONString(DataStructure, HasErrors = False, ErrorCode = "")

    JSONString = "";

    JSONWriterSettings = New JSONWriterSettings(JSONLineBreak.Auto, "", True,
JSONCharacterSetEscaping.None, False, True, True, True, False);

    FileEncoding = "utf-8";
    TempFile = GetTempFileName("txt");

    JSONWriter = New JSONWriter;
    JSONWriter.ValidateStructure = False;
    JSONWriter.Open(TempFile, FileEncoding, False, JSONWriterSettings);

```

```
WriteJSON(JSONWriter, DataStructure);
JSONWriter.Close();

FileWithString = New TextReader(TempFile, FileEncoding);
JSONString = FileWithString.Read();
FileWithString.Close();

// Delete temp file
DeleteFiles(TempFile);

Return JSONString;

EndFunction
```