Міністерство освіти і науки України
Криворізький національний університет
Факультет інформаційних технологій
Кафедра автоматизації, комп'ютерних наук і технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеню вищої освіти – магістр
за освітньо-професійною програмою
«Автоматизація, комп'ютерно-інтегровані
технології та робототехніка»

зі спеціальності
174 – Автоматизація, комп'ютерно-інтегровані
технології та робототехніка

тема роботи:

«Автоматизація керування ланцюгом постачання лікарських препаратів з використанням технології Blockchain»

Виконала студентка гр. АКІТР-23-1м. _____ Пасічна Є. В.

Керівник _____ Рубан С. А.

Нормоконтроль _____ Маринич І. А.

Завідувач кафедри _____ Рубан С. А.

Кривий Ріг – 2024

КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет: *інформаційних технологій*

Кафедра: *автоматизації, комп'ютерних наук і технологій*

Ступінь вищої освіти: *Магістр*

Спеціальність: *174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка.*

ЗАТВЕРДЖУЮ

Зав. кафедри: *к.т.н. Рубан С.А.*

_____

« *5* » *липня* 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра

студентці групи АКІТР-23-1м *Пасічній Єлизаветі Віталіївні*

1. **Тема кваліфікаційної роботи:** *«Автоматизація керування ланцюгом постачання лікарських препаратів з використанням технології Blockchain»*

затверджено наказом по університету № *595с* від *04.07.2024 р.*

2. Термін здачі кваліфікаційної роботи: *01.12.2024 р.*

3. Склад кваліфікаційної роботи: *Пояснювальна записка обсягом 93с., додатки, презентація у Microsoft PowerPoint (12 слайдів) в електронному та друкованому вигляді*

4. Консультанти кваліфікаційної роботи:

Розділ 1-3 к.т.н. Рубан С. А.

_____

Нормоконтроль доц. Маринич І. А.

5. Календарний план:

№	Етапи роботи	Термін виконання
1	*Вступ*	*10.07.24*
2	*Розділ 1*	*15.07.24*
3	*Розділ 2*	*18.08.24*
4	*Розділ 3*	*19.09.24*
5	*Висновки*	*15.10.24*
6	*Оформлення кваліфікаційної роботи*	*20.11.24*
7	*Підготовка презентації та графічного матеріалу*	*28.11.24*
8	*Підготовка доповіді до захисту*	*01.12.24*

6. Дата видачі завдання: *28.06.2024р.*

Керівник _____ /Рубан С. А./

7. Запевнення: *Я, Пасічна Єлизавета Віталіївна, запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.*

Із чинним Положенням про академічну доброчесність Криворізького національного університету ознайомлений.

Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі умисних порушень робота не допускається до захисту або оцінюється незадовільно.

Здобувач _____ /Пасічна Є. В./

АНОТАЦІЯ

Пасічна Є.В. «Автоматизація керування ланцюгом постачання лікарських препаратів з використанням технології Blockchain».

Кваліфікаційна робота на здобуття ступеню вищої освіти магістр за освітньо-професійною програмою «Кіберфізичні системи в промисловості, бізнесі та транспорті» зі спеціальності 174 – Автоматизація, комп'ютерно – інтегровані технології та робототехніка– Криворізький національний університет, Кривий Ріг, 2024.

Мета кваліфікаційної роботи полягає у розробці та впровадженні системи управління ланцюгом постачання лікарських препаратів на основі технології блокчейн для забезпечення прозорості, безпеки та ефективності процесу постачання. Об'єктом дослідження є ланцюг постачання лікарських препаратів, включаючи виробництво, транспортування, зберігання та реалізацію ліків. Предметом дослідження є застосування технології блокчейн для оптимізації та забезпечення безпеки ланцюга постачання лікарських препаратів.

Одержані результати включають розробку та впровадження системи управління ланцюгом постачання лікарських препаратів на основі технології блокчейн, яка забезпечує прозорість, безпеку та ефективність процесу постачання. Система дозволяє автоматизувати процеси в ланцюзі постачання за допомогою смарт-контрактів та забезпечує інтеграцію з веб-додатком для управління ланцюгом постачання.

Ключові слова: БЛОКЧЕЙН, ЛАНЦЮГ ПОСТАЧАННЯ, ЛІКАРСЬКІ ПРЕПАРАТИ, АВТОМАТИЗАЦІЯ, СМАРТ-КОНТРАКТИ, ПРОЗОРІСТЬ, БЕЗПЕКА

ANNOTATION

Pasichna Y.V. "Automation of Pharmaceutical Supply Chain Management Using Blockchain Technology".

Graduation master`s work for obtaining an educational degree «Master» for the educational and professional program « Cyber-physical systems in industry, business and transport » in specialty 174 – «Automation, computer-integrated technologies, and robotics». – Kryvyi Rih National University, Kryvyi Rih, 2024

The objective of this qualification work is to develop and implement a management system for the medicine supply chain based on blockchain technology to ensure transparency, security, and efficiency of the supply process. The object of the research is the medicine supply chain, including the production, transportation, storage, and distribution of medicines. The subject of the research is the application of blockchain technology for optimizing and ensuring the security of the medicine supply chain.

The obtained results include the development and implementation of a management system for the medicine supply chain based on blockchain technology, which ensures transparency, security, and efficiency of the supply process. The system enables the automation of processes in the supply chain through the use of smart contracts and provides integration with a web application for managing the supply chain.

Keywords: BLOCKCHAIN, SUPPLY CHAIN, MEDICAL PRODUCTS, AUTOMATION, SMART CONTRACTS, TRANSPARENCY, SECURITY

ЗМІСТ

ВСТУП

У сучасному світі логістика постачання лікарських препаратів стає все більш складною через зростання попиту на медичні засоби, що ставить під загрозу доступність ліків для пацієнтів. Крім того, складність управління інформацією та даними ускладнює процес координації між виробниками, дистриб'юторами та кінцевими користувачами. Пандемія COVID-19 ще більше підкреслила цю необхідність, наголошуючи на важливості швидкості, прозорості та надійності постачання медичних засобів для боротьби зі стихійними випадками та захворюваннями.

У контексті цих викликів, технологія блокчейн стає надзвичайно привабливою для оптимізації ланцюга постачання лікарських препаратів. Вона забезпечує безпеку, прозорість та ефективність у кожному етапі постачання, починаючи з виробництва та закінчуючи передачею продукту кінцевому споживачу. Використання блокчейну забезпечує достовірність даних, аутентифікацію продуктів та виробників, а також ефективну взаємодію між учасниками ланцюга постачання.

Крім того, технологія блокчейн може вирішити проблему підробки ліків, що є серйозним викликом для медичної індустрії. Завдяки унікальним ідентифікаційним міткам та системі відстеження, вона дозволить ефективно перевіряти автентичність продуктів і уникати поширення фальшивих лікарських засобів в ланцюгу постачання. Таким чином, впровадження технології блокчейн в медичну логістику може допомогти покращити якість, доступність та безпеку медичних засобів для всіх пацієнтів.

Мета кваліфікаційної роботи полягає у розробці та впровадженні системи управління ланцюгом постачання лікарських препаратів на основі технології блокчейн для забезпечення прозорості, безпеки та ефективності процесу постачання.

Об’єктом дослідження є ланцюг постачання лікарських препаратів,

включаючи виробництво, транспортування, зберігання та реалізацію ліків.

Предметом дослідження є застосування технології блокчейн для оптимізації та забезпечення безпеки ланцюга постачання лікарських препаратів.

Завдання кваліфікаційної роботи включають:

1. Аналіз існуючих підходів до організації постачання лікарських препаратів.
2. Вивчення принципів роботи технології блокчейн та її застосування у ланцюзі постачання.
3. Розробка архітектури системи управління ланцюгом постачання на основі блокчейну.
4. Впровадження смарт-контрактів для автоматизації процесів у ланцюзі постачання.
5. Розробка та інтеграція веб-додатку для управління ланцюгом постачання.
6. Оцінка ефективності та безпеки запропонованої системи.

Теоретична значущість роботи полягає у внеску в розвиток знань про застосування технології блокчейн у ланцюзі постачання лікарських препаратів. Результати дослідження можуть бути корисними для подальших наукових досліджень у галузі логістики та медичної індустрії.

Практична значущість роботи полягає у розробці та впровадженні ефективної системи управління ланцюгом постачання лікарських препаратів, що допоможе покращити якість, доступність та безпеку медичних засобів для пацієнтів.

У роботі використовуються наступні основні джерела інформації:

1. Наукові статті та публікації. Дослідження з області застосування технології блокчейн у ланцюзі постачання.
2. Книги та підручники. Література з логістики, управління ланцюгами постачання та технології блокчейн.
3. Інтернет-ресурси.

РОЗДІЛ 1
АНАЛІЗ ПІДХОДІВ ДО ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПРОЦЕСУ

1.1 Технологія блокчейну

1.1.1 Опис принципів роботи блокчейну

Блокчейн – це розподілена база даних, яка спільно використовується вузлами комп’ютерної мережі. Вони найбільш відомі своєю ключовою роллю в системах криптовалют для підтримки безпечного та децентралізованого запису транзакцій, але вони не обмежуються використанням криптовалюти. Блокчейни можна використовувати, щоб зробити дані в будь-якій галузі незмінними.

Оскільки неможливо змінити блок, єдина довіра потрібна до точки, де користувач або програма вводить дані. Цей аспект зменшує потребу в довірених третіх сторонах, якими зазвичай є аудитори чи інші люди, які і роблять помилки.

У звичайних базах даних чи електронних таблицях інформація зберігається централізовано, але у блокчейні вона розподілена між багатьма вузлами мережі. Кожен вузол мережі має копію всієї бази даних, що гарантує її безпеку та доступність. [1]

Сценарії, які виконуються у блокчейні, називаються "смарт-контрактами". Вони відповідають за обробку та зберігання інформації, а також за забезпечення консенсусу між учасниками мережі. Інформація про транзакції збирається у блоки, кожен з яких містить певну кількість даних.

Перед тим як додати блок до ланцюга, він проходить через процес шифрування за допомогою криптографічних алгоритмів. Цей процес генерації унікального хешу, який використовується для ідентифікації блоку та забезпечення його незмінності.

Хеш – це унікальний вихідний код, який генерується за допомогою

криптографічного алгоритму на основі даних у блоку. Він слугує як цифровий відбиток пальця блоку і використовується для перевірки цілісності та недоступності для змінення даних у блокчейні. Кожен блок містить посилання на попередній блок у ланцюгу, що створює послідовний порядок. Така структура гарантує недоступність для змінення даних у блокчейні без зміни всього ланцюга. Таким чином, блокчейн забезпечує надійність, цілісність та безпеку інформації, що зберігається у ньому.

Через децентралізовану природу блокчейну усі транзакції можна прозоро переглядати, маючи особистий вузол або використовуючи дослідники блокчейнів , які дозволяють будь-кому бачити транзакції, що відбуваються в реальному часі. Кожен вузол має власну копію ланцюга, яка оновлюється, коли підтверджуються та додаються нові блоки. [2]

Приклад принципу роботи блокчейна показано на рис. 1.1 де платіж відправляється від A до B, в той час як інші вузли перевіряють транзакцію. У разі збою чи визнання транзакції недійсною, транзакція не підтверджується. Зрештою, всі вузли перевірять і додадуть транзакцію до своєї копії реєстру. Концептуально вона працює шляхом з'єднання або об'єднання блоків інформації про транзакції та їх зберігання у хронологічному порядку і, отже, називається блокчейном. [3]

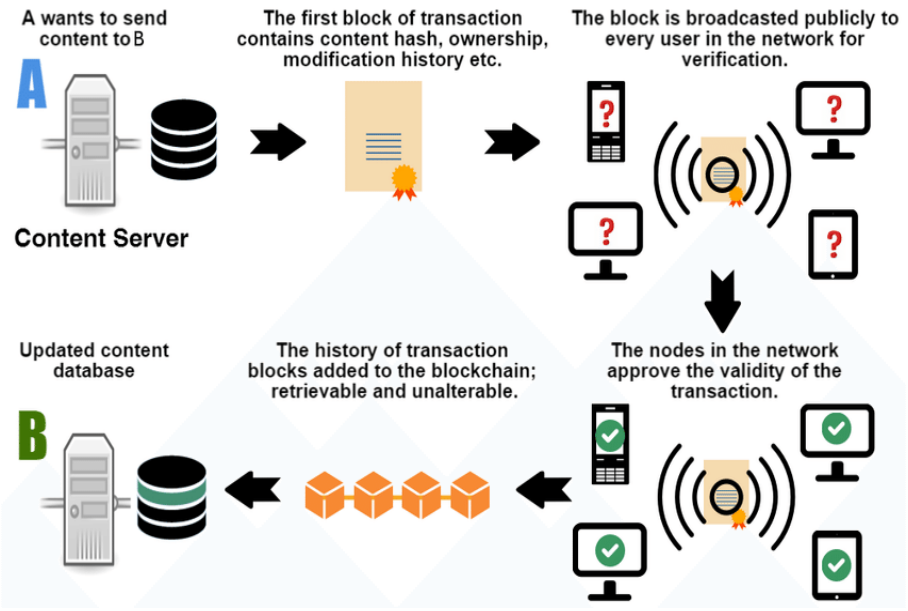


Рисунок 1.1 – Прицип роботи блокчейну

Переваги блокчейнів

– Точність ланцюга. Транзакції в мережі блокчейн схвалюються тисячами комп’ютерів і пристроїв. Це виключає майже всіх людей із процесу перевірки, що призводить до зменшення людських помилок і точного запису інформації. Навіть якби комп’ютер у мережі припустився обчислювальної помилки, помилка буде зроблена лише в одній копії блокчейну й не буде сприйнята рештою мережі.

– Зниження витрат. Як правило, споживачі платять за перевірку транзакції або нотаріусу за підписання документа. Blockchain усуває необхідність сторонньої перевірки, а разом з цим і пов’язані з нею витрати. Наприклад, власники бізнесу стягують невелику комісію, коли приймають платежі кредитною карткою, оскільки банки та компанії з обробки платежів мають обробляти ці транзакції.

– Децентралізація. Blockchain не зберігає свою інформацію в центральному місці. Натомість блокчейн копіюється та поширюється по мережі комп’ютерів. Щоразу, коли до блокчейну додається новий блок, кожен комп’ютер у мережі оновлює свій блокчейн, щоб відобразити зміни. Завдяки поширенню цієї інформації по мережі, а не зберіганню в одній центральній базі даних, блокчейн стає важче підробити.

– Ефективні транзакції. Розрахунок транзакцій, здійснених через центральний орган, може зайняти кілька днів. У деяких блокчейнах транзакції можуть бути завершені за лічені хвилини і вважаються безпечними. Це особливо корисно для транскордонних торгів, які зазвичай займають набагато більше часу через проблеми з часовим поясом і той факт, що всі сторони повинні підтвердити обробку платежу.

– Приватні транзакції. Багато блокчейн-мереж працюють як загальнодоступні бази даних, тобто кожен, хто має підключення до Інтернету, може переглядати список історії транзакцій мережі. Хоча користувачі можуть отримати доступ до деталей транзакцій, вони не можуть отримати доступ до ідентифікаційної інформації про користувачів, які здійснюють ці транзакції.

Поширеною є помилкова думка, що блокчейн-мережі, такі як біткойн, єповністю анонімними; вони фактично псевдонімні, тому що є видима адреса, яка може бути пов’язана з користувачем, якщо інформація стане доступною.

– Безпечні транзакції. Після реєстрації транзакції її автентичність повинна бути перевірена мережею блокчейн. Після підтвердження транзакції вона додається до блоку блокчейну. Кожен блок у блокчейні містить свій унікальний хеш і унікальний хеш блоку перед ним. Таким чином, блоки не можуть бути змінені після підтвердження мережею.

– Прозорість. Більшість блокчейнів є повністю відкритим програмним забезпеченням. Це означає, що кожен може переглянути його код.

Недоліки блокченів

– Вартість технології. Хоча блокчейн може заощадити гроші користувачів на комісії за транзакції, ця технологія далеко не безкоштовна. Наприклад, система підтвердження роботи мережі Bitcoin для підтвердження транзакцій споживає величезну кількість обчислювальної потужності.

– Незаконна діяльність. Хоча конфіденційність у мережі блокчейн захищає користувачів від злому та зберігає конфіденційність, вона також дозволяє здійснювати незаконну торгівлю та діяльність у мережі блокчейн. Найбільш цитованим прикладом використання блокчейну для незаконних транзакцій є, ймовірно, Silk Road , ринок нелегальних наркотиків і відмивання грошей у дарк нет.

Отже, цю систему можна розглядати як плюси, так і мінуси. Це дає будь-кому доступ до фінансових рахунків, але дозволяє злочинцям легше здійснювати операції. Багато хто стверджує, що хороше використання криптовалюти, як-от банківська справа в безбанківському світі, переважує зловмисне використання криптовалюти, особливо коли більшість незаконної діяльності все ще здійснюється за допомогою готівки, яку неможливо відстежити. [4]

1.1.2 Огляд типів блокчейнів, їх особливості, переваги та недоліки

Існують кілька типів блокчейнів, кожен з яких має свої особливості та застосування.

Публічний блокчейн (Public Blockchain):

- Це розподілений реєстр без дозволів, якого кожен може приєднатися і проводити транзакції.
- Це необмежена форма реєстру, в якій кожен вузл має копію. Це також означає, що будь-хто, хто має підключення до Інтернету, може отримати доступ до публічного блокчейну.
- Цей користувач має доступ до історичних та сучасних записів та можливість виконувати операції з видобутку корисних копалин.
- Ці складні обчислення необхідно виконати для перевірки транзакцій та додавання в реєстр.
- У мережі блокчейна ніякий дійсний запис або транзакція не може бути змінено. Оскільки вихідний код зазвичай відкритий, кожен може перевірити транзакції, виявити проблеми та запропонувати виправлення.

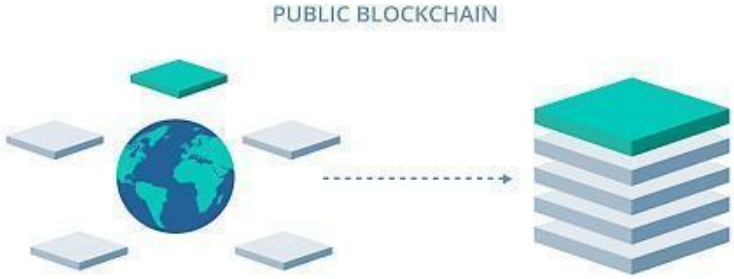


Рисунок 1.2 – Публічний блокчейн

Переваги публічного блокчейну :

1. Надійність: вузлам публічного блокчейну не потрібно знати чи довіряти один одному, оскільки процедура доказу роботи гарантує відсутність шахрайських транзакцій.Безпека: загальнодоступна мережа може мати стільки учасників або вузлів, скільки забажає, що робить її безпечною мережею. Чим більший розмір мережі, тим більше записів поширюється і тим складніше хакерам зламати всю мережу.
2. Відкритість та прозорість: дані у загальнодоступному ланцюжку

блоків прозорі для всіх вузлів-членів. Кожен авторизований вузол має копію записів блокчейна чи цифрового реєстру.

Недоліки публічного блокчейну:

– Нижчий TPS: кількість транзакцій на секунду в загальнодоступному блокчейні надзвичайно низька. Це пов'язано з тим, що це велика мережа з безліччю вузлів, які потребують часу для перевірки транзакції та виконання доказу роботи.

– Проблеми масштабованості: транзакції обробляються та завершуються повільно. Це шкодить масштабованості. Тому що чим більше ми намагаємося розширити розмір мережі, тим повільніше вона ставатиме.

– Високе енергоспоживання. Пристрій перевірки працездатності є дорогим і вимагає багато енергії. Технології, безперечно, вимагатимуть розробки енергоефективних консенсусних методів.

Приватний блокчейн (Private Blockchain):

- Мережа блокчейна працює у приватному контексті, наприклад, у мережі з обмеженим доступом, або контролюється одним ідентифікатором.

- Незважаючи на те, що він має таку ж однорангову сполуку та децентралізацію, що й публічна мережа блокчейнів, цей блокчейн набагато менший.

- Вони часто працюють у невеликій мережі всередині фірми чи організації, а не відкриті для всіх, хто хоче зробити свій внесок у обчислювальну потужність.

- Дозволені блокчейни та бізнес-блокчейни - це ще два терміни для них.

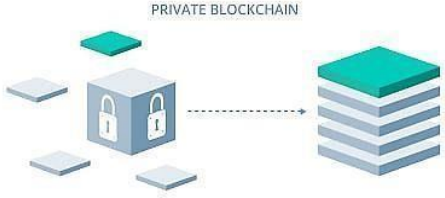


Рисунок 1.3 – Приватний блокчейн

Переваги приватного блокчейну:

– Швидкість: транзакції приватного блокчейну виконуються швидше. Це пов'язано з тим, що приватна мережа має меншу кількість вузлів, що скорочує час, необхідний для перевірки транзакції.

– Масштабованість: ви можете адаптувати розмір вашого приватного блокчейну відповідно до ваших конкретних вимог. Це робить приватні блокчейни, що особливо масштабуються, оскільки вони дозволяють компаніям легко збільшувати або зменшувати розмір своєї мережі.

Недоліки приватного блокчейну:

1. Побудова довіри. У приватній мережі менше учасників, ніж у приватній мережі.

2. Нижча безпека: приватна мережа блокчейнів має менше вузлів або учасників, тому вона є більш вразливою для порушення безпеки.

3. Централізація. Приватні блокчейни обмежені тим, що для їх функціонування потрібна центральна система керування ідентифікацією та доступом (IAM). Ця система надає повні можливості адміністрування та моніторингу.

Гібридний блокчейн (Hybrid Blockchain):

- Організації, які очікують найкращого з обох світів, вибирають гібридний блокчейн, який поєднує в собі функції як приватних, так і публічних блокчейнів.

- Це дозволяє підприємству створювати приватну систему разом на основі дозволів із публічною системою без дозволів, дозволяючи їм вибирати, хто має доступ до певних даних Blockchain і які дані опрацьовуються.

- У гібридному блокчейні транзакції та записи традиційно не оприлюднюються, але їх можна перевірити, обов’язково, за умови надання доступу через смарт-контракт.

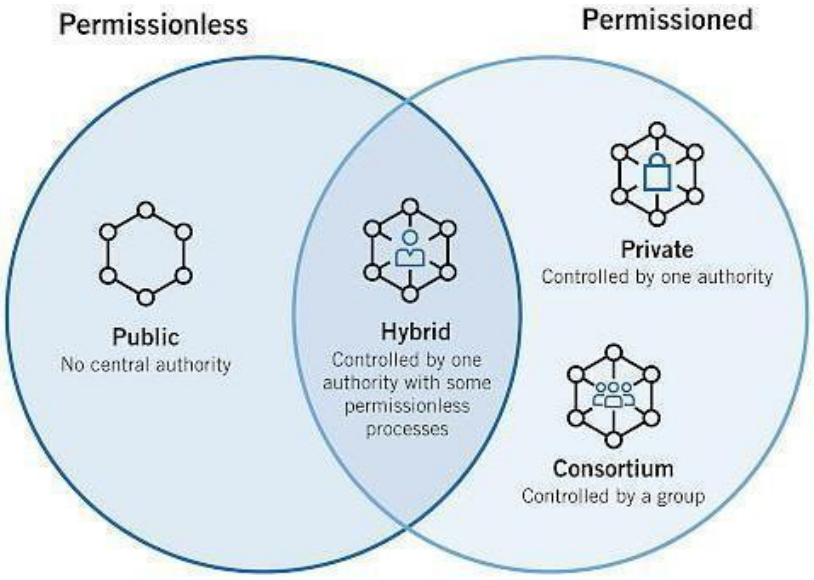


Рисунок 1.4 – Гібридний блокчейн

Переваги гібридного блокчейна:

1. Безпека: Hybrid Blockchain працює в закритому середовищі, не даючи стороннім хакерам здійснити 51-відсоткову атаку в мережі.

2. Економічно: це також захищає конфіденційність, дозволяючи стороннім контактам. Транзакції недорогі, швидкі та масштабуються краще, ніж публічна мережа блокчейн.

Недоліки гібридного блокчейна:

1. Відсутність прозорості: одну інформацію можна приховати, цей тип блокчейна не є повністю прозорим.

2. Менше стимулів: Оновлення може бути важким, і користувачі не мають стимулів брати участь у мережі чи робити свій внесок у неї.

Консорціум блокчейн (Consortium Blockchain):

- Так само, як гібридний блокчейн має як приватні, так і загальнодоступні функції блокчейну, блокчейн Консорціуму, також відомий як об’єднаний блокчейн, має.

- Однак він відрізняється тим, що в ньому беруть участь різні члени організації, які працюють разом у децентралізованій мережі.

- Попередньо визначені вузли контролюють методи консенсусу в блокчейні консорціуму.

- Він має вузол перевірки, відповідальний за ініціювання, отримання та перевірку транзакцій. Транзакції можуть бути ініційовані або отримані вузлами- учасниками.

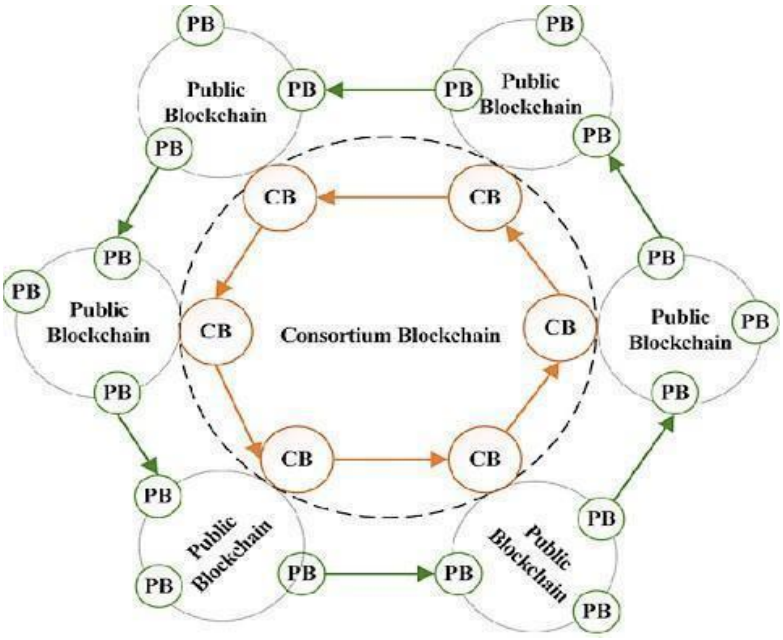


Рисунок 1.5 – Консорціум блокчейн

Переваги Consortium Blockchain:

Безпека: Блокчейн консорціуму є більш безпечним, масштабованим і ефективнішим, ніж публічна мережа блокчейну. Він, як і приватні та змішані блокчейни, має контроль доступу.

Недоліки Consortium Blockchain:

Відсутність прозорості: блокчейн консорціуму має нижчий ступінь прозорості. Якщо вузол-учасник проникнутий, його все одно можна зламати, а правила Blockchain можуть зробити мережу непрацездатною. [5]

1.2 Аналіз ланцюга постачання медичних препаратів

Ланцюг постачання медичних препаратів є життєвоважливою системою,

що забезпечує безперебійне та ефективне постачання лікарських засобів від виробника до кінцевого користувача. Цей складний процес включає в себе низку ключових етапів, кожен з яких відіграє важливу роль у забезпеченні якості, безпеки та доступності медичних препаратів для пацієнтів у всьому світі. Від підготовки та виробництва препаратів до їх доставки та зберігання, кожен етап цього процесу має велике значення для здоров'я та благополуччя пацієнтів. Тільки завдяки добре організованому та ефективному ланцюгу постачання медичних препаратів ми можемо мати впевненість у тому, що надійні та ефективні ліки доступні для всіх, хто потребує їх, коли це необхідно.

1.2.1 Опис ланцюга постачання медичних препаратів

Нижче розглянемо детально структуру та основні компоненти холодного ланцюга постачання медичних препаратів, як складного випадку ланцюга, щоб краще зрозуміти цей процес та можливості його оптимізації (рис. 1.6).

На кожному етапі "холодного ланцюга" необхідно обов'язково проводити реєстрацію в журналах обліку отримання, умов зберігання та переміщення лікарських препаратів до кінцевого споживача. Ця реєстрація повинна включати торгову назву продукту, кількість доз, номер партії, термін придатності, дату отримання, умови зберігання і транспортування, а також дані про контрольні карточки-індикатори, індикатори заморожування або реєструючі пристрої, а також ім'я відповідальної особи. Результати тестів температури, отримані за допомогою термотестерів, термореєстраторів і термографів у вигляді графіків і таблиць, також мають бути збережені разом з журналом реєстрації температур для подальшої звітності. [6]

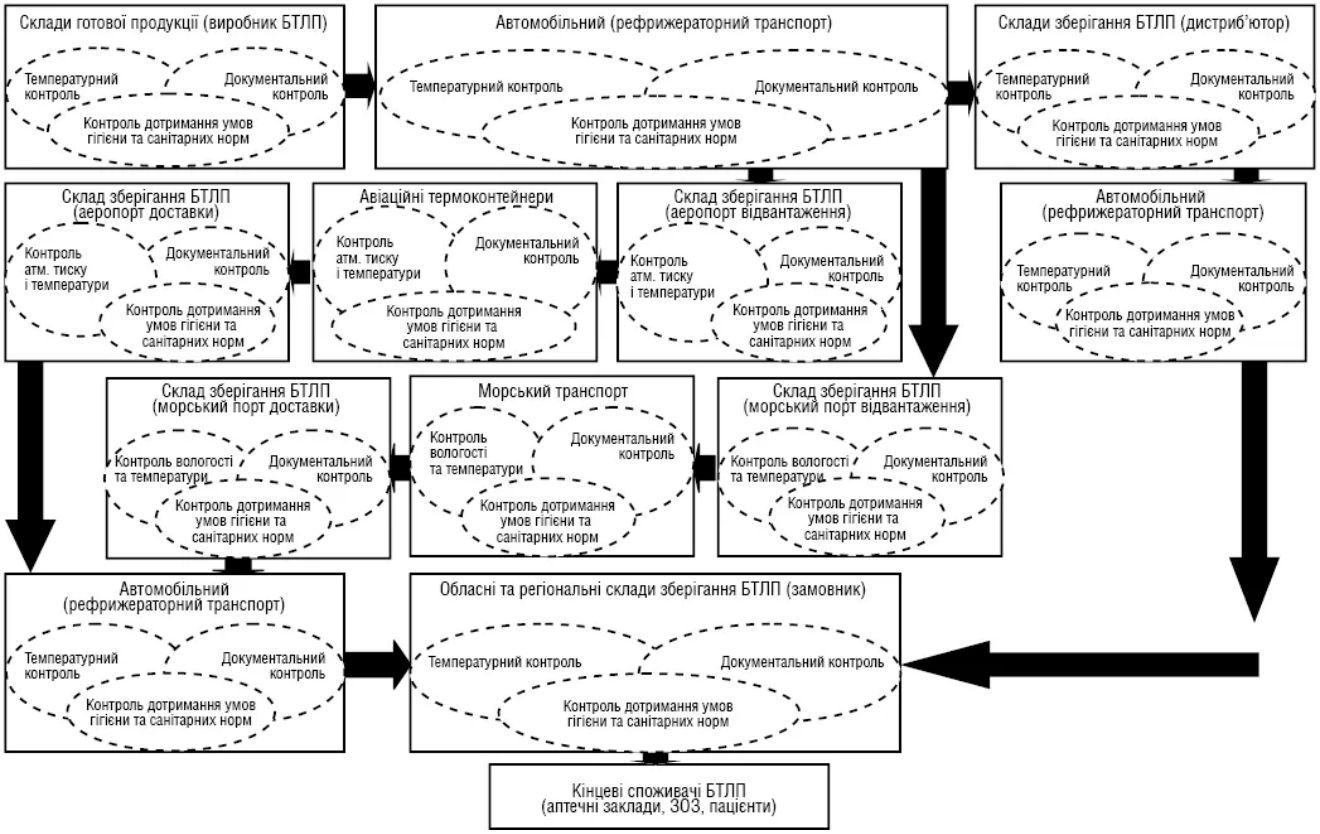


Рисунок 1.6 – Схема інтегрованого «холодного ланцюга»

Ефективне функціонування інтегрованого "холодового ланцюга" при постачанні біотехнологічних лікарських препаратів передбачає наявність ряду важливих складових.[7]

Ефективне створення інтегрованого "холодового ланцюга" для належного постачання біотехнологічних лікарських препаратів включає дотримання ряду важливих правил: 1) приймання лікарських препаратів для доставки здійснюється лише після отримання сертифікатів їх якості та моніторингу температурних умов зберігання, який підтверджує, що препарати зберігалися за необхідних умов для отримання дозволу на їх використання; 2) температурний моніторинг якості лікарських препаратів проводиться з моменту отримання згоди на їх відвантаження до підписання актів передачі-приймання в пунктах призначення; 3) температурний моніторинг якості препаратів повинен забезпечуватися в не менш ніж у двох точках контрольованих обсягів за допомогою сертифікованих електронних термореєстраторів;4) доставка лікарських препаратів повинна здійснюватися

виключно в сертифікованих термоконтейнерах, дозволених для застосування відповідними установами; 5) передача препаратів здійснюється за транспортними накладними з друкованою інформацією щодо моніторингу температурного режиму; 6) препарати зберігаються у спеціальних холодових кімнатах з обов'язковою системою контролю та забезпеченням відповідних температурних умов зберігання; 7) дії щодо забезпечення якості препаратів повинні відповідати програмі санітарногоконтролю та програмі дезінсекції та дератизації; 8) препарати повинні проходити сертифікаційні дослідження перед обігом в обов'язковому порядку; 9) технічні засоби доставки та моніторингу температури повинні бути сертифіковані згідно з рекомендаціями міжнародних організацій; 10) нормативні документи повинні відповідати вимогам єдиного нормативного документа, обов'язкового для всіх учасників інтегрованого фармацевтичного ланцюга постачання; 11) на всіх рівнях "холодового ланцюга" повинні діяти уповноважені особи з чітко визначеними обов'язками; 12) інтегровані "холодові ланцюги" повинні відповідати міжнародним стандартам якості.[8]

1.2.2 Проблеми та виклики, з якими стикається сучасний ланцюг постачання медичних препаратів

Фармацевтичний ланцюг постачання стикається з декількома основними викликами, які впливають на його ефективність, надійність та здатність задовольняти потреби пацієнтів. Серед цих викликів можна виділити наступне:

- Відповідність регулятивним вимогам. Фармацевтична індустрія функціонує в складному регулятивному середовищі. Дотримання правил та стандартів якості, таких як ефективні виробничі практики, ефективні практики дистрибуції та різноманітні національні регуляції, є обов'язковими.
- Фрагментована та глобалізована мережа. Фармацевтичний ланцюг постачання часто є фрагментованим, включаючи численних учасників, таких як постачальники, виробники, дистриб'ютори, оптовики та

роздрібні торговці, в різних географічних регіонах.

- Логістика холодного ланцюга. Багато фармацевтичних продуктів, включаючи вакцини, біологічні препарати та деякі ліки, вимагають певних температурних умов для збереження їх ефективності. Управління логістикою холодного ланцюга, включаючи зберігання, транспортування та моніторинг при контрольованих температурних умовах, ускладнює ланцюг постачання.

- Підробка продукції. Підроблені ліки становлять значний ризик для безпеки пацієнтів та репутації фармацевтичних компаній. Виявлення та запобігання потраплянню підроблених продуктів у ланцюг постачання є постійним викликом.

- Прогнозування попиту та управління запасами. Точне прогнозування попиту є критичним для уникнення дефіциту товарів або перебору запасів. Прогнозування попиту на фармацевтичні продукти може бути складним через такі фактори, як зміна потреб пацієнтів, вибухи захворювань та інші.

- Видимість та відстежуваність у ланцюзі постачання. Забезпечення видимості та відстежуваності на всіх етапах ланцюга постачання важливо для ідентифікації та вирішення проблем, забезпечення якості продукту та проведення вилучень, якщо це необхідно.

- Управління постачальниками та джерелами сировини. Надійність та якість сировини мають прямий вплив на якість кінцевої фармацевтичної продукції.

- Фінансовий тиск. Фармацевтичні компанії стикаються з зростаючим фінансовим тиском через такі фактори, як зростання витрат на дослідження та розробку, ціновий тиск та реформи у галузі охорони здоров'я. [9]

1.3 Огляд існуючих підходів до організації постачання лікарських препаратів

Існують різноманітні підходи до організації постачання лікарських препаратів, які охоплюють широкий спектр методів та технологій. Ці підходи відрізняються застосуванням різних стратегій та інноваційних підходів з метою поліпшення ефективності, безпеки та доступності лікарських засобів для пацієнтів.

Найбільш звичними та розповсюдженими є традиційні підходи. Наприклад, Гударзян та його колеги розробили Sustainable Medical Supply Chain Network (SMSCN) - модель, яка є стійкою, цілочисельною та лінійною (рис. 1.7). Ця модель базується на програмуванні та призначена для визначення оптимального розташування розподільних центрів, управління запасами та мінімізації витрат на всьому ланцюгу поставок у період пандемії. Також, Khan та співавтори провели дослідження щодо впливу COVID-19 на ланцюг поставок, що може бути корисним для формулювання стратегій боротьби з пандемією на рівні розробки політики. [10]

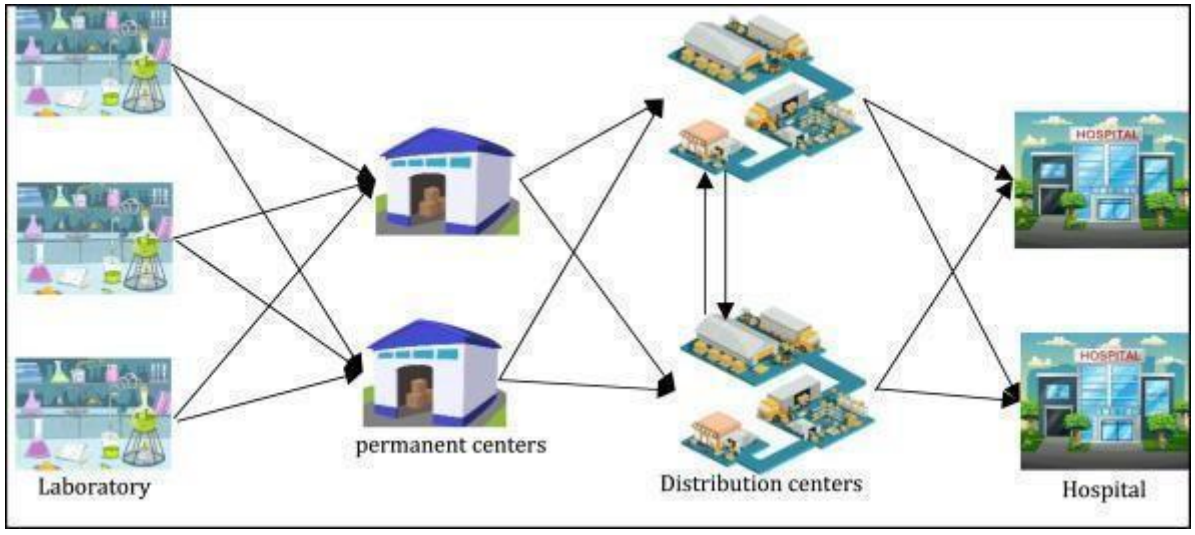


Рисунок 1.7 – Sustainable Medical Supply Chain Network

Інтегрована стійка мережа ланцюжків постачання медичних продуктів під час Covid-19 використовує різноманітні технології для забезпечення ефективності та надійності:

1. Технології IIoT (промислових Інтернет речей):

-Надають цінну інформацію про ланцюжки поставок.

- Забезпечують контроль даних та прозорість у режимі реального часу.
- Дозволяють оптимізувати виробництво та скоротити час.

Прогнозна аналітика:

- Допомагає передбачити потенційні проблеми до їхнього виникнення.
- Дозволяє перейти від планового обслуговування до обслуговування обладнання за станом.

2. Управління та оптимізація активів:

- Використовує дані дляпокращення управління ланцюжком постачання.
- Допомагає мінімізувати відходи та підвищити ефективність.

3. Індустрія 4.0:

- Впровадження мережі інтелектуальних процесів та обладнання у промисловість.
- Створення ефективної екосистеми, що поєднує виробництво та управління ланцюжками постачання.

Ці технології допомагають забезпечити стійкість та надійність ланцюжка постачання медичних продуктів в умовах пандемії.

Наступним є підходи до підробки ліків на основі штучного інтелекту. Технологічний прогрес, включаючи Інтернет речей, хмарні обчислення та штучний інтелект, розкриває нові перспективи у медичному ланцюгу поставок. Корпела та його колеги провели дослідження, аналізуючи дані про різні компанії, щоб визначити значення взаємодії та інтеграції в цифровий ланцюг поставок для різних організацій та систем.

Один з методів, яким можна продемонструвати користь від штучного інтелекту у галузі охорони здоров'я, полягає у використанні мобільних додатків для перевірки та відстеження кожної капсули за допомогою тривимірних флуоресцентних QR-кодів. Ці додатки, доступні для Android, часто дозволяють користувачам перевірити автентичність продукту,

звернувшись до місцевих урядових органів. Також існує структура, яка використовує матрицю даних для виправлення поточних недоліків у вирішенні проблем. За допомогою RFID та алгоритмів аналізу даних полегшено виявлення підроблених ліків. Крім того, методи обробки зображень можуть ідентифікувати підроблені ліки, застосовуючи теплові карти, машини опорних векторів та інші класифікатори для успішної класифікації відповідних областей.

Однак ці рішення можуть стикатися з проблемою недостатньої сумісності та масштабованості через використання різноманітних централізованих баз даних.

Підходи на основі блокчейну. Поточні методи забезпечення відстеження ланцюга постачання ліків часто базуються на централізованих системах, які не завжди забезпечують прозорість серед всіх учасників. Це означає, що централізовані органи можуть змінювати інформацію без попереднього повідомлення інших сторін. З іншого боку, система на основі блокчейну гарантує безпеку даних, прозорість, незмінність, походження та автентичність всіх записів транзакцій. Результати онлайн-опитування свідчать, що більшість клієнтів сприймають цю нову технологію блокчейну дуже позитивно, особливо в контексті медичних поставок.

Дослідники, зокрема Roberto Casado-Vara, активно досліджують вплив блокчейн на різні сфери, включаючи циркулярну економіку. В їхніх роботах обговорюються можливості використання блокчейн для покращення безпеки, прозорості та ефективності в управлінні ланчюгами поставок. Це може включати в себе відстеження походження продуктів, контроль якості, оптимізацію логістики та багато іншого. Такий підхід може сприяти створенню більш стійких та ефективних систем управління ресурсами та продуктами. На рис. 1.7 наведено ланцюг поставок через блокчейн-архітектуру MAS. Кожен рівень надсилає дані своїх транзакцій у блокчейн. Крім того, шари спілкуються один з одним за допомогою смарт-контракту.

Ці контракти призначені для купівлі та продажу товарів. [11]

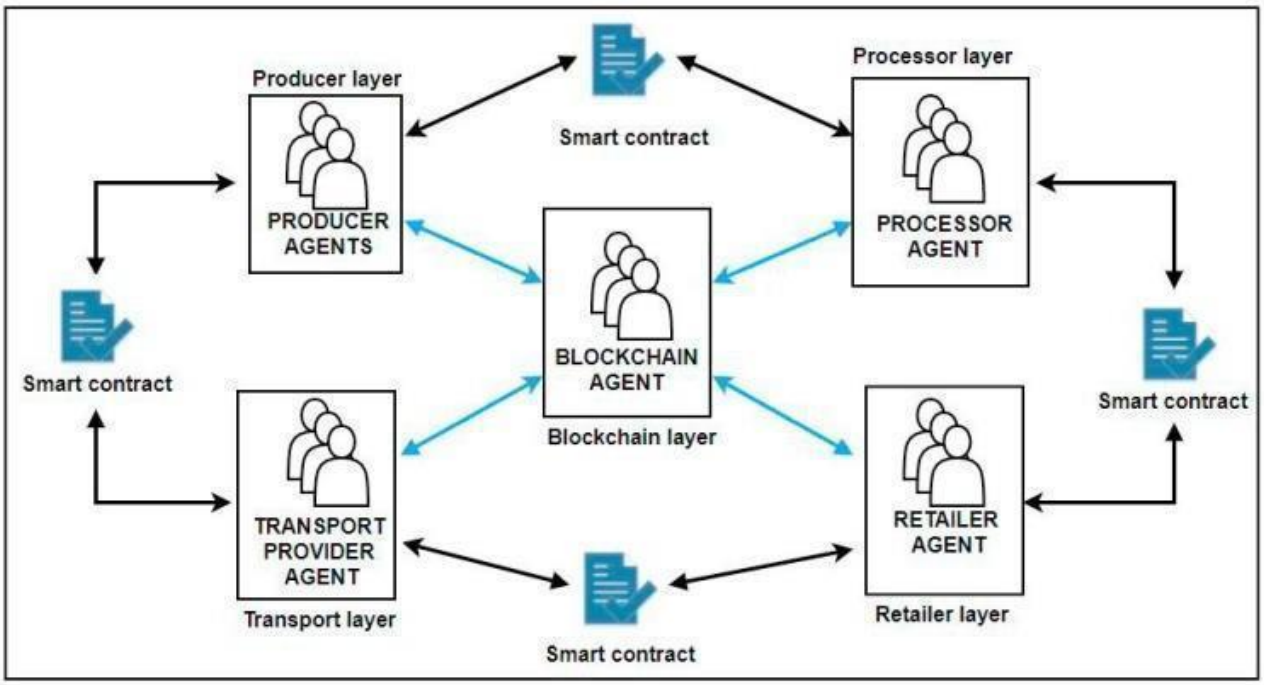


Рисунок 1.7 – Ланцюг поставок через блокчейн-архітектуру MAS

Учасниками ланцюгів постачання ліків є виробники, оптові та роздрібні торговці, аптеки, лікарні та споживачі (споживачі наркотиків). Що стосується ієрархії в системі Gcoin, державні органи повинні стежити за транзакціями та інформацією про ліки, і їм рекомендується взяти на себе роль члена альянсу в системі блокчейну Gcoin. Оскільки виробники ліків є джерелом ліків, визначених системою блокчейну Gcoin, вони повинні взяти на себе роль емітента монет (карбувальника). Рекомендується, щоб майнери, які відповідають за перевірку транзакцій і генерацію блоків, були великими виробниками та державними установами. Решта великих оптовиків, лікарень або третіх осіб можуть бути повноцінними вузлами, відповідальними за зберігання резервної копії історичних транзакцій. Крім того, інші аптеки та споживачі мають бути звичайним вузлом (Wallet), який має повноваження здійснювати транзакції.

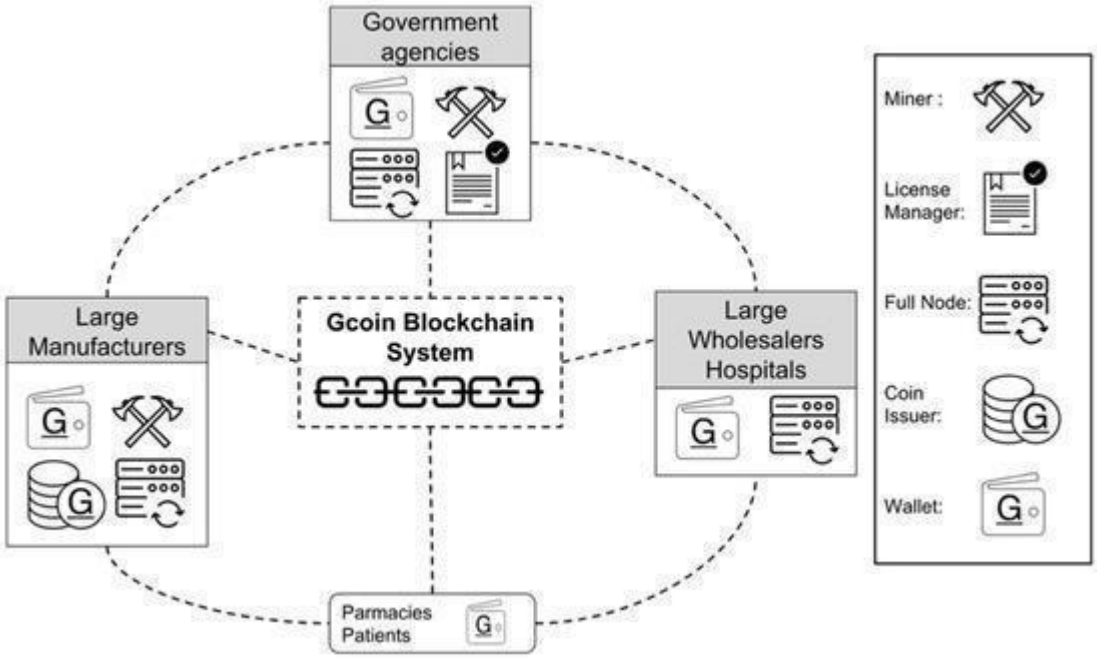


Рисунок 1.8 – Структура системи та ролі учасників.

Персонал, відповідальний за доставку, здійснення транзакцій та перевірку ліків у ланцюжку постачання ліків, отримає відповідні повноваження підписувати ідентичні цифрові підписи під час своєї робочої процедури. Пропонується використовувати дизайн мультипідпису блокчейну Gcoin, який підтверджено успішною заявкою в «Довідковому центрі NTU». [12]

Що стосується ланцюга постачання ліків зверху вниз (від виробників ліків до споживачів), виробники передають дані про свої транзакції безпосередньо одержувачам ліків, і ці дані записуються в блокчейні Gcoin. Дані транзакцій цифрових підписів продавця та покупця ліків, інформація про препарат (включаючи позначку часу, місцезнаходження, назву товару тощо) та кількість ліків перевіряються в ланцюжку. Потім усі ці дані хешуються як дайджест для запису в блокчейн Gcoin (рис. 1.8). Щоразу, коли нелегальний дистриб’ютор хоче продати покупцям підроблені ліки (із «підробленим» ідентифікатором препарату, згаданим вище), транзакція буде визнана недійсною через наявність шахрайської інформації про вихідні дані невитрачених транзакцій (UTXO), що зберігається в блокчейні Gcoin. З

іншого боку, неавторизований персонал не може здійснювати операції з наркотиками в цій системі без правильногозакритого ключа. Отже, покупець/продавець буде негайно знати про будь-які аномалії в транзакціях.

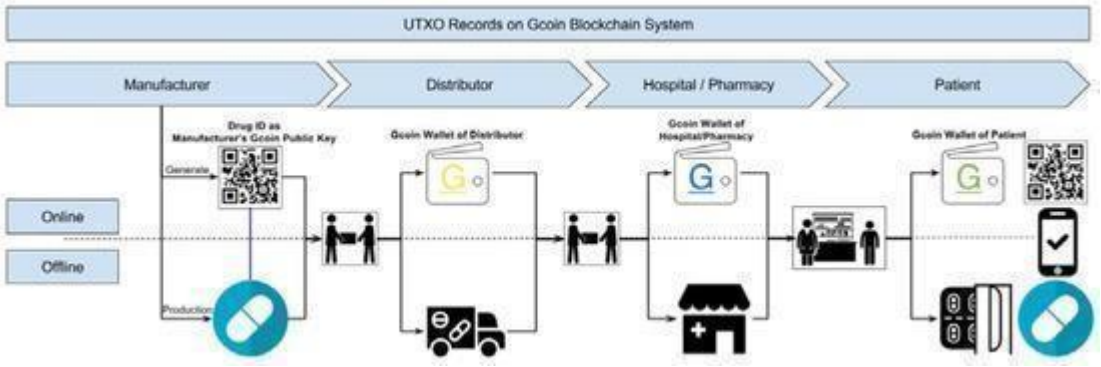


Рисунок 1.8 – Робочий процес блокчейн-системи Gcoin із застосуванням до ланцюжка постачання ліків

Комбіновані підходи. Існують підходи, коли традиційний метод та технологія блокчейн співпрацюють для розробки та полегшення прозорого ланцюга поставок медичних засобів з метою запобігання підробці ліків. З метою підвищення безпеки в ланцюгу поставок лікарських засобів, Schöner та його колеги пропонують програму на базі блокчейну, яка дозволяє відстежувати кожну медичну позицію від її торгової історії до моменту розподілу.Шляхом прикріплення унікальної ідентифікаційної мітки до ліків, вони забезпечують відстеження їх власності як віртуально, так і фізично, і в кінцевому підсумку перевіряють за допомогою смарт-контрактів. Також, для відстеження платформ постачання лікарських засобів, Liu та його колеги розробили структуру, що поєднує технологію блокчейн та Internet of Things (IoT). Вони використовують засоби поза ланцюгом та в самому ланцюзі для зберігання даних та смарт-контрактів з метою забезпечення відстеження. [13]

Приділимо більше уваги методології PharmaChain

Основна технологічна структура рішення ґрунтується на платформі Hyperledger Fabric, що забезпечує надійність автентифікації, масштабованість та можливість обміну важливою інформацією. Розробники можуть розширювати можливості за допомогою модулярної архітектури Fabric. Далі,

щодо каналів, служб замовлення, пірингових вузлів та центрів сертифікації, автори детальнорозглянули мережеву структуру цієї платформи, розділену на фармацевтичні компанії, управління з регулювання лікарських засобів і місцеві постачальники, такі як лікарні, аптеки і т. д., ресурси в ланцюжку використовуються для реєстрації журналів та подій, згенерованих смарт-контрактами, що забезпечує відстеження та контроль. Крім того, система реєстрації та ідентифікації використовується як автономний ресурс для інтеграції децентралізованого зберігання ідентифікаційних адрес різних зацікавлених сторін. Також було впроваджено три модулі управління, які включають участь у мережі, технічну інфраструктуру та бізнес-мережу, з метою підтримки координації між зацікавленими сторонами, формалізації та децентралізації правил, покращеного моніторингу та автоматизації процесів.

Для аналізу результатів генеруються дані у форматі .json окремого продукту, а потім зберігаються у спільній базі в CouchDB. З міркувань безпеки створили хеш-значення кожного фрагмента даних за допомогою алгоритму SHA-256. Для шифрування хеш-значення згенерували дві пари еліптичних ключів, включаючи приватний і відкритий ключі для фармацевтичної компанії та регуляторного органу з лікарських засобів, використовуючи алгоритм цифрового підпису еліптичної кривої. Автори вибрали приватні ключі, згенеровані за допомогою криптографії на основі еліптичної кривої (ECC), для цих двох зацікавлених сторін, а потім сформували для них підписаний документ. Нарешті, хеш-значення можна перевірити, розшифрувавши підписаний документ за допомогою відкритого ключа фармацевтичного підприємства або регуляторного органу. Щоб перевірити чіткість цього відновленого хешу, його можна порівняти з оригінальним хешем у глобальному ланцюжку.

На рис. 1.9 зображено запропоновану високорівневу архітектуру системи відстеження лікарських засобів, включаючи учасників та їхні дії з розумним контрактом. Через веб-інтерфейси API зацікавлені сторони отримають доступ до інтелектуальних контрактів і ресурсів

децентралізованого зберігання (в ланцюжку та поза ним). Журнали, хешовані дані та транзакції генеруватимуться на основі взаємодії з ресурсами в мережі та за її межами. [14]

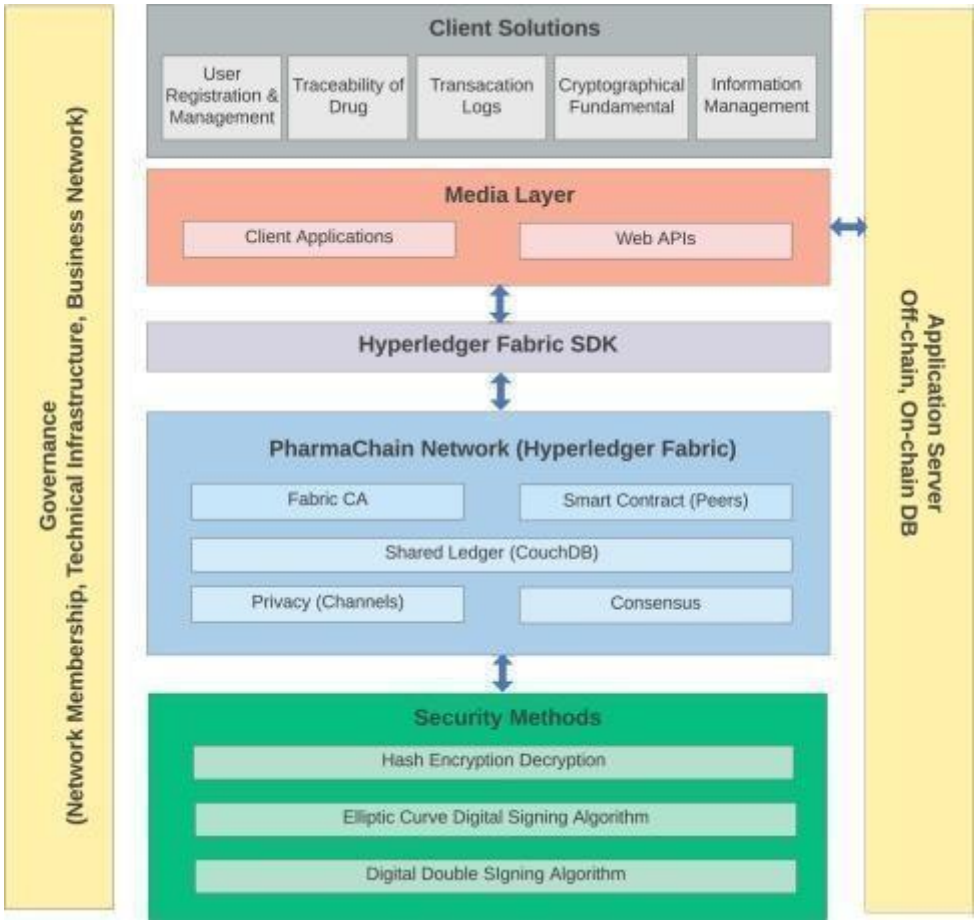


Рисунок 1.9 – Високорівнева архітектура PharmaChain

Висновки до розділу:

У розділі 1 кваліфікаційної роботи було проведено дослідження ключових аспектів технології блокчейн та її застосування в управлінні ланцюгами постачання лікарських препаратів. Основні висновки розділу:

1. Технологія блокчейн є революційним підходом до збереження та обміну даними, який базується на децентралізованості, прозорості та незмінності записів. Було розглянуто її принципи роботи, включаючи механізми консенсусу, криптографічний захист і структуру розподіленого реєстру, які забезпечують безпеку даних і довіру серед учасників.

2. Огляд типів блокчейнів (публічні, приватні та консорціумні) показав їх особливості, переваги та недоліки. Для сфери постачання лікарських препаратів найбільш перспективними виявилися приватні та консорціумні блокчейни, які поєднують контроль доступу з високою ефективністю.

3. Аналіз ланцюга постачання медичних препаратів продемонстрував його складність і багатоступеневість, включаючи виробництво, транспортування, зберігання та реалізацію препаратів. Ключова мета цього процесу – забезпечення якості, безпеки та доступності ліків для кінцевого споживача.

4. Основними проблемами сучасних ланцюгів постачання є відсутність прозорості, ризик підробки ліків, затримки в постачанні та неефективна інтеграція між учасниками. Такі виклики потребують інноваційних підходів для їх подолання.

5. У рамках дослідження було розглянуто сучасні методи оптимізації ланцюгів постачання, включаючи традиційні, штучний інтелект, Інтернет речей, блокчейн і комбіновані підходи. Особливу увагу приділено методології PharmaChain, яка базується на платформі Hyperledger Fabric і демонструє високу ефективність у забезпеченні прозорості та запобіганні підробці ліків.

РОЗДІЛ 2
ОСНОВНІ КОМПОНЕНТИ ТА СТРУКТУРА БЛОКЧЕЙН-СИСТЕМИ ДЛЯ ЛАНЦЮГА ПОСТАЧАННЯ МЕДИЧНИХ ПРЕПАРАТІВ

2.1 Структура ланцюга постачання лікарських препаратів

На основі даних, проаналізованих у першому пункті роботи було розроблено структуру ланцюга постачання лікарських засобів, яка складається з кількох ключових етапів: від постачання сировини до доставки готових препаратів кінцевим споживачам. Кожен етап цього ланцюга є важливим для забезпечення якості та безпеки лікарських засобів. Особливу увагу приділимо тому, як саме на кожному етапі можуть виникати ризики фальсифікації, підробки або маніпуляцій із даними, і яким чином технологія блокчейн здатна усунути ці проблеми, забезпечуючи прозорість та надійний контроль.

Нижче детально проаналізуємо кожен етап ланцюга постачання (рис. 2.1) та вкажемо, як саме технологія блокчейн може запобігти можливим ризикам і порушенням на різних стадіях цього процесу. [15]

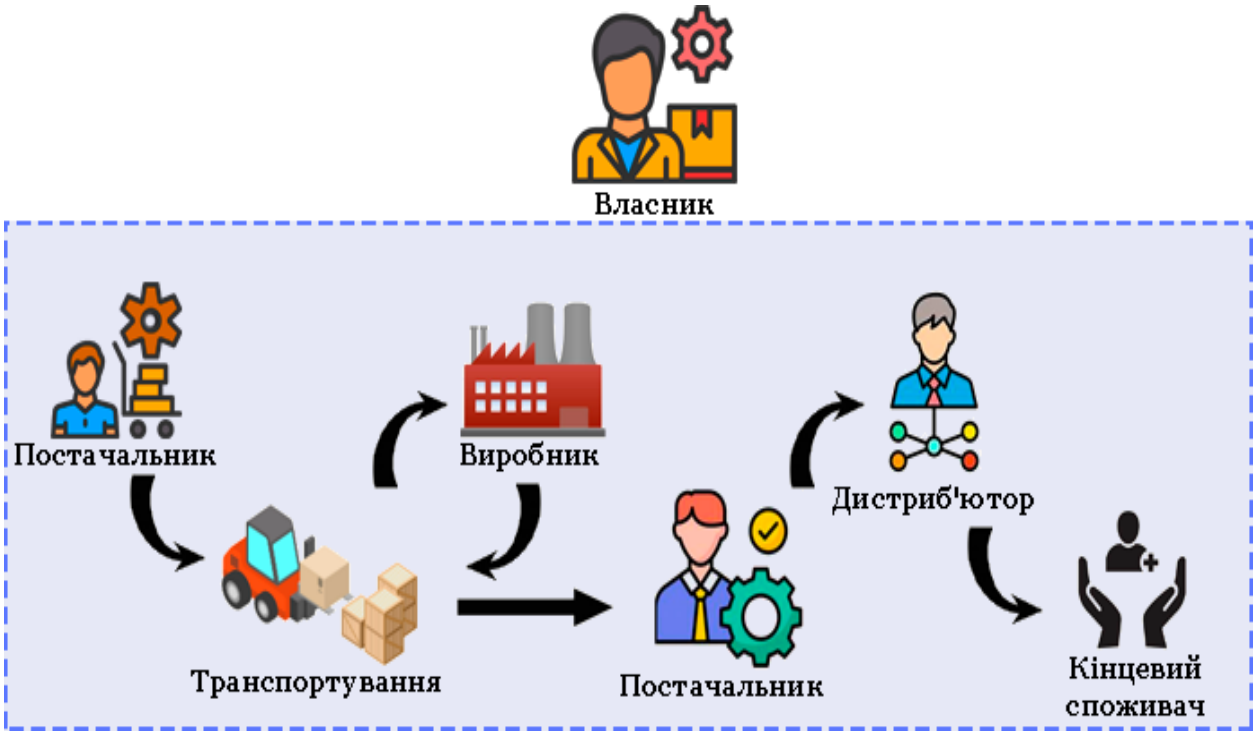


Рисунок 2.1 – Етапи ланцюга постачання лікарських препаратів

Основні етапи ланцюга постачання лікарських препаратів:

Власник (Owner). Власником може бути фармацевтична компанія або інший орган, відповідальний за розробку та ініціалізацію виробництва лікарських препаратів. Саме власник визначає специфікації препаратів, відповідає за дослідження та їх реєстрацію. На цьому етапі питання фальсифікації або маніпуляцій із властивостями препаратів є критичними, оскільки невідповідність складу може призвести до небезпечних наслідків для здоров'я пацієнтів.

Ризики: Порушення прав на інтелектуальну власність, маніпуляції з даними щодо складу препаратів.

Постачальник (Supplier). Постачальники надають сировину, необхідну для виготовлення лікарських засобів. Якість та відповідність цієї сировини є критично важливими для кінцевого продукту. На цьому етапі блокчейн може використовуватися для перевірки походження сировини та уникнення постачання неякісних або підроблених інгредієнтів.

Ризики: Використання підробленої сировини або низькоякісних компонентів.

Транспортування сировини (Transportation). Транспортування сировини до виробника також є важливим етапом, оскільки умови перевезення можуть впливати на якість матеріалів. У разі невідповідних умов транспортування (температура, вологість) сировина може бути зіпсована, що призведе до виробництва неякісних ліків. Блокчейн дозволяє фіксувати дані про стан вантажу в режимі реального часу, забезпечуючи контроль за всіма параметрами перевезення.

Ризики: Пошкодження або втрата сировини під час транспортування, порушення температурного режиму.

Виробник (Manufacturer). Виробник відповідає за безпосереднє виготовлення лікарських препаратів із сировини. Тут важливо забезпечити дотримання всіх норм і стандартів виробництва, оскільки будь-які відхилення можуть призвести до випуску фальсифікованих або неякісних ліків. Впровадження блокчейну допомагає реєструвати кожен етап виробничого процесу та підтверджувати відповідність стандартам.

Ризики: Виготовлення неякісних або підроблених препаратів, маніпуляції з виробничими даними.

Транспортування готових препаратів (Transporter). Після виробництва препарати транспортуються до оптових або дистриб'юторських центрів. Тут також важливо дотримуватися умов зберігання та перевезення, аби зберегти властивості препаратів. Використання блокчейну дозволяє відслідковувати умови транспортування та уникати маніпуляцій із вантажем.

Ризики: Невідповідні умови транспортування, заміна оригінальних препаратів підробками.

Оптовий постачальник (Wholesaler). Оптові постачальники закуповують великі партії ліків і продають їх дистриб'юторам або безпосередньо аптекам. На цьому етапі ризик підробки або маніпуляцій із препаратами збільшується, оскільки обіг великих обсягів ліків ускладнює контроль. Блокчейн дозволяє фіксувати транзакції між учасниками ланцюга і гарантувати оригінальність ліків.

Ризики: Змішування підробок з оригінальними препаратами, підвищення вартості за рахунок фальшивих транзакцій.

Дистриб'ютор (Distributor). Дистриб'ютори відповідають за доставку ліків до роздрібних точок продажу, таких як аптеки або лікарні. Контроль за цілісністю упаковки та якістю ліків є ключовим завданням на цьому етапі. Завдяки блокчейну можна перевіряти кожну партію ліків і підтверджувати її походження.

Ризики: Втручання в упаковку або заміна препаратів під час розподілу, підробка документів.

Транспортування до кінцевого споживача (Transporter). Останнє транспортування ліків від дистриб'ютора до аптек або медичних установ є завершальним етапом. Блокчейн забезпечує повний ланцюг відслідковування від виробника до аптеки, що дозволяє уникати підробок навіть на цьому останньому етапі.

Ризики: Підміна або неякісне зберігання під час останнього транспортування.

Кінцевий споживач (аптека або пацієнт) (Customer). Аптеки або пацієнти отримують ліки для подальшого використання. Тут особливо важливим є

впевненість у тому, що препарат є оригінальним, безпечним та ефективним. Блокчейн може бути використаний для верифікації оригінальності препарату через сканування QR-коду або RFID-тегу, що містить усі дані про ланцюг постачання.

Ризики: Продаж підроблених ліків, недостатня інформація для верифікації походження препарату.

2.2 Структура взаємодії учасників за допомогою смарт-контрактів

Для взаємодії між учасниками ланцюга постачання лікарських препаратів використовується блокчейн-архітектура MAS (Multi-Agent System), яка забезпечує безперервний обмін даними через спільний реєстр транзакцій. Кожен учасник системи (виробник, постачальник, транспортувальник, дистриб'ютор, кінцевий споживач) виконує свої дії через смарт-контракти, що регулюють і автоматизують передачу лікарських засобів, записуючи всі транзакції в блокчейн.[16]

2.2.1 Взаємодія через спільний реєстр

У системі MAS кожен учасник має унікальну роль і доступ до спільного децентралізованого реєстру – блокчейну. Блокчейн є основною базою даних, яка використовується для реєстрації всіх транзакцій. Інформація, яку записує кожен учасник, залишається незмінною після її внесення і доступною для всіх інших учасників, які мають необхідні права доступу. Це дозволяє гарантувати прозорість та достовірність даних на всіх етапах постачання лікарських засобів. [17]

Основними учасниками цієї системи є:

1. Виробник лікарських препаратів.

Виробник ініціює процес, вводячи в систему дані про виготовлені партії ліків. До блокчейну записуються наступні атрибути:

- Ідентифікаційний номер партії.
- Назва препарату.
- Кількість виготовлених одиниць.
- Дата і час виробництва.

Ця інформація підписується цифровим підписом виробника, який гарантує її автентичність та захист від змін. Дані стають частиною блоку в ланцюжку, забезпечуючи подальший облік і контроль на наступних етапах.

2. Транспортування.

Логістичні компанії, що відповідають за доставку ліків від виробника до оптових постачальників або дистриб’юторів, додають у блокчейн дані про транспортування, такі як:

– Транспортний засіб.

– Час і місце відправлення та прибуття.

Дані також підписуються цифровими підписами транспортної компанії, і після їх внесення до реєстру стають доступними для інших учасників ланцюга. Смарт-контракт перевіряє дотримання встановлених умов транспортування і блокує транзакцію у разі відхилення від них, наприклад, у випадку неправильного температурного режиму.

3. Оптовий постачальник та дистриб’ютор. На цьому етапі постачальники та дистриб'ютори отримують товар, після чого вони фіксують інформацію про прийом ліків у блокчейні:

– Ідентифікатори партії, що надійшла.

– Кількість отриманих одиниць.

– Умови зберігання.

Оптовий постачальник також використовує смарт-контракт для перевірки і підтвердження відповідності партії лікарських засобів, що надходять, з тими даними, які були записані виробником і транспортною компанією. Це гарантує, що партія є легітимною і не була підроблена або пошкоджена.

4. Кінцевий споживач. Кінцевий споживач – аптека або пацієнт – через блокчейн може отримати доступ до історії руху ліків. Всі дані, зокрема про виробництво, транспортування, зберігання, стають доступними для кінцевої перевірки. Це дозволяє споживачеві впевнитися, що препарат є оригінальним, а також перевірити дотримання умов транспортування та зберігання.

2.2.2 Смарт-контракти в блокчейн-архітектурі MAS

Ключовим елементом взаємодії учасників у блокчейн-архітектурі MAS є смарт-контракти – автоматизовані програми, які забезпечують виконання умов угод між учасниками без необхідності в посередниках. Кожен агент системи взаємодіє з іншими учасниками через смарт-контракти, які виконують такі функції:

Перевірка умов угоди (відповідність даних про кількість ліків та умови їх транспортування).

Автоматичне виконання умов угоди (наприклад, передача прав власності на ліки після підтвердження відповідності умов).

Реєстрація транзакцій у блокчейні після успішної перевірки.

Використання смарт-контрактів дозволяє мінімізувати ризик людських помилок, шахрайства та сприяє значній автоматизації процесів у ланцюзі постачання. Кожна взаємодія, будь-то передача партії ліків від виробника до постачальника або зміна умов транспортування, підтверджується через смарт-контракт і записується у блокчейн. [18]

2.2.3 Створення та використання QR-кодів

У рамках блокчейн-архітектури MAS Для взаємодії між учасниками ланцюга постачання лікарських препаратів у нашій системі використовуються QR-коди, які автоматизують процес ідентифікації та внесення даних у блокчейн на всіх етапах. Кожен лікарський препарат має унікальний QR-код, що містить інформацію про продукт, а його зчитування автоматично додає дані до реєстру транзакцій. [19]

Створення та використання QR-кодів

Кожна партія ліків, поставка яких була узгоджена, отримує унікальний QR-код, який включає наступні дані:

- Ідентифікатор партії.
- Назву препарату.
- Виробника.
- Дату виробництва.
- Термін придатності.

– Кількість одиниць в партії.

– Дані про транспортування.

– Хеш смарт-контракту.

Цей код генерується під час погодження поставки лікарських препаратів від виробника до наступного отримувача (посередника або кінцевого споживача) разом з першим блоком у блокчейні. Він інтегрований з блокчейн-системою та розміщений на упаковці продукту.

Зчитування цього QR-коду будь-яким учасником автоматично передає відповідні дані до блокчейну і реєструє нову транзакцію.[20]

На кожному етапі ланцюга постачання учасники зчитують QR-код за допомогою спеціальних пристроїв або мобільних додатків. Це дозволяє автоматизувати обробку інформації та внесення даних до блокчейну, забезпечуючи безперервний контроль за кожним препаратом.

Після зчитування:

1. Дані автоматично передаються до блокчейну.

Система фіксує всі атрибути, такі як час і місце зчитування, кількість товару, умови зберігання та перевезення. Це усуває необхідність ручного введення даних, знижуючи ймовірність помилок.

2. Перевіряється автентичність та відповідність інформації. Після зчитування QR-коду система автоматично порівнює отримані дані з тими, що вже були записані у блокчейні на попередніх етапах. Якщо дані збігаються, транзакція підтверджується. У разі виявлення невідповідностей (наприклад, при спробі внести підроблені ліки), транзакція блокується смарт-контрактом.

3. Запис нових даних у блокчейн. Кожен етап постачання ліків – від виробництва до доставки кінцевому споживачеві – фіксується у блокчейні через зчитування QR-кодів. Після підтвердження системою автентичності даних нові транзакції записуються до реєстру, забезпечуючи прозорість і безперервний облік усіх переміщень продукції.

Використання QR-кодів кінцевими споживачами

Кінцеві споживачі – аптеки, медичні установи або пацієнти – також мають можливість зчитувати QR-коди для перевірки автентичності ліків. Після сканування коду вони отримують доступ до всієї історії руху препарату, починаючи з моменту виробництва:

– Ідентифікаційні дані про виробника.
– Інформацію про препарат: назву, дату виготовлення, термін придатності, кількість, ціну тощо.
– Етапи транспортування.

Це забезпечує повну прозорість процесу постачання і дозволяє уникнути ризику отримання контрафактної продукції. QR-коди дозволяють кінцевим споживачам перевірити достовірність ліків безпосередньо на місці продажу.

2.3 Процес верифікації та запису даних у блокчейн

Після того як учасник системи сканував QR-код, що містить інформацію про лікарський препарат, розпочинається послідовність дій, що забезпечує верифікацію даних та їхнє внесення до блокчейн-системи. Цей процес включає кілька ключових етапів (рис. 2.2). [21]

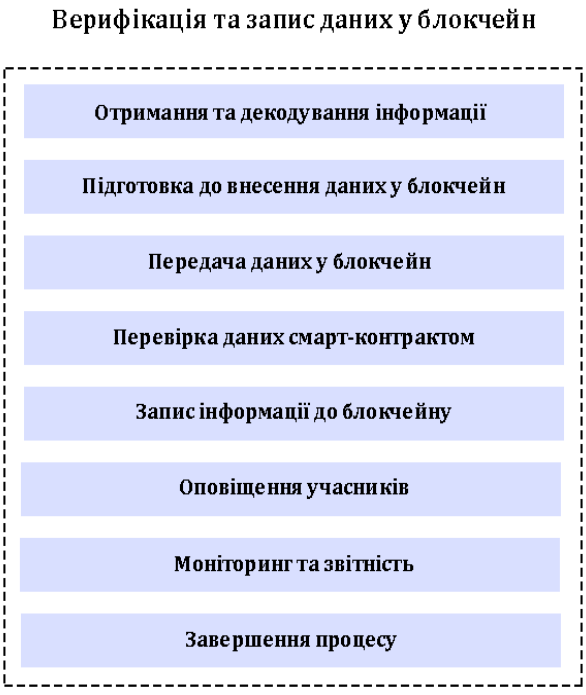


Рисунок 2.2 – Верифікація та запис даних у блокчейн

1. Отримання та декодування інформації

– Учасник використовує спеціалізований пристрій (наприклад, смартфон або планшет) для сканування QR-коду. QR-код містить зашифровану інформацію у форматі JSON, яка включає деталі, такі як ідентифікатор препарату, назва, номер партії, дата виготовлення та термін придатності.

– Після зчитування система декодує цю інформацію та перевіряє цілісність даних, підтверджуючи, що всі необхідні поля заповнені.

2. Підготовка до внесення даних у блокчейн

– Учасник перевіряє, чи всі дані коректні та актуальні. У разі виявлення невідповідностей система надає сповіщення про помилку, що вимагає корекції.

– Формується транзакція, що містить ідентифікаційні дані учасника, інформацію про препарат та метадані, такі як час та тип транзакції.

3. Передача даних у блокчейн

– Учасник викликає смарт-контракт, який відповідає за обробку та верифікацію даних. Цей контракт автоматично перевіряє дані на відповідність установленим критеріям.

– Після цього формовані дані (JSON-об'єкт) передаються в блокчейн, що дозволяє зафіксувати їх у дистрибутивному реєстрі.

4. Перевірка даних смарт-контрактом

– Смарт-контракт проводить автоматизовану перевірку даних, підтверджуючи справжність препарату, дотримання терміну придатності та коректність інформації про учасника.

– На основі результатів перевірки смарт-контракт визначає статус транзакції (успішно/неуспішно) та надає відповідні дані учаснику.

5. Запис інформації до блокчейну

– Якщо смарт-контракт підтверджує коректність даних, інформація про транзакцію записується в блокчейн, створюючи новий блок. Це забезпечує незмінність та доступність інформації для всіх учасників системи.

– Запис в блокчейн також оновлює статус препарату, що дозволяє учасникам бачити актуальну інформацію.

6. Оповіщення учасників

– Учасник отримує сповіщення про результат транзакції. Якщо вона була успішно підтверджена, інформація оновлюється у системі, що дозволяє іншим учасникам бачити актуальний статус препарату.

– У разі невдачі учасник отримує інформацію про помилку, що дозволяє вжити необхідні заходи для виправлення ситуації.

7. Моніторинг та звітність

– Усі дії, пов'язані з транзакцією, фіксуються у журналах блокчейну, що забезпечує прозорість та можливість відстеження історії постачання.

– Учасники можуть генерувати звіти на основі зафіксованих даних, що є корисним для контролю якості, звітності або аудиту.

8. Завершення процесу

– Після підтвердження транзакції учасник може завершити взаємодію, зафіксувавши, що препарат було отримано, відвантажено або використано.

– Вся інформація про препарат залишається доступною для перегляду всім учасникам, що підвищує довіру до системи та зменшує ризики, пов'язані з підробками.

2.4 Механізм взаємодії між базою даних та блокчейном

Впровадження блокчейн-технології для відстеження поставок лікарських засобів потребує ефективної інтеграції з традиційними системами зберігання даних, зокрема базами даних. У цій системі блокчейн виступає гарантом прозорості та незмінності інформації, тоді як база даних забезпечує швидкий доступ і обробку великих обсягів даних.[22] Механізм взаємодії між ними реалізується наступним чином:

1. Збір та внесення даних. Первинні дані про виробництво, транспортування чи розподіл лікарських препаратів вводяться виробниками, складами або іншими учасниками через веб-застосунок. Ця інформація включає

номер партії, кількість одиниць, пункт відправлення, пункт призначення та інші ключові атрибути.

2. Збереження даних у базі. Отримані дані передаються до бази даних через API. База даних виконує функцію централізованого сховища, що забезпечує зручність зберігання та швидкий доступ для операційних потреб.

3. Синхронізація з блокчейном. Для забезпечення незмінності та достовірності даних під час їх збереження в базі, синхронізація з блокчейном виконується через смартконтракти. Смартконтракти, налаштовані в блокчейні, перевіряють:

- правильність структури даних;
- відповідність даних правилам системи (наприклад, коректність номеру партії чи місця призначення);
- відсутність дублювання або помилок.

4. Запис у блокчейн. Після успішної перевірки смартконтрактом блокчейн зберігає хеші ключових даних. Хешування забезпечує конфіденційність (відсутність прямого доступу до деталей партії) та незмінність інформації. Крім того, кожен новий запис у блокчейні доповнює транзакційну історію поставки, створюючи повну хронологію руху партії.

5. Перевірка консистентності. Під час зчитування або оновлення даних із бази виконується звірка між актуальними даними та хешами в блокчейні. Якщо виявлено невідповідність, система блокує запит або генерує сповіщення, сигналізуючи про можливе втручання чи помилку.

6. Моніторинг та аналітика. Інтеграція бази даних із блокчейном дозволяє учасникам ланцюга постачання отримувати звіти в реальному часі через веб-застосунок. Це дозволяє контролювати рух партій, відстежувати критичні точки та перевіряти автентичність лікарських засобів.[23]

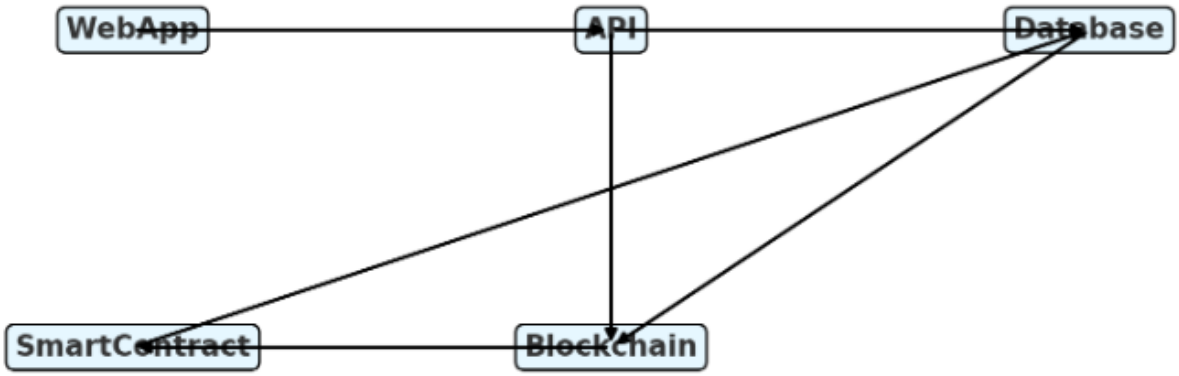


Рисунок 2.3 – Взаємодія між елементами системи

2.5 Вимоги до системи управління ланцюгом постачання лікарських препаратів

2.5.1 Функціональні вимоги

Реєстрація та авторизація користувачів. Система повинна підтримувати реєстрацію нових користувачів та їхню авторизацію з використанням безпечних методів, таких як JWT токени.

Управління ролями та дозволами. Система повинна підтримувати різні ролі користувачів з відповідними дозволами.

Ведення журналу транзакцій. Система повинна вести журнал усіх транзакцій та змін у ланцюзі постачання для забезпечення прозорості та відстежуваності.

Інтеграція з блокчейном. Система повинна підтримувати інтеграцію з блокчейн-платформою для забезпечення незмінності та безпеки даних.

2.5.2 Нефункціональні вимоги

Продуктивність. Система повинна забезпечувати швидку обробку запитів та транзакцій, навіть при великих обсягах даних.

Безпека. Система повинна забезпечувати захист даних від несанкціонованого доступу, зміни або видалення.

Масштабованість. Система повинна бути масштабованою для обробки зростаючих обсягів даних та кількості користувачів.

Надійність. Система повинна бути надійною та стійкою до відмов, забезпечуючи безперервність роботи.

Зручність використання. Система повинна мати інтуїтивно зрозумілий інтерфейс та бути зручною у використанні для всіх категорій користувачів.

Аналіз кібербезпеки: потенційні загрози та способи їх мінімізації

Система управління ланцюгом постачання лікарських препаратів на основі блокчейну повинна забезпечувати високий рівень кібербезпеки, враховуючи наступні загрози:

1. Хакерські атаки. злам системи для зміни даних про постачання, крадіжки конфіденційної інформації або підробки записів.

2. Зловмисне втручання у смарт-контракти. внесення шкідливого коду для порушення роботи контрактів, що регулюють транзакції.

3. Соціальна інженерія: отримання доступу до системи шляхом обману працівників.

4. DDoS-атаки. перевантаження мережі фальшивими запитами, що призводить до її недоступності.

5. Атаки 51%. можливість злому блокчейну через отримання контролю над більшістю обчислювальної потужності мережі.

Способи мінімізації загроз:

Криптографічний захист даних. використання сучасних алгоритмів шифрування для забезпечення безпеки даних і транзакцій.

Аудит смарт-контрактів. регулярна перевірка коду контрактів незалежними експертами на наявність вразливостей.

Багатофакторна автентифікація. впровадження кількох рівнів авторизації для користувачів системи.

Сегментація мережі. поділ мережі на окремі частини для обмеження поширення шкідливого впливу у випадку злому.

Резервне копіювання. створення резервних копій даних та системних налаштувань, щоб забезпечити швидке відновлення після інцидентів.

Навчання персоналу. підвищення обізнаності працівників про методи соціальної інженерії та правила безпечного користування системою.

Застосування механізмів захисту від DDoS. використання спеціальних систем для моніторингу трафіку та відхилення небажаних запитів.

Консенсусний механізм. вибір ефективного механізму консенсусу (наприклад, Proof of Stake) для зменшення ймовірності атак 51%.

Етичні вимоги

Для забезпечення етичності системи управління ланцюгом постачання лікарських препаратів важливо врахувати наступні аспекти:

1. Прозорість даних. Система повинна забезпечувати доступ до інформації про походження та якість лікарських засобів усім зацікавленим сторонам.

   Використання блокчейн-технології дозволяє уникнути маніпуляцій з даними, зберігаючи всі транзакції в незмінному реєстрі.
2. Конфіденційність. Забезпечення захисту персональних даних пацієнтів та комерційної інформації виробників.

   Використання шифрування для захисту конфіденційних даних.
3. Уникнення конфлікту інтересів. Автоматизація процесів за допомогою смарт-контрактів, що виключає суб’єктивне втручання людей.

   Наявність відкритих записів про дії всіх учасників ланцюга, що унеможливлює приховування фактів корупції або неетичної поведінки.
4. Дотримання міжнародних стандартів. Система повинна відповідати принципам етичності, закріпленим у нормативних актах, таких як Good Distribution Practice (GDP) та Good Manufacturing Practice (GMP).

   Забезпечення рівноправного доступу до лікарських засобів для всіх груп населення.
5. Соціальна відповідальність. Впровадження механізмів контролю за дотриманням етичних норм всіма учасниками процесу.

   Запобігання дискримінації у постачанні ліків та гарантування їх доступності для вразливих груп населення.

2.6 Вимоги до блокчейн-платформи

Підтримка смарт-контрактів. Блокчейн-платформа повинна підтримувати створення та виконання смарт-контрактів для автоматизації процесів у ланцюзі постачання.

Інтеграція з зовнішніми системами. Блокчейн-платформа повинна підтримувати інтеграцію з зовнішніми системами, такими як бази даних та веб-додатки.

Прозорість та відстежуваність. Блокчейн-платформа повинна забезпечувати прозорість та можливість відстежування всіх транзакцій у ланцюзі постачання.

Незмінність даних. Блокчейн-платформа повинна забезпечувати незмінність записаних даних для запобігання шахрайства та помилок.

Також сучасних системах управління ланцюгами постачання лікарських засобів масштабованість блокчейн-платформи є ключовим параметром, який визначає її здатність обробляти великі обсяги даних та високу частоту транзакцій.

Проблеми масштабованості.

1. Збільшення кількості учасників системи (виробників, постачальників, аптек) може спричинити затримки у записі транзакцій через високу навантаженість мережі.

2. Високий обсяг транзакцій (постійне оновлення даних про партії ліків, транспортні умови тощо) може перевантажити систему.

Вимоги до масштабованості.

1. Пропускна здатність: блокчейн-платформа повинна підтримувати обробку щонайменше декількох тисяч транзакцій за секунду (TPS), щоб уникнути затримок.

Швидкість підтвердження: мінімізувати час підтвердження транзакції до декількох секунд.

2. Гнучкість масштабування: система повинна дозволяти легке масштабування як горизонтально (додавання вузлів), так і вертикально (збільшення обчислювальних потужностей окремих вузлів).

Можливі рішення:

1. Використання шарів другого рівня (Layer 2): впровадження таких технологій, як State Channels або Rollups, що дозволяють обробляти транзакції поза основним блокчейном.

2. Шардінг: розподіл блокчейн-мережі на сегменти (шарди), що працюють паралельно, щоб розвантажити мережу.

3. Об'єднання обчислювальної потужності: використання консорціумних блокчейнів, де учасники об'єднують ресурси для підвищення продуктивності.

4. Моніторинг продуктивності: блокчейн-платформа повинна включати інструменти для моніторингу продуктивності, щоб виявляти та усувати "вузькі місця" в обробці даних.

2.7 Вимоги до бази даних

Зберігання даних про лікарські препарати. База даних повинна підтримувати зберігання інформації про лікарські препарати, включаючи назву, тип, виробника, умови зберігання та інші атрибути.

Зберігання даних про транзакції. База даних повинна підтримувати зберігання інформації про транзакції у ланцюзі постачання, включаючи дату, час, учасників та статус транзакції.

Зберігання даних про користувачів. База даних повинна підтримувати зберігання інформації про користувачів, включаючи їхні ролі та дозволи.

Інтеграція з блокчейном. База даних повинна підтримувати інтеграцію з блокчейн-платформою для забезпечення незмінності та безпеки даних.

2.8 Вимоги до веб-додатку

Інтуїтивно зрозумілий інтерфейс. Веб-додаток повинен мати інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам легко виконувати необхідні дії.

Управління лікарськими препаратами. Веб-додаток повинен підтримувати управління інформацією про лікарські препарати, включаючи їх додавання, редагування та видалення.

Управління транзакціями. Веб-додаток повинен підтримувати управління транзакціями у ланцюзі постачання, включаючи їх створення, редагування та відстежування.

Інтеграція з блокчейном. Веб-додаток повинен підтримувати інтеграцію з блокчейн-платформою для забезпечення незмінності та безпеки даних.

2.9 Вибір середовища розробки блокчейну

Середовищем для розробки блокчейн-системи було обрано Ganache. Це рішення обумовлено низкою переваг, які надає цей інструмент для розробки та тестування смарт-контрактів.

Ganache – це локальний блокчейн, який дозволяє розробникам створювати, тестувати та розгортати смарт-контракти без необхідності підключення до основної мережі Ethereum. Ganache надає можливість швидко розгортати приватний блокчейн на локальному комп'ютері, що дозволяє розробникам тестувати свої додатки в ізольованому середовищі.[24]

Переваги використання Ganache

Швидкість розгортання. Ganache дозволяє швидко створити локальний блокчейн, що значно прискорює процес розробки та тестування смарт-контрактів.

Немає необхідності чекати підтвердження транзакцій від мережі, як це відбувається в основній мережі Ethereum.

Простота використання. Ganache має інтуїтивно зрозумілий інтерфейс, який дозволяє легко керувати локальним блокчейном.

Процес встановлення та налаштування Ganache є простим і не вимагає глибоких технічних знань.

Контроль над станом блокчейну. Розробники можуть контролювати стан блокчейну, включаючи баланси рахунків, стан смарт-контрактів та інші параметри.

Це дозволяє створювати різні сценарії тестування та перевіряти поведінку смарт-контрактів у різних умовах.

Інтеграція з іншими інструментами. Ganache легко інтегрується з іншими інструментами розробки, такими як Truffle, Remix та MetaMask.

Це дозволяє створювати комплексне середовище розробки, де можна використовувати різні інструменти для різних завдань.

Безпека та приватність. Використання локального блокчейну дозволяє розробникам тестувати свої додатки в ізольованому середовищі, без ризику витоку інформації або втручання зовнішніх факторів.

Це особливо важливо для тестування чутливих даних та конфіденційної інформації.

Використання Ganache в проєкті

Створення локального блокчейну. Для початку роботи з Ganache необхідно встановити програмне забезпечення та створити новий локальний блокчейн.

Після створення блокчейну, Ganache автоматично генерує декілька тестових рахунків з віртуальними етерами, які можна використовувати для тестування.

Розгортання смарт-контрактів. Після створення локального блокчейну, розробники можуть розгортати свої смарт-контракти за допомогою Truffle або Remix.

Ganache надає можливість швидко розгортати та тестувати смарт-контракти, перевіряючи їх функціональність та безпеку.

Тестування та відладка. Використовуючи Ganache, розробники можуть створювати різні сценарії тестування та перевіряти поведінку смарт-контрактів у різних умовах.

Це дозволяє виявляти та виправляти помилки на ранніх етапах розробки, що значно зменшує витрати та час на відладку.

Інтеграція з Truffle. Ganache легко інтегрується з Truffle, що дозволяє автоматизувати процес розгортання та тестування смарт-контрактів.

Використовуючи Truffle, розробники можуть створювати міграції, тести та скрипти для взаємодії зі смарт-контрактами, що значно прискорює процес розробки.

2.10 Правове забезпечення

У системах управління ланцюгом постачання лікарських препаратів важливим аспектом є дотримання юридичних норм і стандартів. Це забезпечує відповідність системи вимогам законодавства, захист прав учасників і пацієнтів, а також підвищує довіру до платформи.

1. Основні принципи дотримання законодавства про конфіденційність даних:

- Система повинна відповідати національним і міжнародним нормативам у сфері захисту персональних даних, наприклад:

GDPR (Загальний регламент захисту даних ЄС).

HIPAA (Закон про портативність та підзвітність страхування здоров'я у США).

- Дані про пацієнтів і лікарські препарати повинні зберігатися та передаватися з використанням надійного шифрування.

2. Механізми забезпечення конфіденційності:

- Використання багаторівневого доступу: кожен учасник отримує доступ лише до тієї інформації, яка стосується його ролі.

- Регулярний аудит системи для перевірки відповідності стандартам безпеки даних.

- Впровадження механізму псевдонімізації: персональні дані пацієнтів замінюються на унікальні ідентифікатори, які не дозволяють ідентифікувати особу без додаткової інформації.

3. Юридична відповідальність:

- Учасники системи повинні підписати угоду про нерозголошення інформації (NDA).

- У випадку витоку даних система повинна передбачати план дій, включаючи повідомлення відповідних органів та постраждалих осіб.

Захист прав пацієнтів

1. Прозорість:

- Пацієнти повинні мати можливість перевірити походження лікарських засобів через відкриті дані системи (наприклад, через QR-код або RFID-тег).

- Інформація про ліки повинна включати дату виробництва, термін придатності, умови зберігання та дистриб'юторів.

2. Право на доступ до даних:

- Пацієнти мають право отримувати доступ до своїх даних, якщо вони зберігаються у системі, з дотриманням вимог конфіденційності.

- Система повинна забезпечувати простий і зрозумілий механізм запиту даних.

3. Право на захист від підроблених ліків:

- Технології блокчейну дозволяють уникнути потрапляння підроблених ліків у ланцюг постачання, що є прямим внеском у безпеку пацієнтів.

- Законодавчі механізми повинні передбачати відповідальність за поширення підробок.

4. Міжнародні стандарти та регулювання

Відповідність міжнародним стандартам:

- Система має враховувати вимоги Good Distribution Practice (GDP) та Good Manufacturing Practice (GMP).

- Використання стандартів відстеження, таких як GS1, для уникнення дублювання або втрати даних.

5. Врахування локального законодавства. Ланцюг постачання лікарських засобів повинен відповідати специфічним вимогам національного законодавства у сфері обігу лікарських засобів, логістики та фармацевтичної практики.[25]

Висновки до розділу:

У другому розділі кваліфікаційної роботи було визначено структуру та основні компоненти блокчейн-системи для ланцюга постачання медичних препаратів. Основні висновки розділу:

1. Структура ланцюга постачання лікарських препаратів: Ланцюг постачання лікарських препаратів включає кілька ключових етапів: від постачання сировини до доставки готових препаратів кінцевим споживачам. Кожен етап має специфічні ризики, такі як фальсифікація та підробка. Технологія блокчейн забезпечує прозорість та надійний контроль на всіх етапах ланцюга постачання.

2. Взаємодія учасників за допомогою смарт-контрактів. Блокчейн-архітектура MAS (Multi-Agent System) забезпечує безперервний обмін даними через спільний реєстр транзакцій. Смарт-контракти регулюють та автоматизують передачу лікарських засобів, мінімізуючи ризик людських помилок та шахрайства.

3. Створення та використання QR-кодів. QR-коди автоматизують процес ідентифікації та внесення даних у блокчейн на всіх етапах. Кожен лікарський препарат має унікальний QR-код, що містить інформацію про продукт.

4. Механізм взаємодії між базою даних та блокчейном. Інтеграція блокчейн-технології з традиційними системами зберігання даних забезпечує прозорість та незмінність інформації, тоді як база даних забезпечує швидкий доступ та обробку даних.

5. Вимоги до системи управління ланцюгом постачання лікарських препаратів. Функціональні вимоги включають реєстрацію та авторизацію користувачів, управління ролями та дозволами, ведення журналу транзакцій та інтеграцію з блокчейном. Нефункціональні вимоги включають продуктивність, безпеку, масштабованість, надійність та зручність використання.

РОЗДІЛ 3
ПРОГРАМНО-ТЕХНІЧНА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Архітектура баз даних

3.1.1 База даних для зберігання даних про поставки, препарати, виробників

Архітектура бази даних [PHARMACY] створена для зберігання та управління даними, пов’язаними з лікарськими засобами, їх виробництвом, транспортуванням, зберіганням і продажем. Вона включає взаємопов’язані таблиці, які забезпечують логічну структуру та підтримують основні бізнес-процеси фармацевтичного ланцюга постачання.

Таблиці бази даних та їх опис

1. Таблиця DrugForms

- Призначена для зберігання інформації про форми лікарських препаратів (наприклад, таблетки, капсули, розчини).

- Основні атрибути:

1. FormID — унікальний ідентифікатор форми (первинний ключ).
2. FormName — назва форми препарату.
3. Description — опис форми, що містить додаткову інформацію.

2. Таблиця Drugs

- Містить детальну інформацію про лікарські препарати.

- Основні атрибути:

1. DrugID — унікальний ідентифікатор препарату (первинний ключ).
2. DrugName — назва препарату.
3. TypeID — зовнішній ключ, який посилається на DrugTypes.
4. FormID — зовнішній ключ, що зв’язує препарат із його формою у DrugForms.
5. ManufacturerID — зовнішній ключ для посилання на Manufacturers.

6. Інші атрибути включають активний інгредієнт, концентрацію, розмір упаковки, дозування, умови зберігання, термін придатності, ціну тощо.

3. Таблиця DrugTypes

- Зберігає інформацію про типи лікарських препаратів (наприклад, антибіотики, анальгетики).

- Основні атрибути:

1. TypeID — унікальний ідентифікатор типу (первинний ключ).

2. TypeName — назва типу препарату.

3. Description — текстовий опис типу.

4. Таблиця Manufacturers

- Містить інформацію про виробників препаратів.

- Основні атрибути:

1. ManufacturerID — унікальний ідентифікатор виробника (первинний ключ).

2. ManufacturerName — назва виробника.

3. Контактна інформація: країна, місто, адреса, телефон, електронна пошта, вебсайт.

4. Інформація про ліцензії: номер і термін дії.

5. Таблиця ShipmentDetails

- Призначена для зберігання даних про поставки препаратів.

- Основні атрибути:

1. ID — унікальний ідентифікатор запису (первинний ключ).

2. BatchID — номер поставки (унікальний ідентифікатор партії у рамках конкретної поставки).

3. DrugID — зовнішній ключ для зв’язку з препаратом.

4. ManufacturerID — зовнішній ключ для зв’язку з виробником.

5. Інші атрибути: кількість, дата виробництва, ціна.

6. Таблиця Shipments

- Відображає інформацію про маршрути доставки лікарських засобів.

- Основні атрибути:

1. ID — унікальний ідентифікатор запису (первинний ключ).

2. BatchID — номер поставки (унікальний для кожної партії в процесі транспортування).

3. Маршрут: адреси відправлення та доставки.

4. Часові мітки: створення запису та останнє оновлення.

7. Таблиця ShipmentStatus

- Містить інформацію про статус поставок.

- Основні атрибути:

1. ID — унікальний ідентифікатор запису (первинний ключ).

2. BatchID — номер поставки (зовнішній ключ для зв’язку із Shipments).

3. Час відправки, прибуття, останнє оновлення.

4. Поточний статус поставки (наприклад, "У дорозі", "Доставлено").

8. Таблиця Warehouse

- Зберігає дані про партії лікарських засобів на складах.

- Основні атрибути:

1. ID — унікальний ідентифікатор запису (первинний ключ).

2. DrugID, ManufacturerID — зовнішні ключі для зв’язку з препаратом і виробником.

3. BatchNumber — номер партії.

4. Інформація про кількість, сертифікат якості, часові мітки.

Взаємодія між таблицями:

Основні зв’язки:

- Drugs пов’язана з DrugForms, DrugTypes та Manufacturers через зовнішні ключі (FormID, TypeID, ManufacturerID).

- ShipmentDetails взаємодіє з Drugs і Manufacturers для деталізації поставок конкретних препаратів.

- Shipments і ShipmentStatus пов’язані через BatchID, забезпечуючи відстеження статусу кожної поставки.

- Warehouse зберігає дані про партії (BatchNumber), посилаючись на конкретні препарати та виробників.

Нижче наведена діаграма зв’язків між таблицями (рис. 3.1).

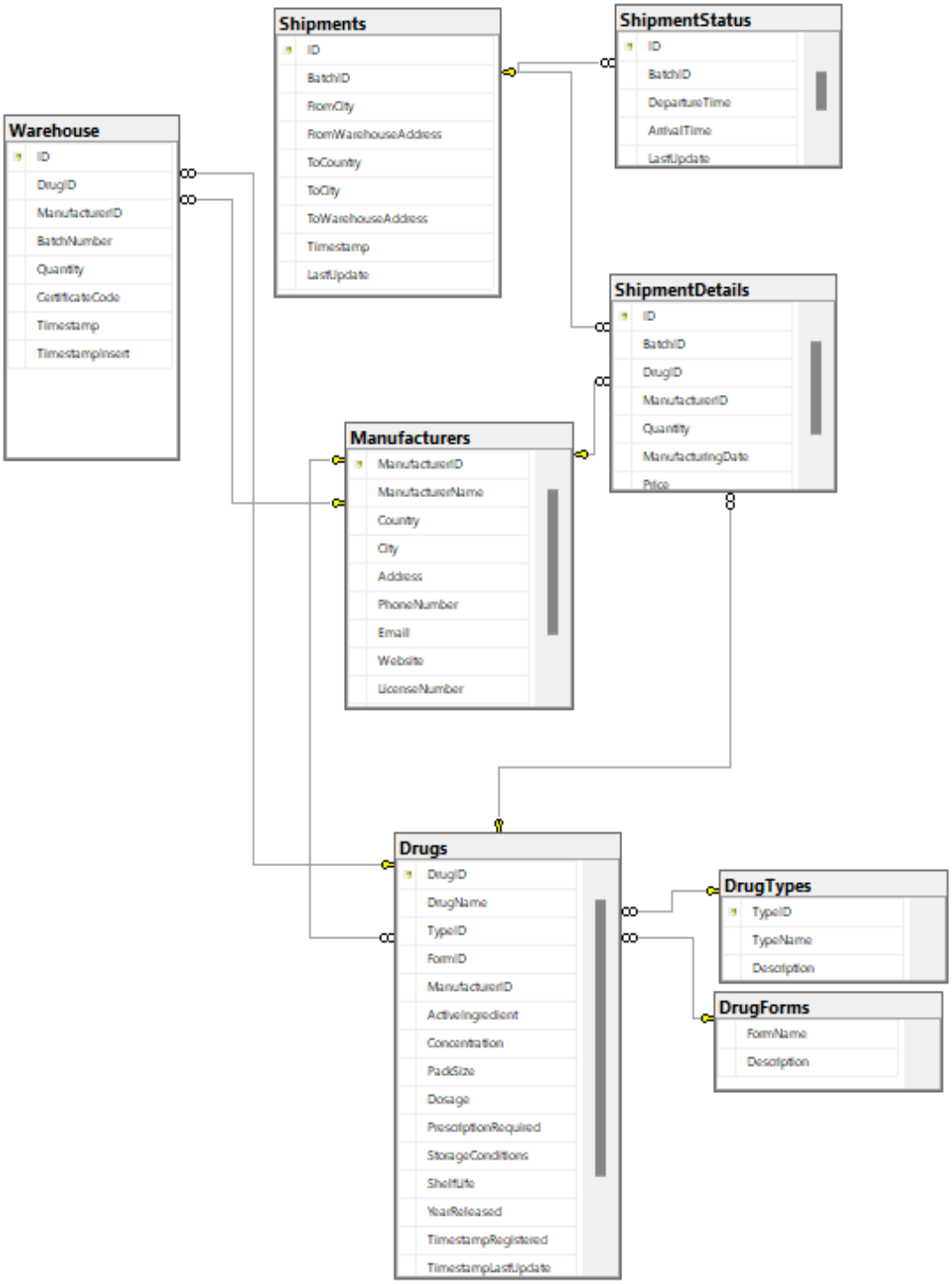


Рисунок 3.1 – Діаграма зв’язків між таблицями бази даних PHARMACY

3.1.2 База даних для авторизації

База даних для авторизації [Auth] створена для зберігання та управління даними, пов’язаними з користувачами, їх ролями та дозволами. Вона включає взаємопов’язані таблиці, які забезпечують логічну структуру та підтримують основні бізнес-процеси системи авторизації.

Причини зберігання даних для авторизації в окремій базі:

1. Безпека. Зберігання даних авторизації в окремій базі даних дозволяє ізолювати чутливу інформацію, таку як паролі та ролі користувачів, від основної бази даних. Це зменшує ризик несанкціонованого доступу та потенційних вразливостей.

2. Масштабованість. Окрема база даних для авторизації дозволяє легко масштабувати систему авторизації незалежно від основної бази даних. Це особливо корисно для великих систем з великою кількістю користувачів.

3. Простота управління. Розділення даних авторизації та основних бізнес-даних дозволяє спростити управління та адміністрування баз даних. Це також полегшує резервне копіювання та відновлення даних.

4. Продуктивність. Окрема база даних для авторизації може бути оптимізована для швидкого доступу та обробки запитів, пов’язаних з авторизацією, без впливу на продуктивність основної бази даних.

Таблиці бази даних та їх опис

Таблиця DataAdditional

Призначена для зберігання додаткової інформації про користувачів, таких як назви виробників, з якими вони пов’язані.

Основні атрибути:

– ID - унікальний ідентифікатор запису (первинний ключ).

– UserID - ідентифікатор користувача (зовнішній ключ, що посилається на таблицю Users).

– ManufacturerName - назва виробника.

Таблиця Permissions

Зберігає інформацію про дозволи, які можуть бути надані користувачам.

Основні атрибути:

– PermissionID - унікальний ідентифікатор дозволу (первинний ключ).

– PermissionName - назва дозволу.

– Description - опис дозволу.

Таблиця RolePermission

Призначена для зберігання зв’язків між ролями та дозволами.

Основні атрибути:

– RoleID - ідентифікатор ролі (зовнішній ключ, що посилається на таблицю Roles).

– PermissionID - ідентифікатор дозволу (зовнішній ключ, що посилається на таблицю Permissions).

Таблиця Roles

Зберігає інформацію про ролі користувачів.

Основні атрибути:

– RoleID - унікальний ідентифікатор ролі (первинний ключ).

– RoleName - назва ролі.

Таблиця Users

Містить інформацію про користувачів системи.

Основні атрибути:

– UserID — унікальний ідентифікатор користувача (первинний ключ).

– Username — ім’я користувача.

– PasswordHash — хеш пароля користувача.

– Email — електронна пошта користувача.

– RoleID — ідентифікатор ролі користувача (зовнішній ключ, що посилається на таблицю Roles).

– CreatedAt — дата та час створення запису.

Взаємодія між таблицями

Основні зв’язки:

– DataAdditional пов’язана з Users через зовнішній ключ UserID.

– RolePermission взаємодіє з Roles та Permissions для деталізації дозволів, наданих кожній ролі.

– Users пов’язана з Roles через зовнішній ключ RoleID.

Діаграма зв’язків між таблицями

Нижче наведена діаграма зв’язків між таблицями (рис. 3.2).

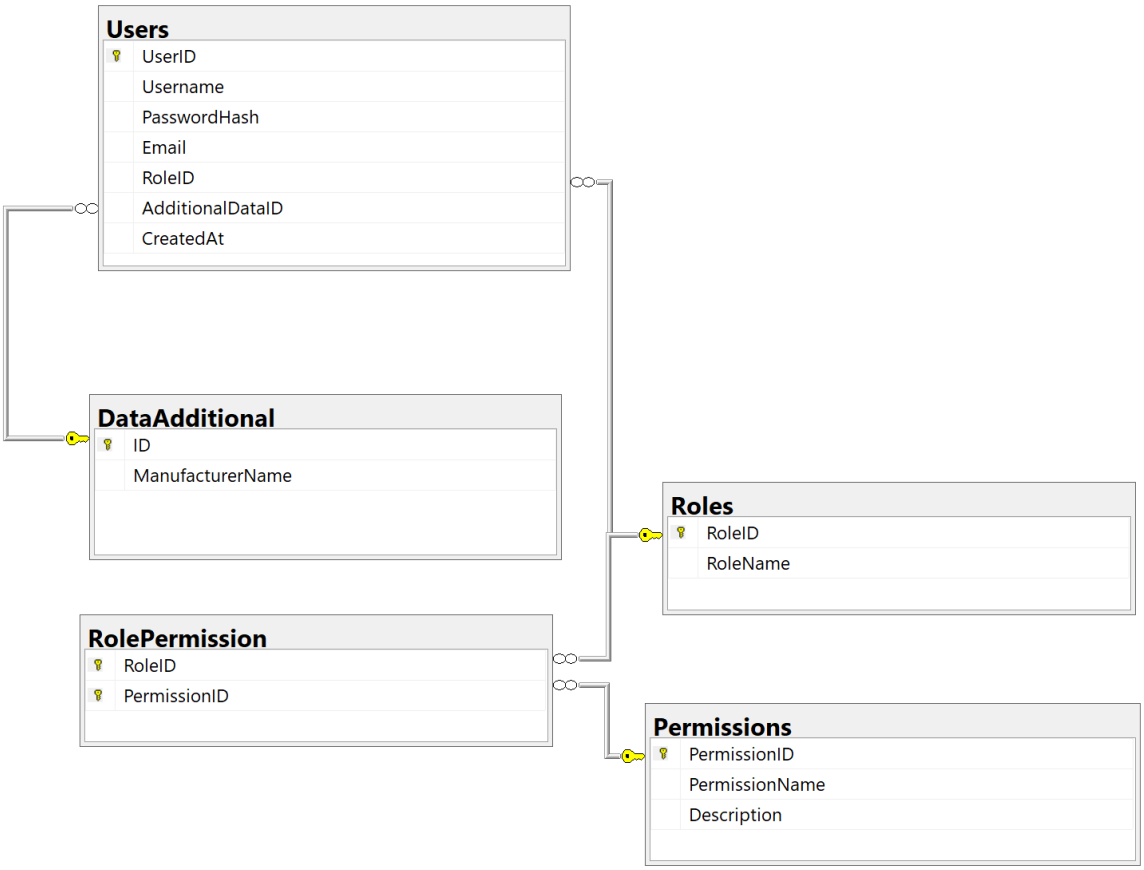


Рисунок 3.2 - Діаграма зв’язків між таблицями бази даних Auth

Інтеграція даних: Архітектура баз даних забезпечує пов’язаність усіх ключових об’єктів фармацевтичного ланцюга постачання. Це дозволяє відстежувати препарати на всіх етапах – від виробництва до кінцевого споживача.

Індексація: Для забезпечення швидкого доступу до даних та оптимізації запитів використовуються індекси. Індекси створюються на первинних ключах та зовнішніх ключах, а також на колонках, які часто використовуються у фільтрах та сортуваннях.

Нормалізація: Бази даних нормалізовані для уникнення дублювання даних та забезпечення цілісності даних. Нормалізація включає розбиття таблиць на менші, пов’язані таблиці, що дозволяє зменшити надмірність та покращити ефективність запитів.

Безпека: Для забезпечення безпеки даних використовуються механізми контролю доступу, шифрування та аудиту. Кожен користувач має свій рівень

доступу, що дозволяє обмежити доступ до певних даних лише авторизованим користувачам.

Резервне копіювання та відновлення: Регулярне резервне копіювання даних забезпечує можливість відновлення даних у разі втрати або пошкодження. Резервні копії зберігаються на окремих серверах та регулярно перевіряються на цілісність.

Масштабованість: Архітектура баз даних розроблена з урахуванням можливості масштабування. Це дозволяє легко додавати нові таблиці та зв’язки, а також збільшувати обсяг даних без значного впливу на продуктивність системи.

3.2 Архітектура WebAPP та API

Веб-застосунок розроблений для автоматизації та оптимізації управління ланцюгом постачання лікарських засобів. Основними його завданнями є забезпечення зручного інтерфейсу для введення, оновлення та перегляду даних, а також інтеграція з блокчейном для підвищення прозорості та достовірності інформації.

Веб-застосунок побудований на основі клієнт-серверної архітектури. Основні компоненти:

Клієнтська частина (Frontend): Реалізована за допомогою React, що забезпечує зручний та інтерактивний інтерфейс користувача. Використовуються сучасні веб-технології (HTML, CSS, JavaScript) для створення адаптивного дизайну. Інтерактивна взаємодія з сервером відбувається через API для обміну даними.

Серверна частина (Backend): Розроблена на Go з використанням фреймворків Gin для обробки HTTP-запитів. Інтеграція з блокчейном реалізована через go-ethereum, що забезпечує роботу зі смарт-контрактами та зберігання даних у розподіленій мережі. Сервер обробляє запити від клієнтської частини, керує бізнес-логікою та взаємодіє з базою даних і блокчейном через API.

3.2.1 Авторизація

При вході в систему виконується авторизація (Рис. 3.3). Алгоритм авторизації:

- Введення даних користувача (логін/пароль): Користувач вводить свої ідентифікаційні дані.
- Перевірка ідентифікаційних даних у базі даних: Система перевіряє надані дані, порівнюючи їх з збереженими в базі даних [Auth].
- Присвоєння ролі та дозволів користувача: Після успішної аутентифікації система присвоює користувачу відповідну роль (наприклад, адміністратор, виробник тощо).
- Генерація JWT токена: Після успішної авторизації система генерує JWT токен, який використовується для аутентифікації користувача в наступних запитах. Токен підписується секретним ключем, що забезпечує його цілісність та автентичність.
- Обробка помилок: Якщо дані, відправлені клієнтом, не можуть бути прив’язані до структури Credentials, повертається помилка 400 Bad Request. Якщо виникає помилка при пошуку користувача або ролі в базі даних, повертається відповідна помилка.

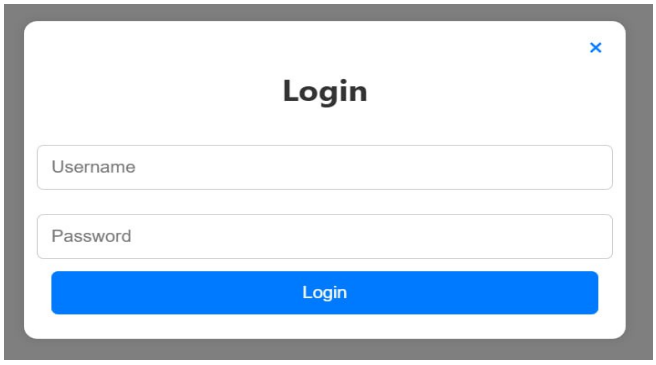


Рисунок 3.3 – Авторизація у систему

3.2.2 Вкладка Shipments

Ця вкладка створена для управління поставками (рис. 3.4). Основні можливості:

Перегляд існуючих поставок:

- Унікальний ідентифікатор партії.
- Статус поставки (Accepted, In Transit, Cancelled).
- Пункти відправлення та призначення (місто, склад).
- Час створення та останнього оновлення запису.

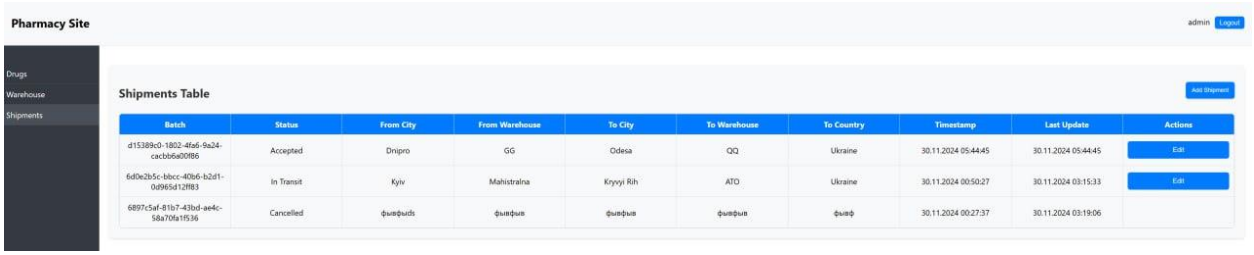


Рисунок 3.4 - Сторінка Shipments

Додавання нових поставок (рис. 3.5):

- Заповнення форми із полями, такими як міста та склади відправлення/призначення, країна отримувач.
- Автоматична фіксація даних про поставку у базі даних і запис транзакції в блокчейн через API.
- Генерація QR-коду для кожної нової поставки, який містить початкові дані та хеш транзакції.

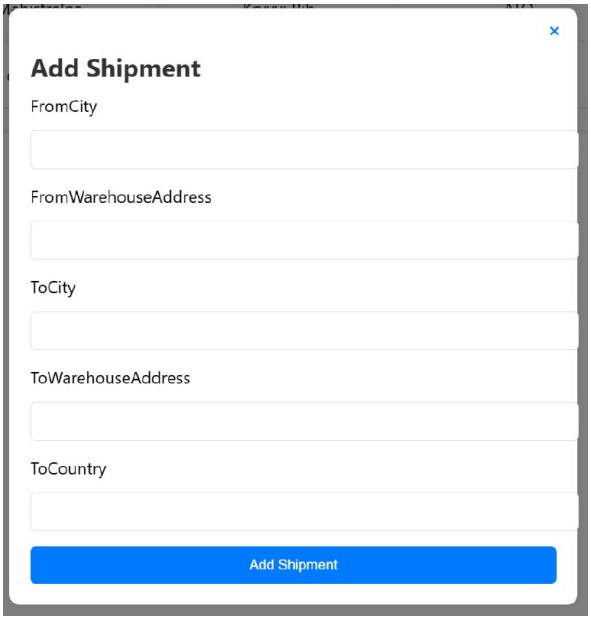


Рисунок 3.5 - Додавання нового запису до Shipments

Редагування поставок (Рисунок3.6):

- Зміна статусу поставки.
- Можливість оновлення адреси складу або міста призначення.
- Оновлення даних автоматично відображається у блокчейні та базах.

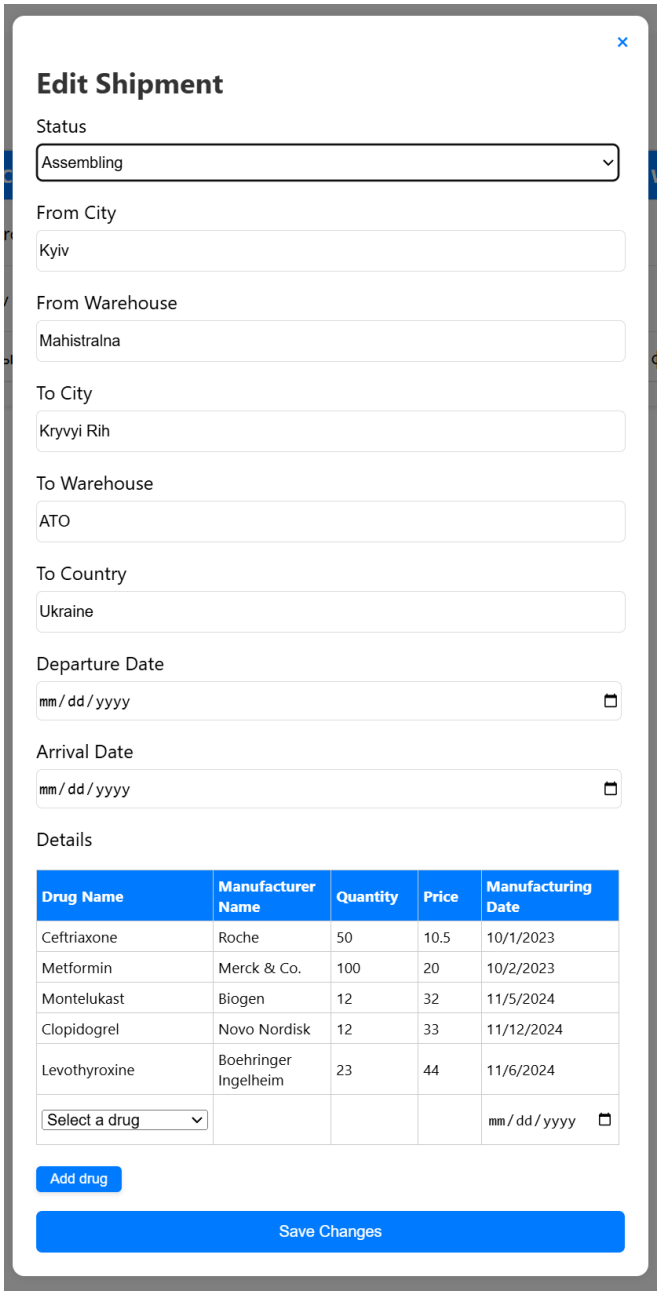


Рисунок 3.6 - Редагування даних поставки

Безпека та захист даних

Хешування паролів:

Бібліотека bcrypt: Використовується для хешування паролів. Це важливий аспект безпеки, оскільки зберігання паролів у відкритому вигляді є вразливістю.

```
hashedPassword, err := bcrypt.GenerateFromPassword([]byte(creds.Password), bcrypt.DefaultCost)
if err != nil {
    c.JSON(http.StatusInternalServerError, gin.H{"error": "Password hashing error"})
    return
}
```

JWT токен:

Використовується для аутентифікації користувача в наступних запитах. Токен підписується секретним ключем, що забезпечує його цілісність та автентичність.

```
token := jwt.NewWithClaims(jwt.SigningMethodHS256, claims)
tokenString, err := token.SignedString(jwtKey)
```

Обробка помилок прив’язки JSON:

Якщо дані, відправлені клієнтом, не можуть бути прив’язані до структури

```
if err := c.BindJSON(&creds); err != nil {
    c.JSON(http.StatusBadRequest, gin.H{"error": "Invalid data format"})
    return
}
```

Обробка помилок бази даних:

Якщо виникає помилка при пошуку користувача або ролі в базі даних, повертається відповідна помилка.

```
// Find the user by username
var user models.User
if err := database.AuthDB.Where("username = ?", creds.Username).First(&user).Error; err != nil {
    c.JSON(http.StatusUnauthorized, gin.H{"error": "Invalid credentials"})
    return
}
```

Управління сесіями

Термін дії токена: Токен дійсний протягом 72 годин. Цей час можна налаштувати в залежності від вимог безпеки та зручності користувачів.

```
expirationTime := time.Now().Add(time.Hour * 72)
```

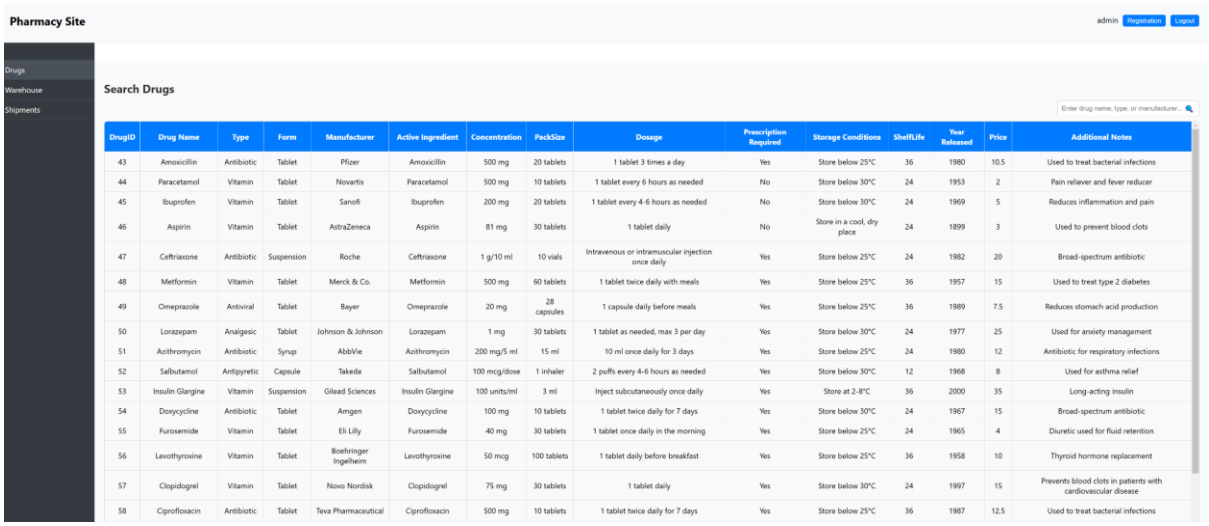
Логування та моніторинг

Для відстеження дій користувачів та виявлення підозрілої активності можна додати логування.

```
log.Printf("Користувач %s успішно увійшов", creds.Username)
```

3.2.3 Вкладка Drugs

Вкладка перегляду всіх ліків призначена для надання користувачам зручного інтерфейсу для перегляду інформації про всі доступні лікарські препарати в системі. Ця вкладка дозволяє користувачам шукати, фільтрувати та сортувати ліки за різними параметрами, такими як назва, тип, виробник, ціна та інші характеристики (рис. 3.7).



DrugID	Drug Name	Type	Form	Manufacturer	Active Ingredient	Concentration	PackSize	Dosage	Prescription Required	Storage Conditions	ShelfLife	Year Released	Price	Additional Notes
43	Amoxicillin	Antibiotic	Tablet	Pfizer	Amoxicillin	500 mg	20 tablets	1 tablet 3 times a day	Yes	Store below 25°C	36	1980	10.5	Used to treat bacterial infections
44	Paracetamol	Vitamin	Tablet	Novartis	Paracetamol	500 mg	10 tablets	1 tablet every 6 hours as needed	No	Store below 30°C	24	1953	2	Pain reliever and fever reducer
45	Ibuprofen	Vitamin	Tablet	Sanofi	Ibuprofen	200 mg	20 tablets	1 tablet every 4-6 hours as needed	No	Store below 30°C	24	1969	5	Reduces inflammation and pain
46	Aspirin	Vitamin	Tablet	AstraZeneca	Aspirin	81 mg	30 tablets	1 tablet daily	No	Store in a cool, dry place	24	1899	3	Used to prevent blood clots
47	Ceftriaxone	Antibiotic	Suspension	Roche	Ceftriaxone	1 g/10 ml	10 vials	Intravenous or intramuscular injection once daily	Yes	Store below 25°C	24	1982	20	Broad-spectrum antibiotic
48	Metformin	Vitamin	Tablet	Merck & Co.	Metformin	500 mg	60 tablets	1 tablet twice daily with meals	Yes	Store below 25°C	36	1957	15	Used to treat type 2 diabetes
49	Omeprazole	Antiviral	Tablet	Bayer	Omeprazole	20 mg	28 capsules	1 capsule daily before meals	Yes	Store below 25°C	36	1989	7.5	Reduces stomach acid production
50	Lorazepam	Analgesic	Tablet	Johnson & Johnson	Lorazepam	1 mg	30 tablets	1 tablet as needed, max 3 per day	Yes	Store below 30°C	24	1977	25	Used for anxiety management
51	Azithromycin	Antibiotic	Syrup	AbbVie	Azithromycin	200 mg/5 ml	15 ml	10 ml once daily for 3 days	Yes	Store below 25°C	24	1980	12	Antibiotic for respiratory infections
52	Salbutamol	Antipyretic	Capsule	Takeda	Salbutamol	100 mcg/dose	1 inhaler	2 puffs every 4-6 hours as needed	Yes	Store below 30°C	12	1968	8	Used for asthma relief
53	Insulin Glargine	Vitamin	Suspension	Gilead Sciences	Insulin Glargine	100 units/ml	3 ml	Inject subcutaneously once daily	Yes	Store at 2-8°C	36	2000	35	Long-acting insulin
54	Doxycycline	Antibiotic	Tablet	Amgen	Doxycycline	100 mg	10 tablets	1 tablet twice daily for 7 days	Yes	Store below 30°C	24	1967	15	Broad-spectrum antibiotic
55	Furosemide	Vitamin	Tablet	Eli Lilly	Furosemide	40 mg	30 tablets	1 tablet once daily in the morning	Yes	Store below 25°C	24	1965	4	Diuretic used for fluid retention
56	Levothyroxine	Vitamin	Tablet	Boehringer Ingelheim	Levothyroxine	50 mcg	100 tablets	1 tablet daily before breakfast	Yes	Store below 25°C	36	1958	10	Thyroid hormone replacement
57	Clopidogrel	Vitamin	Tablet	Novo Nordisk	Clopidogrel	75 mg	30 tablets	1 tablet daily	Yes	Store below 30°C	24	1997	15	Prevents blood clots in patients with cardiovascular disease
58	Ciprofloxacin	Antibiotic	Tablet	Teva Pharmaceutical	Ciprofloxacin	500 mg	10 tablets	1 tablet twice daily for 7 days	Yes	Store below 25°C	36	1987	12.5	Used to treat bacterial infections

Рисунок 3.7 – Вкладка Drugs

Основні можливості вкладки Drugs:

1. Перегляд списку ліків:

- Користувачі можуть переглядати повний список доступних ліків з основною інформацією, такою як назва, тип, виробник, ціна.
- Інформація відображається у вигляді таблиці.

2. Пошук ліків:

- Вкладка містить поле пошуку, де користувачі можуть вводити ключові слова для пошуку ліків за назвою, типом або виробником.

- Результати пошуку оновлюються в реальному часі, що дозволяє користувачам швидко знайти потрібні ліки.

3. Деталі ліків:

- Користувачі можуть переглядати детальну інформацію про кожен препарат.

- Детальна інформація включає всі дані, які є в таблиці Drugs бази даних [PHARMACY].

Нижче наведено запит на вивід даних

```go
func GetDrugs(c *gin.Context) {
    var drugs []models.Drug
    if err := database.DB.Preload("Type").Preload("Form").Preload("Manufacturer").Find(&drugs).Error; err != nil {
        c.JSON(http.StatusInternalServerError, gin.H{"error": "Failed to fetch drugs"})
        return
    }
    c.JSON(http.StatusOK, drugs)
}
```

3.2.4 Вкладка Warehouse

Вкладка Warehouse (рис. 3.8) призначена для управління записами конкретних партій лікарських препаратів. Вона надає зручний інтерфейс для перегляду, додавання та редагування даних про лікарські препарати, що зберігаються на складі. Доступ до функціоналу вкладки надається залежно від ролі користувача.

Рисунок 3.8 – Вкладка Warehouse

Функціональність вкладки Warehouse:

- Усі користувачі, включаючи оператора (operator), можуть переглядати список лікарських засобів.

- Інформація відображається у вигляді таблиці, яка містить такі дані:

1. Назва препарату (Drug Name)
2. Виробник (Manufacturer)
3. Номер партії (Batch Number)
4. Кількість (Quantity)
5. Сертифікат (Certificate)
6. Дата та час додавання (Date and Time)

- Додавання складських даних (Рисунок3.9). Додавати нові записи можуть лише адміністратори (admin) або виробники (manufacturer).

Для цього передбачена кнопка "Add Warehouse", яка відкриває модальне вікно "Add Warehouse Details".

У формі потрібно заповнити такі поля:

1. Drug Name: Вибір назви препарату з випадаючого списку.
2. Manufacturer Name: Виробник препарату (автоматично визначається або вибирається).
3. Batch Number: Унікальний номер партії препарату.
4. Quantity: Кількість одиниць препарату.
5. Certificate: Унікальний номер сертифіката на партію.
6. Date and Time: Час додавання даних (може бути автоматично заповнений або змінений вручну).

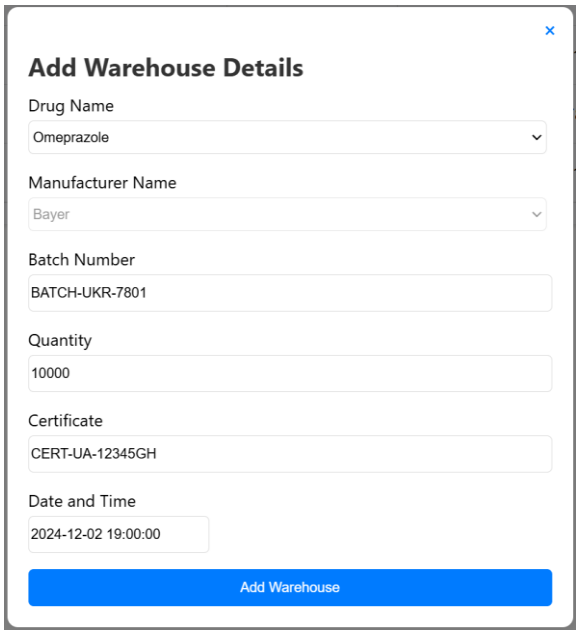


Рисунок 3.8 – Додавання нового запису до таблиці Warehouse

- Редагування складських записів. Редагувати дані можуть тільки адміністратори (admin) або відповідний виробник (manufacturer), що відповідає за препарат.

Форма редагування відкривається після натискання кнопки "Edit". У ній відображаються попередні дані, які можна змінити.

3.2.5 Дозволи та ролі для WebAPP

У системі управління ланцюгом постачання лікарських засобів кожен користувач має певну роль, яка визначає його дозволи на виконання різних дій. Ролі та дозволи забезпечують контроль доступу та безпеку даних, гарантуючи, що користувачі мають доступ лише до тих функцій та інформації, які необхідні для виконання їхніх обов'язків.

Ролі та їх дозволи:

1. Адміністратор (admin)
- Перегляд (show): Дозволено
- Редагування (edit): Дозволено
- Додавання (add): Дозволено
- Додавання користувача (add_user): Дозволено
2. Виробник (manufacturer)
- Перегляд (show): Дозволено
- Редагування (edit): Дозволено
- Додавання (add): Дозволено
3. Оператор (operator)
- Перегляд (show): Дозволено
- Редагування (edit): Дозволено
- Додавання (add): Дозволено
4. Гість (guest)
- Перегляд (show): Дозволено

Нижче наведена таблиця дозволів для кожної ролі табл. 1.

Табл. 1 - Таблиця ролей та дозволів

Дія	Адміністратор (admin)	Виробник (manufacturer)	Оператор (operator)	Гість (guest)
Перегляд (show)	✅	✅	✅	✅
Редагування (edit)	✅	✅	✅	❌
Додавання (add)	✅	✅	✅	❌
Додавання користувача (add_user)	✅	❌	❌	❌

Адміністратор (admin). Має повний доступ до всіх операцій у всіх розділах (Drugs, Warehouse, Shipments), включаючи додавання нових користувачів та всі додаткові функції.

Виробник (manufacturer). Має доступ до розділів Drugs та Warehouse, але не має доступу до розділу Shipments. Має доступ до деяких додаткових функцій, але не до всіх.

Оператор (operator). Має доступу до розділу Drugs, має лише доступ до перегляду в розділі Warehouse, але має повний доступ до розділу Shipments. Має доступ лише до базових додаткових функцій.

Гість (guest). Має доступ лише до перегляду інформації в усіх розділах (Drugs, Warehouse, Shipments). Додаткові функції недоступні.

Табл. 2 - Доступ до функціоналу вкладов відповідно до ролі

Розділ	Адміністратор (admin)	Виробник (manufacturer)	Оператор (operator)	Гість (guest)
Drugs	Повний доступ	Повний доступ	Повний доступ	Перегляд
Warehouse	Повний доступ	Повний доступ	Лише перегляд	Перегляд
Shipments	Повний доступ	Немає доступу	Повний доступ	Перегляд
Додаткові функції	Усі	Частково	Базові	Немає

У системі адміністратор має доступ до функції додавання нових

користувачів через спеціальну форму реєстрації (рис. 3.9).

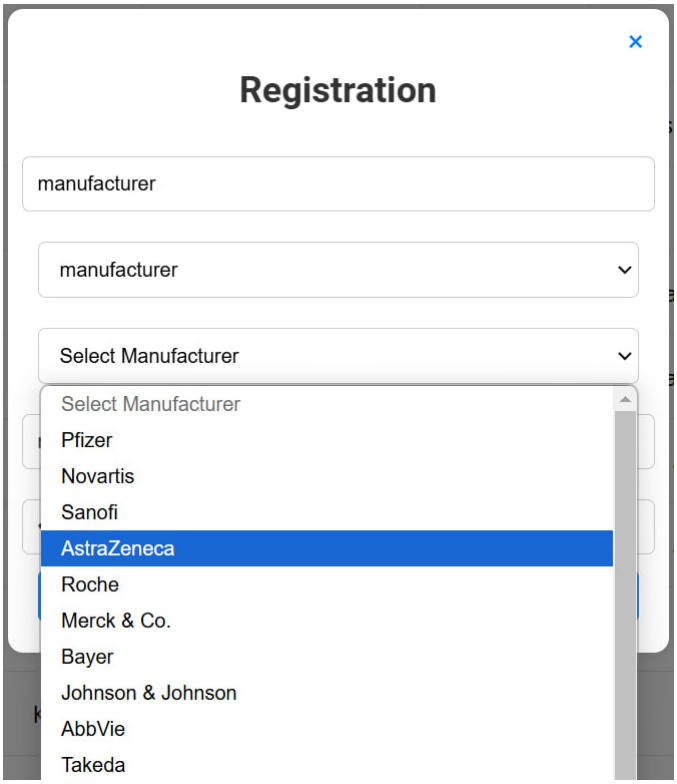


Рисунок 3.9 - Форма додавання нових користувачів (виробника)

У формі реєстрації доступні такі поля:

1) роль користувача (admin, manufacturer, operator, guest);
2) ім’я користувача;
3) email;
4) пароль;
5) додаткова інформація.

3.3Створення блокчейну

3.3.1 Створення проєкту

Створення блокчейн-проєкту починається з налаштування середовища розробки (рис. 3.10). Для цього використовуються такі інструменти, як Ganache і Truffle. Ganache дозволяє створювати локальний блокчейн для тестування, тоді як Truffle є фреймворком для розробки, тестування та розгортання смарт-контрактів.

Для роботи з Ganache достатньо завантажити з офіційного сайту та налаштувати проєкт.

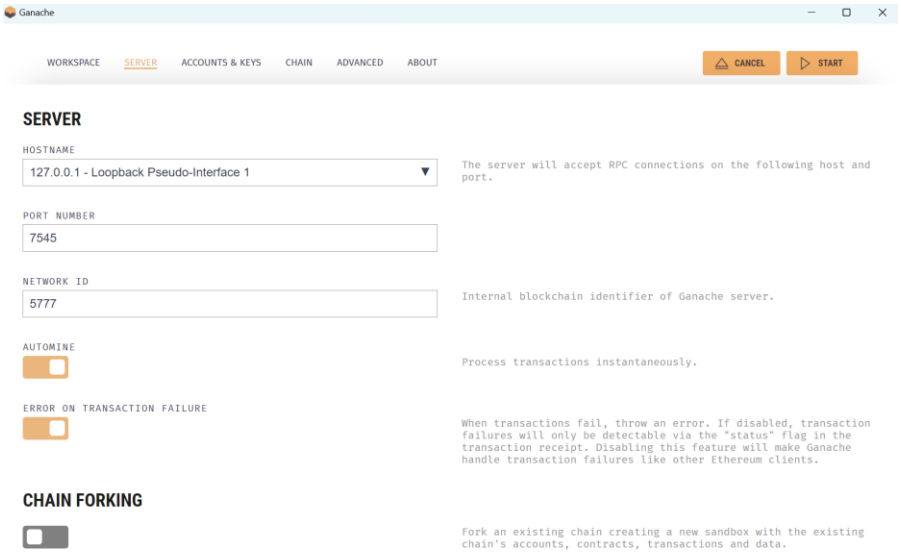


Рисунок 3.10 - Налаштування проєкту

Кроки:

1. Встановити Node.js та npm.
2. Встановити Truffle глобально за допомогою npm: npm install -g truffle
3. Встановити Ganache та створити новий рабочий простір.
4. Створити новий проєкт Truffle: truffle init
5. Налаштувати truffle-config.js для підключення до локального блокчейну Ganache.

backend > shipment-blockchain > truffle-config.js > ...

```
module.exports = {
  networks: {
      development: {
          host: "127.0.0.1", // локальний хост
          port: 7545,        // порт Ganache
          network_id: "*"    // будь-який network_id
      }
  },
  compilers: {
      solc: {
          version: "0.8.0" // версія компілятора Solidity
      }
  }
};
```

3.3.2 Створення смарт-контракту

Смарт-контракт - це самовиконуючий контракт з умовами агрегації коду. Він розробляється на мові Solidity та компілюється для розгортання на блокчейні Ethereum.

Принцип роботи. Смарт-контракт ShipmentTracker відстежує стани відправлення товарів. Він має функції для створення нових відправлень, оновлення їх стану та отримання інформації про відправлення.

Нижче наведено налаштування смарт-контраку на Solidity.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract ShipmentTracker {
    struct Shipment {
        string batchId;
        string fromCity;
        string toCity;
        string status;
        string timestamp;
    }
```

У смарт-контракті є чотири головні події: додавання нової поставки, оновлення поставки, оновлення статусу поставки, отримання історії змін.

```
    event ShipmentUpdated(string batchId, string status, string
timestamp);


    // Додавання нової поставки
    function addShipment(
        string memory batchId,
        string memory fromCity,
        string memory toCity,
        string memory status,
        string memory timestamp
    ) public {
        require(bytes(batchId).length > 0, "Batch ID cannot be empty");
        require(bytes(fromCity).length > 0, "From city cannot be
empty");
        require(bytes(toCity).length > 0, "To city cannot be empty");
        require(bytes(status).length > 0, "Status cannot be empty");
        require(bytes(timestamp).length > 0, "Timestamp cannot be
empty");
```

```
        Shipment memory newShipment = Shipment(
            batchId,
            fromCity,
            toCity,
            status,
            timestamp
        );
        shipments[batchId].push(newShipment);


        emit ShipmentAdded(batchId, fromCity, toCity, status,
timestamp);
```

```
    // Оновлення статусу вантажа
    function updateShipment(
        string memory batchId,
        string memory status,
        string memory timestamp
    ) public {
        require(bytes(batchId).length > 0, "BatchID cannot be empty");
        require(bytes(status).length > 0, "Status cannot be empty");
        require(shipments[batchId].length > 0, "Shipment does not
exist");

        Shipment memory updatedShipment = Shipment(
            batchId,
            shipments[batchId][0].fromCity,
            shipments[batchId][0].toCity,
            status,
            timestamp
        );

        shipments[batchId].push(updatedShipment);

        emit ShipmentUpdated(batchId, status, timestamp);
    }
```

```
    // Отримати історію змін
    function getShipmentHistory(
        string memory batchId
    ) public view returns (Shipment[] memory) {
        require(bytes(batchId).length > 0, "Batch ID cannot be empty");
        return shipments[batchId];
    }
}
```

Після створення смарт-контракту під’єднуємо його до Ganache (рис. 3.11).

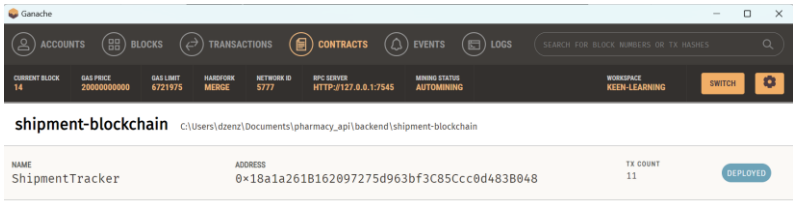


Рисунок 3.11 - Смарт-контракт у Ganache

3.3.3 Створення та підключення бібліотеки

Для взаємодії з блокчейном Ethereum з мови Go використовується бібліотека go-ethereum. Abigen дозволяє генерувати Go-бібліотеки для взаємодії зі смарт-контрактами.

Кроки:

1. Встановити Go та go-ethereum:

go get -u github.com/ethereum/go-ethereum

2. Встановити Abigen:

go get -u github.com/ethereum/go-ethereum/cmd/abigen

3. Створити ABI-файл для смарт-контракту:

truffle compile

4. Згенерувати Go-бібліотеку за допомогою Abigen:

abigen --abi contracts/ShipmentTracker_sol_ShipmentTracker.abi --pkg main --out ShipmentTrac

5. Підключення згенерованих контрактів

```go
package blockchain

import (
    "context"
    "log"

    "pharmacy/shipment-blockchain/contracts" // Generated ABI contracts

    "github.com/ethereum/go-ethereum/accounts/abi/bind"
    "github.com/ethereum/go-ethereum/common"
    "github.com/ethereum/go-ethereum/crypto"
    "github.com/ethereum/go-ethereum/ethclient"
)

var blockchainClient *ethclient.Client

func SetClient(client *ethclient.Client) {
    // Transaction setting
    privateKey := "cfd71a355c8e7edccf0014b37de8a49e3d14958ef9ccc9d9fffd14b8e38133f0"
    // Convert string to bytes
    keyBytes := common.Hex2Bytes(privateKey)
    // Parse private key
    privKey, err := crypto.ToECDSA(keyBytes)
    if err != nil {
        log.Fatalf("Помилка при парсинге приватного ключа: %v", err)
    }

    auth, err := bind.NewKeyedTransactorWithChainID(privKey, chainID)
    if err != nil {
        log.Fatalf("Помилка при створені транзактора: %v", err)
    }

    // Send data to blockchain
    tx, err := instance.AddShipment(auth, batch, fromCity, toCity, status, timestamp)
    if err != nil {
        return "", err
```

```
    blockchainClient = client
}


func AddShipmentToBlockchain(batch string, fromCity string, toCity
string, status string, timestamp string) (string, error) {
    // Connect to contract
    contractAddress :=
common.HexToAddress("0x18a1a261B162097275d963bf3C85Ccc0d483B048") //
Замените на адрес вашего контракта
    instance, err := contracts.NewContracts(contractAddress,
blockchainClient)
    if err != nil {
        return "", err
    }


    chainID, err := blockchainClient.ChainID(context.Background())
    if err != nil {
        log.Fatalf("Failed to get Chain ID: %v", err)
    }
    }


    log.Printf("Transaction sent: %s", tx.Hash().Hex())
    return tx.Hash().Hex(), nil
}
```

3.4 Взаємодія з QR-code

Генерація QR-коду на фронтенді дозволяє користувачам швидко отримувати доступ до інформації про відправлення. Для цього використовується бібліотека qrcode.react, яка дозволяє легко створювати QR-коди у React-застосунках. Дані для QR-коду надходять з бекенду у форматі JSON, і вони включають інформацію про відправлення та хеш-код.

Кроки:

1. Встановити необхідні бібліотеки

Для генерації QR-коду на фронтенді потрібно встановити бібліотеку qrcode.react:

npm install qrcode.react

2. Створити компонент для відображення QR-коду

Створимо компонент ShipmentQRCodeModal, який буде відображати модальне вікно з QR-кодом. Дані для QR-коду будуть надходити з бекенду у форматі JSON.

```
import React from 'react';
import QRCode from 'qrcode.react';



const ShipmentQRCodeModal = ({ shipment, hashcode, onClose }) => {
    // Об'єднуємо дані відправлення та хеш-код у один рядок для QR-коду
    const qrData = JSON.stringify({ shipment, hashcode });



    return (
        <div className="modal-container">
            <div className="modal-header">Add Shipment</div>
            <div className="qr-code">
                <QRCode value={qrData} size={200} />
            </div>
            <div className="modal-footer">
                <button className="ok-button" onClick={onClose}>
                    OK
                </button>
            </div>
        </div>
    );
};



export default ShipmentQRCodeModal;
```

Модуль зчитування QR-коду та прийняття поставки замовником забезпечує автоматизацію процесу зчитування QR-коду для отримання інформації про

поставку, її перевірку та підтвердження прийняття замовником. Основні компоненти та функції:

1. Інтерфейс користувача. Має дві основні кнопки: сканування коду та вибір зображення кода з галереї (рис. 3.12).

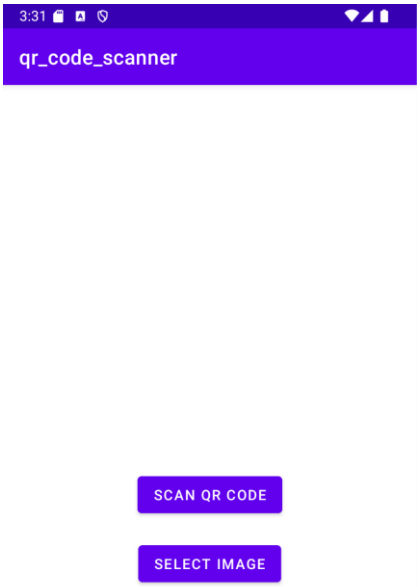


Рисунок 3.12 - Зчитування та завантаження QR-коду

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    scanButton = findViewById(R.id.button_scan)
    selectImageButton = findViewById(R.id.button_select_image)

    scanButton.setOnClickListener {
        Log.d(TAG, msg: "Scan button clicked")
        if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.CAMERA)
            != PackageManager.PERMISSION_GRANTED) {
            Log.d(TAG, msg: "Requesting camera permission")
            ActivityCompat.requestPermissions( activity: this, arrayOf(Manifest.permission.CAMERA),
                CAMERA_PERMISSION_REQUEST_CODE)
        } else {
            Log.d(TAG, msg: "Starting QR scan")
            startQRScan()
        }
    }

    selectImageButton.setOnClickListener {
        Log.d(TAG, msg: "Select image button clicked")
        if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.READ_MEDIA_IMAGES)
            != PackageManager.PERMISSION_GRANTED) {
            Log.d(TAG, msg: "Requesting storage permission")
            ActivityCompat.requestPermissions( activity: this, arrayOf(Manifest.permission.READ_MEDIA_IMAGES),
                READ_EXTERNAL_STORAGE_REQUEST_CODE)
        } else {
            Log.d(TAG, msg: "Opening gallery")
            openGallery()
        }
    }
}
```

2. Перевіряється доступ до камери та галереї. Якщо доступу немає, записуються відповідні дозволи.

```
override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
    when (requestCode) {
        CAMERA_PERMISSION_REQUEST_CODE -> {
            if (grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                Log.d(TAG, msg: "Camera permission granted")
                startQRScan()
            } else {
                Log.d(TAG, msg: "Camera permission denied")
                Toast.makeText( context: this, text: "Camera permission is required to scan QR codes", Toast.LENGTH_SHORT).show()
            }
        }
        READ_EXTERNAL_STORAGE_REQUEST_CODE -> {
            if (grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                Log.d(TAG, msg: "Storage permission granted")
                openGallery()
            } else {
                Log.d(TAG, msg: "Storage permission denied")
                Toast.makeText( context: this, text: "Storage permission is required to select an image", Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

3. Сканування QR-коду

Для зчитування QR-коду Використовується бібліотека JourneyApps Barcode Scanner.

```
private fun startQRScan() {
    val options = ScanOptions()
    options.setPrompt("Point at QR-code")
    options.setBeepEnabled(false)
    options.setOrientationLocked(false)
    options.setCaptureActivity(CustomCaptureActivity::class.java)
    barcodeLauncher.launch(options)
}
```

4. Зчитування QR-коду з галереї

Користувач може обрати зображення з галереї свого пристрою, яке обробляється бібліотекою ZXing для декодування.

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    if (requestCode == SELECT_IMAGE_REQUEST_CODE && resultCode == RESULT_OK && data != null) {
        Log.d(TAG, msg: "Image selected from gallery")
        val selectedImageUri: Uri = data.data!!
        val inputStream: InputStream? = contentResolver.openInputStream(selectedImageUri)
        val bitmap = BitmapFactory.decodeStream(inputStream)
        val qrCodeContent = decodeQRCode(bitmap)
        if (qrCodeContent != null) {
            Log.d(TAG, msg: "QR code content: $qrCodeContent")
            showAuthDialog(qrCodeContent)
        } else {
            Log.d(TAG, msg: "No QR code found in the image")
            Toast.makeText( context: this, text: "No QR code found in the image", Toast.LENGTH_LONG).show()
        }
    }
}
```

5. Обробка даних QR-коду

Отриманий текст розшифровується в об’єкт Java, витягуються ключові дані про поставку.

```
private fun decodeQRCode(bitmap: Bitmap?): String? {
    if (bitmap == null) return null
    val width = bitmap.width
    val height = bitmap.height
    val pixels = IntArray( size: width * height)
    bitmap.getPixels(pixels, offset: 0, width, x: 0, y: 0, width, height)
    val source = RGBLuminanceSource(width, height, pixels)
    val binaryBitmap = BinaryBitmap(HybridBinarizer(source))
    val reader = MultiFormatReader()
    return try {
        val result: Result = reader.decode(binaryBitmap)
        result.text
    } catch (e: NotFoundException) {
        Log.e(TAG, msg: "QR code not found", e)
        null
    }
}
```

6. Відображення необхідної інформації

Інформація про поставку виводиться в діалоговому вікні.

```
private fun showQRCodeInfoDialog(qrCodeContent: String) {
    val qrData = Gson().fromJson(qrCodeContent, QRData::class.java)
    val shipment = qrData.shipment
    val dialogView = layoutInflater.inflate(R.layout.dialog_qr_code_info, root: null)
    val qrCodeTextView: TextView = dialogView.findViewById(R.id.qrCodeContentText)
    val dialogMessage = buildString {
        append("Batch ID: ${shipment.BatchID}\n")
        append("From City: ${shipment.FromCity}\n")
        append("From Warehouse: ${shipment.FromWarehouseAddress}\n")
        append("To City: ${shipment.ToCity}\n")
        append("To Warehouse: ${shipment.ToWarehouseAddress}\n")
        append("To Country: ${shipment.ToCountry}\n")
        append("Creation Date: ${shipment.Timestamp ?: "N/A"}\n")
        append("Last Update: ${shipment.LastUpdate ?: "N/A"}\n")
        append("Status: ${shipment.Status.Status}\n")
        append("Status Last Update: ${shipment.Status.LastUpdate ?: "N/A"}\n")
    }
    qrCodeTextView.text = dialogMessage
    val dialog = AlertDialog.Builder( context: this)
        .setView(dialogView)
        .setCancelable(false)
        .create()
    val acceptButton: Button = dialogView.findViewById(R.id.btnAccept)
    acceptButton.setOnClickListener {
        dialog.dismiss()
        sendQRCodeData(qrCodeContent)
    }
    val closeButton: Button = dialogView.findViewById(R.id.btnClose)
    closeButton.setOnClickListener {
        dialog.dismiss()
    }
    dialog.show()
}
```

7. Підтвердження прийняття поставки

Після підтвердження дані відправляються на сервер через API.

```
private fun sendQRCodeData(qrCodeContent: String) {
    val token = "Bearer $token"
    Log.d(TAG, msg: "Authorization header: $token")

    val qrData = Gson().fromJson(qrCodeContent, QRData::class.java)

    RetrofitClient.api.proceedQR(token, qrData).enqueue(object : Callback<ApiResponse> {
        override fun onResponse(call: Call<ApiResponse>, response: Response<ApiResponse>) {
            if (response.isSuccessful) {
                val apiResponse = response.body()
                Log.d(TAG, msg: "QR Code proceeded: ${apiResponse?.message}")
                Toast.makeText( context: this@MainActivity, text: "QR Code proceeded: ${apiResponse?.message}", Toast.LENGTH_LONG).show()
            } else {
                Log.e(TAG, msg: "Error: ${response.message()}")
                Toast.makeText( context: this@MainActivity, text: "Error: ${response.message()}", Toast.LENGTH_LONG).show()
            }
        }

        override fun onFailure(call: Call<ApiResponse>, t: Throwable) {
            Log.e(TAG, msg: "Failure: ${t.message}")
            Toast.makeText( context: this@MainActivity, text: "Failure: ${t.message}", Toast.LENGTH_LONG).show()
        }
    })
}
```

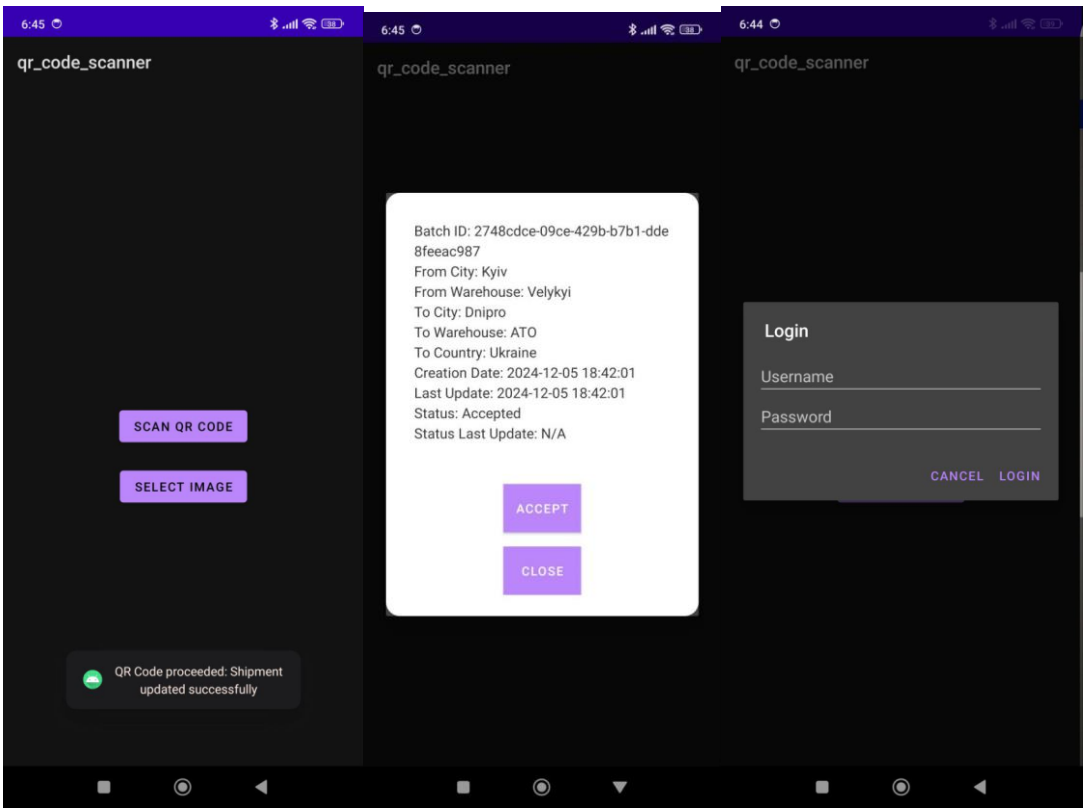


Рисунок 3.13 - Перехід від сканування QR-коду до авторизації

3.4.1 Опис роботи API для обробки QR-кодів

Функція ProcessQRCode обробляє дані зчитаного QR-коду, перевіряє їх відповідність базі даних і оновлює статус поставки. Основні етапи роботи описані нижче.

1. Прийом та перевірка вхідних даних

API отримує JSON-об’єкт, який містить HashCode та інформацію про поставку (Shipment). Перевіряє коректність формату JSON та обов'язкових полів (BatchID, FromCity, ToCity, ToCountry). Якщо дані некоректні, повертає помилку 400 Bad Request.

```go
// Bind JSON data to qrCode struct
if err := c.ShouldBindJSON(&qrCode); err != nil {
    log.Println("Error binding JSON:", err.Error())
    c.JSON(http.StatusBadRequest, gin.H{"error": "Invalid QR code data"})
    return
}
// Validate required shipment fields
missingFields := []string{}
if qrCode.Shipment.BatchID == "" {
    missingFields = append(missingFields, "BatchID")
}
if qrCode.Shipment.FromCity == "" {
    missingFields = append(missingFields, "FromCity")
}
if qrCode.Shipment.ToCity == "" {
    missingFields = append(missingFields, "ToCity")
}
if qrCode.Shipment.ToCountry == "" {
    missingFields = append(missingFields, "ToCountry")
}
```

2. Перевірка відповідності даних у базі

API перевіряє, чи існує поставка з відповідним BatchID у базі. Порівнює дані з QR-коду з даними в базі. Якщо є розбіжності, повертає список невідповідних полів.

```go
// Fetch existing shipment from the database
var existingShipment models.Shipment
if err := database.DB.Preload("Status").First(&existingShipment, "BatchID = ?", qrCode.Shipment.BatchID).Error; err != nil {
    log.Println("Shipment not found:", err.Error())
    c.JSON(http.StatusNotFound, gin.H{"error": "Shipment not found"})
    return
}

// Compare QR data with database data
mismatchedFields := compareShipments(qrCode.Shipment, existingShipment)
if len(mismatchedFields) > 0 {
    c.JSON(http.StatusBadRequest, gin.H{
        "error":             "QR code data does not match database records",
        "mismatched_fields": mismatchedFields,
    })
    return
}
```

3. Перевірка транзакції в блокчейні

Використовує функцію blockchain.VerifyTransaction для перевірки валідності транзакції за HashCode. У разі помилки повертає відповідний статус (400 Bad Request або 500 Internal Server Error).

```
// Verify transaction on blockchain
transactionValid, err := blockchain.VerifyTransaction(qrCode.HashCode)
if err != nil {
    log.Println("Blockchain verification failed:", err.Error())
    c.JSON(http.StatusInternalServerError, gin.H{"error": "Failed to verify transaction on blockchain"})
    return
}
if !transactionValid {
    c.JSON(http.StatusBadRequest, gin.H{"error": "Invalid or unconfirmed blockchain transaction"})
    return
}
```

4. Оновлення статусу поставки

Якщо всі перевірки успішні, API оновлює статус поставки в блокчейні та локальній базі даних. Статус оновлюється до Delivered, а час прибуття (ArrivalTime) та останнього оновлення (LastUpdate) синхронізуються.

```
// Update shipment status on blockchain
currentTime := formatCurrentTime()
txHash, err := blockchain.UpdateShipmentOnBlockchain(qrCode.Shipment.BatchID)
if err != nil {
    log.Println("Blockchain update failed:", err.Error())
    c.JSON(http.StatusInternalServerError, gin.H{"error": "Failed to update shipment on blockchain"})
    return
}

// Synchronize updated data in the local database
existingShipment.Status.Status = "Delivered"
*existingShipment.Status.ArrivalTime = time.Now()
existingShipment.Status.LastUpdate = &currentTime
existingShipment.LastUpdate = &currentTime
```

5. Оновлення в БД.

```
// Save updated status in the database
if err := database.DB.Save(&existingShipment.Status).Error; err != nil {
    log.Println("Failed to update shipment status in database:", err.Error())
    c.JSON(http.StatusInternalServerError, gin.H{"error": "Failed to update shipment status in database"})
    return
}

// Save the updated shipment in the database
if err := database.DB.Save(&existingShipment).Error; err != nil {
    log.Println("Failed to update shipment in database:", err.Error())
    c.JSON(http.StatusInternalServerError, gin.H{"error": "Failed to update shipment in database"})
    return
}
```

6. Повернення успішної відповіді

API повертає статус 200 OK із повідомленням про успішне оновлення та хеш транзакції в блокчейні.

Висновки до розділу:

У третьому розділі кваліфікаційної роботи було реалізовано та досліджено основні технічні аспекти розробки системи управління ланцюгом постачання лікарських препаратів з використанням технології блокчейн. У рамках дослідження вдалося забезпечити інтеграцію різних компонентів системи та створити повністю функціональний прототип, що демонструє можливості оптимізації логістичних процесів за рахунок інноваційних підходів.

Основні результати роботи включають:

1. Розробка архітектури системи:

- Створено багаторівневу базу даних для зберігання детальної інформації про поставки, лікарські препарати, виробників та користувачів системи.
- Реалізовано окрему базу даних для авторизації користувачів, що дозволяє забезпечити високий рівень безпеки та контролю доступу до конфіденційних даних.
- Забезпечено можливість інтеграції бази даних із зовнішніми сервісами для підвищення функціональності системи.

2. Поєднання блокчейн-технологій із веб-додатком:

- Впроваджено смарт-контракти, які автоматизують перевірку виконання умов угод між учасниками ланцюга постачання, реєструють транзакції та забезпечують їх незмінність і прозорість.
- Інтегровано QR-коди для відстеження кожної партії лікарських препаратів, що надає кінцевим користувачам можливість перевірити автентичність ліків та ознайомитися з їх історією на всіх етапах ланцюга постачання.
- Реалізовано інтерфейси веб-додатка для управління ключовими аспектами ланцюга постачання, такими як інформація про поставки, склади та доступні препарати.

3. Підвищення безпеки та відстежуваності:

- Розроблена система дозволяє мінімізувати ризики підробки лікарських засобів за рахунок прозорості процесів та детальної фіксації всіх транзакцій у блокчейні.

- Завдяки впровадженню ролей користувачів та смарт-контрактів забезпечено контроль за діями кожного учасника системи, що мінімізує людські помилки та шахрайство.

4. Ефективність інтеграції технологій:

- Система демонструє високий рівень автоматизації процесів у ланцюгу постачання завдяки використанню смарт-контрактів, баз даних та веб-додатка.

Впроваджена архітектура легко масштабується, дозволяючи адаптувати її до потреб різних фармацевтичних компаній та учасників ланцюга постачання.

Результати роботи підтвердили, що запропонована система є ефективним інструментом для забезпечення прозорості, безпеки та відстежуваності процесів у ланцюзі постачання лікарських препаратів. Її впровадження дозволить значно покращити якість управління, зменшити ризики підробки ліків, оптимізувати логістичні витрати та підвищити довіру кінцевих споживачів до медичної продукції.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено систему автоматизації управління ланцюгом постачання лікарських препаратів на основі технології блокчейн, яка відповідає сучасним викликам у сфері медичної логістики. Основними етапами виконаної роботи є:

1. Дослідження актуальності проблеми:

- Проаналізовано складність сучасного ланцюга постачання лікарських засобів, включаючи багатоступеневість, потребу в прозорості, ризики підробки ліків, низьку інтеграцію між учасниками та недостатню ефективність традиційних систем управління.

- Розглянуто можливості застосування блокчейн-технологій як інноваційного інструменту для вирішення цих проблем.

2. Розробка архітектури системи:

- Побудовано багаторівневу структуру, яка поєднує бази даних, блокчейн-платформу, смарт-контракти та веб-додаток.

- Забезпечено збереження й обробку інформації про поставки, препарати, виробників та користувачів з високим рівнем безпеки.

3. Реалізація основних функцій системи:

- Впроваджено смарт-контракти для автоматизації ключових операцій, зокрема перевірки виконання умов угод, реєстрації транзакцій та забезпечення прозорості взаємодії між учасниками.

- Інтегровано функцію відстеження поставок із використанням QR-кодів, що дозволяє кінцевим користувачам перевіряти автентичність ліків та їхній рух у ланцюзі постачання.

4. Оцінка ефективності системи:

- Тестування підтвердило високу ефективність розробленої системи у забезпеченні прозорості, автоматизації процесів та відстежуваності постачань.

- Система дозволяє знизити ризики підробки ліків, покращити взаємодію між учасниками та забезпечити доступність актуальної інформації для кінцевих споживачів.

Рекомендації щодо вдосконалення функціоналу системи

1. Розширення функціоналу відстеження поставок:

- Впровадження трекера поставок у реальному часі, який відображатиме місцезнаходження вантажу на карті.

- Інтеграція з GPS-модулями для моніторингу ключових параметрів транспортування (температурний режим, вологість тощо).

2. Автоматизація управління процесами:

- Розробка алгоритмів для прогнозування термінів доставки на основі поточного стану поставок і зовнішніх факторів, таких як погода чи завантаженість транспортних маршрутів.

- Додавання функції автоматичного сповіщення користувачів про зміни статусу поставки або виявлення відхилень у транспортних умовах.

3. Покращення інтерактивності користувацького інтерфейсу:

- Створення дашборду для відображення ключових показників у реальному часі (наприклад, статус поставок, інформація про порушення транспортування).

- Розробка мобільного додатка для забезпечення доступу до системи зі смартфонів та планшетів.

4. Розширення функцій відстеження даних у ланцюгу:

- Впровадження автоматизованих звітів для учасників ланцюга постачання, які міститимуть аналіз ефективності доставки та статистику виконання умов транспортування.

Впровадження цих рекомендацій дозволить значно покращити функціональність системи, підвищити її інтерактивність та адаптивність до потреб учасників ланцюга постачання, забезпечуючи вищий рівень надійності та ефективності процесів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Hayes A. Blockchain facts. *investopedia.com*. URL: https://www.investopedia.com/terms/b/blockchain.asp (date of access: 15.07.2024).

2. S R. A. What is Blockchain Technology? How Does Blockchain Work?. *Simplilearn.com*. URL: https://www.simplilearn.com/tutorials/blockchain-tutorial/blockchain-technology (date of access: 15.07.2024).

3. THE JPEG-BLOCKCHAIN FRAMEWORK FOR GLAM SERVICES / D. Bhowmik et al. URL: https://www.researchgate.net/publication/324834631_The_Jpeg-Blockchain_Framework_For_Glam_Services (date of access: 15.07.2024).

4. Pros and Cons of Blockchain Technology: An Overview | Shardeum. Shardeum | EVM based Sharded Layer 1 Blockchain. URL: https://shardeum.org/blog/pros-and-cons-of-blockchain/ (date of access: 16.07.2024).

5. Jha S. A Comprehensive Overview of Blockchain Types | Simplilearn. Simplilearn.com. URL: https://www.simplilearn.com/tutorials/blockchain-tutorial/types-of-blockchain (date of access: 17.07.2024).

6. «Холодовая цепь» | Фармацевтическая отрасль. Фармацевтична галузь | Об'єктивний інформаційний канал для професіоналів. URL: https://promoboz.com/ru/journal/2023/1-94-2023/holodovyj-lantsyug/ (дата звернення: 18.07.2024).

7. Холодовий ланцюг. Центр Валідації. URL: https://val-center.com/ua/post/cold-chain (дата звернення: 18.07.2024).

8. Team C. What is Cold Chain Management? | Cold Chain Technologies. Thermal Packaging & Insulated Shippers | Cold Chain Technologies. URL: https://www.coldchaintech.com/blog/knowledge/what-is-cold-chain-management (date of access: 19.07.2024).

9. Team C. What is Cold Chain Management? | Cold Chain Technologies. Thermal Packaging & Insulated Shippers | Cold Chain Technologies. URL: https://www.coldchaintech.com/blog/knowledge/what-is-cold-chain-management (date

of access: 19.07.2024).

10. An integrated sustainable medical supply chain network during COVID-19. Engineering Applications of Artificial Intelligence. 2021. URL: https://www.sciencedirect.com/science/article/pii/S095219762100035X (date of access: 23.07.2024).

11. Dynamic distributed iterative computational model for payment information management in shared logistics using blockchain-assisted Internet of Things approach. 2021. URL: https://www.researchgate.net/publication/353036831_Dynamic_distributed_iterative_computational_model_for_payment_information_management_in_shared_logistics_using_blockchain-assisted_Internet_of_Things_approach (date of access: 25.07.2024).

12. Khurshid A. Applying Blockchain Technology to Address the Crisis of TrustDuring the COVID-19 Pandemic. 2020. URL: https://www.researchgate.net/publication/344144279_The_crisis_of_trust_in_COVID-19_pandemic_can_blockchain_technology_help_Preprint (date of access: 27.07.2024).

13. Governance on the Drug Supply Chain via Gcoin Blockchain / J.-H. Tseng et al. International Journal of Environmental Research and Public Health. 2018. Vol. 15, no. 6. P. 1055. URL: https://doi.org/10.3390/ijerph15061055 (date of access: 05.08.2024).

14. PharmaChain: Blockchain-based drug supply chain provenance verification system / Sarmistha Sarna Gomasta et al. 2023. URL: https://www.cell.com/heliyon/fulltext/S2405-8440(23)05165-4?_returnURL=https://linkinghub.elsevier.com/retrieve/pii/S2405844023051654?showall=true (date of access: 15.08.2024).

15. Blockchain for drug traceability: Architectures and open challenges / M. Uddin et al. Health Informatics Journal. 2021. Vol. 27, no. 2. P. 146045822110112. URL: https://doi.org/10.1177/14604582211011228 (date of access: 18.08.2024).

16. What is a Multi Agent System - Relevance AI. Relevance AI - Build your AI Workforce - AI for Business. URL: https://relevanceai.com/learn/what-is-a-multi-agent-system (date of access: 20.08.2024).

17. Multi-agent Systems. Site not found · GitHub Pages. URL: https://langchain-ai.github.io/langgraph/concepts/multi_agent/ (date of access: 20.08.2024).

18. Kalpesh Lad, M. Ali Akber Dewan, Fuhua Lin. Trust Management for Multi-Agent Systems Using Smart Contracts. 2020. URL: https://www.researchgate.net/profile/M-Dewan/publication/343486098_Trust_Management_for_Multi-Agent_Systems_Using_Smart_Contracts/links/5f3addb5a6fdcccc43d0e166/Trust-Management-for-Multi-Agent-Systems-Using-Smart-Contracts.pdf (date of access: 24.08.2024).

19. Blockchain-Based Architecture: A MAS Proposal for Efficient Agri-Food Supply Chains / Yeray Mezquita Martín et al. 2020. URL: https://www.researchgate.net/publication/333968344_Blockchain-Based_Architecture_A_MAS_Proposal_for_Efficient_Agri-Food_Supply_Chains (date of access: 25.08.2024).

20. Mahendrian D. N., VS K. R. Fake Product Detection Using QR Code. International Journal of Research Publication and Reviews. 2024. Vol. 5, no. 4. P. 1093–1096. URL: https://doi.org/10.55248/gengpi.5.0424.0931 (date of access: 26.08.2024).

21. SR. Enhancing Security with TOTP, Blockchain, and QR Codes: Preventing Cloning in Modern Applications. Medium. URL: https://medium.com/@deepml1818/enhancing-security-with-totp-blockchain-and-qr-codes-preventing-cloning-in-modern-applications-1f141934b642 (date of access: 26.08.2024).

22. Kademeteme E., Bvuma S. Using Blockchain Technology to Improve the Integrity and Transparency of Procurement Processes between SMMEs and Government: A Systematic Literature Review. The Journal of The British Blockchain Association. 2023. Vol. 7, no. 1. P. 1–12. URL: https://doi.org/10.31585/jbba-7-1-(1)2024 (date of access: 27.08.2024).

23. Gumerov E. A. DEVELOPMENT TRENDS OF BLOCKCHAIN SYSTEMS. Educational resources and technology. 2019. P. 59–63. URL:

https://doi.org/10.21777/2500-2112-2019-2-59-63 (date of access: 27.08.2024).

24. What is Ganache Blockchain. 101 Blockchains. URL: https://101blockchains.com/ganache-blockchain/ (date of access: 28.08.2024).

25. Жорняк А. ЗАКОНОДАВЧЕ (ПРАВОВЕ) РЕГУЛЮВАННЯ ТЕХНОЛОГІЇ БЛОКЧЕЙН. СУЧАСНІ ВИКЛИКИ ТА МІЖНАРОДНИЙ ДОСВІД. Open Science and Innovation. 2024. Т. 1, № 1. URL: https://doi.org/10.62405/osi.2024.01.06 (дата звернення: 29.08.2024).

26. Маринич І. А., Тронь В. В. Методичні рекомендації до виконання кваліфікаційної роботи магістра для студентів спеціальності 151 “Автоматизація та комп’ютерно-інтегровані технології”. Кривий Ріг : Видавничий центр КНУ, 2022. 50с.

27. ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура і правила оформлення. Київ, ДП «УкрННЦ», 2015. 26с. (Інформація та документація).

28. ДСТУ 8302:2015. Бібліографічне посилання. Загальні вимоги та правила складання Київ, ДП «УкрННЦ», 2016. 16 с. (Інформація та документація).

29. ДСТУ 3582:2013. Бібліографічний опис. Скорочення слів і словосполучень в українській мові. Загальні вимоги та правила. Київ, ДП «УкрННЦ», 2013. 23 с. (Інформація та документація)

30. ДСТУ 3651.0-97 Метрологія. Одиниці фізичних величин. Основні одиниці фізичних величин Міжнародної системи одиниць. Основні положення, назви та позначення Київ, Держстандарт України, 1998. 27 с.