

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»  
Кафедра інформатики факультету інформатики

## **ТОКЕНІЗАЦІЯ CLEANTECH ЕФЕКТУ ПІДПРИЄМСТВА ЗА ДОПОМОГОЮ ТЕХНОЛОГІЇ БЛОКЧЕЙН**

**Текстова частина до курсової роботи  
за спеціальністю „Комп’ютерні науки” 121**

Керівник курсової роботи

ас.

Гороховський К.С.

\_\_\_\_\_

*(підпис)*

“\_\_” \_\_\_\_\_ 2021 р.

Виконав студент 3 курсу

Холодов Д.В.

“\_\_” \_\_\_\_\_ 2021 р.

Київ 2021

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»  
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ  
Зав. Кафедри інформатики,  
Гороховський С.С.  
(підпис) \_\_\_\_\_

„\_\_” \_\_\_\_\_ 2021 р.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**  
на курсову роботу

студенту 3-го курсу, факультету інформатики, Холодову Дмитру Віталійовичу

ТЕМА Токенізація cleantech ефекту підприємства за допомогою технології блокчейн

Вихідні дані: Створення проєктної заявки, підтвердження cleantech ефекту, перегляд проєктів, продаж токенів інвестору

Зміст ТЧ до курсової роботи:

Перелік умовних позначень

Анотація

Вступ

Розділ 1: Аналіз предметної області. Постановка завдання курсової роботи.

Розділ 2: Теоретичні відомості про розробку блокчейн систем.

Розділ 3: Опис реалізації програмного продукту.

Висновки

Список літератури

Додатки

Дата видачі „\_\_” \_\_\_\_\_ 2021 р. Керівник \_\_\_\_\_  
(підпис)

Завдання отримав \_\_\_\_\_  
(підпис)

**Тема:** Токенізація cleantech ефекту підприємства за допомогою технології блокчейн

**Календарний план виконання роботи:**

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на дипломну роботу.	09.10.2020	
2.	Огляд технічної літератури за темою роботи.	12.11.2020	
3.	Виконання аналізу наявних рішень на ринку.	25.12.2020	
4.	Розгортання блокчейн мережі	30.01.2021	
5.	Розробка схеми бази даних	24.02.2021	
6.	Програмування смарт-контракту	01.03.2021	
7.	Реалізація веб-додатку. Написання серверу.	07.04.2021	
8.	Реалізація веб-додатку. Створення сторінок.	24.04.2021	
9.	Надання сторінкам функціоналу та стилів.	05.05.2021	
10.	Аналіз отриманих результатів з керівником	28.05.2021	
11.	Написання пояснювальної роботи.	01.05.2021	
12.	Попередній захист курсової роботи.	14.05.2021	
13.	Корегування роботи за результатами попереднього захисту.	15.05.2021	
14.	Створення слайдів для доповіді та написання доповіді.		
15.	Остаточне оформлення пояснювальної роботи та слайдів.	17.05.2021	
16.	Захист курсової роботи	30.05.2021	

Студент Холодов Д.В.

Керівник Гороховський К.С.

“ \_\_\_\_\_ ”

## ЗМІСТ

<b>Перелік умовних позначень:</b>	5
Анотація	6
<b>Вступ</b>	7
<b>РОЗДІЛ 1. Дослідження предметної області</b>	9
1.1 Аналіз сучасного стану питання та обґрунтування теми	9
1.2. Огляд існуючих рішень на ринку	11
1.3 Постановка завдання	14
1.4 Висновки до розділу 1	14
<b>РОЗДІЛ 2. Ключові конструктивні особливості фреймворку Hyperledger Fabric</b>	15
2.1 Hyperledger Fabric як корпоративне блокчейн рішення	15
2.2 Fabric Certification Authority	18
2.3 Висновки до розділу 2	22
<b>РОЗДІЛ 3. Опис підходу та реалізація</b>	23
3.1 Аналіз технічного завдання	23
3.2 Реалізація токенизації cleantech ефекту	27
3.3 Обґрунтування вибору засобів розробки	33
3.4 Тестування програми і результати її виконання	36
3.5 Висновки до розділу 3	43
<b>Висновки</b>	44
Список використаної літератури	45
Додаток А. Схема блокчейн мережі	47
Додаток Б. Взаємодія з чейнкодом на сервері	48
Додаток В. Схема бази даних	49
Додаток	50
Додаток	51
Додаток Г. Взаємодія з базою даних LevelDB	52

**Перелік умовних позначень:**

1. Cleantech – загальний термін, що охоплює клас інвестиційних активів, технологій та секторів бізнесу, які включають в себе чисту енергетику, екологічну продукцію та екологічні послуги.
2. Токен – інструмент для передачі права власності в цифровому вигляді.
3. Токенізація - це конвертація будь-якого активу в токени.
4. Зелені активи – фінансовий інструмент, що використовується для отримання фінансування проєктами, які здійснюють позитивний вплив на довкілля, реалізований у вигляді токенів на блокчейні.
5. Зелені проєкти – проєкти з позитивним екологічним впливом.
6. Смарт-контракт - це програмне забезпечення, яке визначає умови договору між користувачами системи. Код і угоди, що містяться в ньому, існують у розподіленій, децентралізованій блокчейн-мережі.

## Анотація

Курсова робота присвячена проблемам інвестицій в екологічно чисті проєкти. За мету курсової роботи було поставлено розробка механізму стимулювання інвестицій в проєкти з позитивним екологічним впливом. За метод вирішення проблеми було обрано випуск токенів у власному приватному блокчейні з подальшим продажом токенів інвестору.

Було розроблено платформу токенизації cleantech ефекту підприємства за допомогою технології блокчейн. Блокчейн, як децентралізована система, задовільнила всі поставлені вимоги до проєкту. Проаналізувавши рішення на ринку, для створення власної блокчейн системи було обрано фреймворк Hyperledger Fabric. Текстова частина курсової містить опис дослідження проблем інвестицій в екологічно чисті проєкти та спосіб їх вирішення. Текстова частина курсової також містить теоретичні відомості про фреймворк Hyperledger Fabric.

Ключові слова: токенизація, cleantech, блокчейн, Hyperledger Fabric, електронний сертифікат, екологічний вплив, смарт-контракт

## Вступ

Протягом тривалого часу в Україні приймалися управлінські рішення, що не враховували операційні показники споживання енергії під час експлуатації обладнання та викидів вуглекислого газу. Поступово це призвело до зростання енергоємності суб'єктів господарювання в Україні, внаслідок чого щорічні втрати економіки оцінюються експертами в USD 9,2 млрд.

Зростаюча кількість інвесторів шукає можливості зробити інвестиції у зелені проекти. Компанії, що не надають відповідну інформацію про екологічні проекти, створюють додаткову пошукову вартість, що встановлює перешкоди у фінансуванні зелених проектів. Окрім цього, інвестори стурбовані можливістю зрозуміти та продемонструвати позитивний вплив своїх зелених інвестицій. Однак, коли компанії або проекти надають екологічну інформацію, відсутнє стале та надійне "маркування" зелених активів. Це призвело до підвищеного попиту на зелені активи випущені у вигляді токенів на блокчейні, оскільки вони являють собою один з небагатьох надійних інвестиційних інструментів, які дотримуються певних принципів і вимагають перевірки третьою стороною проектів щодо екологічного використання інвестицій [1].

Виходячи з попиту на нові інвестиційні інструменти за мету даної роботи було поставлено дослідження токенизації нематеріальних активів підприємства та створення платформи токенизації cleantech ефекту підприємства, яка дозволить компаніям надавати гарантії довіри своїм інвесторам шляхом сприяння сторонній перевірці проектів та надасть інвесторам доступ до інформації про компанії та їх проекти, тим самим забезпечить прозорість і впевненість інвесторів, що гроші спрямовуються на фінансування саме екологічно чистих проектів та сприятиме притоку інвестицій в зелені проекти.

Завдання: Розробка блокчейн системи, що забезпечить компаніям можливість заявляти про створення зелених проектів, можливість підтвердження їх

потенційного позитивного екологічного впливу незалежними спеціалістами з подальшим випуском зазначеної компанією кількості токенів з метою залучення інвестицій в проєкт. Надасть інвесторам можливість отримати доступ до прозорих та незмінних даних про проєкти.

Об'єктом дослідження є токенизація нематеріальних активів підприємства, а саме екологічного ефекту.

Методом дослідження є створення приватного блокчейну за допомогою фреймворку Hyperledger Fabric.

Робота складається з анотації, вступу, трьох основних розділів, кожен з яких містить підрозділи, висновків, списку літератури та додатків.

Перший розділ присвячено аналізу сучасного стану проблеми фінансування екологічних проєктів. Наведено огляд категорій екологічних проєктів. Коротко розглянуто існуючі рішення на ринку.

Другий розділ описує технічні особливості розробки блокчейн систем у фреймворку Hyperledger Fabric. Наведено коротку характеристику моделі даних в системі. Також описані загальні відомості про центр сертифікації фреймворку Hyperledger Fabric.

Третій розділ присвячено реалізації проєкту. Описані функціональні вимоги до системи та вимоги до даних. Описано процес токенизації cleantech ефекту. Виділено та проілюстровану роботу основних модулів програми. Обґрунтовано вибір засобів розробки. Наведено результати тестування системи з описанням кожного кроку.



## РОЗДІЛ 1. Дослідження предметної області

### 1.1 Аналіз сучасного стану питання та обґрунтування теми

Екологічні переваги продуктів чи послуг Cleantech це вищий рівень переробки та енергоефективності, зменшення використання та впливу на природні ресурси, зниження рівня шкідливих для довкілля викидів [2].

Фірма може розглядатися як “Cleantech”, якщо вона допомагає захищати навколишнє середовище, виробляючи продукцію, послуги або процеси, що зменшують споживання невідновлюваних ресурсів та / або створюють значно менше відходів, ніж загальноприйняті пропозиції на ринку [3].

Категорії "зелених проєктів" включають, але не обмежуються:

а) відновлювана енергія (включаючи виробництво, передачу, прилади та вироби);

б) енергоефективність (наприклад, у нових та відремонтованих будівлях, накопичувачі енергії, централізоване опалення, розумні мережі, побутова техніка);

в) запобігання забрудненню та контроль (включаючи зменшення викидів у атмосферу, контроль парникових газів, санація ґрунту, зменшення відходів, переробка відходів);

г) екологічно стійке управління життям, природними ресурсами та землекористуванням (в тому числі екологічно стійке сільське господарство, тваринне господарство, кліматично розумні ферми, екологічно стійке рибальство та аквакультура, екологічно стійке лісове господарство, включаючи лісовідновлення та збереження або відновлення природних ландшафтів);

г) чистий транспорт (наприклад, електричний, гібридний, громадський, залізничний, немоторизований, інфраструктура для екологічно чистих транспортних засобів та зменшення шкідливих викидів);

д) стале управління водними ресурсами (включаючи стійку інфраструктуру для питної води, очищення стічних вод) [4].

Зелені активи – найсучасніший спосіб монетизації позитивного впливу на довкілля, реалізовані у вигляді токенів на блокчейні. Використовуються для отримання фінансування проектами, які здійснюють позитивний екологічний вплив. Використовуються інвесторами для торгівлі на міжнародних біржах. У 2013 році зареєстровано перший муніципальний проект зелених активів у штаті Массачусетс, США.

Поряд із широким розвитком світової економіки, проблеми, пов'язані з охороною навколишнього середовища, стали більш нагальними, ніж будь-коли раніше. Ці тенденції викликають гострий науковий інтерес до економічних питань та сталого розвитку, привертаючи таким чином увагу країн з різних куточків світу [5].

В останнє десятиліття ідея сталого розвитку та стійкої економіки все частіше обговорюється на міжнародних форумах, об'єднуючи уряди, а також міжнародні та національні організації в спільні ініціативи з охорони навколишнього середовища. Для переходу до стійкої світової економіки потрібні фінансові джерела для інвестицій, що забезпечують екологічні вигоди. Зелені активи є основними фінансовими інструментами, які розглядаються як джерела фінансування зелених проектів [6].

У фінансових колах підвищення обізнаності про проблеми, пов'язані з охороною навколишнього середовища, призвело до встановлення практики зелених активів. Незважаючи на існуючі перешкоди, ринок зелених активів надзвичайно розширився за останні кілька років [7].

Це зростання підкреслює необхідність вивчення цієї практики, щоб краще зрозуміти пов'язану з цим динаміку та механізми.

## **1.2. Огляд існуючих рішень на ринку**

**Green Assets Wallet** - це перша у світі платформа блокчейнів для перевірки та звітування про фінансові наслідки. Green Assets Wallet розроблений для забезпечення прозорості та впевненості інвесторів на ринку зелених боргів, що зелені гроші спрямовуються на фінансування екологічно чистих проектів та забезпечують обіцяні екологічні вигоди [8].

### **Вплив**

Метою технології є підтримка масштабування ринку зеленого боргу шляхом об'єднання зелених інвесторів із зеленими інвестиційними можливостями. Вводячи економічну ефективність, прозорість та довіру, гаманець Green Assets має на меті зробити широкий спектр інвестиційних можливостей доступним для інвесторів.

### **Технологічні характеристики**

Гаманець Green Assets використовує технологію реляційного блокчейна Chromia, енергоефективний блокчейн, розроблений шведськими інженерами блокчейну ChromaWay. Користувачі не бачать блокчейн, але інформація, опублікована на гаманці Green Assets, зберігається в журналі змін, яким не можна маніпулювати, тим самим інвестори, а також широкий ринок можуть отримати доступ до надійних, перевірених та незмінюваних даних впливу на оточуюче середовище.

**EverGreen** – екологічна організація, що емітувала однойменну криптовалюту у 2015 році.

## **Вплив**

Сама валюта EverGreenCoin - це механізм, який використовується для підживлення уваги, відповідальності та турботи про наше довкілля та світ, в якому ми живемо, допомагаючи збирати кошти на екологічні зелені проекти.

## **Технологічні особливості**

Proof of Environment - це алгоритм, за допомогою якої EverGreenCoin винагороджує екологічні завдання. EverGreenCoin - нащадок біткойна, і EverGreenCoin успадкував деякі чудові риси від своїх предків: можливість переказу в будь-яку точку світу майже з нульовою платою, нульовий ризик втрати або крадіжки особистої інформації, оскільки особиста інформація ніколи не потрібна. А також: нульові маніпуляції з боку урядів та банків, оскільки EverGreenCoin не друкується та не видобувається, залежно від обставин.

EverGreenCoin можна обмінювати на державні валюти на біржах криптовалют або витрачати на оплату сервісів компаній, які їх приймають.

Недоліком цього токена є те, що він є нащадком біткоіна, тобто окрім переваг перейняв і його недоліки, а саме алгоритм консенсусу Proof of Work – який є енерговитратним та дорогим, окрім того таким, що сприяє видобутку викопного палива, особливо вугілля, яке має найгірші викиди серед усіх видів палива [9].

**TokenD** - центр обробки цифрових активів, який одночасно виконує функції системи управління гаманцями та біржі.

### **Вплив**

TokenD призначений для підприємств, які готові скористатися перевагами токенизації або експериментувати з технологією блокчейну без необхідності утримувати дорогу команду експертів з блокчейну та протягом найкоротшого часу на ринок.

Дозволяє випускати, передавати та обмінювати свої активи з високим рівнем конфіденційності, безпеки та аудиту, дотримуючись норм юрисдикції компанії [10].

### **Технологічні особливості**

TokenD використовує запатентовану технологію блокчейну для обслуговування гаманців, ідентифікаційних даних, цифрових активів та книги транзакцій.

Недолік фреймворку полягає в тому, що для емісії більше ніж двох тисяч токенів, компанія має оплатити ліцензію, яка коштує 50000\$.

### **1.3 Постановка завдання**

1. Аналіз бізнес-логіки процесу токенизації cleantech ефекту.
2. Дослідження засобів розробки блокчейн систем та вибір найбільш актуальних.
3. Розробка власної блокчейн системи та веб-додатку для взаємодії з нею, основна мета якої – токенизація позитивного екологічного впливу з метою залучення інвестицій в екологічні проєкти.
4. Розробка смарт-контракту на основі технології блокчейн, що забезпечить впровадження механізму фіксації при трансформації потенційного екологічного ефекту проєкту в практичний екологічний ефект під час їх реалізації.

### **1.4 Висновки до розділу 1**

У цьому розділі було проаналізовано актуальний стан проблеми, проаналізовано наявні рішення на ринку та поставлено завдання.

## РОЗДІЛ 2. Ключові конструктивні особливості фреймворку Hyperledger Fabric

### 2.1 Hyperledger Fabric як корпоративне блокчейн рішення

Блокчейн – розподілена система обміну даними, в якій всі учасники обробки операцій поєднані єдиними алгоритмами для збереження, передачі і підтвердження транзакцій. Кожна транзакція в системі підтверджена, не може бути змінена або видалена, може бути перевірена публічно. Підтвердження транзакцій виконується самими учасниками мережі за алгоритмом консенсусу в мережі. Транзакції формуються в блоки і доєднуються до єдиного ланцюга блоків (блокчейну).

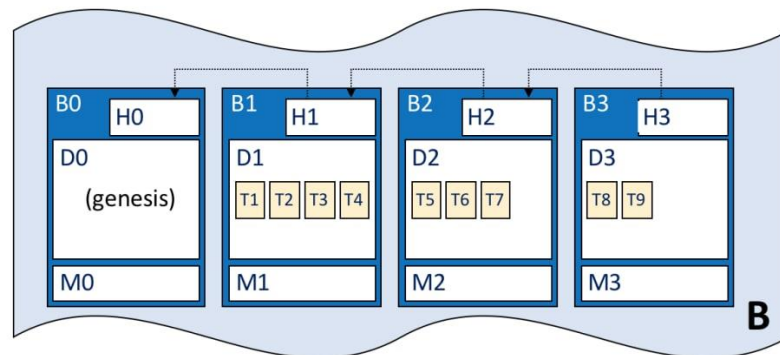


Рисунок 2.1 – Ланцюг блоків [11]

B – Блокчейн,

B0, B1, B2, B3 – блоки,

H0, H1, ... - заголовки блоків, що містять хеш попереднього блоку, поточного блоку та нонс-число (з англ. nonce — «number that can only be used once»),

D1 – тіло блоку, яке містить у собі транзакції T1, T2, T3, T4,

M0, M1, M2, M3 – метадані блоків.

Hyperledger – це спільнота з відкритим кодом, орієнтована на розробку набору стабільних фреймворків, інструментів та бібліотек для розгортання блокчейнів корпоративного рівня [12].

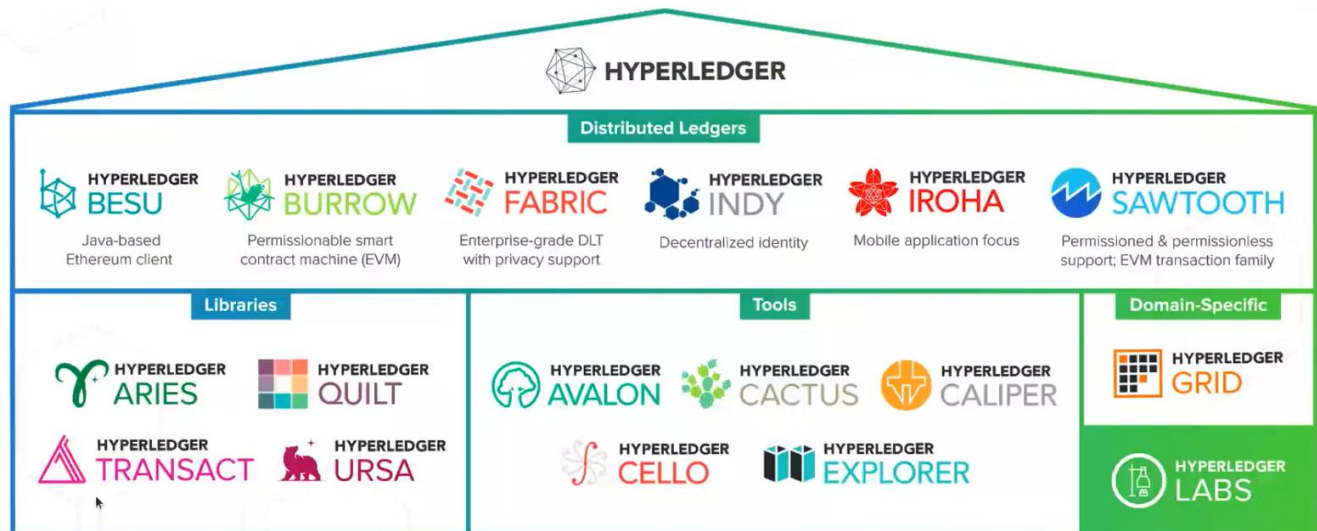


Рисунок 2.2 – Проєкти Hyperledger

Hyperledger Fabric – фреймворк для розробки блокчейн додатків з модульною архітектурою.

Його модульна конструкція задовольняє широкий спектр галузевих випадків використання. Його можна використовувати на підприємствах які розподілені в різних містах країни або навіть в різних країнах та видавати доступ лише авторизованим особам у яких є власний електронний сертифікат.

Також він забезпечую дуже зручну масштабованість: дозволяється оновлювати бізнес логіку, додавати нові сервери до мережі не перезапускаючи всю систему, що є дуже важливим для підприємств, оскільки це допомагає дешевше обслуговувати програмне забезпечення.



### 2.1.1 Активи в Fabric

Активи можуть бути як матеріальними (нерухомість та обладнання) так і нематеріальні (контракти та інтелектуальна власність).

Активи представлені в Hyperledger Fabric як колекція пар ключ-значення, із змінами стану, записаними як транзакції в журналі транзакцій каналу. Канал-механізм комунікації учасників системи по мережі для взаємодії з фабриком.

Активи можуть бути представлені у бінарній та / або JSON формі.

Hyperledger Fabric надає можливість модифікувати активи за допомогою функцій chaincode.

### 2.1.2 Chaincode

Chaincode - це програмне забезпечення, яке визначає актив чи активи, а також інструкції транзакції для модифікації активу (активів); іншими словами, це бізнес-логіка. Результатом виконання функцій чейнкоду є набір записів ключ-значення, який можна подати в мережу та записати до журналу транзакцій всіх учасників мережі. Chaincode є аналогом смарт-контракту в мережі Ethereum.

### 2.1.3 Ledger

Леджер (Ledger) - це послідовний, стійкий до втручання запис усіх переходів стану в фабрику. Переходи (зміни) станів є результатом викликів функцій чейнкоду («транзакцій»), підтверджених учасниками системи. Кожна транзакція призводить до набору пар активів ключ-значення, які фіксуються в леджері як створення, оновлення або видалення даних.

В кожному каналі існує тільки один леджер. Кожен учасник системи (Peer Node) зберігає копію леджера для кожного каналу, членом якого він є. Така реалізація необхідна для досягнення консенсусу в мережі. Консенсус – набір правил

для прийняття рішень стосовно запису інформації в леджер та її синхронізації. Fabric використовує алгоритм дозволеного голосування (“permissioned voting-based”) для досягнення консенсусу [13].

## **2.2 Fabric Certification Authority**

### **2.2.1 Загальні відомості**

Важливим компонентом фреймворку є Fabric Certification Authority. Fabric Certification Authority – частина фреймворка, завданням якої є підтвердження справжності ключів шифрування за допомогою сертифікатів електронного підпису.

Цифровий сертифікат - це документ, що містить набір атрибутів, що стосуються власника сертифіката.

Всі взаємодії в блокчейні Fabric виконуються за допомогою цих сертифікатів. Тобто, Fabric CA (Certification Authority) надає наступний функціонал:

- Видача реєстраційних сертифікатів;
- Валідація сертифікатів;
- Продовження і скасування сертифікатів.

Fabric CA працює на основі X.509 стандарту і інфраструктури публічних ключів – PKI. Інфраструктура публічних ключів (PKI) - це сукупність Інтернет-технологій, що забезпечує безпечний зв'язок у мережі.

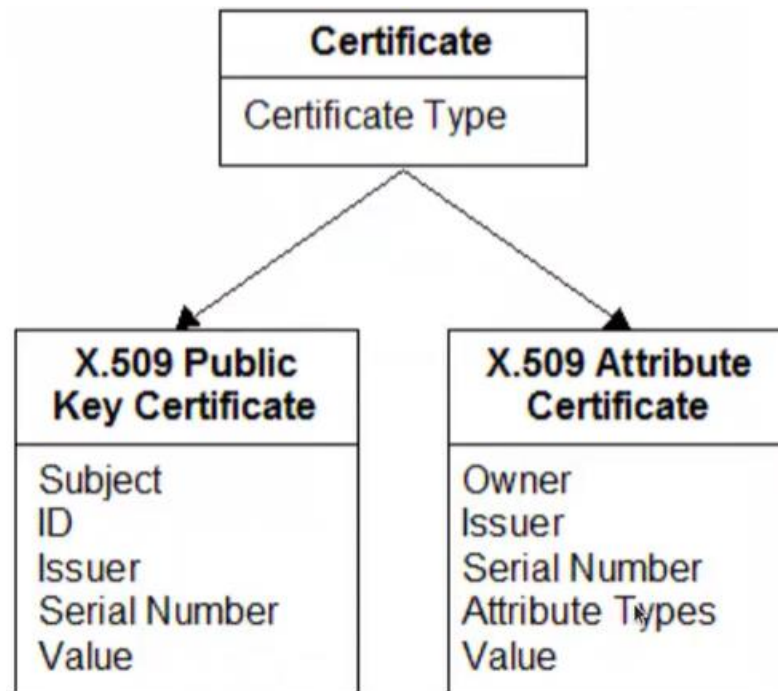


Рисунок 2.3 – Схема X.509 сертифікату

Сертифікат містить інформацію:

- Хто видав сертифікат;
- Хто є власником сертифікату;
- Термін дії;
- Організація до якої належить даний суб'єкт;
- Країна;
- Місто;
- Департамент організації;

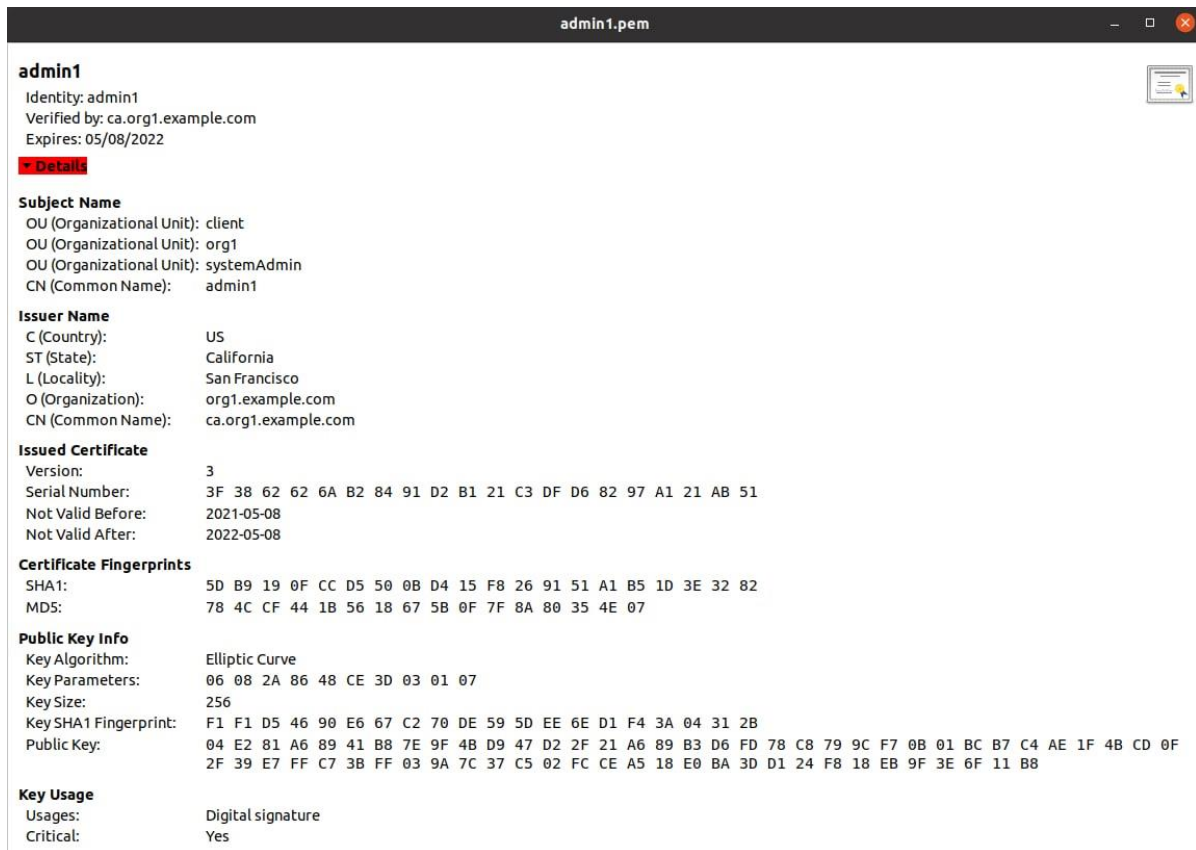
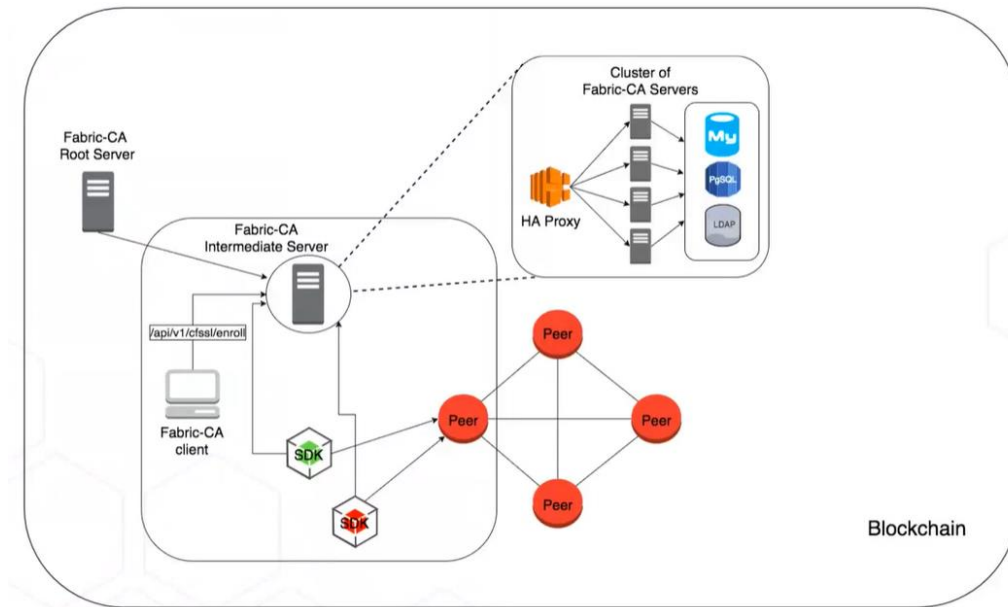


Рисунок 2.4 – Електронний сертифікат особи

## 2.2.2 Архітектура Fabric CA



Fabric-CA Root Server – головний сервер, один в системі.

Cluster of Fabric-CA Servers – дочірні сертифікаційні сервери, що зберігають ідентифікатори сертифікатів та розвантажують головний.

Fabric-CA Client – консольний додаток для взаємодії з Certification Authority.

Peer – учасник системи: клієнтський застосунок або сервер. У випадку сервера Peer відповідає за обробку транзакцій, у випадку клієнтського застосунку – взаємодіє з Fabric мережею.

Рисунок 2.5 – Архітектура Fabric-CA

В застосунку було обрано спосіб взаємодії з Fabric CA за допомогою Fabric-CA Client.

### **2.3 Висновки до розділу 2**

У цьому розділі було наведено основні теоретичні відомості про технологію Блокчейн, конструктивні особливості фреймворку Hyperledger Fabric та центру сертифікації Fabric-CA. Було описано загальні відомості про електронні сертифікати та стандарт X.509.

## **РОЗДІЛ 3. Опис підходу та реалізація**

### **3.1 Аналіз технічного завдання**

В системі наявні наступні ролі користувачів: інвестор, компанія, валідатор, системний адміністратор. У веб-додатку реалізовані інтерфейси для кожної ролі користувачів.

#### **3.1.1 Технічні вимоги користувачів до функціональності системи**

##### **3.1.1.1 Інвестор**

- Реєстрація та автентифікація;
- Перегляд власного портфоліо (крипто-гаманця);
- Перегляд списку проєктів з підтвердженим потенційним cleantech ефектом та інформації про ці проєкти, а саме:
  - 1) назва проєкту та компанії;
  - 2) опис проєкту;
  - 3) ім'я валідатора, що підтвердив позитивний екологічний вплив проєкту;
  - 4) назва токєну;
  - 5) заявлена емітована кількість токєнів;
  - 6) вартість в USDT – криптовалюти Тезер;
  - 7) документація по проєкту.
- Інвестиції в обрані проєкти.

### 3.1.1.2 Валідатор

- Реєстрація та автентифікація;
- Перегляд власного балансу;
- Перегляд списку проєктів з непідтвердженим потенційним cleantech ефектом та інформації про ці проєкти, а саме:
  - 1) назва проєкту та компанії;
  - 2) опис проєкту;
  - 3) ім'я валідатора, що підтвердив позитивний екологічний вплив проєкту;
  - 4) назва токєну;
  - 5) заявлена емітована кількість токєнів;
  - 6) вартість в USDT – криптовалюти Тезер;
  - 7) документація по проєкту;
- Підтвердження позитивного екологічного впливу проєкту;
- Перегляд підтверджених проєктів цим валідатором.

### 3.1.1.3 Компанія

- Створення проєктної заявки;
- Завантаження документації проєкту;
- Перегляд списку власних проєктів компанії та інформації про ці проєкти, а саме:
  - 1) назва проєкту та компанії;
  - 2) опис проєкту;
  - 3) статус проєкту;
  - 4) ім'я валідатора, що підтвердив позитивний екологічний вплив проєкту (якщо підтверджений);
  - 5) назва токєну;



- 6) заявлена емітована кількість токенів;
  - 7) вартість в USDT – криптовалюти Тезер;
  - 8) документація по проєкту;
  - 9) Перегляд портфоліо (крипто-гаманця).
- Нарахування коштів на гаманець проєкту;
  - Перегляд всіх інвестицій по всім своїм проєктам.

#### **3.1.1.4 Адміністратор**

- Перегляд списку інвесторів;
- Перегляд гаманця інвестора;
- Нарахування коштів на гаманець інвестора;
- Перегляд списку валідаторів;
- Перегляд підтверджених проєктів валідатором;
- Перегляд списку всіх проєктів та інформації про ці проєкти, а саме:
  - 1) назва проєкту та компанії;
  - 2) опис проєкту;
  - 3) статус підтвердження екологічного впливу;
  - 4) ім'я валідатора, що підтвердив позитивний екологічний вплив проєкту (якщо підтверджено);
  - 5) назва токєну;
  - 6) заявлена емітована кількість токенів;
  - 7) вартість в USDT – криптовалюти Тезер;
  - 8) документація по проєкту;

### **3.1.2 Специфікації вимог до даних**

#### **3.1.2.1 Інвестор**

Про інвестора зберігається його унікальний ідентифікатор, повне ім'я та його крипто-гаманець. Гаманець інвестора, окрім нього самого, може переглядати також адміністратор системи.

#### **3.1.2.2 Валідатор**

Про валідатора зберігається його унікальний ідентифікатор, повне ім'я, його крипто-гаманець та список підтверджених ним проєктів. Список підтверджених ним проєктів, окрім нього самого, може переглядати також адміністратор системи.

#### **3.1.2.3 Компанія**

Про компанію зберігається її унікальний ідентифікатор, повна назва та її проєкти. Інвестори бачать в системі тільки проєкти з підтвердженим позитивним екологічним впливом. Валідатори бачать в системі тільки проєкти з непідтвердженим позитивним екологічним впливом. Компанія бачить в системі тільки свої проєкти. Адміністратор системи бачить усі проєкти всіх компаній.

#### **3.1.2.4 Проєкт**

Про проєкт зберігається його ім'я, опис, загальна емітована кількість токенів, кількість вже проданих токенів, назва токenu, вартість токenu в USDT, статус підтвердження cleantech ефекту, ім'я валідатора який підтвердив (якщо підтверджено), крипто-гаманець проєкту та історія інвестицій в проєкт.

### 3.2 Реалізація токенизації cleantech ефекту

#### 3.2.1 Принцип дії платформи токенизації cleantech ефекту



Рисунок 3.1 – Процес токенизації cleantech ефекту

Проаналізувавши схему процесу на рисунку 3.1, можна виокремити ключові фрагменти системи: реєстрація користувачів, створення заявки компанією, підтвердження позитивного екологічного ефекту експертами, випуск tokenів, проведення транзакції crypto-to-crypto між інвестором та компанією.

### 3.2.2 Реалізація ключових функцій чейнкоду

Нижче наведено реалізацію ключових функцій чейнкоду в процесі токенизації cleantech ефекту підприємства.

#### 3.2.2.1 Створення проєктної заявки

Компанія заявляє про створення нового проєкту з потенційним позитивним екологічним впливом. По ключу з бази отримується колекція компаній, далі за унікальним ідентифікатором компанії, що записаний в її електронному сертифікаті, отримуємо масив всіх проєктів цієї компанії. До цього масиву додається новий об'єкт, в якому записані всі необхідні дані про новий проєкт компанії. Статус підтвердження – false. Нова пара ключ-значення записується в леджер.

```

async createNewProject(ctx, projectName, projectDescription, totalSupply, tokenName, priceInUSDT) {
  const identity = new ClientIdentity(ctx.stub);
  if (identity.cert.subject.organizationalUnitName !== UNIT_COMPANY) {
    throw new Error('Current subject does not have access to this function');
  }
  const companiesAsBytes = await ctx.stub.getState(COMPANIES_COLLECTION);
  const companiesAsObject = JSON.parse(companiesAsBytes.toString());

  companiesAsObject[identity.cert.subject.commonName].projects.push(
    {
      projectName: projectName,
      projectDescription: projectDescription,
      totalSupply: totalSupply,
      supply: 0,
      tokenName: tokenName,
      priceInUSDT: priceInUSDT,
      approved: "false",
      wallet: []
    }
  );

  await ctx.stub.putState(COMPANIES_COLLECTION, Buffer.from(JSON.stringify(companiesAsObject)));
  return JSON.stringify( { value: { data: ctx.stub.getTxID() }, replacer: null, space: 2 });
}

```

Рисунок 3.2 – Створення проєктної заявки

### 3.2.2.2 Підтвердження cleantech ефекту

Користувач валідатор, після перегляду проєктної документації, підтверджує позитивний екологічний вплив проєкту. За свої послуги валідатор отримує фіксовану кількість USDT, яка знімається з рахунку проєкту.

На початку отримується колекція компаній, з якої за унікальним ідентифікатором компанії ми отримуємо потрібну нам компанію та масив її проєктів. В цьому масиві за назвою знаходимо потрібний нам проєкт. Після необхідних перевірок на можливість проведення транзакції, статус підтвердження позитивного екологічного впливу проєкту стає true, а на рахунок валідатора зараховується плата за послуги. Функція повертає ID транзакції.

```

async approveProject(ctx, projectName, companyUID) {
  const identity = new ClientIdentity(ctx.stub);
  if (identity.cert.subject.organizationalUnitName !== UNIT_VALIDATOR) {
    throw new Error('Current subject does not have access to this function');
  }

  const companiesAsBytes = await ctx.stub.getState(COMPANIES_COLLECTION);
  const companiesAsObject = JSON.parse(companiesAsBytes.toString());

  companiesAsObject[companyUID].projects.forEach(proj => {
    if (proj.projectName === projectName) {
      if (proj.wallet.length === 0) {
        throw new Error('Project's portfolio is empty');
      } else {
        let usdtRecord = false;
        proj.wallet.forEach((record :T) => {
          if (record.currencyName === VALIDATOR_FEE_CURRENCY) {
            if (Number(record.amount) - Number(VALIDATOR_FEE) < 0) {
              throw new Error('Project does not possess enough USDT');
            } else {
              record.amount = Number(record.amount) - Number(VALIDATOR_FEE);
              proj.approved = true;
              proj["approvedBy"] = identity.cert.subject.commonName;
            }
            usdtRecord = true;
          }
        })
        if (!inArr) {
          throw new Error('Project does not possess USDT');
        }
      }
    }
  })
  try {
    await this.depositInvestor(ctx, identity.cert.subject.commonName, VALIDATOR_FEE_CURRENCY, VALIDATOR_FEE);
  } catch (e) {
    throw new Error(e.stack);
  }
  await ctx.stub.putState(COMPANIES_COLLECTION, Buffer.from(JSON.stringify(companiesAsObject)));
  return JSON.stringify({ value: {data: ctx.stub.getTxID()}, replacer: null, space: 2});
}

```

Рисунок 3.3 - Підтвердження cleantech ефекту проєкту

### 3.2.2.3 Отримання проєктів з підтвердженням cleantech ефектом

Користувач інвестор бачить тільки підтверджені проєкти компаній. На початку транзакції отримується колекція компаній, по ключам та значенням якій відбувається ітерація. Ключом є ідентифікатор компанії, значенням – об’єкт, що містить дані про компанію. Далі відбувається ітерація по проєктам компанії. Якщо проєкт підтверджено валідатором, він додається до результуючого масиву. Після проходження по всім компаніям, функція повертає результуючий масив.

```
async getAllProjects(ctx) {  
  const identity = new ClientIdentity(ctx.stub);  
  if (identity.cert.subject.organizationalUnitName !== UNIT_INVESTOR) {  
    throw new Error('Current subject does not have access to this function');  
  }  
  const companiesAsBytes = await ctx.stub.getState(COMPANIES_COLLECTION);  
  const companiesAsObject = JSON.parse(companiesAsBytes.toString());  
  let result = []  
  
  Object.entries(companiesAsObject).forEach(([key, value]) => {  
    if (value.projects.length === 0) return;  
    value.projects.forEach(project => {  
      if (project.approved === true) {  
        result.push(  
          {  
            companyUID: key,  
            companyName: value.companyName,  
            ...project  
          }  
        )  
      }  
    })  
  })  
};  
return JSON.stringify(result, {replaced: null, space: 2});  
}
```

Рисунок 3.4 - Отримання проєктів з підтвердженням cleantech ефектом

### 3.2.2.4 Фінансування проєкту

На початку роботи функції отримується колекція інвесторів, з якої за унікальним ідентифікатором отримуються дані про інвестора та його гаманець. У випадку успішних перевірок, з гаманця інвестора знімається зазначена сума інвестиції

```

async investToProject(ctx, companyUID, projectName, currency, amount) {
  const identity = new ClientIdentity(ctx.stub);
  if (identity.cert.subject.organizationalUnitName !== UNIT_INVESTOR) {
    throw new Error('Current subject does not have access to this function');
  }
  const investorUID = identity.cert.subject.commonName;

  const investorsAsBytes = await ctx.stub.getState(INVESTORS_COLLECTION);
  const investorsAsObject = JSON.parse(investorsAsBytes.toString());
  const investorFullName = investorsAsObject[investorUID].investorFullName;
  const investorWallet = investorsAsObject[investorUID].wallet;

  //withdraw investor's tokens
  if (investorWallet.length === 0) {
    throw new Error("Current investor's portfolio is empty");
  } else {
    let inArr = false;
    investorWallet.forEach((record :T) => {
      if (record.currencyName === currency) {
        if (Number(record.amount) - Number(amount) < 0) {
          throw new Error('Current investor does not possess enough ${currency}');
        } else {
          record.amount = Number(record.amount) - Number(amount);
        }
        inArr = true;
      }
    })
    if (!inArr) {
      throw new Error("Current investor does not possess this token");
    }
  }
}

```

Рисунок 3.5 – Транзакція crypto-to-crypto

Далі з бази даних отримується проєкт, фінансування якого відбувається. Вираховується сума токенів проєкту для продажу інвестору. У випадку успішних перевірок, ця сума записується в поле «постачання» («supply») проєкту та відправляється на гаманець інвестора. Відбувається перерахунок інвестиції на гаманець проєкту. Функція повертає повне ім'я інвестора, назву компанії та ID транзакції.

```
const projToInvestAsString = await this.getProject(ctx, projectName, companyUID);
const projToInvest = JSON.parse(projToInvestAsString);
const projectTokenAmount = Number(amount) / Number(projToInvest.priceInUSDT);

const companiesAsBytes = await ctx.stub.getState(COMPANIES_COLLECTION);
const companiesAsObject = JSON.parse(companiesAsBytes.toString());

//write updated supply
companiesAsObject[companyUID].projects.forEach(proj => {
  if (proj.projectName === projectName) {
    if (Number(proj.totalSupply) - (Number(proj.supply) + Number(projectTokenAmount)) < 0) {
      throw new Error("Impossible to withdraw project token, not enough!");
    } else {
      proj.supply = Number(proj.supply) + Number(projectTokenAmount);
    }
  }
})

try {
  await this.depositInvestor(ctx, investorUID, projToInvest.tokenName, projectTokenAmount);
  await this.depositCompanyProject(ctx, companyUID, projectName, currency, amount);
} catch (e) {
  throw new Error(e.stack);
}

await ctx.stub.putState(INVESTORS_COLLECTION, Buffer.from(JSON.stringify(investorsAsObject)));
await ctx.stub.putState(COMPANIES_COLLECTION, Buffer.from(JSON.stringify(companiesAsObject)));
return JSON.stringify({ value: {
  investorFullName: investorFullName,
  companyName: companiesAsObject[companyUID].companyName,
  transactionId: ctx.stub.getTxID()
}, replacer: null, space: 2});
}
```

Рисунок 3.5 (продовження)



### 3.3 Обґрунтування вибору засобів розробки

Облік цифрових активів містить ряд ризиків. Важливо мати систему, де видача, зберігання, передача активів здійснюється з достатнім рівнем надійності, ефективності та прозорості для всіх учасників. Саме тому, для поставленого завдання добре підходить технологія блокчейн; безпеку даних гарантує криптографія, а будь-які спроби махінацій стають очевидні для аудитора.

Узгодження та дотримання стандартів є складним завданням, особливо одразу по всьому світу. Перевагою блокчейну є те, що ця технологія здатна забезпечити дотримання заданих вами стандартів програмно. Це суттєва перевага у контексті зеленого фінансування, де інвестиції часто робляться через національні кордони, і де наскільки бажано, настільки ж і складно забезпечити, щоб різні суб'єкти в різних країнах дотримувалися зазначених стандартів та вимог. [14, 1]

Ця особливість блокчейну підвищує якість даних, і в підсумку виникає багатий інформаційний ресурс для дослідження якості та впливу різних видів екологічної діяльності в різних сферах. Часто досить суперечливим та неоднозначним є те, чи щось дійсно є екологічним чи ні, або наскільки екологічно чистим це є насправді. Високоякісні дані можуть допомогти прояснити це. Система управління (платформа) побудована на таких принципах означає, що ці стандарти можуть природньо виникати з колективних дій консорціуму зацікавлених сторін, на відміну від введення їх однією організацією. Теоретично цього можна було б досягти за допомогою централізованої платформи, але прецедентів цього мало [14, 2].

Саме тому, в своїй системі я використовую наступні ключові особливості технології блокчейн:

- Безпечна та прозора реплікація (синхронізація) даних;
- Дотримання стандартів щодо достовірності даних;
- Цифрові сертифікати, що підтверджують особу.

Для розгортання власної блокчейн мережі я використав фреймворк Hyperledger Fabric. Однією з ключових особливостей фреймворку, якою я скористався, це розподіл даних «за необхідністю» («Need-To-Know Basis»). Цей принцип реалізовано за допомогою каналів в мережі. Він дозволяє нам надавати користувачам лише ті дані, до яких їм необхідно мати доступ, тим самим захищаючи дані, оскільки інколи самої тільки криптографії буває недостатньо. Окрім цього, модульна архітектура дозволяє розгорнути нову мережу та підключити її до вже існуючого застосунку. Модульна архітектура також надає можливість створювати надбудови над фреймворком, такі як власна система управління ідентифікацією користувачів. Особливо корисною перевагою такої архітектури фреймворку для мене була можливість створювати та оновлювати чейнкод (chaincode) в каналі, без необхідності заново розгортати всю блокчейн мережу. Схему блокчейн мережі наведено в додатку А.

Окремі компоненти фреймворку мають різні залежності та вимоги до системи. Наприклад chaincode у Fabric повинен працювати у власному ізолюваному середовищі і не взаємодіяти з іншими, якщо це не вимагається розробником. Це робиться для підтримання цілісності даних, а також для збереження конфіденційності даних. Для ізоляції цих компонентів з їх особливостями, було використано Docker та Docker-compose, а саме створено їх в окремих контейнерах. Контейнер - є одиницею програмного забезпечення, яке містить код та всі його залежності, завдяки чому програма працює швидко та надійно незалежно від обчислювального середовища. Docker-compose було використано задля полегшення запуску декількох контейнерів одночасно та їх взаємодії.

Застосунок побудовано за клієнт-серверною архітектурою. На сервері застосовано середовище Node.js. Головними перевагами цього середовища є те що воно повністю керується подіями (event driven), і більшість коду виконується на основі зворотних викликів (callbacks). Такий підхід допомагає програмі не робити паузи під час обробки запиту від клієнта, а бути доступним і для інших запитів.

Також, Node.js використовує JavaScript-рушій V8, що робить це середовище швидшим за Ruby або Python.

Чейнкод написано на мові JavaScript. Взаємодію з чейнкодом на сервері наведено в додатку Б.

Для розробки front-end було використано мову розмітки - HTML, та мову стилю сторінок – CSS. Для надання динамічності сторінок було використано JavaScript. Для роботи з даними на стороні клієнта було використано бібліотеку React.

Для збереження даних в системі використовується NoSQL база даних ключ-значення LevelDB. Кожне оновлення леджеру містить набір пар ключ-значення, пов'язаних із транзакцією. Ця база даних використовується в Hyperledger Fabric за замовченням. Схему бази даних наведено в додатку В. В самій базі, дані зберігаються у вигляді масиву байтів. Взаємодію з базою даних наведено в додатку Г.

### 3.4 Тестування програми і результати її виконання

Незареєстрований користувач потрапляє на сторінку реєстрації. У разі вдалої реєстрації користувачу на пристрій завантажуються два файли: сертифікат та приватний ключ.

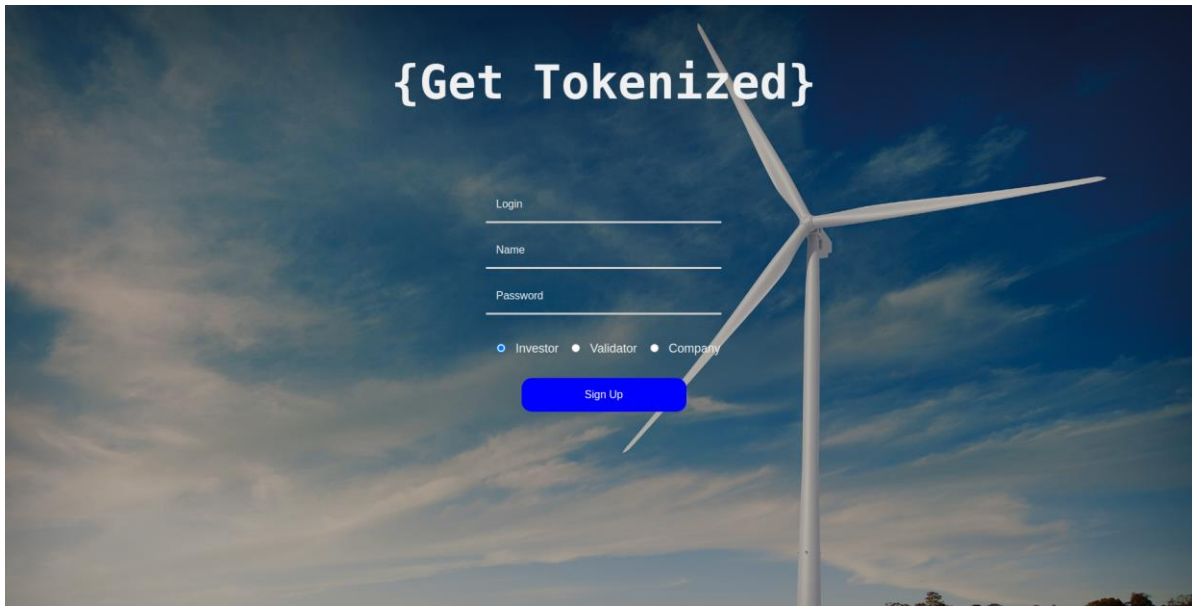


Рисунок 3.6 – Сторінка реєстрації користувача

Після реєстрації, користувач має увійти в систему, обравши у файловій системі свого пристрою файл, що містить сертифікат та файл що містить приватний ключ.

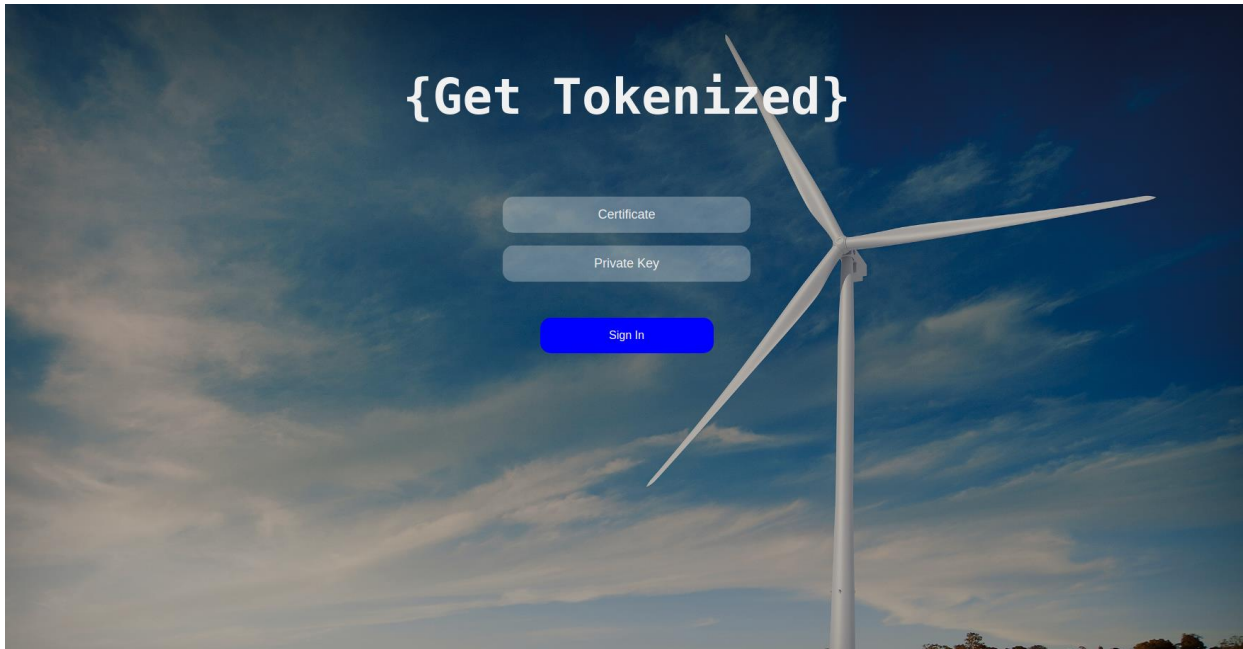


Рисунок 3.7 – Сторінка автентифікації користувача

Після автентифікації, користувач, що має роль в системі – компанія, бачить загальну суму інвестицій в її проєкти та кожну окрему транзакцію окремо.

Home	Projects	Virtu101	Log out
------	----------	----------	---------

Virtu101 total investments:	
Currency	Amount
USDT	480

Рисунок 3.8 – Сторінка гаманця компанії

В меню Projects компанія бачить усі свої проєкти. Вона може відсортувати їх за одним із параметрів.

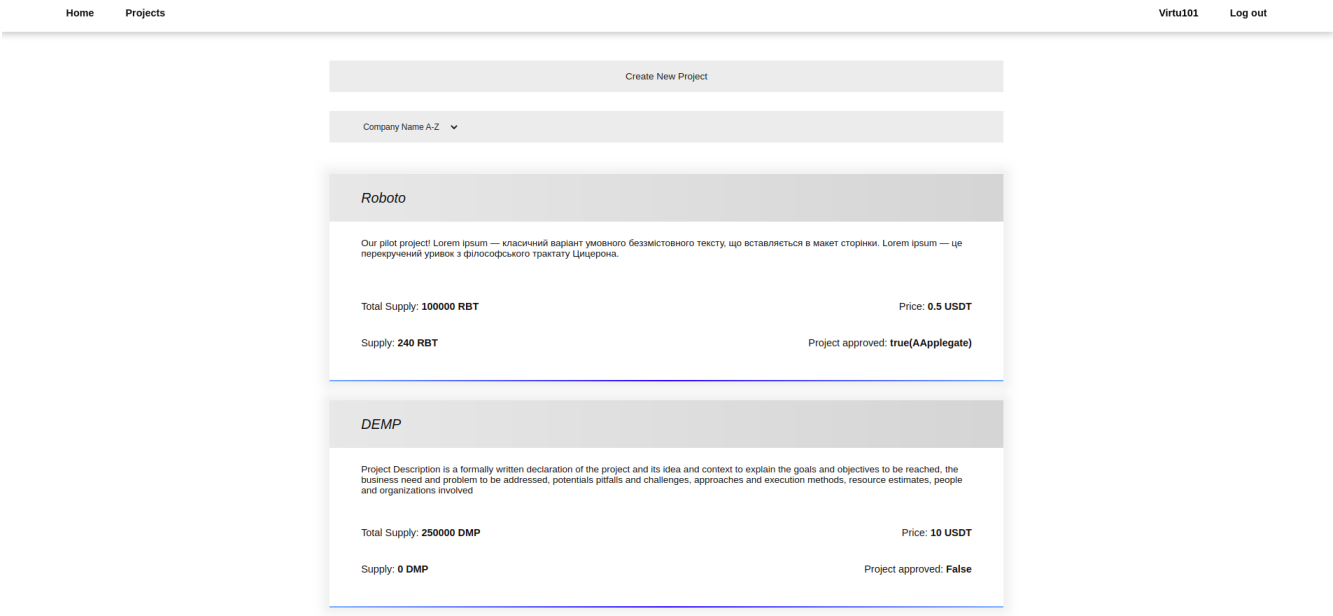


Рисунок 3.9 – Сторінка всіх проєктів компанії

Компанія створює проектну заявку.

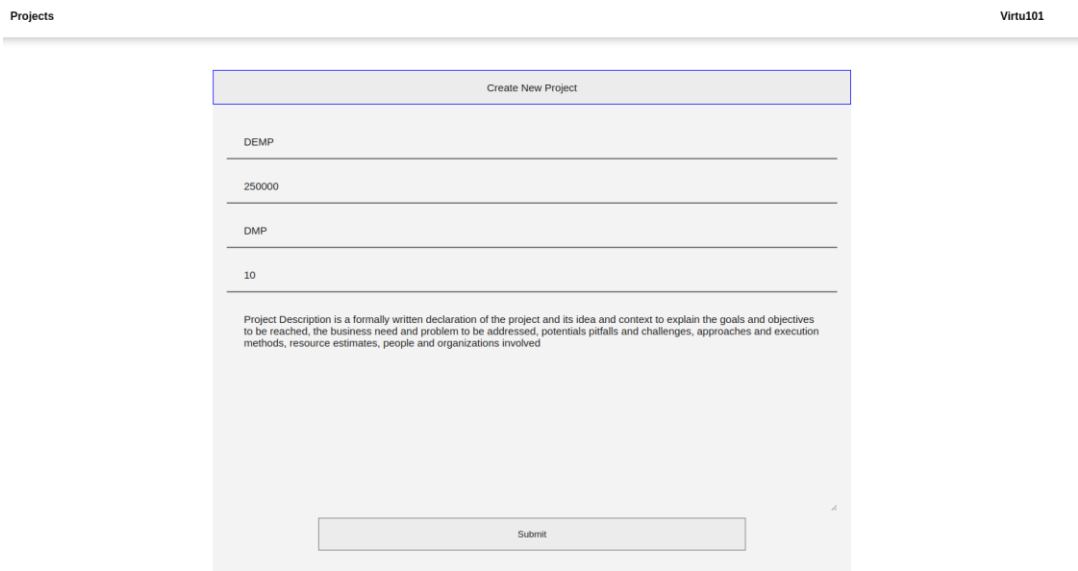


Рисунок 3.10 – Створення проєктної заявки

Компанія завантажує проектну документацію.

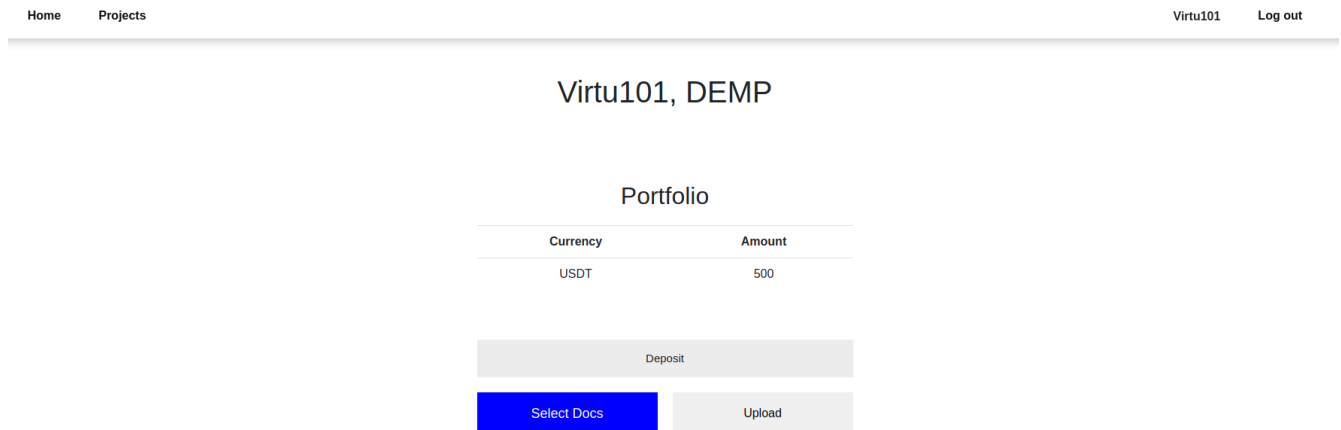


Рисунок 3.11 – Завантаження проєктної документації

Користувач валідатор бачить нову непідтверджену заявку.

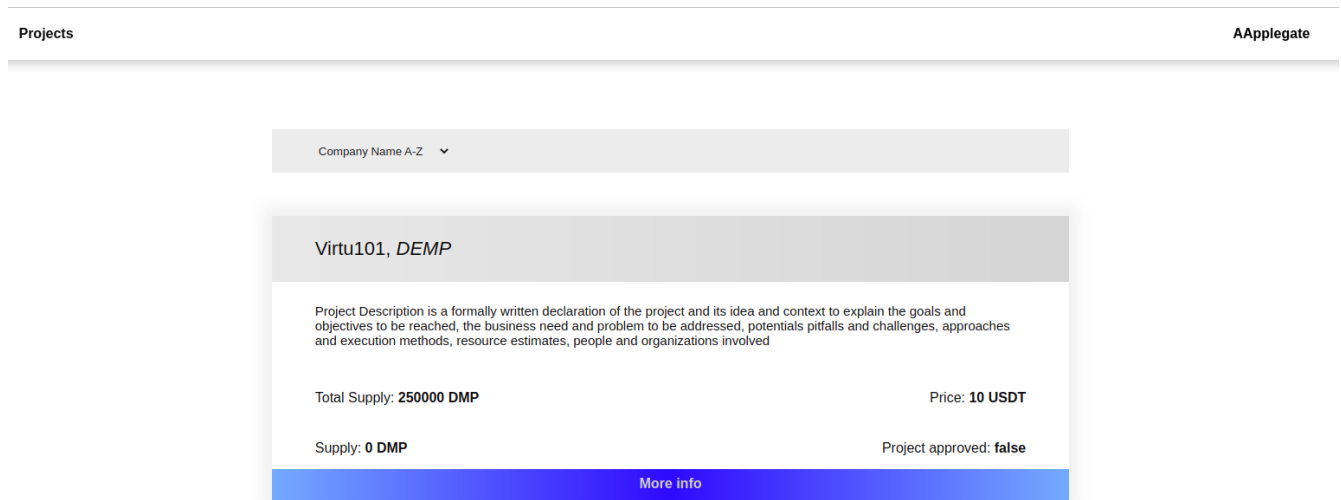


Рисунок 3.12 – Перелік непідтверджених проєктів в інтерфейсі валідатора

Після кліку на More Info користувач валідатор може ознайомитися з проектною документацією та підтвердити позитивний екологічний вплив проєкту в системі.

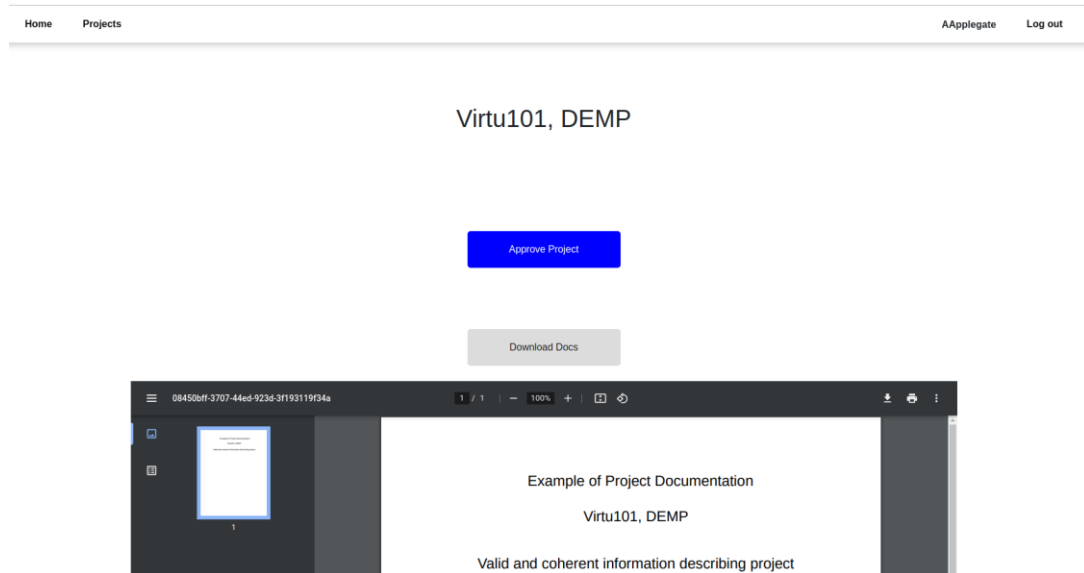


Рисунок 3.13 – Підтвердження проєктної заявки

Адміністратор системи заводить кошти на рахунок інвестора.

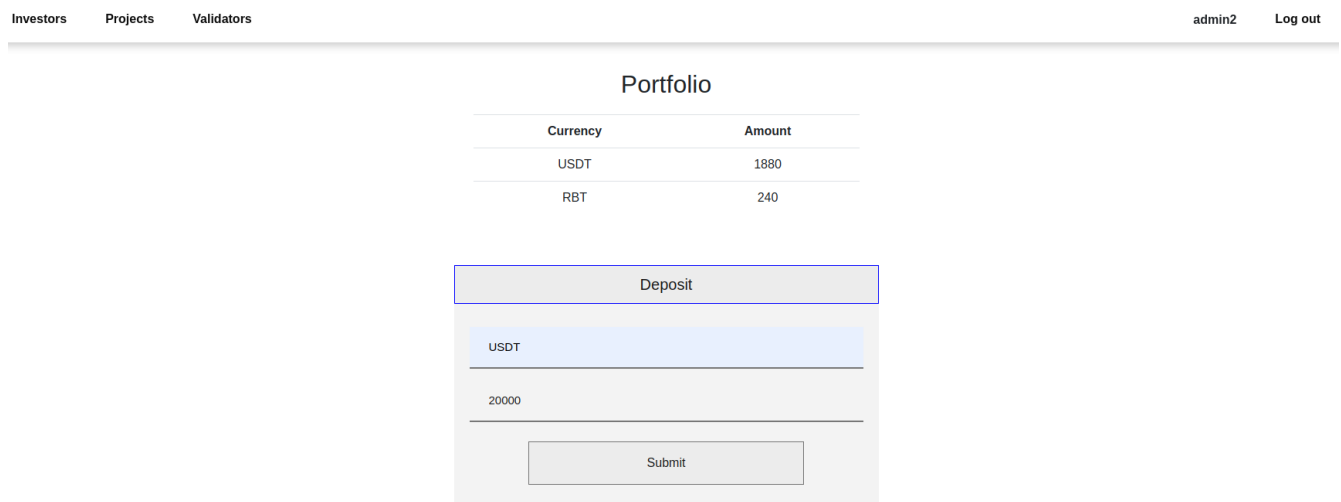


Рисунок 3.14 – Заведення коштів на гаманець інвестора в інтерфейсі адміністратора



Інвестор бачить свій крипто-гаманець.

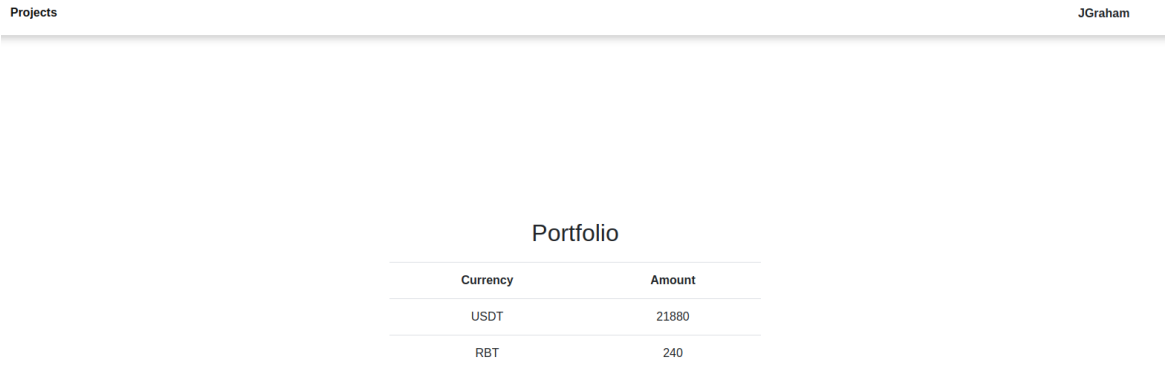


Рисунок 3.15 – Сторінка гаманця інвестора

Інвестор бачить перелік проєктів з підтвердженням позитивним екологічним впливом.

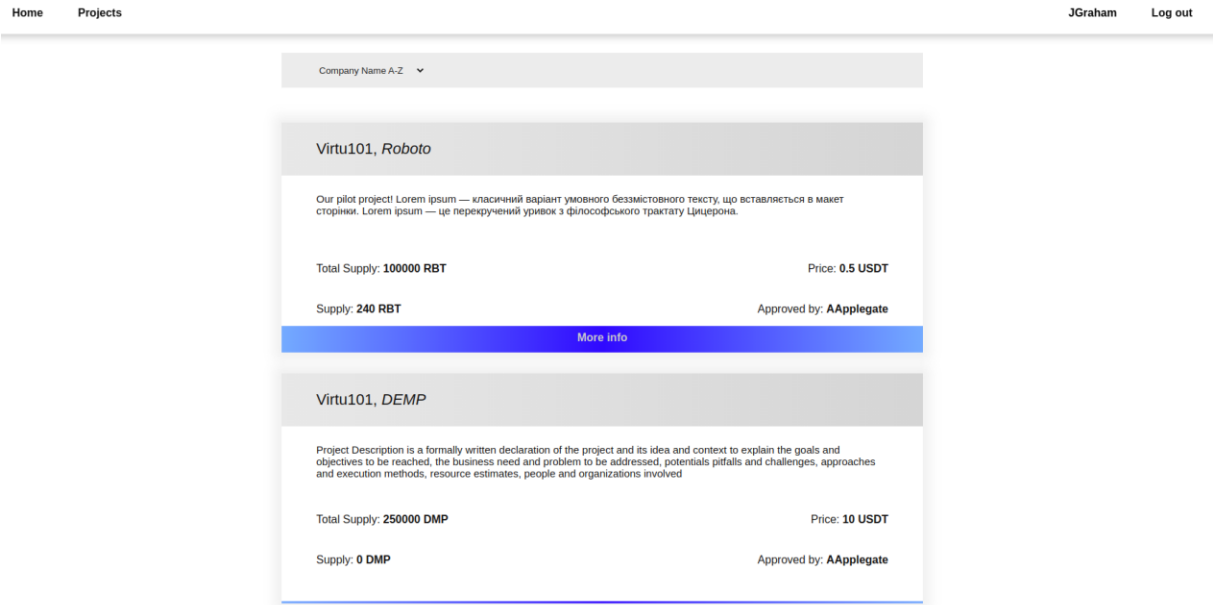


Рисунок 3.16 – Перелік проєктів з підтвердженням cleantech ефектом

Після кліку на More Info інвестор потрапляє на сторінку проєкту та може завантажити проєктну документацію або ознайомитися з нею онлайн.

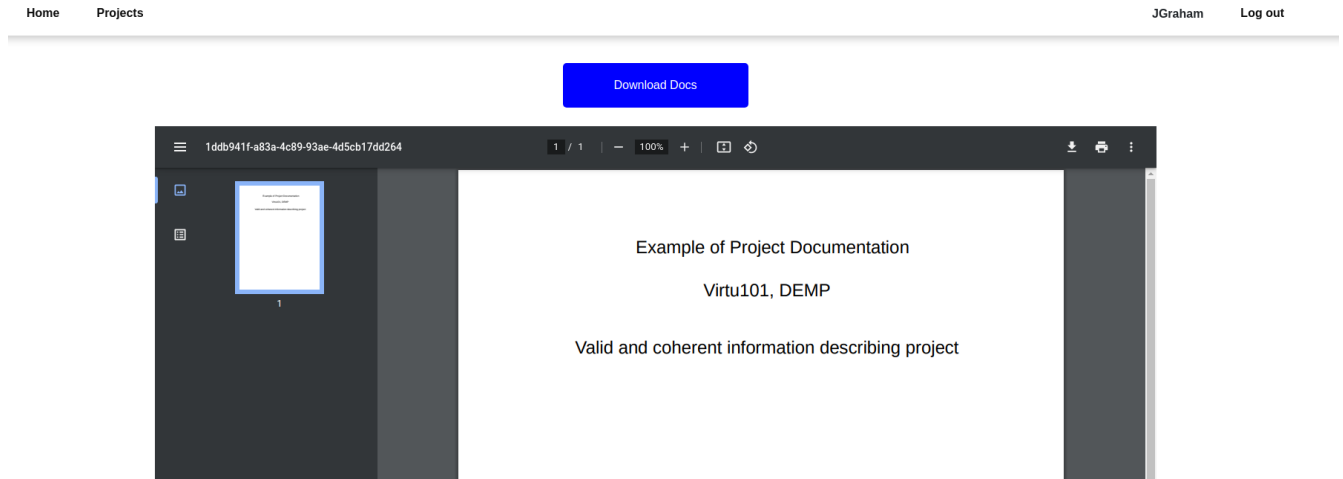


Рисунок 3.17 – Завантаження проєктної документації

Після ознайомлення з документацією інвестор може провести транзакцію, а саме зробити інвестицію.

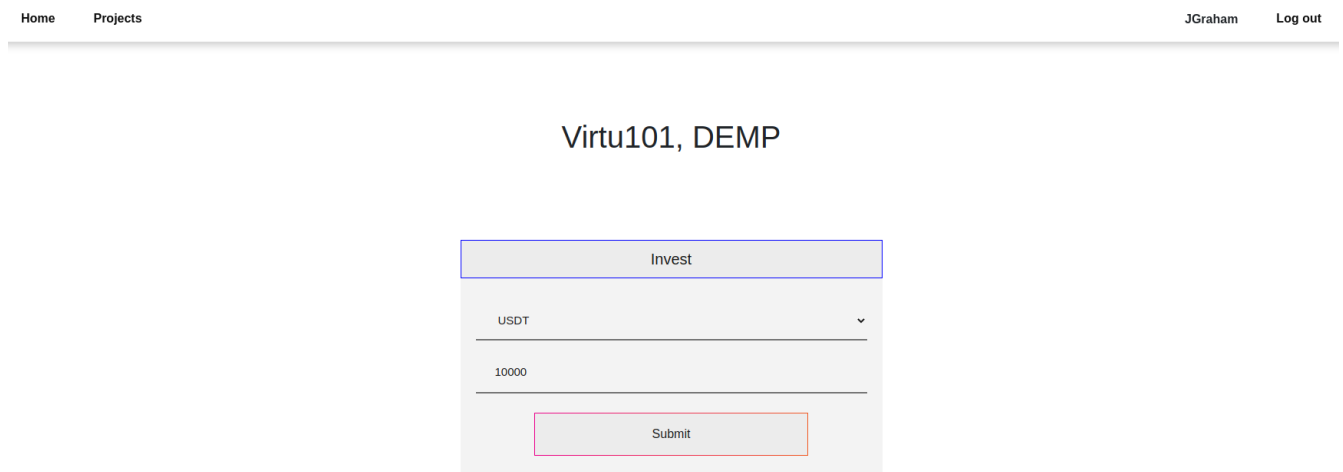


Рисунок 3.18 – Фінансування проєкту

У разі успішної транзакції інвестор отримує сертифікат. На рахунок інвестора перераховуються придбані токени проєкту. Проєкт отримує цільове фінансування.

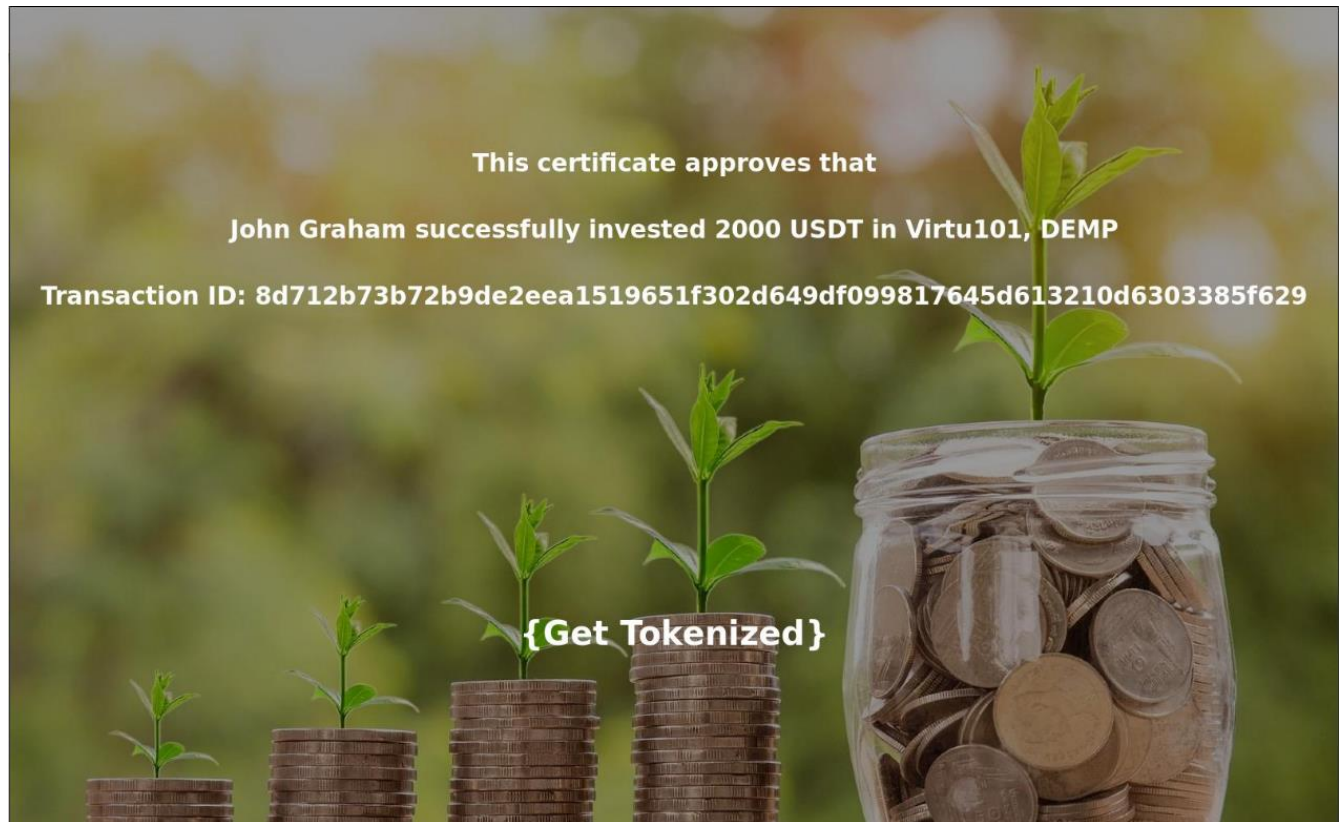


Рисунок 3.19 – Сертифікат про фінансування екологічного проєкту

### 3.5 Висновки до розділу 3

У цьому розділі були описані вимоги до системи, алгоритм токенизації cleantech ефекту підприємства, ключові функції чейнкоду та наведено результати роботи системи. Також було обгрунтовано вибір засобів розробки.

## Висновки

Отже, поставлене завдання було виконано, а саме: було розроблено механізм стимулювання інвестицій у проєкти з позитивним екологічним впливом. Було проаналізовано бізнес-логіку процесу токенизації cleantech ефекту підприємства, існуючі рішення на ринку, їх переваги та недоліки. За метод вирішення поставленого завдання було обрано токенизацію позитивного екологічного впливу (cleantech ефекту) підприємства за допомогою технології блокчейн. Було досліджено засоби розробки блокчейн систем та обрано найбільш актуальний – фреймворк Hyperledger Fabric.

Обрані засоби розробки задовільнили технічні вимоги до системи. Hyperledger Fabric надав можливість створити приватний блокчейн, доступ до якого мають лише особи та підприємства з електронним сертифікатом. Було забезпечено прозорість та достовірність інформації якою послуговуються інвестори та валідатори. Було забезпечено механізм фіксації позитивного екологічного впливу проєктів компаній, що надає можливість компаніям продавати право власності на екологічний вплив своїх проєктів, з метою залучення цільового фінансування.

Створена система є розподіленою мережею серверів, з якою надається можливість взаємодіяти за допомогою розробленого веб-додатку. Система має напрямки для подальшого вдосконалення: створення вторинного ринку активів (біржі) та розширення функціоналу системного адміністратора – надавати дозволи на реєстрацію в системі, тобто реалізація політики «знай свого користувача» («Know Your Customer» Policy).

## Список використаної літератури

1. Розблокування потенціалу зеленого фінтеху //stockholmgreenfin.tech/. 2017.  
<https://static1.squarespace.com/static/59b29215c027d84ada066d3b/t/5a4f73e6e4966b7a764114ba/1515156458061/stockholm-green-digital-finance-insight-brief-no1-2017.pdf>
2. Нові підприємства в Cleantech: можливості та інноваційні результати // Business Strategy and the Environment. – 2019.  
[https://www.researchgate.net/publication/337492008\\_New\\_ventures\\_in\\_Cleantech\\_Opportunities\\_capabilities\\_and\\_innovation\\_outcomes](https://www.researchgate.net/publication/337492008_New_ventures_in_Cleantech_Opportunities_capabilities_and_innovation_outcomes)
3. Pernick R. Революція чистих технологій / R. Pernick, W. Clint., 2007.
4. Добровільні керівні принципи випуску зелених облігацій ICMA. – 2018.  
<https://www.icmagroup.org/assets/documents/Regulatory/Green-Bonds/Green-Bonds-Principles-June-2018-270520.pdf>.
5. Зелені інвестиції - Між необхідністю, фіскальними обмеженнями та прибутком. Procedia Economics and Finance. 2015.
6. Bieliński T., Mosionek-Schweda M. Зелені облігації як фінансовий інструмент фінансування екологічних проєктів. 2018.  
[https://www.academia.edu/37954934/Green\\_bonds\\_as\\_a\\_financial\\_instrument\\_for\\_environmental\\_projects\\_funding](https://www.academia.edu/37954934/Green_bonds_as_a_financial_instrument_for_environmental_projects_funding).
7. Hachenberg B., Schiereck D. Чи відрізняються зелені облігації від звичайних облігацій? 2018.  
[https://www.researchgate.net/publication/326857253\\_Are\\_green\\_bonds\\_priced\\_differently\\_from\\_conventional\\_bonds](https://www.researchgate.net/publication/326857253_Are_green_bonds_priced_differently_from_conventional_bonds)
8. Green Assets Wallet [Електронний ресурс]  
<https://greenassetswallet.org/>
9. EverGreenCoin [Електронний ресурс]  
<https://evergreencoin.org/>

10. TokenD [Електронний ресурс]

<https://tokend.io/>

11. Леджер в Hyperledger Fabric [Електронний ресурс]

<https://hyperledger-fabric.readthedocs.io/en/release-2.2/ledger/ledger.html>

12. Hyperledger [Електронний ресурс]

<https://www.hyperledger.org/>

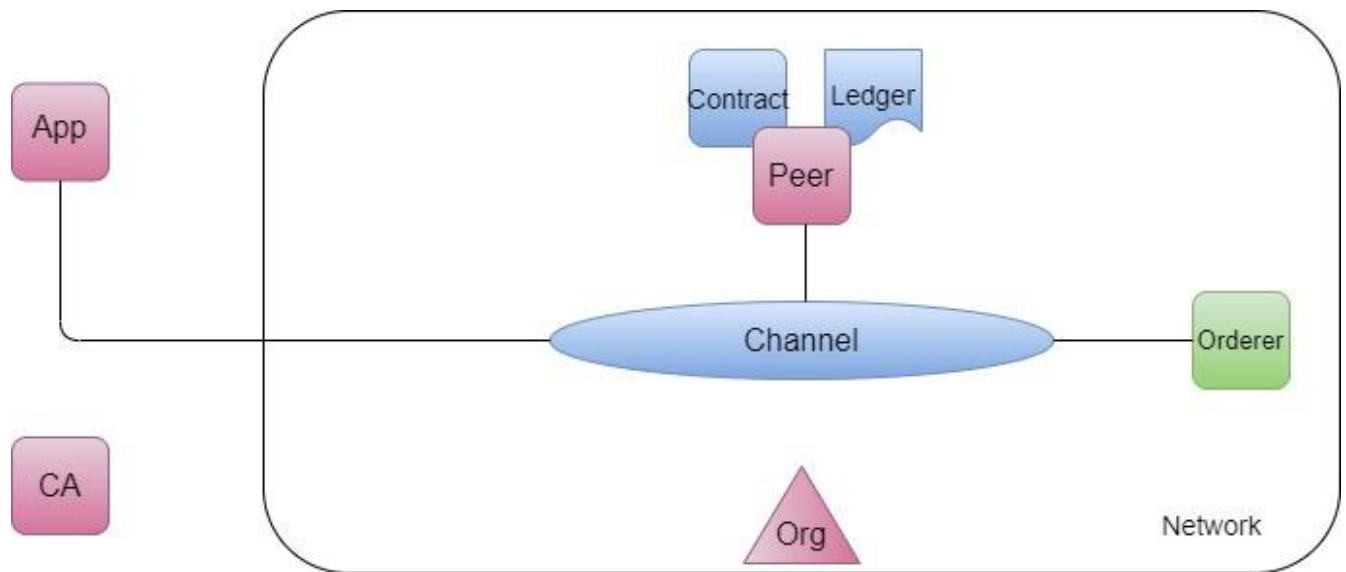
13. Модель Hyperledger Fabric [Електронний ресурс]

[https://hyperledger-fabric.readthedocs.io/en/release-2.2/fabric\\_model.html](https://hyperledger-fabric.readthedocs.io/en/release-2.2/fabric_model.html)

14. Документація Green Assets Walleet [Електронний ресурс]

<https://github.com/greenassetwallet/greenassetwallet>

## Додаток А. Схема блокчейн мережі



## Примітки

- 1 До мережі входить одна організація.
- 2 В мережі присутній один Peer Node що обробляє запити від клієнта.
- 3 Peer має локальну копію леджера (Ledger).
- 4 Застосунок (App) взаємодіє з каналом Channel за відправляючи запити до Peer.
- 5 В каналі Channel та в Peer встановлено чейнкод Contract.
- 6 В мережі створено вузол CA, що підписує електронні сертифікати.
- 7 Вузол Orderer формує сервіс замовлення (ordering service) транзакцій.

## Додаток Б. Взаємодія з чейнкодом на сервері

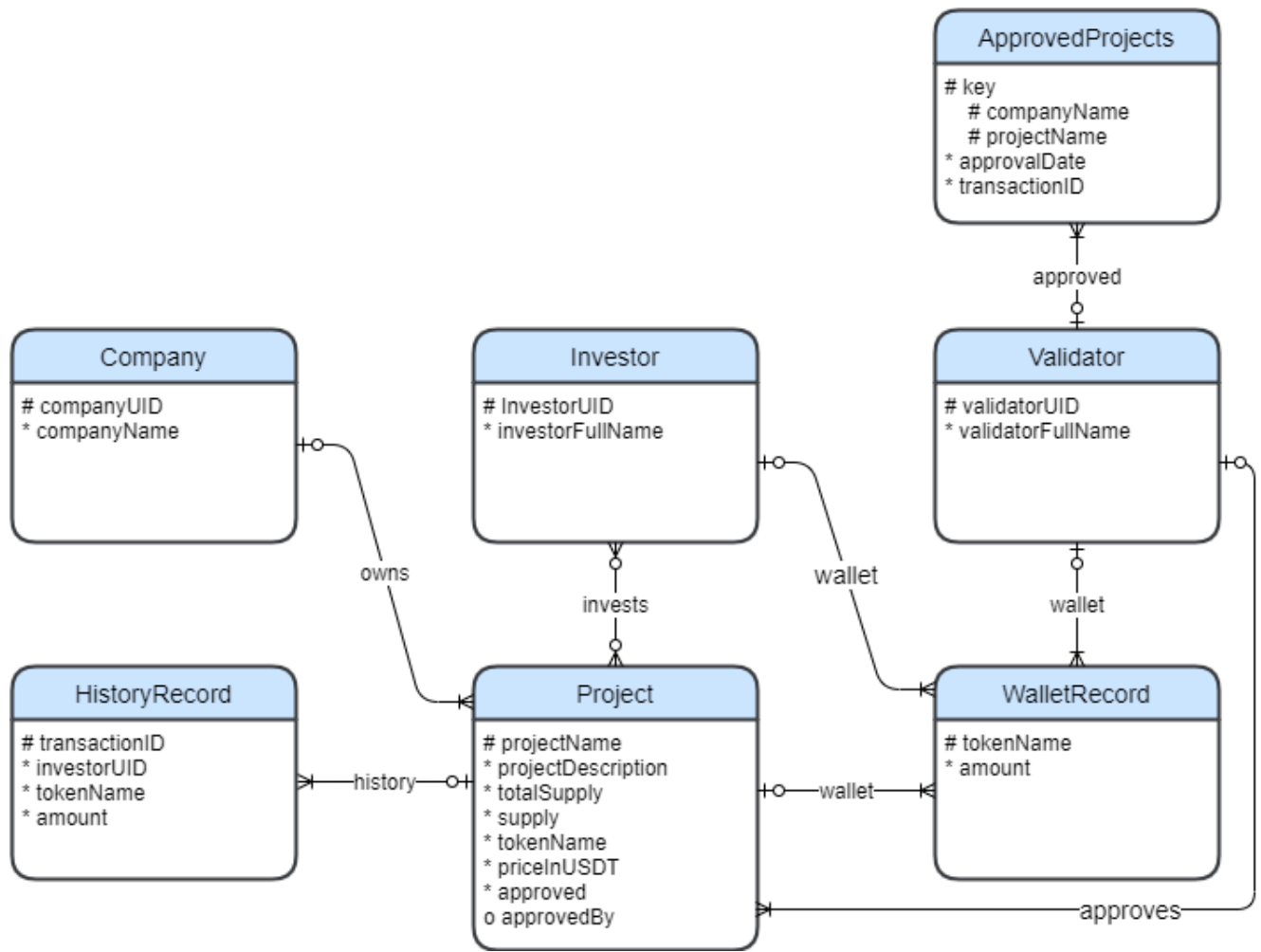
```
export const createInvestor = async (gateway, investorUID, investorFullName) => {  
  try {  
    const result = await sendTransaction(gateway, transaction: {  
      name: 'createInvestor',  
      props: [investorUID, investorFullName]  
    })  
    gateway.disconnect()  
    return result;  
  } catch (e) {  
    console.log(e);  
  }  
};
```

### Примітки

- 1 Відбувається виклик методу чейнкоду «createInvestor» для створення в базі нового інвестора.
- 2 Параметрами передаються унікальний ідентифікатор користувача та повне ім'я користувача.



## Додаток В. Схема бази даних



## Додаток В (продовження)

```

{
  "investors": {
    "investorUID": {
      "investorFullName": "investorFullName",
      "wallet": [
        {
          "tokenName": "tokenName1",
          "amount": 123
        },
        {
          "tokenName": "tokenName2",
          "amount": 456
        }
      ]
    }
  },
  "validators": {
    "validatorUID": {
      "validatorFullName": "validatorFullName1",
      "validatedProjects": [
        {
          "companyName": "companyName",
          "projectName": "projectName",
          "approvalDate": 1620751366232,
          "transactionId": "ba781d7588cbe79460eeb249134e8e411211a4a8a5dbc0c1
d769d5299e57ea12"
        }
      ],
      "wallet": [
        {
          "tokenName": "tokenName1",
          "amount": 0
        },
        {
          "tokenName": "tokenName2",
          "amount": 0
        }
      ]
    }
  },
  "companies": {
    "companyUID": {
      "companyName": "companyName1",
      "projects": [

```

## Додаток В (продовження)

```

{
  "projectName" : "projectName2",
  "projectDescription": "Some text goes here",
  "totalSupply": 100000,
  "supply": 5000,
  "tokenName": "FOO",
  "priceInUSDT": 10,
  "approved": true,
  "approvedBy": "Validator Name",
  "wallet": [
    {
      "tokenName": "USDT",
      "amount": 50000
    }
  ],
  "investmentHistory": [
    {
      "transactionId": "ba781d7588cbe79460eeb249134e8e411211a4a8a
5dbc0c1d769d5299e57ea12",
      "investorUID": "JGraham",
      "tokenName": "USDT",
      "amount": "50000"
    }
  ]
}

```

## Додаток Г. Взаємодія з базою даних LevelDB

```
async createInvestor(ctx, investorUID, investorFullName) {
  const investorsAsBytes = await ctx.stub.getState(this.INVESTORS_COLLECTION);

  const investorsAsObject = JSON.parse(investorsAsBytes.toString());
  if (!!investorsAsObject[investorUID] && investorsAsObject[investorUID].toString().length !== 0) {
    throw new Error('Investor with current name already exists');
  }

  investorsAsObject[investorUID] = {
    investorFullName: investorFullName,
    wallet: []
  }

  const newRecordInBytes = Buffer.from(JSON.stringify(investorsAsObject));
  await ctx.stub.putState(this.INVESTORS_COLLECTION, newRecordInBytes);
  return JSON.stringify(investorsAsObject, { replacer: null, space: 2 });
}
```

### Примітки

- 1 На початку отримується колекція інвесторів у вигляді масиву байтів.
- 2 Відбувається розбір (parsing) байтів у формат JSON.
- 3 Відбувається перевірка чи є користувач з таким ідентифікатором в системі.
- 4 Якщо ні, створюється новий користувач інвестор та його гаманець.
- 5 Пара ключ-значення записується в базу даних.