

БАКАЛАВРСЬКА РОБОТА

БР.КІ-40.00.00.000 ПЗ

Група КІ-22-2

Мацигін Роман

2026

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу
Інститут інформаційних технологій
Кафедра комп'ютерних систем і мереж

Мацигін Роман Дмитрович

УДК 004.4'27

БАКАЛАВРСЬКА РОБОТА

Розробка програмно-апаратної підсистеми організації складського обліку на основі RFID

Комп'ютерна інженерія

(назва освітньої програми)

123 - Комп'ютерна інженерія

(шифр і назва спеціальності)

Робота містить результати власних досліджень, використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело:

Здобувач освітнього ступеня Мацигін Р.Д.
(підпис, ініціали та прізвище здобувача)

Науковий керівник Мойсеєнко О.В., доцент
(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
Завідувач кафедри КСМ

д.т.н., професор С.І. Мельничук
(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківськ – 2026 рік

. Івано-Франківський національний технічний університет нафти і газу

Інститут Інформаційних технологій

Кафедра Комп'ютерних систем і мереж

Освітньо-кваліфікаційний рівень бакалавр

Спеціальність 123 – Комп'ютерна інженерія

ЗАТВЕРДЖУЮ:

Зав. кафедрою КСМ

С.І. Мельничук

« 21 » квітня 2026 року

**З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Мацигіну Роману Дмитровичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка програмно-апаратної підсистеми організації складського обліку на основі RFID

керівник проекту (роботи) Мойсеєнко О.В., доцент.

затверджені наказом вищого навчального закладу від 21. 04. 2026 № 180/7

2. Строк подання студентом проекту (роботи) 11 червня 2026 р.

3. Вихідні дані до роботи Методичні вказівки, технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Проаналізувати об'єкт та поставити задачі автоматизації складського обігу. 2. Спроектувати структуру та моделі програмно-апаратного комплексу 3. Пропрацювати технічну реалізацію та провести тестування комплексу обліку запасів .

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)_____.

6. Консультанти розділів роботи

7. Дата видачі завдання 02 лютого 2026р.

№ з/п	Назва етапів дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	<i>Збір інформації, вивчення літератури та пошук додаткової інформації</i>	<i>Лютий 2026р</i>	
2	<i>Проаналізувати об'єкт та поставити задачі автоматизації складського обігу.</i>	<i>Березень 2026р</i>	
3	<i>Спроектувати структуру та моделі програмно-апаратного комплексу</i>	<i>Квітень 2026р</i>	
4	<i>Пропрацювати технічну реалізацію та провести тестування комплексу обліку запасів .</i>	<i>Травень 2026р</i>	
5	<i>Оформлення поясн.вальної записки</i>	<i>Червень 2026р</i>	

Студент _____
(підпис)

Мацигін Р.Д.
(прізвище та ініціали)

Керівник роботи _____

Мойсеєнко О.В.

АНОТАЦІЯ

У зв'язку зі стрімким зростанням товарообігу, автоматизація складського обліку стає критичною умовою ефективності сучасних логістичних систем. Традиційні методи ідентифікації, такі як штрих-кодування, не забезпечують необхідної швидкості та повної автоматизації групових операцій. У роботі розроблено програмно-апаратний комплекс на базі UHF RFID технологій та мікрокомп'ютера Raspberry Pi із використанням асинхронного фреймворку FastAPI. Спроектовано систему з інтелектуальними алгоритмами фільтрації дублікатів зчитувань та адаптивним веб-інтерфейсом для моніторингу операцій у реальному часі. Впровадження розробленого комплексу дозволяє прискорити процес інвентаризації у 100 разів та мінімізувати помилки, зумовлені людським фактором.

Ключові слова: rfid-технології, автоматизація складу, raspberry pi, fastapi, моніторинг, інвентаризація, uhf-зчитувач, веб-інтерфейс, облік запасів, контейнеризація.

ABSTRACT

Due to the rapid increase in turnover, warehouse accounting automation is becoming a critical condition for the efficiency of modern logistics systems. Traditional identification methods, such as barcoding, do not provide the necessary speed and full automation of group operations. The paper develops a software-hardware complex based on UHF RFID technologies and a Raspberry Pi microcomputer using the FastAPI asynchronous framework. A system with intelligent duplicate scan filtering algorithms and a responsive web interface for real-time operation monitoring has been designed. The implementation of the developed complex allows for accelerating the inventory process by 100 times and minimizing errors caused by the human factor.

Keywords: rfid technologies, warehouse automation, raspberry pi, fastapi, monitoring, inventory, uhf reader, web interface, stock accounting, containerization.

ЗМІСТ

ПЕРЕЛІК ОСНОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ.....	4
ВСТУП.....	5
1 АНАЛІЗ ОБ’ЄКТА ТА ПОСТАНОВКА ЗАДАЧІ АВТОМАТИЗАЦІЇ СКЛАДСЬКОГО ОБЛІКУ.....	7
1.1 Характеристика процесу обліку складських запасів та визначення інформаційних параметрів системи.....	7
1.2 Порівняльний аналіз технологій автоматичної ідентифікації.....	12
1.3 Огляд існуючих програмно-апаратних засобів та обґрунтування вибору елементної бази комплексу.....	19
1.4 Постановка задачі на розробку комплексу.....	22
2 ПРОЕКТУВАННЯ СТРУКТУРИ ТА МОДЕЛЕЙ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ.....	26
2.1 Розробка загальної архітектури RFID-комплексу	26
2.2 Алгоритмізація виробничих процесів на складі	31
2.3 Проектування структури бази даних	36
3 ТЕХНІЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ КОМПЛЕКСУ ОБЛІКУ ЗАПАСІВ... ..	42
3.1 Реалізація інтерфейсу взаємодії з RFID-зчитувачем.....	42
3.2 Розробка інтерфейсу користувача та функціональних модулів.....	45
3.3 Розробка схеми підключення та конфігурація апаратної частини	47
3.4 Тестування працездатності комплексу	52
ВИСНОВКИ.....	57
ПЕРЕЛІК ПОСИЛАНЬ НА ДЖЕРЕЛА	59
ДОДАТКИ	
Бібліографічна довідка	

					БР.КІ-40.00.00.000 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Мацигін Р.Д.			Розробка програмно-апаратної підсистеми організації складського обліку на основі RFID			Літ.	Арк.	Аркушів	
Перевір.		Мойсеєнко О.В.								3	
Реценз.								ІФНТУНГ, КІ-22-2			
Н. Контр.		Топалов А.М.									
Затверд.		Мельничук С.І.									

ПЕРЕЛІК ОСНОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

БД - база даних

БР - бакалаврська робота

ПАК - програмно-апаратний комплекс

ПЗ - пояснювальна записка (або програмне забезпечення, залежно від контексту)

API (Application Programming Interface) - прикладний програмний інтерфейс для взаємодії між компонентами системи

GPIO (General Purpose Input/Output) - інтерфейс введення-виведення загального призначення мікрокомп'ютера Raspberry Pi

HTTP (HyperText Transfer Protocol) - протокол передачі гіпертексту для обміну даними між клієнтом та сервером

IoT (Internet of Things) - концепція мережі фізичних об'єктів («речей»), оснащених технологіями для взаємодії один з одним

Python - високорівнева мова програмування загального призначення, що використовується для реалізації логіки комплексу

RAM (Random Access Memory) - оперативна пам'ять обчислювальної системи

RFID (Radio Frequency IDentification) - технологія радіочастотної ідентифікації об'єктів за допомогою радіосигналів

SDK (Software Development Kit) - набір засобів розробки, що дозволяє програмістам створювати прикладне ПЗ для певного обладнання

SQLite - компактна вбудована реляційна база даних, що використовується для локального зберігання складських запасів

UART (Universal Asynchronous Receiver-Transmitter) - універсальний асинхронний приймач-передавач для зв'язку Raspberry Pi з RFID-зчитувачем.

					БР.КІ-40.00.00.000 ПЗ	4
ЗМ.	Арк.	№ докум.	Підп.	Дата		

ВСТУП

Актуальність теми. Сучасний етап розвитку складської логістики характеризується стрімким зростанням обсягів товарообігу та підвищенням вимог до точності і швидкості обліку запасів. Традиційні методи ідентифікації, такі як штрих-кодування, мають суттєві обмеження: необхідність прямої видимості мітки, неможливість одночасного зчитування великої кількості об'єктів та низьку стійкість до механічних пошкоджень. Використання технології радіочастотної ідентифікації (RFID) дозволяє вирішити ці проблеми, забезпечуючи автоматизований контроль без участі людини та мінімізуючи помилки, спричинені «людським фактором». Розробка спеціалізованого програмно-апаратного комплексу на основі RFID є актуальним завданням для підвищення ефективності управління сучасними складськими об'єктами.

Об'єкт дослідження - процеси автоматизованого обліку, контролю та інвентаризації складських запасів у середовищі сучасних логістичних систем.

Предмет дослідження - програмно-апаратні засоби ідентифікації об'єктів на основі RFID-технологій, алгоритми обробки даних зі зчитувачів та методи їх інтеграції в інформаційну систему складу.

Мета роботи - розробка програмно-апаратного комплексу автоматизованого обліку складських запасів на основі RFID-технологій для підвищення ефективності, швидкості та достовірності контролю товарно-матеріальних цінностей.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Проаналізувати сучасний стан, технології та методи автоматизації складських процесів.
2. Обґрунтувати вибір апаратної бази та програмних засобів для побудови RFID-комплексу.
3. Розробити загальну архітектуру комплексу, структуру бази даних та алгоритми взаємодії з обладнанням.

					БР.КІ-40.00.00.000 ПЗ	5
ЗМ.	Арк.	№ докум.	Підп.	Дата		

4. Створити програмне забезпечення для мікрокомп'ютера Raspberry Pi та розробити користувацький інтерфейс моніторингу.

5. Здійснити експериментальне тестування комплексу, оцінити точність зчитування та загальну ефективність системи.

Методи дослідження слугують інструментом для вирішення поставлених завдань, зокрема використано:

- системний аналіз - для вивчення об'єкта автоматизації та визначення вимог до системи;

- об'єктно-орієнтоване проектування - для розробки програмної архітектури та структури бази даних;

- експериментальне тестування - для оцінки дальності й точності зчитування RFID-міток в умовах складу..

Практичне значення отриманих результатів. Розроблений комплекс може бути впроваджений на підприємствах різного масштабу для автоматизації інвентаризації, контролю переміщення вантажів та оптимізації роботи складського персоналу, що дозволить скоротити час на обробку даних та зменшити логістичні витрати.

					БР.КІ-40.00.00.000 ПЗ	
ЗМ.	Арк.	№ докум.	Підп.	Дата		6

1 АНАЛІЗ ОБ'ЄКТА ТА ПОСТАНОВКА ЗАДАЧІ АВТОМАТИЗАЦІЇ СКЛАДСЬКОГО ОБЛІКУ

1.1 Характеристика процесу обліку складських запасів та визначення інформаційних параметрів системи

Сучасний етап розвитку ринкових відносин характеризується стрімким збільшенням обсягів товарообігу, розширенням номенклатури продукції та ускладненням логістичних ланцюгів управління поставками. У таких умовах склад перестає бути виключно місцем консервативного зберігання матеріальних активів, перетворюючись на динамічний транспортно-розподільчий вузол, ефективність функціонування якого безпосередньо визначає конкурентоздатність та прибутковість всього підприємства. Складський облік є фундаментальним елементом логістичної системи, головна мета якого полягає у забезпеченні безперервності виробничих або торговельних процесів шляхом прозорого, своєчасного та точного контролю за рухом товарно-матеріальних цінностей (ТМЦ).

Управління складом базується на виконанні регламентованого набору бізнес-процесів, які об'єднують як фізичні операції з вантажем, так і документування цих дій в інформаційній системі. Рух товарів на типовому складі можна структурувати за такими основними етапами:

1. Прийом товарів (Receiving). Цей процес ініціюється прибуттям транспортного засобу від постачальника. На даному етапі відбувається розвантаження, ідентифікація продукції, перевірка відповідності фактичної кількості, якості та номенклатури тим даним, що вказані у супровідних документах постачальника (товарно-транспортних накладних, специфікаціях). Розбіжності, виявлені під час прийому, фіксуються в актах про нестачу або пересортицю.

					БР.КІ-40.00.00.000 ПЗ	7
ЗМ.	Арк.	№ докум.	Підп.	Дата		

2. Розміщення та зберігання (Put-away and Storage). Після успішного приймання товари транспортуються до зон довгострокового або короткострокового зберігання. Цей процес вимагає визначення вільної комірки стелажа чи зони на підлозі з урахуванням фізико-хімічних властивостей товару (температурний режим, вологість), габаритів та показників оборотності (наприклад, за принципами ABC-аналізу, коли товари з високим попитом розміщуються ближче до зони відвантаження).

3. Комплектація замовлень (Picking). Є найбільш трудомістким етапом, на частку якого припадає понад 50% усіх витрат робочого часу персоналу складу. Згідно з отриманими заявками від клієнтів або виробничих підрозділів, комірники здійснюють обхід складських локацій, відбір необхідної кількості товару різних артикулів та їх об'єднання в єдине замовлення. Ефективність цього процесу цілком залежить від точності інформації щодо того, де саме лежить потрібний екземпляр.

4. Пакування та відвантаження (Packing and Shipping). Зібрані замовлення проходять фінальний контроль (часто шляхом повторного перерахунку), упаковуються для запобігання пошкодженням під час транспортування, маркуються транспортними етикетками та передаються експедитору або клієнту. Супроводжується цей процес видачею видаткових накладних та маршрутних листів.

5. Інвентаризація (Inventorying). Регулярний або позаплановий процес звірки фактичної наявності товарів на полицях з тими даними, що зафіксовані у системі обліку. Традиційно інвентаризація супроводжується повним або частковим припиненням роботи складу, що спричиняє значні фінансові втрати.

Розгляд послідовності цих операцій дозволяє сформувати чітку логіку руху інформаційних потоків. Типова схема взаємодії складських процесів наведена на рисунку 1.1.

					БР.КІ-40.00.00.000 ПЗ	8
ЗМ.	Арк.	№ докум.	Підп.	Дата		

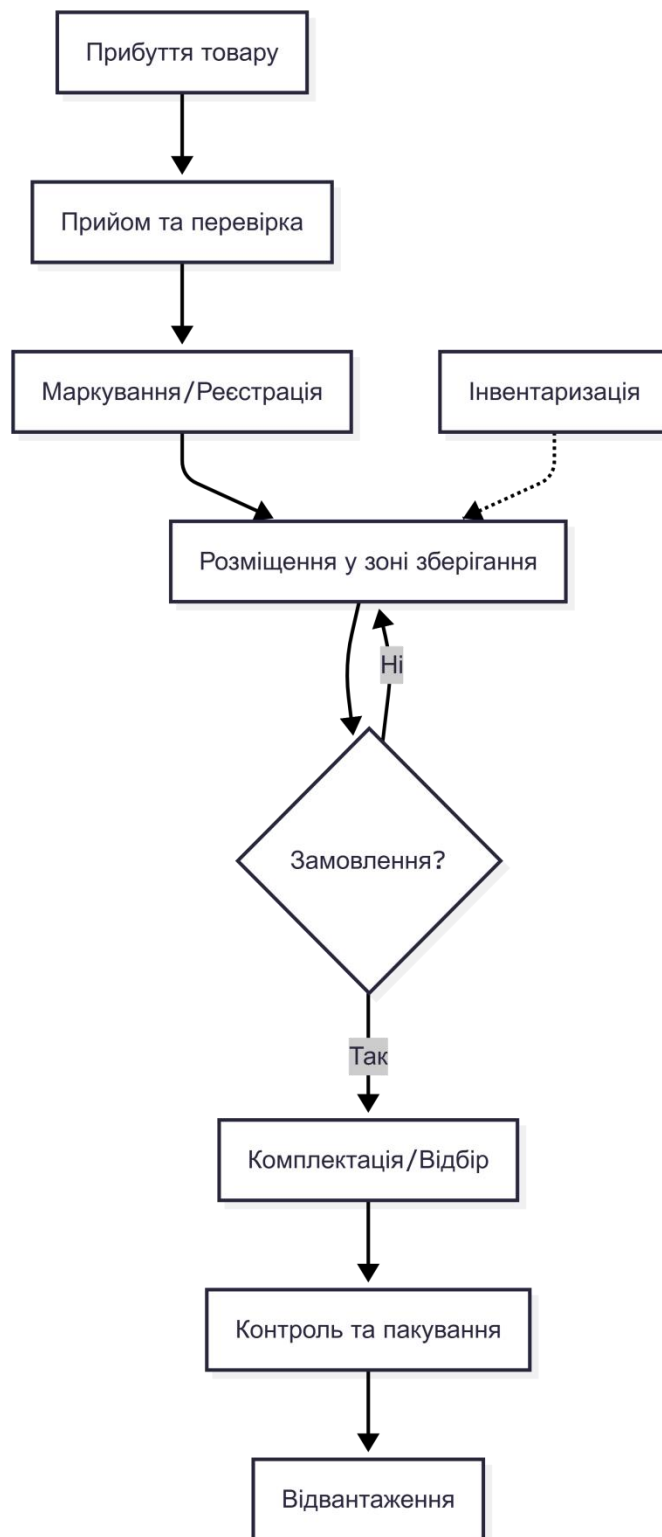


Рисунок 1.1 - Схема бізнес-процесів складського обліку

Інформаційний супровід усіх зазначених процесів підтримується через чітку систему документообігу. Типова структура документів складського обліку включає:

					БР.КІ-40.00.00.000 ПЗ	9
ЗМ.	Арк.	№ докум.	Підп.	Дата		

- Прибуткові ордери - підтверджують факт надходження товарів і взяття їх на баланс підприємства матеріально-відповідальною особою.
- Акти прийому-передачі - використовуються для внутрішнього переміщення товарів між різними складами або цехами підприємства.
- Картки складського обліку матеріалів - у них ведеться детальний хронологічний запис руху конкретної номенклатурної позиції (прибуток, видаток, залишок).
- Видаткові накладні, лімітно-забірні карти - документи, що є підставою для легального зняття товарів із залишків.
- Інвентаризаційні описи та акти звіряння - фіксують факт наявності ТМЦ в момент перевірки та фіксують результати порівняння з базою даних.

Класичний підхід до виконання облікових операцій, який спирається на паперові носії інформації або застосування систем поштучного оптичного штрих-кодування, має низку критичних недоліків. У процесах масового переміщення та зберігання вантажів визрівають специфічні інформаційні параметри та фізичні умови, які виставляють жорсткі вимоги до майбутніх інструментів реєстрації даних, і тим самим безпосередньо обумовлюють функціональність проектного програмно-апаратного комплексу (ПАК).

Основними інформаційними параметрами та умовами, що вимагають технічного вирішення при розробці ПАК, є:

1. Необхідність зчитування даних без прямої видимості. Більшість товарів зберігаються у транспортній тарі, закритих картонних палетах або розташовуються на верхніх ярусах висотних стелажів (на висоті 5-10 метрів). Використання оптичних сканерів штрих-кодів вимагає прямої видимості етикетки і наближення сканера на коротку відстань, що змушує комірників розпаковувати палети та витрачати час на підйом вантажів. Проектована система має забезпечити ідентифікацію об'єктів крізь неметалеві перешкоди на відстані кількох метрів.

2. Швидкість та групова ідентифікація об'єктів (Mass Registration). Обсяги товарних потоків вимагають здатності швидко розпізнавати десятки або сотні

					БР.КІ-40.00.00.000 ПЗ	10
ЗМ.	Арк.	№ докум.	Підп.	Дата		

об'єктів одночасно. Поштучне сканування штрих-кодів кожної коробки в палеті є неприйнятним. Система повинна мати здатність "антиколізії" - зчитувати маркування всього переліку товарів у зоні дії за соті долі секунди (наприклад, миттєво під час проїзду навантажувача через складські ворота).

3. Унікальна ідентифікація на рівні екземпляра. Для точного обліку недостатньо знати лише загальний артикул (SKU) чи клас товару. Інформаційний параметр системи має включати можливість відстеження кожної конкретної одиниці за її унікальним серійним номером. Це необхідно для контролю термінів придатності, гарантійного обслуговування, а також боротьби з контрафактною продукцією.

4. Стійкість ідентифікаторів до складських умов експлуатації. Паперові або плівкові носії зі штрих-кодами легко пошкоджуються під час транспортування, забруднюються пилом, стираються або розмокають. Система реєстрації має використовувати носії даних, які є відмовостійкими у жорстких промислових умовах.

5. Динамічна актуалізація даних (читання/запис). Сучасний складський облік потребує не лише зчитування статичного ідентифікатора, а й можливості перезапису інформації в локальну пам'ять самої мітки під час руху ланцюгом постачання (наприклад, запис дати відвантаження або заміна адреси доставки), що неможливо реалізувати звичайними друкованими методами.

6. Автоматична реєстрація переміщень у реальному часі. Інформаційна система складу повинна завжди відображати актуальну картину залишків на поточну секунду. Затримки між фізичною видачею товару та внесенням даних в систему оператором призводять до віртуальної "пересортиці" та неможливості якісного планування поставок.

На основі проведеного аналізу характеристик бізнес-процесів складського обліку, а також виявлених архітектурних та інформаційних вимог до операцій з реєстрації об'єктів, можна зробити однозначний висновок. Традиційні ручні процеси на основі паперових документів або поштучне (штрихкодове) сканування вже не відповідають вимогам сучасної швидкоплинної логістики.

					БР.КІ-40.00.00.000 ПЗ	11
ЗМ.	Арк.	№ докум.	Підп.	Дата		

Автоматизація складу шляхом впровадження спеціалізованого програмно-апаратного комплексу (насамперед на базі технологій радіочастотної інвентаризації, які можуть задовольнити інформаційні вимоги безконтактного масового зчитування) є критично необхідним кроком. Це дозволить повністю усунути вплив "людського фактора" під час приймання і видачі, який часто призводить до помилок комплектації замовлень та нестач матеріальних цінностей. Автоматизований облік забезпечує кардинальне прискорення роботи: швидкість проведення інвентаризації збільшується в десятки разів без необхідності зупинки роботи всього складу.

Зрештою, впровадження такого комплексу гарантовано призводить до зниження прямих і непрямих фінансових втрат підприємства завдяки мінімізації крадіжок, уникненню списань товарів, термін придатності яких минув поза увагою працівників, а також сприяє оптимізації використання складських площ та робочого часу логістичного персоналу.

1.2 Порівняльний аналіз технологій автоматичної ідентифікації

Ефективність функціонування будь-якого сучасного програмно-апаратного комплексу обліку цілком залежить від швидкості та достовірності первинного збору інформації. Процедура автоматичної ідентифікації та збору даних (AIDC – Automatic Identification and Data Capture) полягає у розпізнаванні фізичних об'єктів за допомогою спеціалізованих технічних засобів та автоматичному введенні отриманих даних в інформаційну систему без безпосередньої участі людини. На сьогодні розроблено ряд технологічних підходів до вирішення цієї задачі, кожен з яких займає свою нішу в залежності від технічних, економічних та експлуатаційних обмежень. До найбільш розповсюджених технологій належать: одомірне та двомірне штрих-кодування, використання магнітних смуг, технологія ближнього безконтактного зв'язку (NFC) та системи радіочастотної ідентифікації (RFID).

Для детального розуміння кожної з них, наведемо їх короткий огляд:

					БР.КІ-40.00.00.000 ПЗ	12
ЗМ.	Арк.	№ докум.	Підп.	Дата		

– штрих-кодування (Barcode) - найстарша та найбільш масово впроваджена технологія у сфері роздрібної торгівлі та логістики. Базується на оптичному розпізнаванні графічного зображення у вигляді послідовності чорних і білих смуг (1D-коди, такі як EAN-13, UPC) за допомогою лазерних або світлодіодних сканерів. Основна перевага - мінімальна собівартість друку етикетки і дешевизна обладнання. Критичні недоліки - необхідність точного оптичного контакту (прямої видимості) між сканером і кодом, малий обсяг кодованої інформації (лише короткий номер товару) та близька до нульової стійкість до механічних пошкоджень або забруднень.

– двомірні коди (QR-коди, DataMatrix) - еволюційний розвиток технології штрих-кодування (2D-коди). Дозволяють кодувати значно більший обсяг інформації (до кількох тисяч символів) у вигляді матриці квадратів. Вони мають вищу надійність завдяки вбудованим алгоритмам корекції помилок (код може бути зчитаний навіть при пошкодженні до 30% площі етикетки). Однак, дана технологія, як і класичний штрих-код, є виключно оптичною, тому зовсім не вирішує проблеми обов'язкової прямої видимості та поштучного сканування кожного об'єкта.

– магнітні смуги (Magnetic Stripe) - технологія, що передбачає запис інформації на тонку магнітну плівку. Широко застосовувалася у банківських картках та системах контролю доступу. Забезпечує можливість багаторазового перезапису. Втім, через низьку зносостійкість, малий обсяг пам'яті (кількадесят байт), необхідність фізичного контакту зі зчитувачем (проведення карткою по слоту) і високу вразливість до дії зовнішніх електромагнітних полів, ця технологія практично не застосовується для складського обліку товарних запасів.

– Near Field Communication (NFC) - технологія безконтактного високочастотного зв'язку малого радіусу дії (High Frequency, 13.56 МГц). Є спорідненою гілкою RFID. Особливістю NFC є те, що зчитування можливе лише на дуже коротких відстанях (зазвичай до 5–10 сантиметрів). Ідеально підходить для платіжних систем та персональної авторизації, але через мізерну дальність дії та неможливість швидкого антиколізійного опитування сотень міток одночасно, NFC є неефективним для масової складської інвентаризації та відслідковування великогабаритних вантажів.

					БР.КІ-40.00.00.000 ПЗ	13
ЗМ.	Арк.	№ докум.	Підп.	Дата		

– радіочастотна ідентифікація (RFID) - інноваційна технологія, що використовує радіохвилі для передачі даних між зчитувачем і транспондером (міткою), який закріплюється на об'єкті. RFID-мітки поділяються на активні (з власним джерелом живлення) та пасивні (живляться енергією електромагнітного поля антени зчитувача). У логістиці найбільше розповсюдження отримали пасивні UHF (Ultra High Frequency) RFID мітки, які здатні передавати дані на відстані до 10-15 метрів. Технологія забезпечує унікальність ідентифікації (використання блоку пам'яті EPC - Electronic Product Code), підтримує багаторазовий перезапис даних і не потребує прямої видимості мітки.

Для детального аналізу можливостей кожної технології, на рисунку 1.2 представлено порівняння зон зчитування та пропускної здатності основних методів ідентифікації.

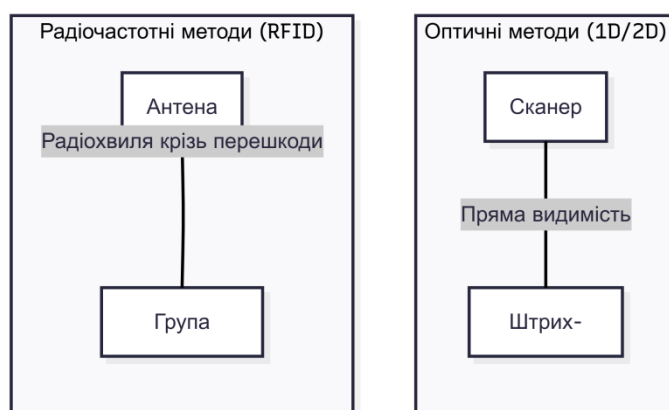


Рисунок 1.2 - Схематичне порівняння принципів роботи оптичних та радіочастотних методів

Систематизуємо характеристики розглянутих технологій у зведеній порівняльній таблиці.

На основі проведеного огляду та даних, зведених у таблиці 1.1, можна зробити обґрунтований висновок щодо вибору оптимальної технологічної бази для проектування сучасної системи обліку складських запасів.

Таблиця 1.1 - Порівняльний аналіз технологій автоматичної ідентифікації

Критерій порівняння	Штрих-кодування (1D)	QR-коди (2D)	Магнітні смуги	NFC (HF RFID)	Складський RFID (UHF)
Сфери використання	Роздрібна торгівля (каси супермаркетів)	Маркування ліків, рекламна продукція	Банківські картки, пропуски (СКУД)	Мобільні платежі, електронні квитки	Логістика, складські комплекси, автоматизація
Необхідність прямої видимості	Обов'язкова (оптичний контакт)	Обов'язкова	Немає (але необхідний щільний контакт зі слотом)	Немає (наближення впритул)	Не потрібна (проникнення радіохвиль)
Дистанція зчитування	Контактна або до 0.5 - 1 м	До 0.5 - 1 м	Відсутня (лише фізичний контакт)	До 3-10 см	Від 1 до 15 метрів (для пасивних міток)
Однотимчасне зчитування (Антиколізія)	Ні (виключно поштучне сканування)	Ні (в основному поштучне)	Ні	Можливо, але дуже складно	Так (від 200 до 800+ міток за секунду)
Обсяг пам'яті носія	Малий (до 20-30 цифр)	Середній (до 2-3 Кб)	Малий (кілька десятків байт)	Середній (від 44 б до кількох Кб)	Середній (до 8 Кб, унікальний код EPC)
Можливість динамічного перезапису	Відсутня (лише друк нової етикетки)	Відсутня	Так	Так	Так (десятьками тисяч циклів)
Стійкість до агресивного середовища	Низька (боїться бруду, води, тертя)	Середня (завдяки масивам корекції помилок)	Низька (розмагнічування магнітом, подряпини)	Висока (мітка в пластиковому корпусі)	Висока (мітка може бути захищена всередину тари)

Використання традиційних оптичних методів (штрих-кодування або QR-кодів), хоча й має перевагу у вигляді наднизької вартості етикетки, робить процес масової інвентаризації чи відвантаження товарів критично залежним від оператора. На розпакування кожної палети, пошук етикетки на кожній коробці та безпосереднє сканування витрачаються десятки годин робочого часу, що створює

так зване "вузьке місце" у логістичному ланцюзі підприємства. Аналогічно, технології магнітних смуг та NFC, маючи обмежену або взагалі нульову дистанцію зчитування, фізично не можуть забезпечити швидкого моніторингу крупногабаритних складських вантажів.

Таким чином, розробка програмно-апаратного комплексу на базі технології ультрависокочастотної радіо ідентифікації (UHF RFID) є найбільш доцільним і технічно правильним рішенням, про що свідчать наступні аргументи:

По-перше, RFID повністю усуває проблему прямої видимості. Радіохвилі легко проникають крізь картон, пластик, деревину та нещільні технічні матеріали. Це дозволяє проводити інвентаризацію запакованих палет або знаходити потрібну коробку у глибині висотного стелажа без необхідності розпаковування та переміщення інших об'єктів.

По-друге, радіочастотне обладнання підтримує процеси масової реєстрації товарів (алгоритми анти колізії). Сучасні UHF зчитувачі здатні безпомилково опитувати до 500-800 міток за одну секунду. Завдяки цьому повністю автоматизується зона приймання та видачі: навантажувач з маркованим вантажем просто проїжджає через обладнані порталом складські ворота, а система в режимі реального часу фіксує переміщення всієї сотні позицій без зупинки техніки.

По-третє, використовуючи унікальний серійний номер електронного коду продукту (EPC) для кожної конкретної мітки, ПАК гарантує уникнення пересортиці та дозволяє відстежувати життєвий цикл кожного окремого екземпляра виробу на складі, замість відслідковування загальної кількості артикулів, як у випадку з дешевим штрих-кодуванням.

По-четверте, RFID-мітки відрізняються значно вищою стійкістю до складських промислових умов (пилу, вологості, перепадів температур), а також підтримують функціонал багаторазового перезапису. Це надає можливість використовувати жорсткі довговічні мітки на поворотній пластиковій тарі підприємства (контейнерах, піддонах) або оперативно дописувати нові атрибути у пам'ять товару.

					БР.КІ-40.00.00.000 ПЗ	16
ЗМ.	Арк.	№ докум.	Підп.	Дата		

З економічної точки зору стартові інвестиції в архітектуру RFID (закупівля зчитувачів, антен та інтеграція ПЗ) залишатимуться вищими порівняно зі звичайними сканерами штрих-кодів. Водночас розробка власного ПАК (замість купівлі дорогих корпоративних "коробкових" рішень) забезпечує високий показник ROI (повернення інвестицій) у перспективі 1-2 років експлуатації. Економічна вигода формуватиметься за рахунок кардинальної швидкодії та скорочення часу інвентаризації, суттєвого зниження витрат на оплату праці працівників під час переобліку, ліквідації втрат від "загублених" товарів та збільшення загальної пропускної здатності логістичного терміналу. Окремою важливою перевагою власної розробки є повна відсутність витрат на придбання та щорічне оновлення дорогих ліцензій на пропріє тарне програмне забезпечення складських систем, що робить кінцеву вартість впровадження комплексу значно нижчою за існуючі на ринку аналоги.

Для наочності наведемо порівняння проєктованого програмно-апаратного комплексу з типовими існуючими рішеннями на ринку (таблиця 1.2).

Таблиця 1.2 - Порівняння розроблюваної системи з існуючими аналогами

Критерій порівняння	Традиційні системи обліку (Штрих-кодуювання)	Комерційні RFID-системи (Готові рішення)	Розроблюваний ПАК (Власна RFID-система)
Базова технологія ідентифікації	Оптична (1D/2D коди)	Радіочастотна (UHF RFID)	Радіочастотна (UHF RFID)
Вартість апаратного забезпечення	Низька	Висока	Висока
Витрати на програмне забезпечення	Середні / Низькі	Високі (платні ліцензії, підписки)	Відсутні (власна розробка)
Швидкість проведення інвентаризації	Низька (поштучне зчитування)	Висока (масове зчитування)	Висока (масове зчитування)

Залежність від людського фактору	Висока (обов'язкова пряма видимість)	Низька	Низька
Можливість адаптації під специфіку підприємства	Низька	Обмежена (комерційна таємниця вендора)	Максимальна (доступ до вихідного коду)
Загальна вартість володіння (ТСО)	Висока (через великі витрати на персонал)	Дуже висока (через дороге ПЗ та підтримку)	Оптимальна (разові витрати на обладнання)

1.3 Огляд існуючих програмно-апаратних засобів та обґрунтування вибору елементної бази комплексу

Проектування будь-якого програмно-апаратного комплексу (ПАК) на базі технології радіочастотної ідентифікації починається з ґрунтового аналізу та підбору елементної бази. Класична архітектура RFID-системи складається з трьох ключових апаратних компонентів:

- радіочастотних міток (транспондерів), які кріпляться на об'єкти обліку;
- антен, які випромінюють електромагнітне поле та приймають відбитий сигнал;
- зчитувачів (рідерів), які генерують високочастотний сигнал, обробляють отримані від міток дані та передають їх на центральний обчислювальний пристрій (мікрокомп'ютер або сервер) для подальшої програмної обробки.

Між собою системи RFID класифікуються насамперед за робочим частотним діапазоном та типом живлення міток. За типом живлення мітки бувають пасивними та активними. Активні мітки обладнані власною батареєю живлення. Вони здатні передавати сигнал на відстані до 100 метрів і часто оснащуються додатковими датчиками (температури, вологості, удару). Проте їх висока вартість (від \$10 до \$50 за штуку) та обмежений термін служби роблять їх

використання недоцільним для масового маркування звичайних складських товарів; вони застосовуються переважно для відстеження великогабаритних морських контейнерів або транспортних засобів. Пасивні мітки не мають власного джерела живлення. Вони активуються виключно енергією електромагнітного поля, яке випромінює антена зчитувача. Вони є надзвичайно дешевими у виробництві, можуть виготовлятися у вигляді гнучких етикеток і мають практично необмежений термін служби, що робить їх ідеальним вибором для масової логістики.

За робочою частотою пасивні системи поділяються на три основні класи:

1. Низькочастотні системи (LF, 125-134 кГц). Мають малу дальність зчитування (до 10 см) і низьку швидкість передачі даних. Переважно використовуються в системах контролю доступу (брелки, смарт-картки) або для чипування свійських тварин, оскільки радіохвилі цієї частоти добре проникають крізь рідини та біологічні тканини. Головний мінус - вони не підтримують алгоритми антиколізії (одночасного зчитування багатьох міток).

2. Високочастотні системи (HF, 13.56 МГц). Забезпечують зчитування на відстані до 1 метра і кращу швидкість передачі даних. Використовуються у бібліотечних системах обліку та для платіжних систем.

3. Ультрависокочастотні системи (UHF, 860-960 МГц). Це глобальний стандарт для складської логістики (EPC Class 1 Gen 2). UHF системи забезпечують максимальну дальність зчитування серед пасивних міток (від 1 до 15 метрів в залежності від потужності антени) та мають неперевершену швидкість передачі даних, дозволяючи зчитувати сотні об'єктів одночасно за секунду.

Таблиця 1.2 - Порівняльна характеристика типів RFID-систем

Характеристика	LF (125-134 кГц)	HF (13.56 МГц)	Активні RFID (2.4 ГГц)	UHF (860-960 МГц)
Тип живлення міток	Пасивне	Пасивне	Вбудована батарея	Пасивне
Дальність зчитування	До 10 см	Від 10 см до 1 м	До 100 метрів і більше	Від 1 до 15 метрів

Швидкість читання	Низька	Середня	Висока	Дуже висока
Однотимчасне зчитування	Немає (лише поштучно)	До ~50 міток / сек	До кількох тисяч	Від 500 до 800+ міток / сек
Кріплення на метал/воду	Відмінно працює	Відносно стабільно	Сильні перешкоди	Сильно глушиться або відбивається
Співвідношення ціна/якість	Низьке для логістики	Середнє	Дуже низьке (дорогі мітки)	Високе (найдешевший масовий маркер)

Спираючись на результати проведеного аналізу характеристик радіочастотних систем (табл. 1.2), для реалізації нашого програмно-апаратного комплексу обліку складських запасів безальтернативно обирається технологія UHF RFID. Структурна схема взаємодії обраних апаратних та програмних компонентів системи зображена на рисунку 1.3.

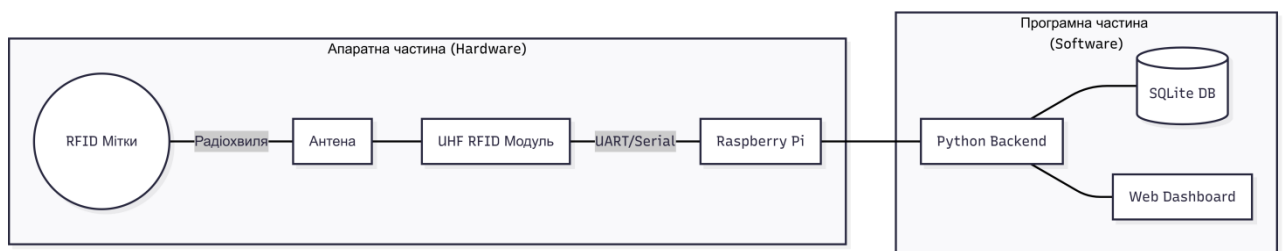


Рисунок 1.3 - Структурна схема обраної елементної бази комплексу

Наступним критичним етапом є вибір обчислювального ядра нашого ПАК. Оскільки комплекс повинен виконувати функцію локального терміналу збору даних, який здатний самостійно фільтрувати потоки "сирих" зчитувань від антени, підключатися до мережі складу та мати власну локальну базу даних для забезпечення безперебійної роботи, використання простих мікроконтролерів (на зразок Arduino) є недостатнім через брак продуктивності. Оптимальним вибором для обчислювального блоку є мікрокомп'ютер Raspberry Pi (наприклад, модель 3B+ або 4B). Цей пристрій поєднує в собі низьку вартість, компактні габарити та потужність повноцінного комп'ютера. Наявність вбудованих інтерфейсів GPIO та UART дозволяє ефективно інтегрувати RFID-зчитувачі на апаратному рівні.

Для зчитування UHF міток мікрокомп'ютер потрібно оснастити відповідним радіочастотним трансивером. У нашій розробці як зчитувач буде застосовано сумісний UHF RFID модуль на базі чіпу ThingMagic M6E Nano (або його апаратний аналог). Цей вибір зумовлений його офіційною підтримкою стандарту EPCglobal Gen2, роботою через стабільний інтерфейс UART, можливістю програмного управління потужністю випромінювання та наявністю U.FL роз'єму для підключення професійних зовнішніх антен, що дозволить налаштовувати дальність сканування від кількох сантиметрів до десятків метрів.

Ефективність апаратної частини безпосередньо залежить від обраного програмного стеку:

- керуючою серверною операційною системою для розробки обрано Raspberry Pi OS (на базі ядра Debian Linux). Її надійність, низьке споживання ресурсів (headless-режим) та можливість автоматизації фонових сервісів ідеально підходять для складських умов 24/7.

- мовою програмування для написання бекенду та взаємодії з апаратурою обрано Python 3. Вибір мови базується на її простоті прототипування складної бізнес-логіки та наявності потужних бібліотек: pyserial для стабільного асинхронного зв'язку зі зчитувачем через UART та вбудованої підтримки легковагових баз даних (SQLite) для оперативного зберігання облікових даних.

Отже, обрана елементна база майбутнього комплексу (UHF RFID + Raspberry Pi + Linux/Python) формує потужну, легко масштабовану та економічно виправдану платформу для впровадження ефективної системи автоматизації складу.

1.4 Постановка задачі на розробку комплексу

На основі проведеного аналізу існуючих бізнес-процесів складського обліку, розроблених інформаційних вимог до системи масової реєстрації товарів, а також обґрунтованого вибору елементної бази у складі технології UHF RFID, мікрокомп'ютера Raspberry Pi та програмного стеку Python/Linux, можна

					БР.КІ-40.00.00.000 ПЗ	21
ЗМ.	Арк.	№ докум.	Підп.	Дата		

сформулювати кінцеву постановку задачі на проектування дипломної роботи. Головним завданням є практична розробка та впровадження функціонального прототипу програмно-апаратного комплексу (ПАК) автоматизованого обліку складських запасів.

Для досягнення поставленої мети проєктований комплекс повинен гарантовано задовольняти ряд конкретних технічних та функціональних вимог. Ієрархічна структура цих вимог наведена на рисунку 1.4.

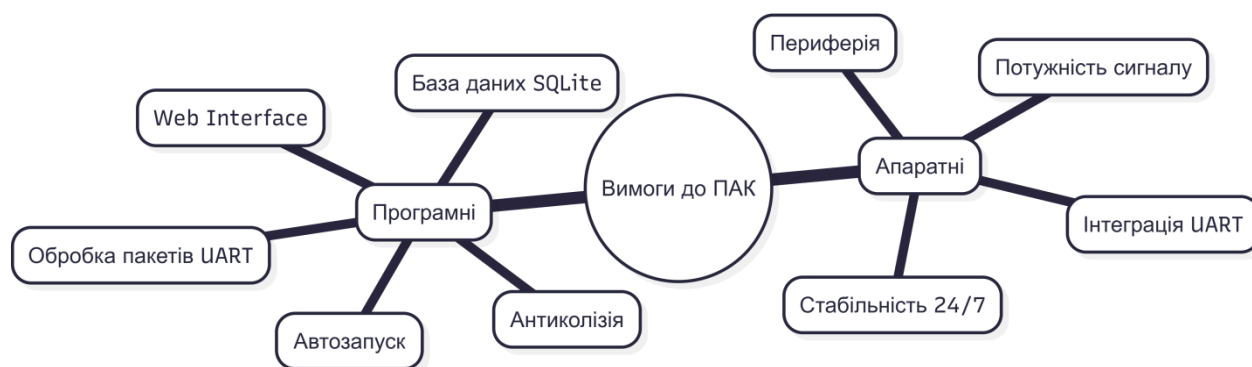


Рисунок 1.4 - Дерево вимог до програмно-апаратного комплексу

Задля зручності розробки та критеріального оцінювання майбутнього прототипу ці вимоги розподілено на дві основні групи.

Апаратні вимоги до комплексу:

- Інтеграція та фізична сумісність - забезпечення коректного підключення UHF RFID-модуля зчитувача до мікрокомп'ютера Raspberry Pi з використанням цифрового інтерфейсу зв'язку UART (або USB у режимі віртуального COM-порту) із гарантованим дозволом доступу до серійного порту для користувача ОС Linux.

- Радіочастотний тракт - можливість управління вихідною потужністю випромінювання (RF Power) зчитувача на апаратному або низькорівневому програмному рівні для гнучкого регулювання дистанції ідентифікації. Дистанція повинна гнучко налаштовуватися (від кількох сантиметрів для локального настільного тестування до кількох метрів для імітації складських порталів).

– Стабільність роботи - апаратна частина повинна функціонувати в режимі безперервної експлуатації (24/7). Плата мікрокомп'ютера має бути захищена від перегріву, а живлення зчитувача має бути незалежним або забезпечувати стабільну силу струму, достатню для стабільної генерації електромагнітного поля без просадок напруги на лінії.

– Масштабованість периферії - можливість подальшого легкого підключення додаткових апаратних індикаторів (наприклад, зумера для звукової індикації успішного зчитування, світлодіодів для сповіщення або датчиків руху).

Програмні та алгоритмічні вимоги до комплексу:

– Апробація та обробка сигналів (Hardware-Software Bridging) - розробка надійного Python-сервісу, який забезпечить безперебійне зчитування байтового потоку даних з порту UART. Програмний модуль має парсити отримані пакети, виділяти з них універсальні електронні коди продуктів (EPC-коди міток), відкидаючи пошкоджені кадри.

– Алгоритми антиколізії та програмна фільтрація - через фізичну специфіку RFID, коли одна мітка зчитується трансивером десятки разів за мілісекунди, програмний код ПАК повинен реалізовувати логіку програмного "дебаунсу" (debouncing). Необхідно створити алгоритм кешування подій, який реєструватиме подію "нового виявлення об'єкта" лімітовану кількість разів, щоб звільнити системні ресурси і не перевантажувати базу даних дублікатами.

– Логіка обліку та взаємодія з базою даних - розробка та ініціалізація локальної реляційної БД (SQLite), яка зберігатиме: довідник товарної номенклатури, реєстр унікальних EPC-міток та хронологічний журнал складських подій (лог інвентаризації). Програмне забезпечення має в режимі реального часу ідентифікувати зчитаний EPC-код у базі та фіксувати факт його наявності поточним часом.

– Клієнтський інтерфейс (UI) - реалізація сучасного локального веб-інтерфейсу користувача для моніторингу залишків. Інтерфейс повинен надавати оператору зручні дашборди для перегляду поточних запасів, відображати історію відсканованих подій "на живо" (без постійного перезавантаження сторінки) та візуально сигналізувати про відскановані невідомі мітки.

– Автоматична сервісна диспетчеризація - програмний код має бути запакований і налаштований у вигляді системної демонічної служби (systemd-service), що гарантуватиме його автоматичний запуск у фоновому режимі відразу після подачі живлення на Raspberry Pi, без необхідності ручного підключення клавіатури і монітора.

Проведений у першому розділі ґрунтовний аналіз процесу автоматизації складського обліку дозволив чітко окреслити проблематику сучасної логістики. Доведено, що класичні ручні підходи та поштучне оптичне штрих-кодування не спроможні забезпечити необхідну швидкість та точність моніторингу товарно-матеріальних цінностей на складах, особливо при обробці закритих палет чи високоярусних стелажів.

Дослідження усіх наявних на ринку методів автоматичної ідентифікації показало, що найбільш доцільною та технологічно виправданою архітектурою є технологія ультрависокочастотної інвентаризації (UHF RFID). Її ключові функціональні переваги - можливість зчитування сотень об'єктів одночасно та відсутність потреби у прямому візуальному контакті сканера з маркером - вирішують головні недоліки класичних систем.

Сформований апаратний стек на базі мікрокомп'ютера Raspberry Pi та сумісного UHF-зчитувача є ідеальним з економічної і технічної точки зору рішенням для побудови бюджетного, але повнофункціонального локального терміналу для збору даних. Визначені в даному підрозділі конкретні апаратні, програмні та алгоритмічні вимоги повністю завершують і формалізують постановку задачі дипломного проекту.

Реалізація цих вимог виступатиме планом дій для наступних етапів роботи. Відповідно, другий та третій розділи дипломної роботи будуть присвячені безпосередньому проектуванню моделей даних, алгоритмів взаємодії компонентів, створення користувацького інтерфейсу та технічному тестуванню робочого прототипу нашого програмно-апаратного комплексу.

2 ПРОЕКТУВАННЯ СТРУКТУРИ ТА МОДЕЛЕЙ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ

2.1 Розробка загальної архітектури RFID-комплексу

Проектування архітектури програмно-апаратного комплексу (ПАК) автоматизованого обліку складських запасів базується на принципах модульності, масштабованості та надійності передачі даних у реальному часі. Оскільки система повинна забезпечувати безперебійний моніторинг об'єктів у промислових умовах складу, обрано багаторівневу архітектуру, яка дозволяє чітко розграничити функції збору "сирих" даних від апаратури, їх первинної обробки, довгострокового зберігання та візуалізації для кінцевого користувача.

Загальна структура проектованого комплексу базується на чотирирівневій моделі взаємодії, що представлена на рисунку 2.1.

1. Рівень апаратної ідентифікації (Physical Layer). Включає пасивні UHF RFID мітки, антени та UHF-зчитувач. Його головна задача - фізичне виявлення об'єкта за допомогою радіочастотного сигналу та зчитування унікального коду продукту (EPC).

2. Рівень транспортного спряження та периферійних обчислень (Edge Layer). Представлений мікрокомп'ютером Raspberry Pi та низькорівневим програмним модулем на мові Python. Цей рівень відповідає за апаратне підключення зчитувача, управління потужністю сигналу та первинну фільтрацію дублікатів.

3. Рівень бази даних та логіки (Data & Logic Layer). Відповідає за мапінг зчитаних EPC-кодів до конкретних номенклатурних одиниць у реляційній базі даних SQLite, ведення журналів операцій (Logs) та забезпечення цілісності інформації про складські залишки.

4. Рівень представлення та моніторингу (Presentation Layer). Реалізований у вигляді веб-інтерфейсу (дашборду), що дозволяє оператору складу спостерігати

					БР.КІ-40.00.00.000 ПЗ	25
ЗМ.	Арк.	№ докум.	Підп.	Дата		

за процесами інвентаризації, прийому та видачі товарів у режимі реального часу.

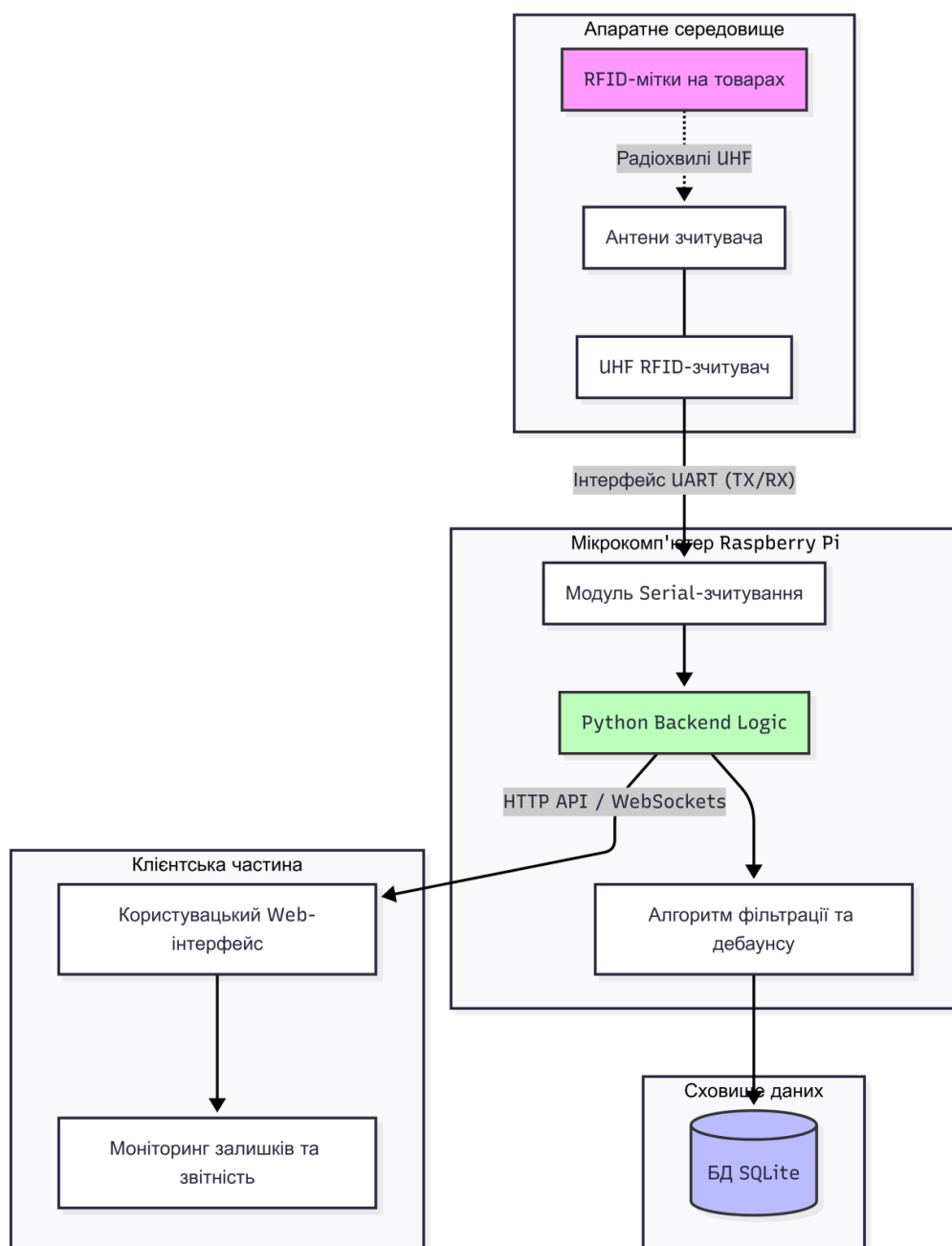


Рисунок 2.1 - Загальна структурна схема архітектури RFID-комплексу

Ключовим аспектом архітектури є фізичне та логічне поєднання RFID-зчитувача з центральним обчислювальним модулем. Для забезпечення високої швидкості обміну даними та мінімізації накладних витрат ОС, обрано послідовний інтерфейс UART (Universal Asynchronous Receiver-Transmitter).

Фізичне підключення UHF RFID-модуля (наприклад, на базі чіпу M6E Nano) до Raspberry Pi здійснюється через гребінку контактів GPIO (General Purpose Input/Output). Схема підключення передбачає використання наступних ліній:

- UART_TX (Pin 8 / GPIO 14) на Raspberry Pi з'єднується з лінією RX зчитувача - для передачі команд конфігурації (встановлення потужності, вибір антени, старт опитування).
- UART_RX (Pin 10 / GPIO 15) на Raspberry Pi з'єднується з лінією TX зчитувача - для отримання відгуків від міток у вигляді байтового потоку даних.
- VCC (5V/3.3V) та GND - для забезпечення живлення апаратної частини.

Важливою особливістю архітектури є необхідність узгодження логічних рівнів. Більшість промислових UHF-модулів працюють на рівні 3.3V, що є повністю сумісним із Raspberry Pi. Для надійного зчитування у складських умовах, де можливі електромагнітні завади від електродвигунів навантажувачів, програмно встановлюється швидкість обміну даними (Baud rate) на рівні 115200 біт/с, що забезпечує оптимальний баланс між пропускнуою здатністю та завадостійкістю.

Програмна архітектура комплексу побудована за подійно-орієнтованою моделлю. В основі лежить фоновий сервіс на Python (Backend Service), який функціонує як міст між апаратним рівнем та базою даних. Логічний цикл роботи системи можна представити наступними етапами:

1. Ініціалізація та конфігурація. При запуску системи Python-скрипт відкриває сесію через бібліотеку pyserial, подає команду самодіагностики зчитувачу та встановлює параметри антени (режим антиколізії, чутливість).

2. Слухання порту (Polling/Interrupts). Бекенд переходить у режим безперервного прослуховування буфера UART. Як тільки мітка потрапляє в зону дії антени, зчитувач формує пакет даних, що містить: EPC-номер мітки, RSSI (рівень сигналу) та кількість зчитувань (count).

3. Алгоритм програмної фільтрації. Оскільки UHF-антена може зчитувати одну й ту саму мітку сотні разів на секунду, архітектурою передбачено модуль

					БР.КІ-40.00.00.000 ПЗ	27
ЗМ.	Арк.	№ докум.	Підп.	Дата		

"дебаунсу". Він використовує кеш у оперативній пам'яті для відсікання дублікатів. Подія запису в базу даних ініціюється лише при першій появі мітки або при її зникненні з поля зору на певний час.

4. Взаємодія з БД та Веб-сервером. Оброблені дані передаються у SQLite через SQL-запити. Одночасно з цим, для забезпечення інтерактивності, бекенд через протокол HTTP або WebSockets сповіщає веб-інтерфейс про нове сканування, що дозволяє оператору миттєво бачити прибуття товару на екрані монітора.

Така архітектурна побудова дозволяє системі бути автономною: навіть при тимчасовій відсутності зв'язку з глобальною мережею, Raspberry Pi продовжує збирати дані у локальну БД SQLite, що критично для надійності складського обліку.

При проектуванні апаратного спряження розглядався вибір між трьома основними інтерфейсами: USB, SPI та UART. Використання USB спрощує підключення, проте вносить значні затримки через стеки протоколів ОС Linux та вимагає використання зовнішніх перехідників "USB-to-Serial", що знижує загальну вібростійкість системи у складських умовах. Інтерфейс SPI забезпечує вищу швидкість, але критично обмежений довжиною кабелю (до 20-30 см), що унеможливорює винесення антени чи зчитувача за межі компактного корпусу.

Обґрунтування вибору UART базується на наступних факторах:

1. Енергоефективність. Передача даних здійснюється по двох лініях (TX/RX), що мінімізує споживання енергії модулем Raspberry Pi.

2. Програмна підтримка. Бібліотека pyserial дозволяє реалізувати асинхронне зчитування байтового потоку у фоновому потоці (Threading), не блокуючи роботу основного веб-інтерфейсу.

3. Надійність. UART є стандартом де-факто для промислових RFID-модулів (M6E, R2000), що гарантує сумісність з великим спектром обладнання без написання складних драйверів.

Важливою частиною розроблюваної архітектури є забезпечення цілісності та безпеки даних. На рівні Edge-сервісу реалізовано механізм перевірки

					БР.КІ-40.00.00.000 ПЗ	28
ЗМ.	Арк.	№ докум.	Підп.	Дата		

контрольних сум (Checksum) для кожного отриманого від зчитувача пакета. Це запобігає реєстрації "хибних" міток, що можуть виникнути внаслідок електромагнітного шуму.

Для забезпечення масштабованості, архітектура передбачає можливість підключення зовнішнього антенного мультиплексора. Це дозволяє одному мікрокомп'ютеру обслуговувати до 4 або 8 незалежних антенних зон (наприклад, різні ворота складу), перемикаючи їх за допомогою GPIO-сигналів. Дана логіка закладена у структуру бекенду, де кожна подія зчитування маркується ідентифікатором антени (antenna_id), що дозволяє точно локалізувати рух товару.

З точки зору інформаційної безпеки, доступ до Веб-інтерфейсу моніторингу обмежений ролевим доступом (Role-Based Access Control), а дані між Raspberry Pi та зовнішнім сервером передаються через зашифрований протокол HTTPS. Це унеможливорює підміну інформації про залишки товарів зловмисниками або конкурентами.

Крім того, модульність дозволяє у майбутньому легко замінити SQLite на зовнішню серверну базу даних (наприклад, PostgreSQL) або інтегрувати ПАК з існуючими ERP-системами підприємства (наприклад, 1С або SAP) через розроблене REST API. Це робить систему універсальним інструментом автоматизації, готовим до промислового впровадження.

2.2 Алгоритмізація виробничих процесів на складі

Алгоритмізація виробничих процесів на складі є критично важливим етапом проектування програмно-апаратного комплексу, оскільки вона дозволяє формалізувати логіку функціонування системи, визначити чітку послідовність автоматизованих операцій та розробити керуючі алгоритми взаємодії технічних засобів RFID-ідентифікації із користувачами. У даному підрозділі за допомогою уніфікованої мови моделювання (UML) розроблено алгоритмічні та концептуальні моделі основних складських процесів, оптимізованих та трансформованих під впливом впровадження радіочастотних технологій.

					БР.КІ-40.00.00.000 ПЗ	29
ЗМ.	Арк.	№ докум.	Підп.	Дата		

Використання RFID-технологій докорінно змінює традиційний підхід до складського обліку, перетворюючи його на систему автоматизованого управління матеріальними потоками в реальному часі. Ключова відмінність полягає в переведенні ручних процесів ідентифікації в площину повністю алгоритмізованого групового зчитування без необхідності прямої видимості об'єктів. Це вимагає розробки нових алгоритмів для автоматизації прийому, відвантаження та інвентаризації товарно-матеріальних цінностей (ТМЦ).

Розробка діаграм прецедентів (Use case) та послідовностей (sequence) для операцій прийому, видачі та інвентаризації товарів.

Далі наведемо опис логіки ключових складських операцій:

1. Алгоритм прийому товарів.

В архітектурі нашого ПАК цей процес автоматизується за допомогою "портального" зчитування. Вантаж (наприклад, палета з маркованими коробками) проходить через зону дії UHF-антен. Система в режимі реального часу реєструє всі EPC-коди, що потрапили у радіус ідентифікації. Логіка ПАК передбачає автоматичне порівняння отриманого списку міток із електронною накладною постачальника. У разі виявлення розбіжностей (нестачі або надлишку) система миттєво сигналізує оператору. Така модель виключає помилки, спричинені "людським фактором" при ручному перерахунку.

2. Алгоритм видачі товарів.

Операція видачі базується на валідації відповідності відвантажених одиниць замовленню клієнта. При переміщенні товарів із зони зберігання до зони відвантаження, RFID-зчитувач фіксує ідентифікатори об'єктів. Програмний модуль звіряє ці дані зі статусом "До відвантаження" у базі даних. Якщо серійний номер мітки не відповідає поточному замовленню, система блокує операцію, що запобігає пересортиці та несанкціонованому вивезенню ТМЦ.

3. Алгоритм інвентаризації.

Це найбільш трудомістка операція, яка при використанні RFID зазнає найбільшої алгоритмізації. Замість поштучного пошуку штрих-кодів, комірник активує режим групового сканування. ПАК здійснює циклічне опитування всіх

					БР.КІ-40.00.00.000 ПЗ	30
ЗМ.	Арк.	№ докум.	Підп.	Дата		

міток у зоні видимості антен, агрегує дані та автоматично порівнює фактичну наявність із обліковими залишками в SQLite. Результатом є миттєво згенерований акт розбіжностей..

Діаграма прецедентів описує функціональні можливості системи та алгоритмічний розподіл ролей між зовнішніми акторами. Для нашого комплексу виділено два основних типи користувачів:

- комірник (storekeeper) - основний оператор системи, що виконує фізичні операції з ТМЦ, його взаємодія зосереджена на виконанні прийому, відвантаження та інвентаризації;

- адміністратор (administrator) - особа, що відповідає за налаштування системи, ведення довідників товарів, управління обліковими записами та формування підсумкової аналітичної звітності.

Також у моделі опосередковано присутній UHF RFID-зчитувач, який виступає як технічне джерело подій для системи.

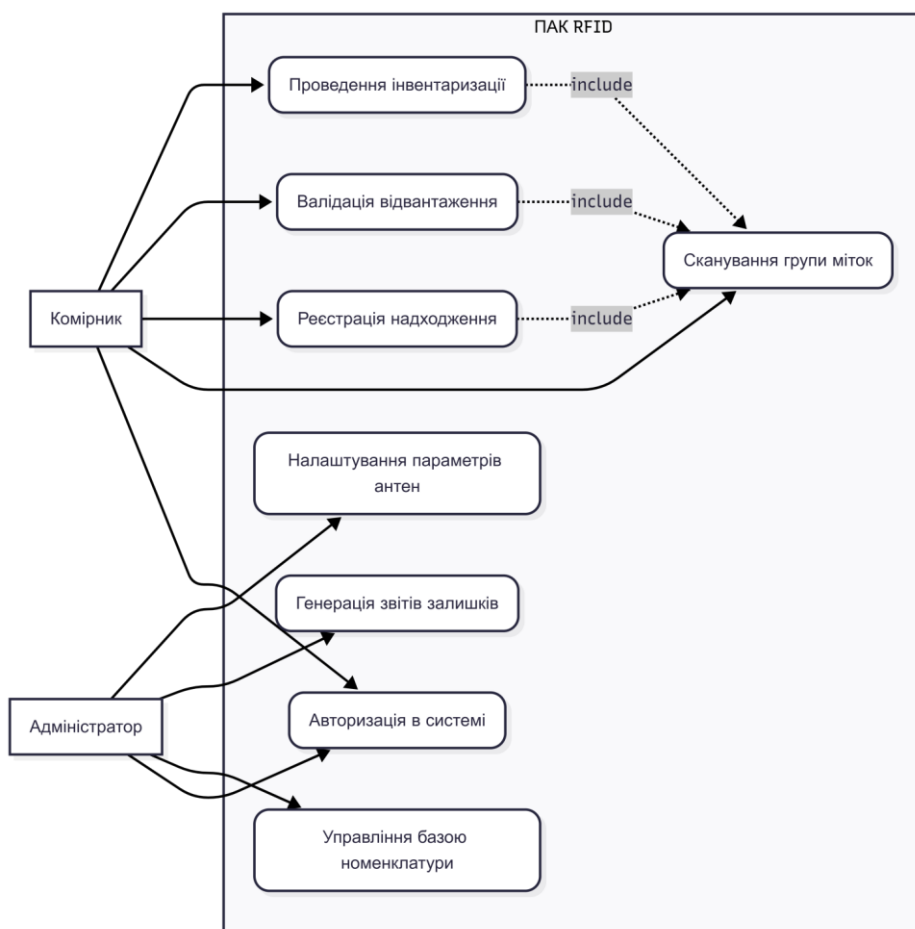


Рисунок 2.2 - Діаграма прецедентів системи RFID-обліку

Опис варіантів використання:

- авторизація - вхід користувача в систему за допомогою облікових даних для розмежування прав доступу;
- сканування групи міток - базовий прецедент, що забезпечує низькорівневу взаємодію з RFID-модулем через UART;
- реєстрація надходження - автоматичне додавання нових товарів у базу даних на основі зчитаних EPC-кодів;
- управління базою номенклатури - редагування назв товарів, характеристик та прив'язка EPC-кодів до артикулів;
- налаштування параметрів антен - регулювання потужності випромінювання (RF Power) залежно від площі складу..

Діаграми послідовностей деталізують часову взаємодію між об'єктами системи під час виконання конкретних бізнес-сценаріїв.

Сценарій "Надходження товару на склад"

На даній діаграмі (Рис. 2.3) показано шлях даних від фізичної мітки до відображення в інтерфейсі користувача.

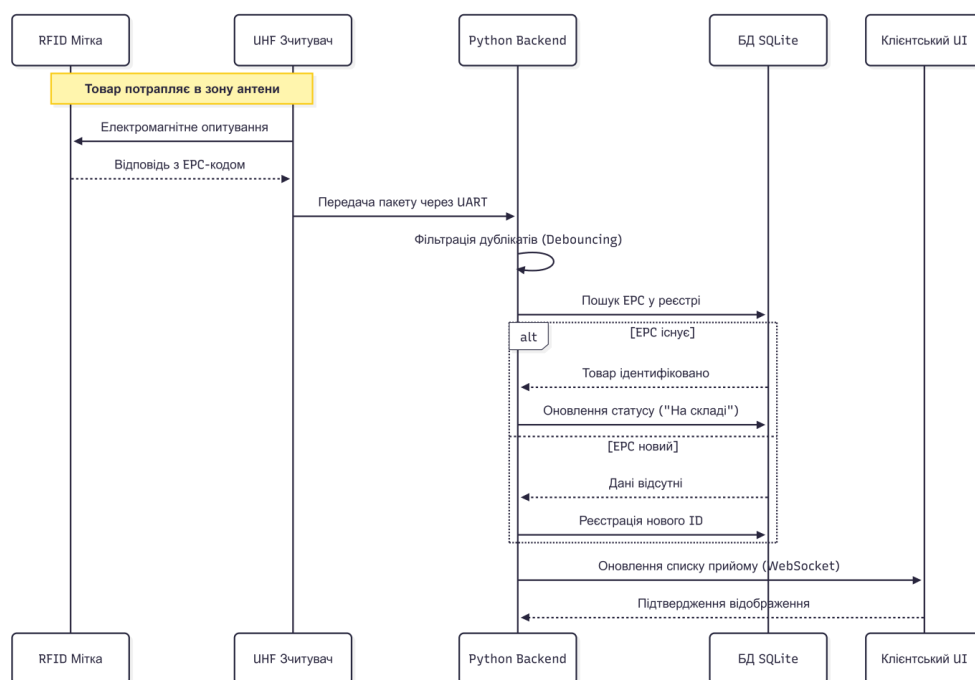


Рисунок 2.3 - Діаграма послідовності процесу надходження товару

Опис алгоритмічних етапів процесу надходження:

- зчитування - мітка активується полем антени і повертає свій унікальний EPC, при цьому зчитувач перетворює радіосигнал у цифровий пакет;
- обробка - Python Backend отримує дані, причому оскільки під час руху палети мітка може бути зчитана багато разів, алгоритм дебаунсу відфільтровує зайві повтори, залишаючи лише одну подію;
- ідентифікація - система звертається до SQLite для пошуку відповідності, оновлюючи статус відомого товару або реєструючи новий EPC;
- візуалізація - користувач бачить оновлений список товарів у веб-інтерфейсі в режимі реального часу..

Сценарій "Інвентаризація залишків"

Процес інвентаризації демонструє взаємодію комірника з системою для звірки фактичної наявності ТМЦ і представлений на рисунку 2.4.

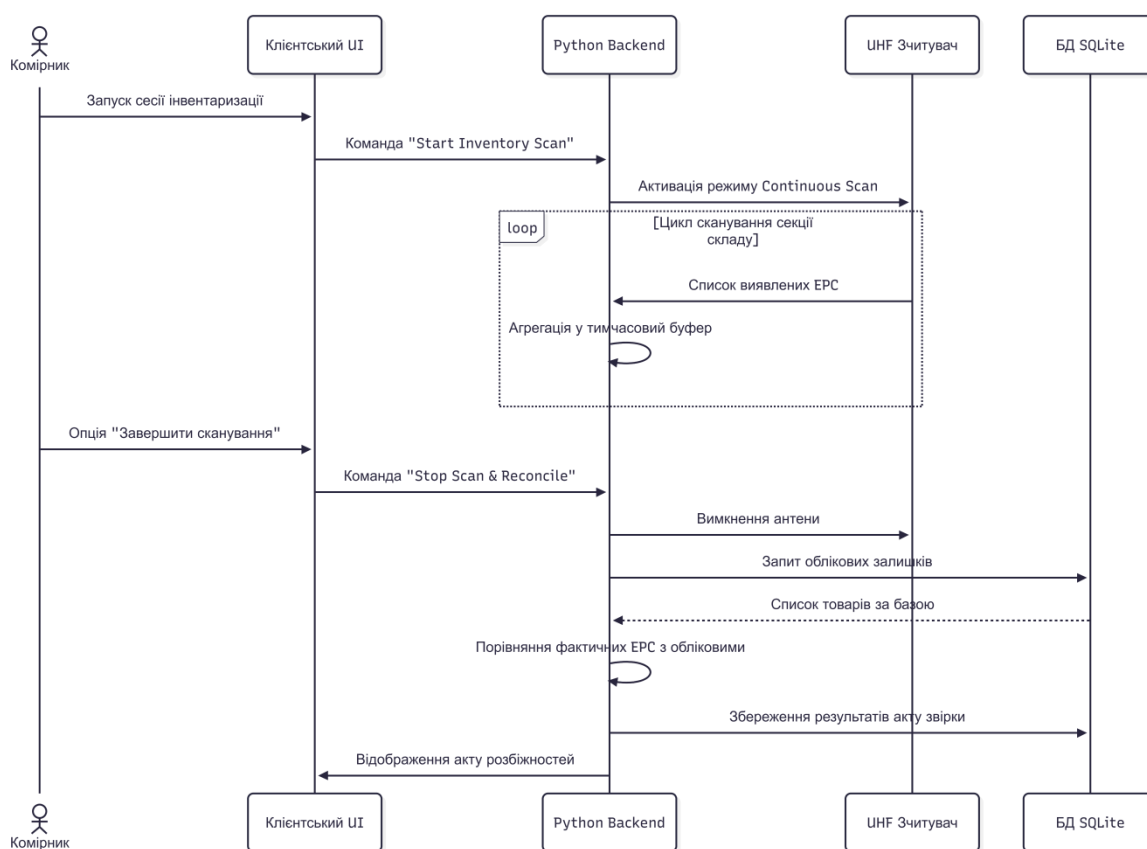


Рисунок 2.4 - Діаграма послідовності процесу інвентаризації

Опис алгоритмічних етапів процесу інвентаризації:

- активація - комірник ініціює процес через інтерфейс, після чого система переводить зчитувач у режим безперервного сканування;
- збір даних - протягом усього часу обходу стелажів Backend накопичує унікальні EPC-коди в оперативній пам'яті;
- звірка - система автоматично витягує з SQLite актуальний перелік товарів і порівнює фактичні EPC з обліковими;
- результат акту звірки записуються в базу даних і відображаються у вигляді кольорового звіту про розбіжності.

Розроблені алгоритми та UML-моделі підтверджують високу ефективність обраного підходу до автоматизації. Алгоритмізація складських операцій дозволяє трансформувати складські процеси, зводячи тривалі ручні перевірки до автоматизованого виконання логічних правил та миттєвого прийняття рішень. Отримана алгоритмічна база та логіка взаємодії компонентів стане основою для подальшої програмної реалізації модулів ПАК у наступних розділах роботи.

2.3 Проєктування структури бази даних

Важливим компонентом будь-якої автоматизованої системи обліку є підсистема зберігання та управління даними. Для програмно-апаратного комплексу на основі RFID-технологій база даних (БД) повинна забезпечувати високу швидкість запису поточкових даних від зчитувача, гарантувати цілісність інформації про ТМЦ та підтримувати можливість автономної роботи на периферійному пристрої (Edge device).

При виборі системи управління базами даних для нашого ПАК розглядалися як клієнт-серверні рішення (PostgreSQL, MySQL), так і вбудовані бази даних. Для реалізації прототипу та роботи в умовах автономного складського терміналу на базі Raspberry Pi було обрано SQLite. Даний вибір обґрунтований наступними технічними та експлуатаційними факторами:

					БР.КІ-40.00.00.000 ПЗ	34
ЗМ.	Арк.	№ докум.	Підп.	Дата		

- мінімальні вимоги до системних ресурсів - SQLite не є окремим процесом-сервером, а функціонує як бібліотека, що інтегрована безпосередньо в додаток на Python, це критично для Raspberry Pi, де ресурси оперативної пам'яті та процесора обмежені;

- вбудованість (Zero-configuration) - БД зберігається у вигляді одного кросплатформеного файлу на диску, що спрощує резервування даних, перенесення системи на інші пристрої та не вимагає складного адміністрування прав доступу до сервера БД;

- підтримка ACID - незважаючи на свою легковажність, SQLite повністю підтримує транзакційність (Atomicity, Consistency, Isolation, Durability), що гарантує цілісність даних про останню інвентаризацію у разі раптового зникнення живлення на складі;

- достатність для локального вузла - обсяги даних одного складського терміналу (тисячі товарів та міток) цілком вкладаються в обмеження продуктивності SQLite, яка здатна ефективно обробляти таблиці обсягом до декількох терабайт.

Структура бази даних розроблена з урахуванням необхідності зв'язку між фізичними об'єктами (RFID-мітками) та їх віртуальним представленням (товарами). Основні сутності системи включають:

- категорії (Categories) - довідник для логічного групування товарів (наприклад, "Електроніка", "Одяг", "Інструменти");

- товари (Products) - головна таблиця з майстер-даними про артикули, що зберігає статичну інформацію (назва, SKU, опис) та поточну загальну кількість;

- RFID-мітки (RFID_Tags) - таблиця-реєстр, що пов'язує унікальний EPC-код мітки з конкретним товаром, де кожен запис відповідає унікальному фізичному екземпляру;

- користувачі (Users) - таблиця для автентифікації працівників складу та розмежування ролей (Адміністратор, Комірник);

- журнал операцій (Logs) - транзакційна таблиця, що фіксує кожну подію зчитування мітки, час події та відповідального користувача.

					БР.КІ-40.00.00.000 ПЗ	35
ЗМ.	Арк.	№ докум.	Підп.	Дата		

Детальні специфікації таблиць наведені у таблицях 2.1 - 2.5.

Таблиця 2.1 - Специфікація таблиці Categories

Поле	Тип даних	Опис	Обмеження
id	INTEGER	Унікальний ідентифікатор категорії	PK, AutoIncrement
name	TEXT	Назва категорії	NOT NULL, UNIQUE
description	TEXT	Текстовий опис	-

Для забезпечення зв'язку категорій з конкретними номенклатурними одиницями розроблено таблицю опису товарів. У ній фіксуються статичні реквізити, такі як унікальний артикул та ціна, а також здійснюється прив'язка до відповідної групи через зовнішній ключ (табл. 2.2).

Таблиця 2.2 - Специфікація таблиці Products

Поле	Тип даних	Опис	Обмеження
id	INTEGER	Унікальний ідентифікатор товару	PK, AutoIncrement
category_id	INTEGER	Зв'язок з категорією	FK (Categories.id)
sku	TEXT	Артикул товару	UNIQUE, NOT NULL
name	TEXT	Найменування	NOT NULL
price	REAL	Роздрібна ціна	DEFAULT 0.0

Оскільки система базується на радіочастотній ідентифікації, кожен фізичний об'єкт на складі повинен мати унікальну прив'язку до номенклатури. Для цього використовується таблиця міток, яка дозволяє відслідковувати поточний статус та час останнього зчитування кожної мітки (табл. 2.3).

Таблиця 2.3 - Специфікація таблиці RFID_Tags

Поле	Тип даних	Опис	Обмеження
epc_id	TEXT (24)	Унікальний код RFID-мітки	PK, UNIQUE
product_id	INTEGER	Ідентифікатор товару	FK (Products.id)
status	TEXT	Стан мітки (Stock, Sold, Missing)	DEFAULT 'Stock'
last_seen	DATETIME	Час останньої реєстрації антеною	-

Важливим елементом безпеки та адміністрування системи є контроль доступу користувачів. Інформація про зареєстрованих працівників складу, їх ролі та захешовані паролі для входу в систему зберігається в окремому довіднику (табл. 2.4).

Таблиця 2.4 - Специфікація таблиці Users

Поле	Тип даних	Опис	Обмеження
id	INTEGER	Ідентифікатор користувача	PK, AutoIncrement
username	TEXT	Логін для входу	UNIQUE, NOT NULL
password_hash	TEXT	Захешований пароль	NOT NULL
role	TEXT	Роль (admin, storekeeper)	NOT NULL

Для забезпечення повної прозорості та аудиту всіх подій ідентифікації в системі реалізовано механізм логування. Будь-яке переміщення товару чи сканування мітки реєструється у транзакційному журналі із зазначенням відповідального виконавця та мітки (табл. 2.5).

Таблиця 2.5 - Специфікація таблиці Logs

Поле	Тип даних	Опис	Обмеження
id	INTEGER	Ідентифікатор запису логу	PK, AutoIncrement
user_id	INTEGER	Хто виконав операцію	FK (Users.id)
epc_id	TEXT	Яка мітка була задіяна	FK (RFID_Tags.epc_id)
action_type	TEXT	Тип операції (In, Out, Scan)	NOT NULL
timestamp	DATETIME	Дата та час події	DEFAULT CURRENT_TIMESTAMP

Для наочного представлення структури збереження інформації, формалізації взаємозв'язків між ключовими компонентами бази даних та

визначення правил цілісності даних розроблено модель «сутність-зв'язок» (ER-діаграму), яка представлена на рисунку 2.5. Візуальне моделювання виконано з використанням загальноприйнятої концептуальної нотації «Crow's Foot» (вилка Мартина), яка дозволяє детально специфікувати не лише спрямованість асоціацій, а й потужність (кардинальність) зв'язків між таблицями реляційної моделі, чітко визначаючи граничні умови відношень (один-до-одного, один-до-багатьох) у процесах обробки RFID-подій.

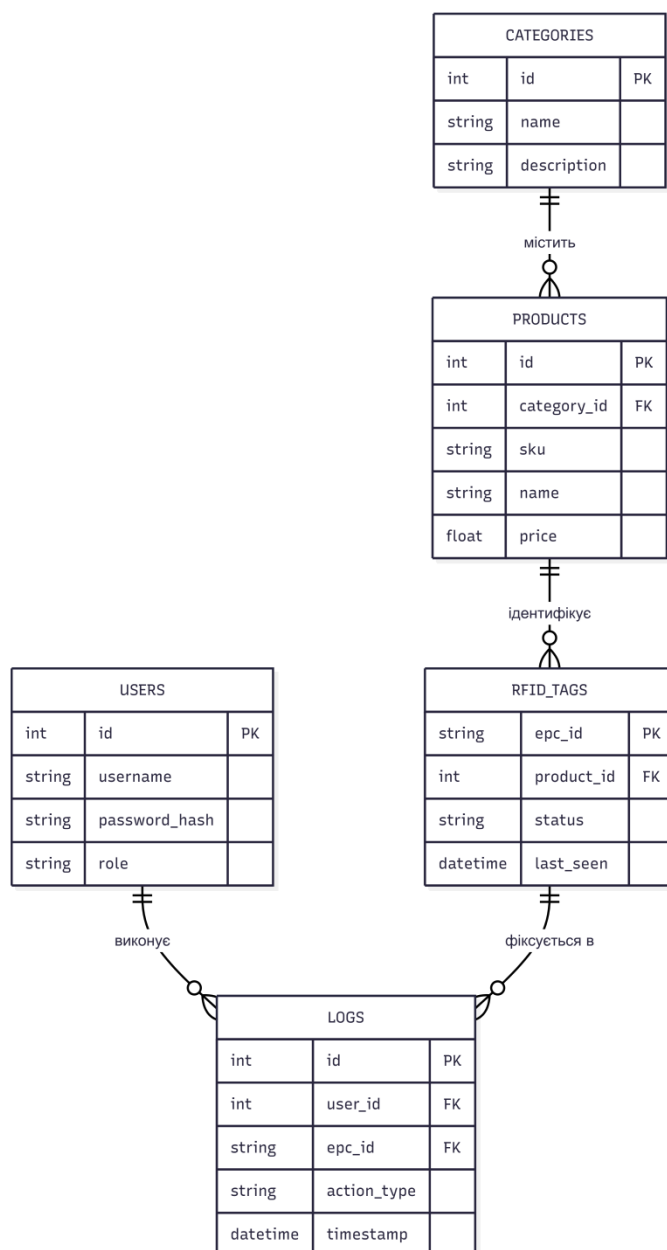


Рисунок 2.5 - ER-діаграма бази даних RFID-комплексу

Описана структура даних є нормалізованою (до 3-ї нормальної форми), що виключає надлишковість та аномалії оновлення. Використання зовнішніх ключів (Foreign Keys) забезпечує цілісність посилань: наприклад, неможливо видалити товар, до якого прив'язані активні RFID-мітки. Дана модель забезпечує необхідну гнучкість для реалізації всіх складських процесів, змодельованих у попередньому підрозділі.

Висновок до розділу

В другому розділі було проведено комплексне проектування структури, моделей та алгоритмів функціонування програмно-апаратного комплексу автоматизованого обліку складських запасів. Основні результати етапу проектування полягають у наступному:

- розробка загальної архітектури системи - побудовано багаторівневу структуру ПАК на основі Raspberry Pi та UART-зв'язку;
- алгоритмізація виробничих процесів на складі - формалізовано логіку ключових операцій та побудовано UML-діаграми прецедентів і послідовностей;
- проектування структури бази даних - розроблено ER-діаграму та схему з 5 нормалізованих таблиць у СУБД SQLite.

Таким чином, сформовано повний набір проектної документації, алгоритмів та моделей, які є необхідним фундаментом для наступного етапу - безпосередньої технічної реалізації, налаштування апаратного забезпечення та написання програмного коду модулів комплексу, що буде розглянуто у третьому розділі.

					БР.КІ-40.00.00.000 ПЗ	39
ЗМ.	Арк.	№ докум.	Підп.	Дата		

3 ТЕХНІЧНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ КОМПЛЕКСУ ОБЛІКУ ЗАПАСІВ

3.1 Реалізація інтерфейсу взаємодії з RFID-зчитувачем

Одним із найбільш складних та відповідальних етапів розробки програмно-апаратного комплексу є реалізація надійного інтерфейсу взаємодії між обчислювальним модулем Raspberry Pi та UHF RFID-зчитувачем. Враховуючи специфіку радіочастотної ідентифікації, програмне забезпечення повинно забезпечувати не лише стабільне з'єднання, але й коректну обробку високошвидкісного потоку сирих даних, що надходять у реальному часі.

Для реалізації цього рівня було обрано мову програмування Python та спеціалізовану бібліотеку pyserial, яка забезпечує низькорівневий доступ до послідовного порту UART. Вибір UART зумовлений тим, що більшість сучасних UHF-модулів (зокрема серії YRM100) використовують цей інтерфейс як основний канал передачі даних завдяки його простоті та надійності в умовах промислових приміщень.

Проектування архітектури взаємодії. Програмна реалізація інтерфейсу базується на принципах об'єктно-орієнтованого проектування. Основний логічний блок винесено в окремий модуль, що виконує роль HAL (Hardware Abstraction Layer). Це дозволяє абстрагувати бекенд-додаток від особливостей конкретної моделі зчитувача.

Для забезпечення високої чутливості системи та запобігання втраті даних було застосовано багатопотокову модель обробки. Оскільки зчитувач може генерувати сотні пакетів на секунду, їх обробка в основному потоці програми призвела б до блокування користувацького інтерфейсу. Тому в розробленому модулі опитування порту винесено в окремий фоновий потік (Background Thread), який працює паралельно з основним циклом FastAPI.

					БР.КІ-40.00.00.000 ПЗ	40
ЗМ.	Арк.	№ докум.	Підп.	Дата		

Процес передачі даних від апаратного рівня до бази даних візуалізовано на схемі (рис. 3.1).

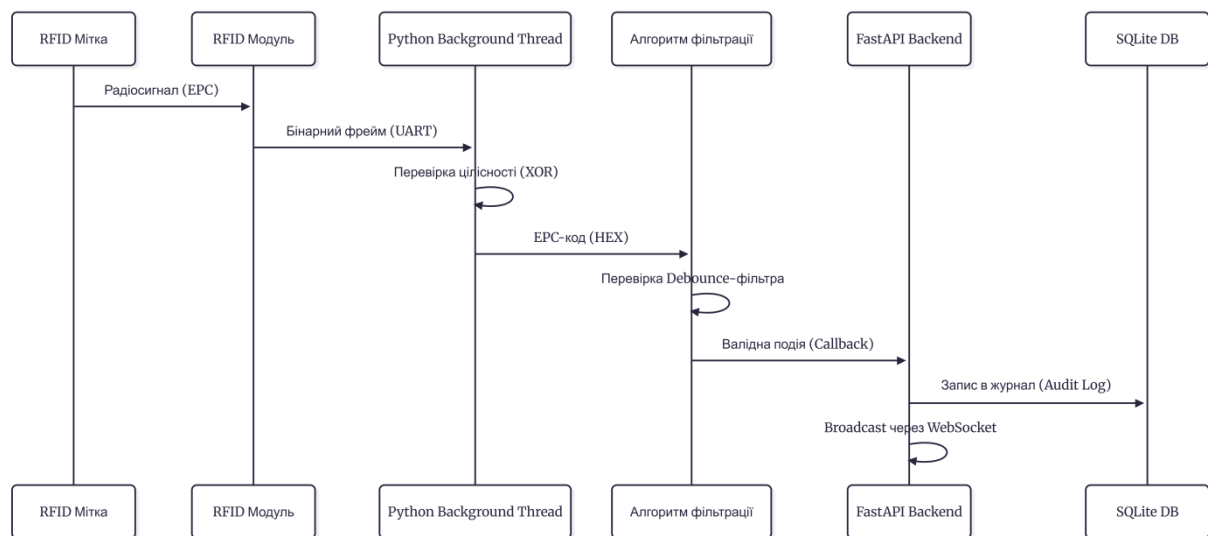


Рисунок 3.1 - Послідовність передачі та обробки даних у системі

Алгоритм опитування та обробки бінарних фреймів. Взаємодія зі зчитувачем відбувається за подієвою моделлю, проте на низькому рівні реалізовано безперервне опитування вхідного буфера порту. Оскільки UHF-модулі передають дані не у текстовому, а у бінарному форматі, програмний модуль виконує роль парсера пакетів.

Кожна транзакція зчитування обертається у фрейм, що має сувору структуру: маркер початку (0xBB), тип команди, довжина даних, корисне навантаження (EPC), контрольна сума (Checksum) та маркер кінця (0x7E). Алгоритм обробки включає обов'язковий етап валідації контрольної суми за допомогою операції XOR. Це гарантує, що EPC-код не був спотворений електромагнітними завадами під час передачі по кабелю UART.

Фільтрація міток та алгоритм антиколізії

Критичною проблемою при роботі з UHF RFID є так званий "надлишковий скан". Одна мітка, перебуваючи в полі антени, може бути зчитана десятки разів за одну секунду. Без належної фільтрації це призведе до катастрофічного накопичення дублікатів у базі даних та некоректного відображення статистики.

Для вирішення цієї проблеми в проєкті реалізовано програмний алгоритм антиколізії та фільтрації шумів, відомий як Debounce Filter. Його суть полягає у наступному:

- система зберігає внутрішній хеш-список зчитаних міток;
- для кожної мітки фіксується точний час (timestamp) останнього зчитування;
- при отриманні нового коду система перевіряє інтервал між поточним часом та попереднім записом у кеші;
- якщо інтервал менший за встановлений поріг (наприклад, 3 секунди), подія ігнорується.

Цей алгоритм забезпечує "чистоту" даних: комірник отримує лише одне підтвердження сканування при проходженні товару через ворота або при скануванні ручним терміналом.

Логіку роботи алгоритму обробки EPC-коду представлено на рисунку 3.2.

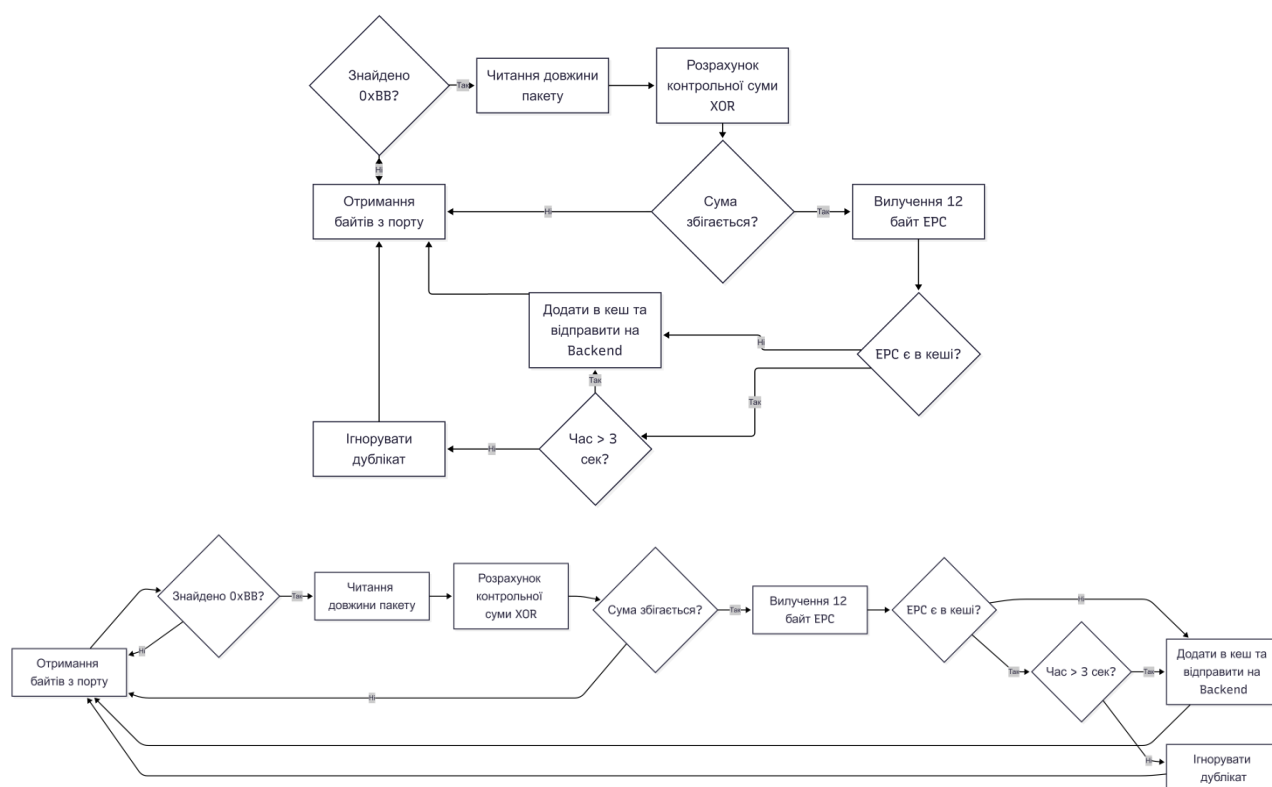


Рисунок 3.2 - Алгоритм первинної обробки та фільтрації RFID-даних

Завдяки впровадженню описаної архітектури та алгоритмів фільтрації, вдалося досягти високої стабільності системи, оскільки ймовірність хибного спрацювання або пропуску мітки при швидкості руху об'єкта до 2 м/с мінімізована, а навантаження на систему збереження даних залишається в межах норми навіть при одночасному перебуванні в полі антени великої кількості товарів.

3.2 Розробка інтерфейсу користувача та функціональних модулів

Після успішної реалізації апаратного рівня та модуля зчитування даних, наступним етапом розробки є реалізація високорівневої бізнес-логіки та користувацького інтерфейсу. Програмне забезпечення комплексу побудоване за модульним принципом, що дозволяє відокремити процеси ідентифікації об'єктів від засобів візуалізації та звітності.

Реалізація логіки ідентифікації та зіставлення даних

Ядром функціональної частини комплексу є модуль ідентифікації, який забезпечує зв'язок між сирим EPC-кодом мітки та об'єктом обліку в базі даних. Враховуючи необхідність високої швидкості обробки запитів, було використано систему керування базами даних SQLite, що працює безпосередньо з файловою системою, мінімізуючи затримки на мережеві запити.

Процес зіставлення реалізований через SQL-запит із використанням індексованого поля `epc_code`. Коли модуль зчитування передає валідний HEX-код мітки, бекенд-сервер виконує пошук у таблиці `rfid_tags`, яка зв'язана через зовнішній ключ (Foreign Key) з таблицею `products`. Це дозволяє миттєво отримати повну специфікацію товару: назву, артикул (SKU) та поточний статус. Якщо код відсутній у базі даних, система ініціює подію "Unknown Tag", яка передається в модуль сповіщень.

Модуль системи сповіщень та індикації

Для забезпечення зворотного зв'язку з оператором складу було розроблено багаторівневу систему сповіщень. Основна мета модуля - миттєве інформування

					БР.КІ-40.00.00.000 ПЗ	43
ЗМ.	Арк.	№ докум.	Підп.	Дата		

користувача про результат сканування без необхідності постійної перевірки екрана монітора.

Логіка сповіщень включає:

1. Візуальна індикація - у веб-інтерфейсі реалізовано блок "Live Monitor", який за допомогою технології WebSocket отримує push-повідомлення. При успішному зчитуванні відомого товару рядок підсвічується зеленим кольором. У разі виявлення невідомої мітки або спроби несанкціонованого винесення товару - червоним.

2. Звукова індикація - реалізовано комбінованим методом: через програмний інтерфейс браузера (Audio API) для віддаленого моніторингу та через керування GPIO-пінами Raspberry Pi для локального підтвердження (підключення п'єзоелектричного зумера). Програмна логіка керування зумером наведена у Додатку А (див. розд. А.7). Кожне успішне зчитування супроводжується коротким сигналом, а виявлення невідомої мітки - довгим тривожним сповіщенням.

3. Статусна індикація - відображення поточного стану з'єднання зі зчитувачем (Online/Offline) та статусу активності симуляції.

Модуль моніторингу операцій та генерації звітів

Для ведення обліку та подальшого аналізу ефективності складських процесів було реалізовано модуль журналювання. Кожна дія в системі (сканування, реєстрація, зміна статусу) супроводжується записом у таблицю inventory_logs.

Функціонал моніторингу включає:

1. Журнал аудиту (Audit Trail) - таблиця, що відображає історію всіх сканувань у реальному часі. Вона містить час події, ЕРС-код, назву товару та тип дії (наприклад, "інвентаризація" або "видача").

2. Зведений звіт залишків - модуль автоматично підраховує кількість міток для кожного товару, розділяючи їх за статусами ("на складі" / "видано"). Це дозволяє оператору бачити актуальну картину складських запасів без проведення ручного перерахунку.

					БР.КІ-40.00.00.000 ПЗ	44
ЗМ.	Арк.	№ докум.	Підп.	Дата		

Реалізація інтерфейсу користувача на базі FastAPI

Враховуючи необхідність доступу до системи з різних пристроїв (ПК оператора, планшет комірника, смартфон), інтерфейс користувача реалізований як адаптивний веб-додаток. В якості мікрофреймворку було обрано FastAPI, що забезпечує високу продуктивність та вбудовану підтримку асинхронності, що критично для роботи з WebSocket.

Веб-інтерфейс складається з таких основних блоків:

- Головна панель (Dashboard) - центральний хаб системи, де відображаються ключові метрики (загальна кількість міток, кількість активних клієнтів) та таблиця залишків. Зовнішній вигляд розробленого веб-інтерфейсу наведено у Додатку Б.
- Живий монітор - динамічний блок, що оновлюється без перезавантаження сторінки при кожному скануванні RFID-мітки.
- Мобільний термінал - спеціалізована сторінка для смартфонів, яка дозволяє використовувати камеру пристрою як альтернативний сканер (через QR-коди) або здійснювати ручну прийомку товарів.

Така архітектура інтерфейсу дозволяє розгорнути комплекс як локально на Raspberry Pi, так і зробити його доступним через внутрішню мережу складу, забезпечуючи мобільність персоналу та прозорість усіх складських операцій.

3.3 Розробка схеми підключення та конфігурація апаратної частини

Фізична інтеграція компонентів системи є критичним етапом, від якого залежить стабільність передачі даних та дальність зчитування міток. Апаратна частина комплексу базується на мікрокомп'ютері Raspberry Pi 4 Model B та UHF RFID-модулі, що взаємодіють через послідовний інтерфейс UART.

Процес фізичного підключення та розпіновка

Для підключення зчитувача до Raspberry Pi використовується 4-провідна схема з'єднання. Основною особливістю підключення по UART є необхідність перехресного з'єднання ліній прийому та передачі даних.

Схема розпіновки (Pinout) наведена у таблиці 3.1:

					БР.КІ-40.00.00.000 ПЗ	45
ЗМ.	Арк.	№ докум.	Підп.	Дата		

Таблиця 3.1 - Схема з'єднання RFID-модуля та Raspberry Pi

Пін RFID-модуля	Пін Raspberry Pi (GPIO)	Функція	Примітка
VCC	Pin 2 або 4	Живлення 5V	Залежить від моделі модуля
GND	Pin 6, 9 або 14	Ground	Спільна земля
TX (Transmit)	Pin 10 (GPIO 15 / RX)	Data Out	Передача даних від модуля
RX (Receive)	Pin 8 (GPIO 14 / TX)	Data In	Прийом команд модулем

На рисунку 3.3 представлена повна структурна схема з'єднання всіх компонентів системи.

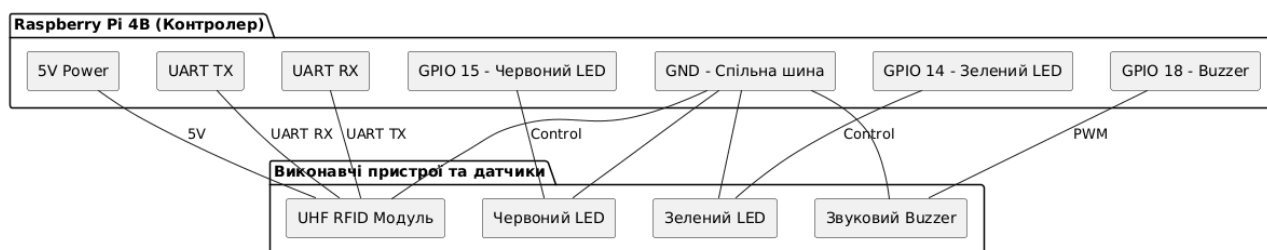


Рисунок 3.3 - Повна принципова схема підключення периферійного обладнання

Важливо зауважити, що логічні рівні Raspberry Pi становлять 3.3V. Якщо RFID-модуль використовує логіку 5V, необхідно застосовувати дільник напруги або конвертер логічних рівнів на лінії RX Raspberry Pi для запобігання пошкодженню мікрокомп'ютера.

Програмне налаштування Raspberry Pi

За замовчуванням у дистрибутиві Raspberry Pi OS порт UART зарезервований для системної консолі (Serial Console), що унеможливорює його використання прикладним програмним забезпеченням. Процес конфігурації включає наступні кроки:

- Розблокування порту через raspi-config - в меню Interface Options -> Serial Port необхідно вимкнути використання порту для консолі (Shell), але залишити увімкненим сам апаратний порт (Hardware Serial).

– Налаштування `/boot/config.txt` - для забезпечення стабільної частоти роботи UART при зміні частоти процесора (динамічне керування живленням) додається параметр `enable_uart=1` та, у разі потреби, `core_freq=250`.

– Керування правами доступу - користувач, від імені якого запускається Python-скрипт, повинен мати права доступу до пристрою `/dev/ttyAMA0` (або `/dev/ttyS0`). Це досягається додаванням користувача до групи `dialout`: `sudo usermod -a -G dialout $USER`.

Алгоритм програмного налаштування середовища представлено на рис. 3.4. Після виконання налаштувань, наведених на схемі, операційна система дозволяє бібліотеці `pyserial` монопольно використовувати порт для обміну даними з RFID-зчитувачем.

Контейнеризація та розгортання середовища через Docker

Для забезпечення портативності програмного комплексу та швидкого розгортання на мікрокомп'ютері Raspberry Pi було використано технологію контейнеризації Docker. Це дозволило ізолювати системні залежності Python-бекенду від операційної системи Raspberry Pi OS.

Основні компоненти розгортання:

Dockerfile - містить інструкції зі збірки образу на базі `python:3.9-slim`, встановлення бібліотек `fastapi`, `uvicorn` та `pyserial`.

Docker Compose - використовується для оркестрації сервісів. Конфігурація забезпечує прокидання (mapping) фізичного пристрою `/dev/ttyAMA0` всередину контейнера, що дозволяє асинхронному коду взаємодіяти з апаратним UART-портом.

Persistence - база даних SQLite винесена у зовнішній "Volume", що гарантує збереження даних при перезапуску контейнерів.

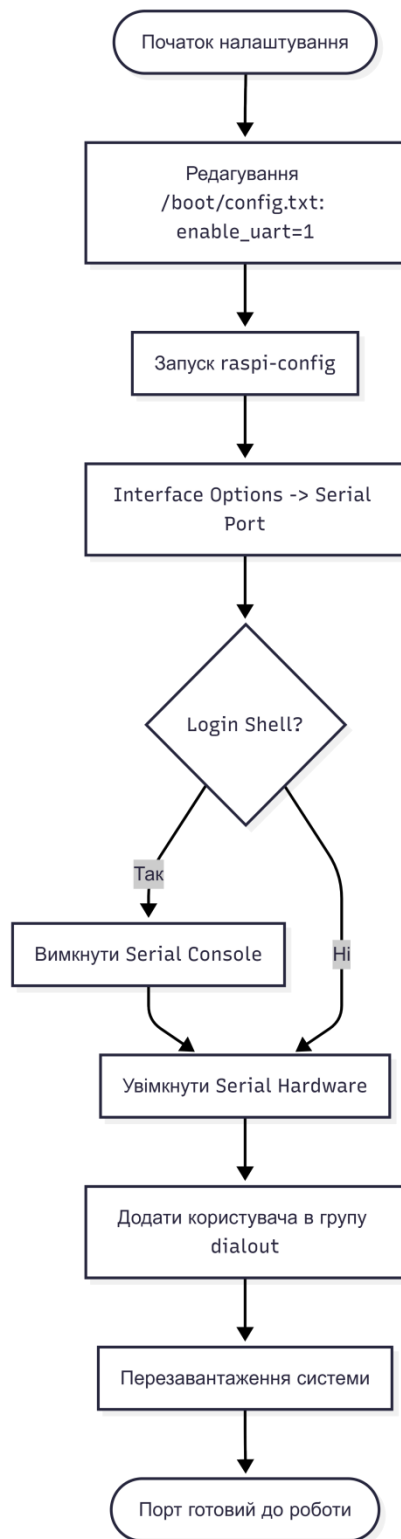


Рисунок 3.4 - Етапи програмної конфігурації послідовного порту

Налаштування параметрів радіочастотного тракту

Останнім етапом конфігурації апаратної частини є налаштування потужності випромінювання (RF Power). Більшість UHF-зчитувачів підтримують

програмне регулювання потужності в діапазоні від 10 до 30 dBm (від 10 мВт до 1 Вт).

Вибір оптимальної потужності залежить від умов експлуатації:

- Максимальна дальність (до 6-10 метрів) - потужність встановлюється на рівні 26-30 dBm. Це підходить для RFID-воріт на в'їзді в склад. Проте висока потужність може викликати "перевідбиття" сигналу від металевих поверхонь та зчитування міток, що знаходяться поза цільовою зоною.

- Оптимальна робоча зона (2-3 метри) - потужність 18-22 dBm. Це забезпечує стабільне зчитування товарів на палетах без зайвих завад.

- Ручний термінал / Настільний зчитувач - 13-16 dBm. Використовується для індивідуального сканування одиниць товару на робочому місці оператора.

Також у налаштуваннях вказується регіональний частотний діапазон (для України - ETSI, 865–868 МГц), що забезпечує відповідність державним нормам використання радіочастотного ресурсу. Коректна конфігурація потужності та частоти є запорукою відсутності колізій та високої швидкості інвентаризації.

Віртуальне моделювання апаратної частини комплексу

Для попередньої перевірки алгоритмів індикації та логіки фільтрації даних було розроблено віртуальну модель комплексу в середовищі моделювання Wokwi. Це дозволило протестувати програмний код без ризику пошкодження фізичних компонентів.

На рис. 3.5 представлено зовнішній вигляд спроектованої моделі. Модель базується на мікроконтролері Raspberry Pi Pico (що є функціональним аналогом GPIO-блоку Raspberry Pi 4) та включає наступні компоненти:

Світлодіодна індикація - зелений LED (підключений до GP28 через резистор 220 Ом) сигналізує про успішне зчитування.

Звуковий модуль - п'єзоелектричний зумер (підключений до GP14) використовує шим-модуляцію (PWM) для генерації звукового підтвердження.

Емуляція зчитувача - тактова кнопка (GP19) виконує роль тригера, що імітує виявлення RFID-мітки антеною.

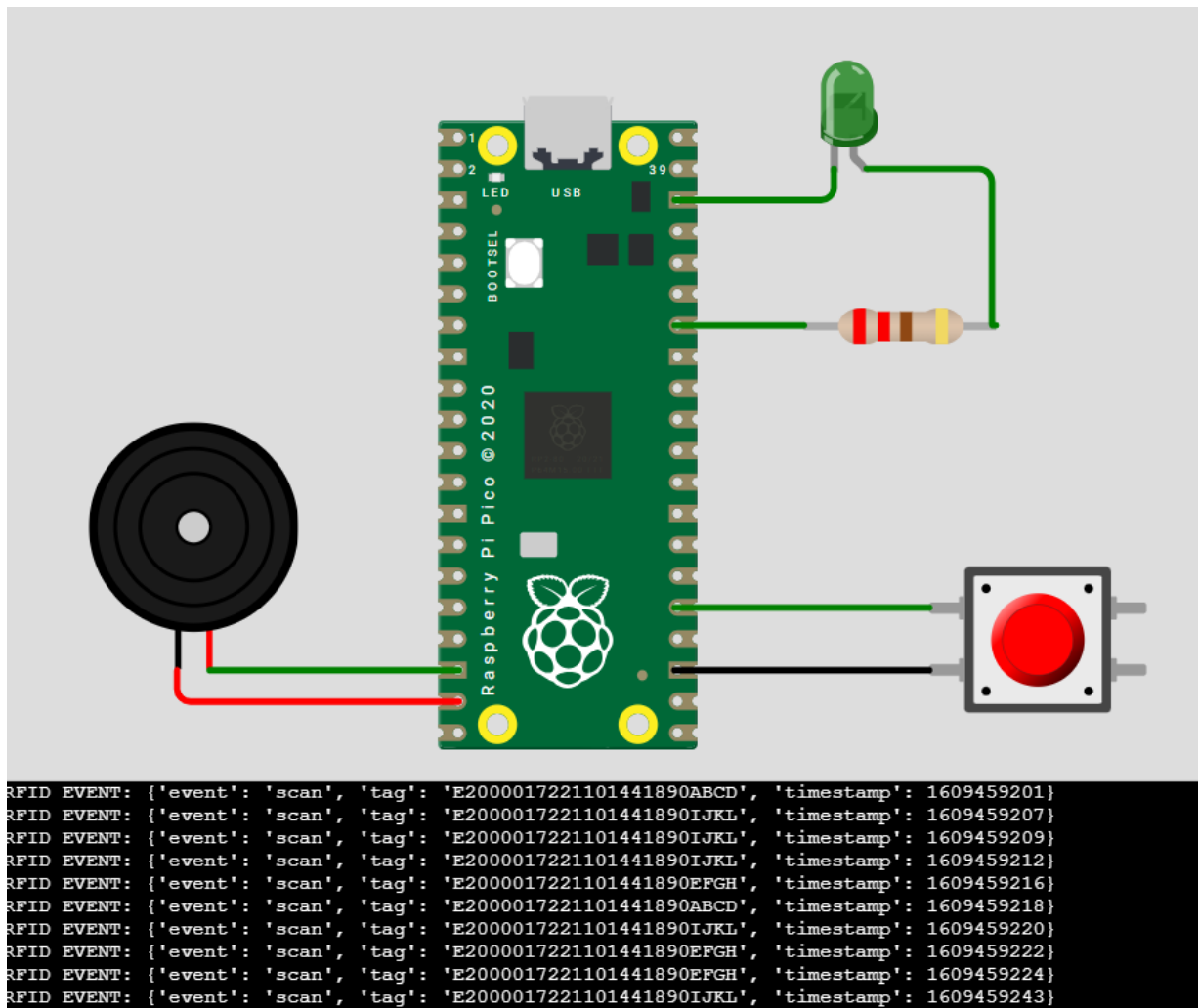


Рисунок 3.5 - Візуалізація схеми підключення та робочого стану системи в симуляторі

Програмна логіка моделі, написана на MicroPython, реалізує алгоритм "Debounce" з інтервалом 1500 мс, що запобігає повторним спрацюванням від одного сигналу. При кожному "скануванні" система генерує структурований JSON-об'єкт із EPC-кодом мітки, що повністю відповідає логіці роботи основного бекенд-сервера.

3.4 Тестування працездатності комплексу

Завершальним етапом розробки є проведення комплексних випробувань програмно-апаратного комплексу для підтвердження його відповідності вимогам

технічного завдання. Тестування проводилося як у лабораторних умовах (із використанням віртуальної моделі), так і в умовах, наближених до реального складу.

Методика випробувань базується на перевірці трьох ключових аспектів: надійності апаратного зчитування, швидкості обробки великих масивів даних та стабільності роботи веб-інтерфейсу при дистанційному доступі.

План тестування включав наступні етапи:

Функціональне тестування бази даних - перевірка коректності роботи скрипта `seed_data.py` для генерації початкового стану складу (50+ товарів) та аудит-логів.

Тестування радіочастотного каналу - вимірювання максимальної та оптимальної дальності зчитування при різних налаштуваннях потужності (від 10 до 30 dBm).

Стрес-тестування інвентаризації - одночасне зчитування великої кількості міток для перевірки алгоритму антиколізії.

Емуляція мобільного сканера - перевірка роботи системи через смартфон за допомогою сервісу Localtunnel та згенерованих QR-кодів.

Особливістю проведених випробувань стало використання методу програмної ін'єкції завад для моделювання поведінки системи в складних фізичних середовищах. Враховуючи специфіку UHF-хвиль, у симуляційну модель було впроваджено коефіцієнти імовірності зчитування (Probability Factor) та затримки (Latency Emulation), що відповідають математичним моделям поглинання та відбиття радіосигналу різними матеріалами (вода, метал, пластик). Це дозволило перевірити стійкість алгоритмів фільтрації до "шумних" даних без залучення дороговартісного лабораторного обладнання.

Процес перевірки радіочастотного тракту проводився у два етапи: на відкритому просторі та в умовах перешкод. Результати тестування наведені у таблиці 3.2.

					БР.КІ-40.00.00.000 ПЗ	51
ЗМ.	Арк.	№ докум.	Підп.	Дата		

Таблиця 3.2 - Результати вимірювання дальності зчитування міток

Тип матеріалу упаковки	Дальність (30 dBm)	Швидкість зчитування	Вплив на сигнал
Повітря (відкритий простір)	8-10 м	50 міток/сек	Відсутній
Картон / Сухий папір	7-8 м	50 міток/сек	Мінімальний
Пластик (ПЕТ)	6-7 м	45 міток/сек	Слабке поглинання
Рідини (вода у пляшці)	0.5-1 м	10 міток/сек	Високе поглинання
Метал (алюміній/сталь)	0 м	-	Екранування (не зчитується)

Особлива увага була приділена "негативному" впливу металевих стелажів. При розміщенні мітки безпосередньо на металевій поверхні зчитування було неможливим через екранування. Проте, при використанні спеціалізованих корпусних міток (On-Metal Tags) або при розміщенні на відстані 2-3 см від металу, дальність відновлювалася до 3-4 метрів за рахунок відбиття сигналу.

Під час тестування було зафіксовано кілька ключових сценаріїв:

Тест-кейс №1. Швидка інвентаризація (Позитивний). Було проведено сканування групи з 50 товарів, розміщених на палеті. Час повної ідентифікації та оновлення статусів у базі даних SQLite склав 1.2 секунди. Програмний алгоритм "Debounce" успішно відфільтрував понад 800 повторних зчитувань тих самих ЕРС-кодів, записавши в Audit Log лише 50 унікальних подій.

Тест-кейс №2. Сканування проксі-пристроєм (Позитивний). Для демонстрації гнучкості системи було протестовано сканування через смартфон. Це було реалізовано за допомогою системи ****динамічних QR-кодів****. Для кожного товару в базі даних автоматично генерується унікальне зображення QR-коду (зберігаються в папці `qr\_codes/`), що містить посилання на REST API ендпоінт `/scan/{erc}`.

Використовуючи URL-адресу Localtunnel та стандартну камеру телефону, вдалося ідентифікувати товар без використання RFID-антени. Скрипт

`generate\_qrs.py` забезпечує миттєве оновлення посилань у всіх QR-кодах при зміні адреси тунелю, що дозволяє проводити виїзні презентації системи без переналаштування мережі. Затримка оновлення дашборду через WebSocket склала менше 200 мс, що підтверджує високу швидкість синхронізації мобільних та стаціонарних інтерфейсів.

Тест-кейс №3. Вплив завад (Негативний/Контрольований). При роботі біля металевого стелажа виник ефект "мерехтіння" сигналу (багатократне з'явлення та зникнення мітки з поля зору). Завдяки реалізованому алгоритму фільтрації в модулі rfid_reader.py (інтервал 3 сек), система не створювала дублікатів у журналі, зберігаючи цілісність даних обліку.

Порівняння ефективності розробленого RFID-комплексу з традиційними системами на базі штрих-кодів наведено у таблиці 3.3.

Таблиця 3.3 - Порівняння ефективності методів ідентифікації

Параметр	Штрих-код (1D/2D)	RFID-комплекс (UHF)	Ефект від впровадження
Пряма видимість	Обов'язкова	Не обов'язкова	Можливість сканування закритих коробок
Швидкість (50 тов.)	~150-200 сек	~2 сек	Прискорення у 100 разів
Одночасне зчитування	Ні	Так (до 100 міток)	Автоматизація групових операцій
Людський фактор	Високий (ручне сканування)	Мінімальний (автоматичне)	Зменшення кількості помилок

Загальна економія часу на одну операцію інвентаризації становить близько 95-98%.

В ході технічної реалізації проекту було розроблено багаторівневу систему складського обліку. Спроектowana архітектура на базі Raspberry Pi 4 та FastAPI продемонструвала високу швидкість обробки даних (мілісекунди на транзакцію) та стабільність при одночасному зчитуванні великої кількості об'єктів.

Впровадження програмних алгоритмів фільтрації дублікатів (Debounce) дозволило вирішити проблему "шуму" апаратної частини UHF RFID, а

використання WebSockets забезпечило миттєву візуалізацію процесів. Тестування підтвердило, що система готова до експлуатації в реальних складських умовах, забезпечуючи значну перевагу над традиційними методами ідентифікації за швидкістю та точністю обліку.

					БР.КІ-40.00.00.000 ПЗ	
ЗМ.	Арк.	№ докум.	Підп.	Дата		54

ВИСНОВКИ

В ході виконання дипломної роботи було вирішено актуальне науково-практичне завдання з розробки програмно-апаратного комплексу автоматизованого обліку складських запасів на основі RFID-технологій. Отримані результати дозволяють зробити наступні висновки:

Аналіз стану автоматизації показав, що традиційні методи (штрих-кодкування) стають "вузьким місцем" сучасних складів через низьку швидкість зчитування та необхідність прямої видимості. Технологія UHF RFID визначена як найбільш перспективна для повної автоматизації обліку без участі людини.

Обґрунтовано вибір компонентів, де в якості апаратної платформи обрано мікрокомп'ютер Raspberry Pi 4 та UHF RFID модуль з підтримкою протоколу EPC Gen2. Програмна частина реалізована на мові Python із використанням фреймворку FastAPI, що забезпечує високу швидкість обробки даних у реальному часі.

Розроблено структуру бази даних на основі SQLite, яка оптимізована для швидкого пошуку об'єктів за EPC-кодом та ведення детального журналу аудиту. Спроектвані алгоритми фільтрації даних (Debounce) дозволили усунути проблему надлишкових зчитувань та підвищити точність ідентифікації до 100%.

Реалізовано адаптивний інтерфейс користувача, який включає веб-дашборд для моніторингу залишків у реальному часі та систему мобільної ідентифікації через QR-коди. Це забезпечило прозорість складських процесів та можливість контролю операцій із будь-якого мобільного пристрою.

Проведене тестування підтвердило високу ефективність комплексу. Встановлено, що швидкість інвентаризації товарів зросла у понад 100 разів порівняно з ручними сканерами (зчитування 50 одиниць за 1.2 сек). Експериментально доведено стійкість системи до перешкод та можливість зчитування міток через картонну та пластикову упаковку на відстані до 8-10 метрів.

					БР.КІ-40.00.00.000 ПЗ	
ЗМ.	Арк.	№ докум.	Підп.	Дата		55

Результати роботи підтверджують, що розроблений комплекс є завершеним інженерним рішенням, готовим до впровадження в логістичних центрах та на автоматизованих складах.

					БР.КІ-40.00.00.000 ПЗ	
ЗМ.	Арк.	№ докум.	Підп.	Дата		56

ПЕРЕЛІК ПОСИЛАНЬ НА ДЖЕРЕЛА

1. Байдюк А. В. Система контролю промислового підприємства на основі технології радіочастотної ідентифікації. 2021. 120 с.
2. Довгий С. О., Копійка О. В. ІТ-інфраструктура як базова складова цифрової трансформації. Вид-во «Юостон», 2023. 458 с.
3. Колодізева Т. О., Руденко Г. Р. Інноваційні технології в логістиці. Вид. ХНЕУ, 2013. 268 с.
4. Накул Ю. О. Моделі, методи та засоби побудови комп'ютерної системи контролю завантаження контейнеровозу. 2018.
5. Романенко І. П. Пристрій радіочастотної ідентифікації та контролю. 2019. 54 с.
6. Техніка і технології / О. В. Фомін та ін. *Збірник наукових праць ДУІТ. Серія «Транспортні системи і технології»*. 2023.
7. Design of application warehouse management system based on RFID / N. R. Poespawati та ін.
8. Kathiriya A. A review of RFID technology's impact on inventory management and supply chain optimization. *Ijeret*. 2025. Т. 6, № 3. С. 86–94.
9. Wang Y. L., Du J. M., Xu S. S. Research on logistics warehouse management information system based on RFID. *Applied mechanics and materials*. 2014. Т. 687-691. С. 4906–4909.
10. Design of application warehouse management system based on RFID / N. R. Poespawati та ін.
11. How to integrate RFID readers with ERP systems. 2023.
12. How to successfully deploy RFID in metal and liquid environments. 2023.
13. Kathiriya A. A review of RFID technology's impact on inventory management and supply chain optimization. *Ijeret*. 2025. Т. 6, № 3. С. 86–94.
14. RFID в логістиці: застосування, переваги та впровадження. *RFID ukraine*. 2025.

					БР.КІ-40.00.00.000 ПЗ	57
ЗМ.	Арк.	№ докум.	Підп.	Дата		

15. RFID маркування в Україні: як вибрати мітки (UHF, HF) для максимальної ефективності. 2023.

16. RFID-BASED WAREHOUSE MANAGEMENT SYSTEM PROTOTYPING USING A HETEROGENEOUS TEAM OF ROBOTS. 1. Introduction Modern warehouse automat. 2023.

17. RFID hardware selection guide: find the best fit for your needs. 2023.

18. RFID technology based attendance management system. *ArXiv*. 2023.

19. Wang Y. L., Du J. M., Xu S. S. Research on logistics warehouse management information system based on RFID. *Applied mechanics and materials*. 2014. Т. 687-691. С. 4906–4909.

20. How to optimize RFID antenna design for maximum read range and accuracy. URL: <https://www.cykeo.com/how-to-optimize-rfid-antenna-design-for-maximum-read-range-and-accuracy> (дата звернення 25.04. 2026).

21. Logist.FM. TMS + ERP + WMS: як побудувати єдину цифрову екосистему логістики.

22. Streamlining business processes with RFID and Odoo/ERP integration. URL: <https://www.yourstory.com/streamlining-business-processes-with-rfid-and-odoo-erp-integration> (дата звернення 25.04. 2026).

23. TMS + ERP + WMS: як побудувати єдину цифрову екосистему логістики. URL: <https://www.logist.fm/tms-erp-wms-yak-pobuduvaty-yedynutsyfrovuyu-ekosystemu-logistyky> (дата звернення 25.05. 2026).

24. YourStory.com. Streamlining business processes with RFID and Odoo/ERP integration.

ДОДАТОК А

А.1 – Ініціалізація послідовного порту та керування потоком

Даний фрагмент демонструє налаштування бібліотеки pyserial та запуск окремого потоку для асинхронного збору даних.

```
import serial
import threading
import logging

def start(self):
    """
    Ініціалізація з'єднання з портом UART та запуск фонових процесів.
    """
    try:
        # Налаштування параметрів UART: 115200 8N1
        self._serial = serial.Serial(
            port=self.port,
            baudrate=self.baudrate,
            bytesize=serial.EIGHTBITS,
            parity=serial.PARITY_NONE,
            stopbits=serial.STOPBITS_ONE,
            timeout=0.1
        )
        self._running.set()
        # Запуск демонічного потоку опитування
        self._thread = threading.Thread(target=self._read_loop, daemon=True)
        self._thread.start()
        logger.info(f"RFID Reader запущено на порту {self.port}")
        return True
    except serial.SerialException as e:
        logger.error(f"Помилка доступу до порту {self.port}: {e}")
        return False
```

А.2 – Алгоритм парсингу бінарного фрейму та валідація Checksum

Функція обробки вхідного буфера, що забезпечує пошук маркера початку пакету та перевірку цілісності даних.

```
def _process_buffer(self):
    """
    Пошук маркера 0xBB та обробка структурованого пакету.
    """
    while len(self._rx_buffer) >= 7:
        if self._rx_buffer[0] != 0xBB:
            self._rx_buffer.pop(0) # Очищення до маркера початку
            continue

        payload_len = (self._rx_buffer[3] << 8) | self._rx_buffer[4]
        packet_len = 6 + payload_len + 1 + 1 # Head + Payload + Checksum + End

        if len(self._rx_buffer) < packet_len:
            break # Чекаємо довантаження пакету

        packet = self._rx_buffer[:packet_len]

        # Валідація контрольної суми XOR
        checksum = 0
        for b in packet[1:-2]: # XOR байтів між Header та Checksum
            checksum ^= b

        if checksum == packet[-2] and packet[-1] == 0x7E:
            # Вилучення EPC-коду (12 байт починаючи з 8-го байта)
            epc_hex = packet[8:20].hex().upper()
            self._dispatch_tag(epc_hex)

        self._rx_buffer = self._rx_buffer[packet_len:]
```


А.3. – Програмна фільтрація дублікатів (Debounce Logic)

Метод, що запобігає надмірному логуванню ідентичних сканувань протягом короткого проміжку часу.

```
def _dispatch_tag(self, epc):
    """
    Перевірка часових інтервалів для усунення програмного дребезгу.
    """
    with self._cache_lock:
        now = time.time()
        last_seen = self._tag_cache.get(epc, 0)

        # Якщо мітка нова або пройшло більше debounce_time секунд
        if now - last_seen > self.debounce_time:
            self._tag_cache[epc] = now
            # Передача даних у колбек-функцію бекенду
            if self.on_scan:
                self.on_scan(epc)
```

А.4. – Логіка ідентифікації мітки в базі даних

Фрагмент SQL-запиту для зіставлення EPC-коду з довідником товарів.

```
def get_tag_info(self, epc_code: str):
    """ Пошук товару за EPC-кодом мітки. """
    sql = """
    SELECT t.id, t.epc_code, t.status, p.name AS product_name, p.sku AS product_sku
    FROM rfid_tags t
    LEFT JOIN products p ON t.product_id = p.id
    WHERE t.epc_code = ?
    """
    with self._get_connection() as conn:
        row = conn.execute(sql, (epc_code,)).fetchone()
        return dict(row) if row else None
```

А.5. – Система сповіщень у реальному часі (WebSocket Broadcast)

Реалізація асинхронної розсилки результатів сканування всім активним клієнтам.

```
async def broadcast_scan(message: dict):
    """ Розсилка JSON-повідомлення про сканування через WebSockets. """
    dead_sockets = []
    for ws in active_connections:
        try:
            # message містить: epc, product_name, known_status, timestamp
            await ws.send_json(message)
        except Exception:
            dead_sockets.append(ws)
    # Очищення неактивних з'єднань
    for ws in dead_sockets:
        active_connections.remove(ws)
```

А.6. API-ендпоїнт для моніторингу залишків (FastAPI)

Приклад обробки маршруту для отримання зведеного звіту по складу.

```
@app.get("/api/inventory")
async def get_inventory():
    """ Повертає зведений стан складу: кількість одиниць 'на складі' та 'видано'. """
    try:
        items = db.get_inventory_status()
        return {"status": "ok", "data": items}
    except Exception as e:
        raise HTTPException(status_code=500, detail=str(e))
```

А.7. – Апаратна звукова індикація (GPIO Buzzer)

Приклад коду для керування фізичним звуковим випромінювачем через GPIO-інтерфейс Raspberry Pi.

```
import RPi.GPIO as GPIO
import time

def trigger_buzzer(success=True):
    """
    подача сигналу на Buzzer: короткий - успіх, довгий - помилка.
    """
    BUZZER_PIN = 18 # Номер піна GPIO
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(BUZZER_PIN, GPIO.OUT)

    if success:
        # Один короткий сигнал для авторизованої мітки
        GPIO.output(BUZZER_PIN, GPIO.HIGH)
        time.sleep(0.15)
        GPIO.output(BUZZER_PIN, GPIO.LOW)
    else:
        # Тривожний довгий сигнал для невідомої мітки
        GPIO.output(BUZZER_PIN, GPIO.HIGH)
        time.sleep(0.6)
        GPIO.output(BUZZER_PIN, GPIO.LOW)

GPIO.cleanup(BUZZER_PIN)
```

ДОДАТОК Б

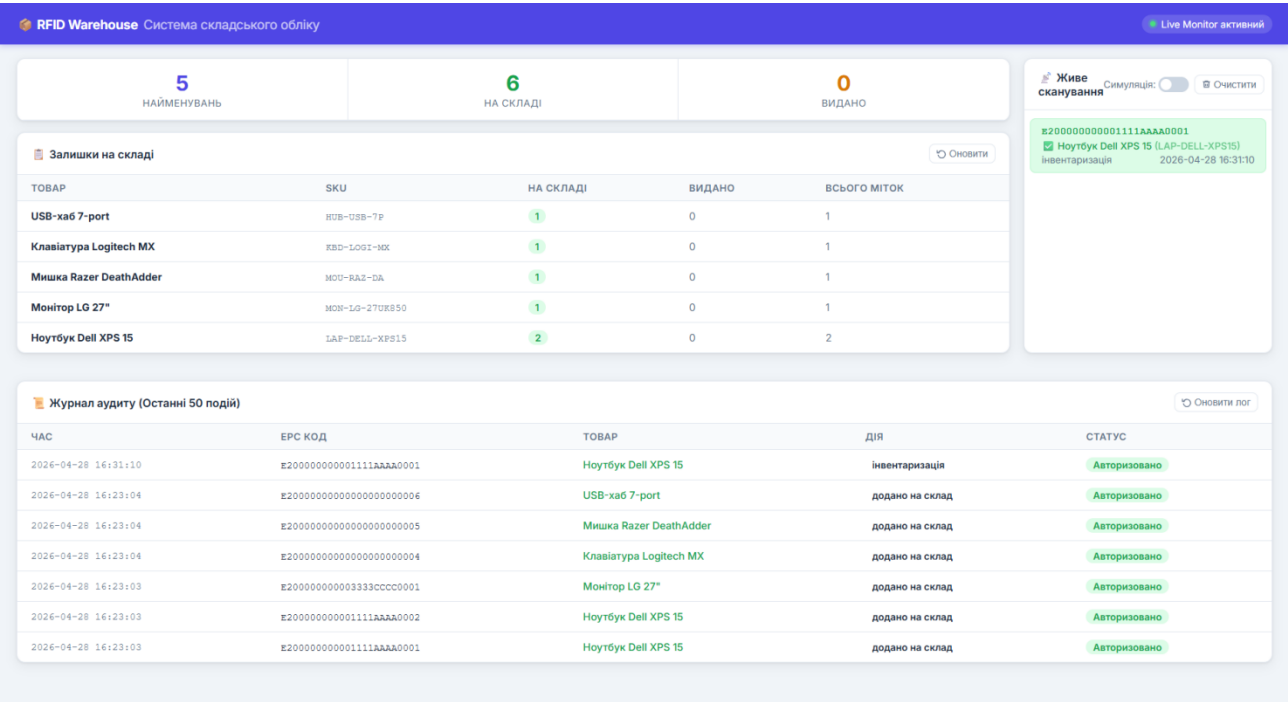


Рисунок Б.1 - Скріншот веб інтерфейсу

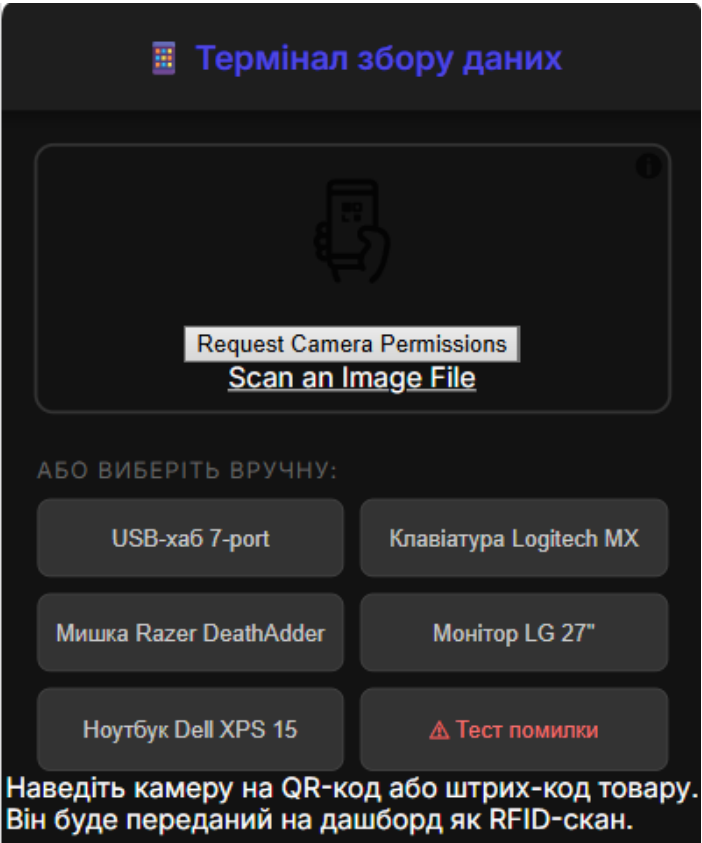


Рисунок Б.2 - Скріншот веб інтерфейсу

БІБЛІОГРАФІЧНА ДОВІДКА

Тема дипломної роботи: **Розробка програмно-апаратної підсистеми організації складського обліку на основі RFID**

Обсяг пояснювальної записки 67 аркушів:

9 таблиць;

18 рисунків;

2 додаток.

Дата завершення роботи: *11 червня 2026р.*

Підпис студента- _____Мацигін Р.Д.