Хмельницький національний університет

Факультет інформаційних технологій

Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

Лагоди Аліси Павлівни

на здобуття ступеня вищої освіти Бакалавра

Система контролю доступу на базі блокчейн-технології

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

Освітня програма Кібербезпека

Шифр КРБКБ.200111.20.01.15 ПЗ

Виконала студентка 4 курсу група КБ-20-1 Аліса ЛАГОДА

Керівник канд. техн. наук, доцент Вікторія ОРЛЕНКО

Нормоконтролер старший викладач Сергій МОСТОВИЙ

До захисту допускаю:
Завідувач кафедри кібербезпеки Юрій КЛЬОЦ

19 06 2024 р.

Хмельницький 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Кібербезпеки

Рівень вищої освіти Бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

Освітня програма Кібербезпека

ЗАТВЕРДЖУЮ

Завідувач кафедри кібербезпеки

Юрій КЛЬОЦ

15 лютого 2024 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Лагоді Алісі Павлівній

1 Тема роботи Система контролю доступу на базі блокчейн-технології

Керівник роботи Орленко В.С.

Затверджено наказом ректора університету від 15 лютого 2024 № 8

2 Строк подання студентом кваліфікаційної роботи на кафедру

3 Вихідні дані до роботи Теоретичні поняття та принципи блокчейну. Нормативні документи та стандарти щодо інформаційної безпеки. Технічнф характеристики програмного обланання.

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Розглянути технології блокчйн, функціонування та роботу вже існуючих систем. Аналіз уже існуючих систем контролю доступу. Можливі варіанти інтеграції блокчейн-технології в систму контрою доступу. Тестування та аналіз системи, чи відповідає вона необхідним норма безпеки.

5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень)
В роботі присутні обов'язкові додатки А та Б. Додаток А містить чотири креслення по роботі. Додаток Б містиць презентаційні матеріали.

6 Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Мостовий С.В., старший викладач кафедри кібербезпеки		

7 Дата видачі завдання 16 лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
Вибір і затвердження теми кваліфікаційної роботи	Січень-Лютий	
Ознайомлення з предметною областю	Лютий	
Дослідження існуючих рішень	Березень	
Постановка задачі	Березень	
Визначення загальних принципів рішення задачі	Березень-Квітень	
Деталізація принципів рішення задачі	Квітень	
Розробка проєктних рішень	Квітень	
Апробація проєктних рішень	Травень	
Оформлення пояснювальної записки згідно вимог	Травень	
Оформлення графічної частини	Червень	
Захист КР	Червень	

Студентка Аліса ЛАГОДА

Керівник кваліфікаційної роботи Вікторія ОРЛЕНКО

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Система контролю доступу на базі блокчейн-технології».

Автор проекту: Лагода Аліса Павлівна

Керівник проекту: Орленко Вікторія Сергіївна

Пояснювальна записка: 63 с., 10 рис., 16 табл., 2 дод., 30 джерела.

Графічна частина: 16 презентаційних слайдів.

Метою даного проєкту є дослідження та створення програмного забезпечення для системи контролю доступу. Було проведене дослідження системи контролю доступу на базі блокчейн технології. Традиційні централізовані системи мають низку вразливостей, що робить їх менш ефективними в умовах сучасних кіберзагроз. Блокчейн технологія пропонує децентралізований підхід, який забезпечує високий рівень безпеки, прозорості та незмінності даних.

Особлива увага приділяється питанням безпеки, прозорості та незалежності системи. Розглянуті основні виклики, з якими можна стикнутися при впровадженні блокчейн у системи контролю доступу, зокрема проблеми скалюваності, конфіденційності та витрат на обробку транзакцій.

Результати дослідження свідчать про значний потенціал блокчейн-технології у сфері контролю доступу та можливість покращення існуючих систем завдяки її впровадженню. Використання блокчейн для контролю доступу може суттєво підвищити рівень безпеки даних, знизити ризики несанкціонованого доступу та забезпечити високий рівень довіри до системи з боку користувачів.

1.06.24

Дата

Підпис

ANNOTATION

Course project:: "Access control system based on blockchain technology".

Project author: Alisa Pavlivna Lagoda

Project manager: Orlenko Victoria Serhiivna

Explanatory note: 63 p., 10 figures, 16 tables, 2 appendices, 30 sources.

Graphic part: 16 presentation slides.

The purpose of this project is to research and create software for the access control system. A study of the access control system based on blockchain technology was conducted. Traditional centralized systems have a number of vulnerabilities that make them less effective in the face of modern cyber threats. Blockchain technology offers a decentralized approach that ensures a high level of security, transparency and immutability of data.

Special attention is paid to issues of security, transparency and independence of the system. The main challenges that can be faced when implementing blockchain in access control systems are considered, including scalability, privacy and transaction processing costs.

The results of the study indicate the significant potential of blockchain technology in the field of access control and the possibility of improving existing systems thanks to its implementation. The use of blockchain for access control can significantly increase the level of data security, reduce the risks of unauthorized access and ensure a high level of trust in the system by users.

1.06.24
Date


Signature

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

БЧ – блокчейн;

БД – база даних;

СКД – система контролю доступу;

ПЗ – програмне забезпечення;

RFID – англ. Radio Frequency IDentification, радіочастотна ідентифікація;

UID – англ. унікальний цифровий підпис;

ВСТУП

Що таке інформація і яку цінність вона собою представляє? Дати визначення інформації можна таким способом: це відомості, незалежно від їх форми подання. Основною властивістю є те, що вона не матеріальна, тобто її параметри неможливо виміряти фізичними методами, але вона передається за допомогою матеріальних носіїв. Представниками цього визначення можуть бути: людський мозок, паперовий носій, машинні носії і тому подібне. Інформація це цінний ресурс компанії, тому вона потребує захисту.

Під інформаційною безпекою розуміється стан захищеності інформації від внутрішніх та зовнішніх загроз за такими критеріями як конфіденційність, доступність та цілісність.

Оскільки даний ресурс майже цілком переведений у цифровий вигляд та зберігається в комп'ютерах, тому виникає потреба у віртуальних засобах контролю.

На сьогоднішній день системи контролю доступу використовуються на багатьох підприємствах, не залежно від їх розміру чи області, у якій вони звершують свою діяльність. СКД потрібні для захисту цінної інформації, тобто це являється фундаментом збереження конфіденційності та цілісності інформації. Дана система реалізує у собі комплекс програмних та технічних засобів безпеки, спрямованих на обмеження, реєстрації та контролю входу-виходу об'єктів на базі підприємства. Вони перешкоджають несанкціонованому проникненню у зони з обмеженим доступом. Аналогічно можуть використовуватися і різні складові обмеження доступу. Існує безліч різних видів СКД від багаторівневих до найпрстіших за допомогою камери та охоронця що відчиняє двері. Оскільки ми живемо в вік технологій то переплачувати робочому це не актуально, все частіше на входах керують машини. Існує також багато компаній, для яких встановлення даних систем є основним прибутком.

Однією з найбільш захищених фінансових систем є біткоїни, вони використовують технологію блокчейн. Простими словами це є інструмент для передачі та зберігання даних. ЇЇ принцип роботи полягає у тому, що база даних зберігається не на одному комп'ютері, а розподілена між усіма учасниками системи. Цей принцип полягає у тому, що наша розподілена база даних, як певна книга у копіях існуюча в кількості рівній користувачам. Дані копії, вони пов'язані між собою і регулярно перевіряють один-одного, оновлюючись. Ця система піддавалася неодноразовим атакам, але вони не осягали успіху, оскільки для успішного проведення, має спрацьовувати принцип 51%. Суть полягає у тому, щоб одна людина або група контролювала більшу частину вузлів. Якщо усі вузли погодяться то блок приймається істинним, що фактично не можливо із-зі величезної кількості вузлів.

Вже сьогодні багато держав[17] використовують дану систему в свої цілях, наприклад для реалізації документального простору у лікарнях і в тому подібних. Але для розвитку повного потенціалу необхідні зусилля міжнародної спільноти, а не окремих країн.

Дану технологію можна впроваджувати в навчальний процес, продовольчий, а також безпека цієї системи дозвоює інтеграцію у банківські системи[19]. Ви тільки уявіть наскільки швидко можна було би купувати, якби все виконувалося на рівні смарт-контрактів. Тут ви володієте машиною, але хтось забажав купити і в долі хвилини відбувається транзакція. Ви отримуєте кошти, а хтось право власності.

Метою кваліфікаційної роботи є створення системи контролю доступу на базі блокчейн технології, що буде керувати доступом до приміщень у будь-якому підприємстві. Для цього будуть використані схожі методи як на платформі Bitcoin. Буде розроблено не тільки програмно-апаратну частину, але й інтерфейсне використання.

Щоб досягнути мети будуть проведені наступні кроки:

– аналіз існуючих систем контролю доступу, що використовуються на всесвітніх рівнях;

– дослідження блокчейн-технологій;

– оцінка СКД на технології блокчейн;

– створення архітектури та прогнозування роботи СКД;

– розробка моделі створюваної системи на різних етапах;

– здобування нових навичок програмування;

– тестування розробленої системи.

1 ОСНОВНІ ПОНЯТТЯ ТА ПРИНЦИПИ БЛОКЧЕЙН ТЕХНОЛОГІЙ ТА ВИДИ СИСТЕМ КОНТРОЛЮ ДОСТУПУ

1.1 Історія та визначення блокчейну

Вперше ідея даних технологій була описана в 1991 р. [2] вченим Стюартом Хабером та Скоттом Сторнеттом, вони спробували застосувати криптографічні методи для захисту цифрового документу від фальсифікації. Ще через рік систему вдосконалили деревами Меркла, що дозволило складати кілька документів у блок. Але дана технологія залишалася в затінку аж до 2008 року, хоча ще у 2004 була представлена, котру вважають прототипом крипто валют, та перший запуск відбувся у грудні 2008 електронних грошей Bitcoin. Після гучного фіаско система почала використовуватися різними розробниками.

Вже у 2015 році на ринок вийшла платформа Ethereum. Її фундаментальною особливістю стали «Smart Contracts». Існує можливість використовувати дані програми або скрипти, що компілюються в байт-код, для здійснення транзакції при дотримання певних умов.

Блокчейн став конкурентом традиційної фінансової системи при появі у 2019 році DeFi (фінансових сервісів та інструментів, побудованих на блокчейні).

Беручи до уваги дані порталу Google Trends [2] частота запитів при пошуку у Google за словом ‘blockchain’ змінювалася за графіко представленим на рисунку 1.1

Рисунок 1.1 – Кількість запитів у Google при пошуку за словом “blockchain”

Зазначені технології являють собою децентралізовану базу даних, яка вміщає в собі безпеку та прозорість (рисунок 1.2). ЇЇ Децентралізованість полягає у тому що база даних підтримується та синхронізується на кількох вузлах (комп'ютери, що зберігають копію бази даних). Дані у блокчейні не можна видалити чи змінити і саме це забезпечує їхню точність та надійність. Також усі дії видно й іншим учасникам що представляє собою прозорість та полегшує перевірку.

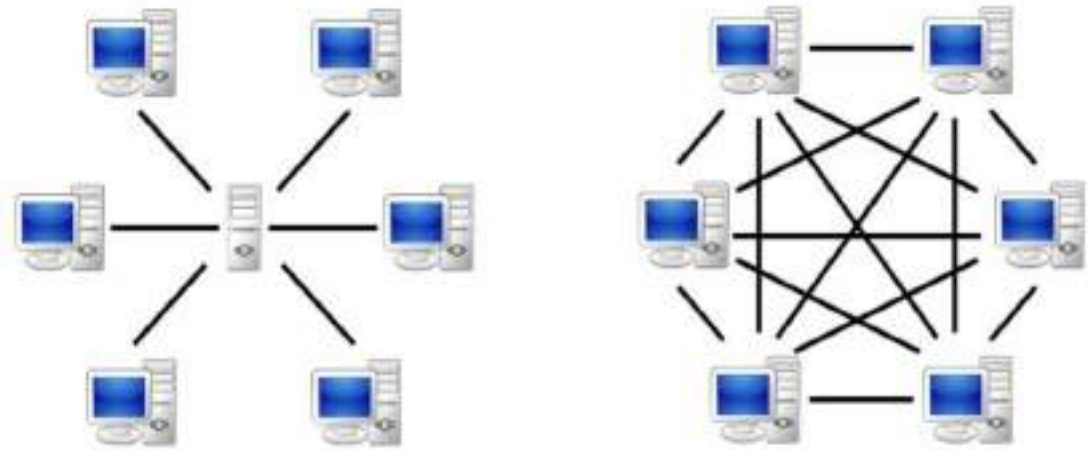


Рисунок 1.2 – Децентралізованість у БЧ

Існує три види блочейну:

- публічні блочейни;
- приватні блокчейни;
- блокчейн-соціум.

Публічні реалізують в собі неконтрольовані жодною організацією або особою блокчейни, а користувачі мають право лишитися анонімними. Будь-хто маючи підключення до інтернету може використовувати даний вид. Тут усі транзакції записуються і ніхто не може їх змінити. Але при великому потоці вузлів існує низька пропускна здатність.

Приватні використовуються певною організацією для власних цілей і вони не включають аспект загальнодоступності. У них мережа підтримується

консорціумом організацій чи приватних осіб і лиш затвердженим учасникам можна приєднуватися до мережі. Тобто це вже являє собою контрольоване середовище з вищою швидкістю транзакцій, що є також конфіденційним.

У блокчейн-консорціумі група організацій є у основі перевірки транзакцій та обслуговування мережі. Організації мають різний ступінь контролю залежно від ролі та рівня довіри. Тобто це щось середнє між публічними та приватними, оскільки тут присутнє більш безпечне та контрольоване середовище на відмінно від загальнодоступних, залишаючи збереженим при цьому рівень децентралізації та довіри.

Переваги БЧ:

– у системі не існує єдиної точки контролю, і це забезпечує їй стійкість до атак і витоку даних;

– видимість транзакцій усім учасникам, що забезпечує їх прозорість і робить легшою перевірку;

– виникає можливість створення швидших та надійніших транзакцій, оскільки працює без посередників;

– не можливість змінення або видалення транзакцій, що фундаментує надійність та точність;

– блокчейн здатен зменшити витрати, усуваючи посередників;

– можливість взаємодії учасників мережі один з одним без неоюхідності довіри один одному.

Отже, цілком очевидним є те, що використання БЧ відкриває перед людством нові можливості, які покращать наше життя. З плином часу дана технологія входить у нові сфери діяльності. Розлогим застосуванням технологій до сьогодення є облік криптовалютних активів. Вже розпочали тестувати або впроваджувати елементи блокчейну у власних напрямках велика кількість державних та комерційних організацій. Завдяки своїм технологіям блокчейн отримує все більше прихильників.

1.2 Основні складові технології

Структура блокчейну може змінюватися від галузі використання та вимог щодо системи. Але як правило тут обов'язково мають бути блоки – це набір даних, що містить сукупність транзакцій та унікальний хеш. Він включає у себе посилання на хеш попереднього блоку, створюючи сам ланцюжок блоків тобто блокчейн.

Транзакції[19] – це запис про обмін цінностями, саме вони групуються у блоки та додаються в ланцюжки. Щоб підробити транзакцію в одному блоці, потрібно змінити всі файли, що знаходяться за ним і це робить задачу практично неможливою. Робота та складові транзакції наведено у табл. 1.1.

Таблиця 1.1 – Архітектура транзакції

Складові транзакції	Роль у транзакції	Значення
Вхідні дані	Участь у створенні транзакції та валідації при додачі в блок	включають інформацію про попередні транзакції, також є цифровий підпис відправника, що підтверджує його право на здійснення операції
Вихідні дані	Відправка активу та формування транзакції	Вказують на власників активів та їх адреси, також містить умови витрат, які отримувач повинен виконати для використання активів
Сума	Перевірка балансів при валідації транзакції	кількість переданих активів або криптовалюти
Підпис	Підтверджує ініціалізацію власником	цифровий підпис, що забезпечує автентичність транзакції

Криптографічний підпис – це математичні функції, що існують для перевірки дійсності транзакцій. Також дана структура може включати в себе таку функцію як смарт-контракти. В основі їх роботи полягає ідея автоматичного виконання угоди між продавцем та клієнтом, при якому усі дані вписуються в код, а також з використанням закритого та відкритого ключа, що є гарантом безпеки передачі даних.

Хеш-функції – це криптографічні алгоритми, що змінюють дані на рядок символів фіксованої довжини. Їх використання забезпечує створення унікального хешу для кожного блоку це і є підставою безпеки. Вхідні дані – це ключ, а результатом є хеш. За рахунок зв'язку блоків цю систему нікому ще не вдавалося взламати. Приклади популярних хеш-функцій наведено у табл. 1.2.

Таблиця 1.2 – Види хеш-функцій

Назва	SHA-256	Keccak-256 (SHA-3)	Blake2b
Характеристики	Використовується в Bitcoin і багатьох інших блокчейнах. Генерує 256-бітний хеш. Застосовується для хешування транзакцій, блоків і в алгоритмі Proof of Work.	Використовується в Ethereum. Генерує 256-бітний хеш. Забезпечує високий рівень безпеки та стійкості до криптографічних атак.	Використовується в деяких сучасних блокчейнах, таких як Zcash. Відомий своєю високою швидкістю і безпекою.

Вузли – це комп'ютери на яких працює програмне забезпечення (пз) їх також називають нодами. Їх роль полягає у видобуванні нових блоків, підтверджені або скасуванні транзакцій, передаванні даних та збереженні копії всього ланцюжка. Існують такі види Вузлів як повні вузли, легкі вузли, архівні вузли та майнерські вузли. Їх функції описані у табл. 1.3.

Таблиця 1.3 – Типи вузлів

Назва	Функції	Переваги	Недоліки
Повні вузли (Full Nodes)	Зберігають повну копію блокчейну, перевіряють та валідують всі транзакції та блоки, забезпечують максимальну безпеку та децентралізацію мережі	Високий рівень безпеки, перевірка всіх транзакцій і блоків, незалежність від інших вузлів	Великі вимоги до зберігання даних та обчислювальних ресурсів
Легкі вузли (Light Nodes)	Зберігають часткову інформацію про блокчейн, зазвичай лише заголовки блоків. Використовують інші вузли для підтвердження транзакцій	Менші вимоги до зберігання даних та обчислювальних ресурсів, швидке налаштування	Менша безпека, залежність від повних вузлів для підтвердження транзакцій
Архівні вузли (Archive Nodes)	Зберігають повну історію всіх станів блокчейну, включаючи всі проміжні стани та історичні дані	Доступ до повної історії транзакцій та станів, корисні для аналітики та аудиту	Дуже високі вимоги до зберігання даних
Майнерські вузли (Mining Nodes)	Виконують спеціальні обчислення для створення нових блоків (майнінг). Вони підтверджують транзакції та додають їх до блокчейну	Забезпечують безпеку та створення нових блоків, отримують винагороду за майнінг	Високі вимоги до обчислювальних ресурсів та енергії

Як нода виступати може будь-яка людина з присутнім пз. Забезпечення безпеки мережі та узгодження даних між вузлами залежить від принципу роботи блокчейну, тобто алгоритму консенсусу. Завдяки ньому система не потребує адміністраторів та центральних сховищ тому що, вона підтверджує правильність інформації у кожному блоці.

Загалом структура та дизайн блокчейну спрямована на впровадження безпечної та децентралізованої системи передачі та зберігання інформації, що відбувається без необхідності посередника.

1.3 Принцип роботи

Оскільки вся система формується на принципі блоків і як вже неодноразово згадано що їй притаманна Децентралізованість, то доречно зазначити яким способом розташовуються блоки. Вони організовані у хронологічній послідовності, об'єднанні один з одним та захищені криптографічним методами, що можна побачити на рисунку 1.3.

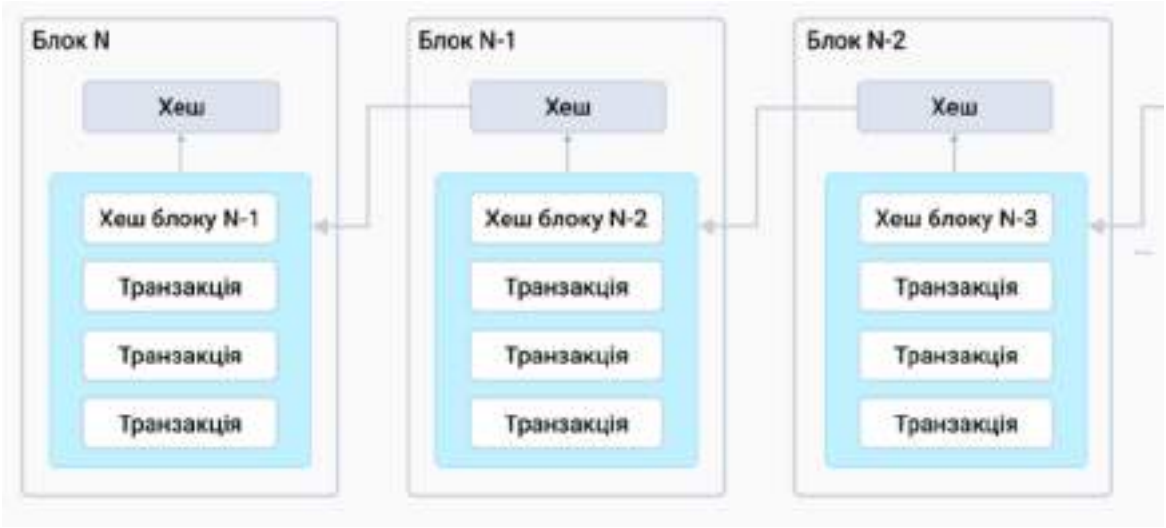


Рисунок 1.3 – Формування блоків

Кожен даний блок мість хеш попереднього та корисне наповнення, тобто інформація про транзакції, угоди, дані про фізичну особу чи об'єкт і тому подібне. І тут можна лише додавати блоки, таких дій як видалення чи змінення не ісує. Якщо спробувати змінити інформацію паралельно відбудеться і зміна хешу, тому даний блок буде відкинений мережею. При підтверджені істиності блоку він розповсюджується по всій мережі, верифікація реалізується за допомогою консенсусних алгоритмів.

Блокчейн може підтримувати смарт-контракти, що є програмами, які автоматизують та контролюють виконання угод між сторонами. Смарт-контракти дозволяють автоматизувати та узгоджувати умови виконання угод, що забезпечує більшу ефективність та надійність. Загальна робота блокчейну продемострована на рисунку 1.4.

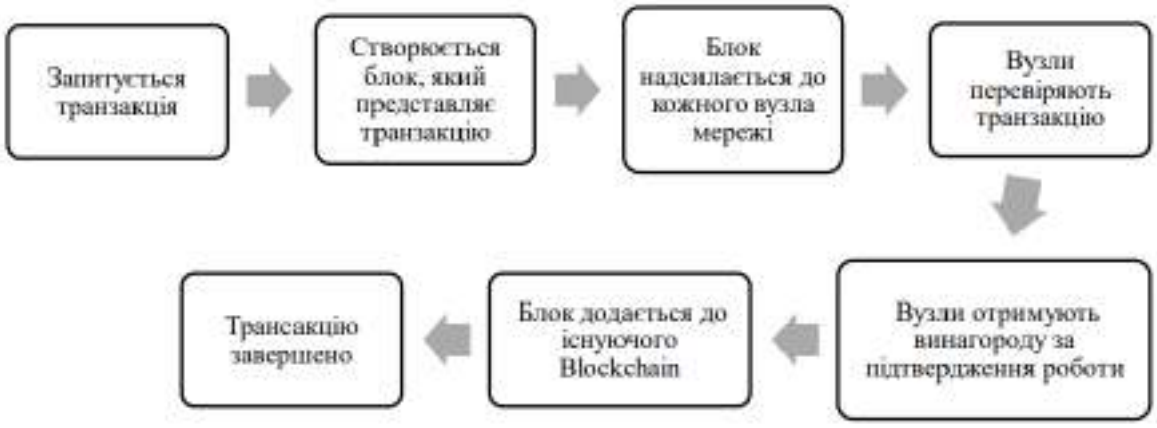


Рисунок 1.4 – Функціонування технології блокчейн

Отже, сукупність цих процесів створює, децентралізовану, безпечну та надійну інфраструктуру,я ка дозволяє учасникам мережі довіряти один одному та здійснювати безпечні та швидкі транзакції без посередників. Що в котре доводить надійність та безпеку даної технології на фоні існуючих рішень та систем, що використовуються в індустрії.

1.4 Огляд традиційних систем контролю доступу

Система контролю доступу[1] – це комплекс технічних та програмних засобів безпеки, що виконує регулювання входу та виходу приміщень, транспортних об’єктів чи людей на територіях, що знаходяться під охороною, для адміністративного монторингу та попереджень несанкціонованого доступу. Приклад роботи СКД ми бачимо на рисунку 1.5.

СКД можна класифікувати на:

- локальні та мережеві;
- дротові та радіоканальні;
- біометричні та з застосуванням пін-коду або магнітної карти.

Рисунок 1.5 – Схема роботи простої СКД

Локальні керують лише одним конкретним об’єктом, не під’єднуючись до комп’ютера. Навідмінно від них мережеві дозволяють тримати під контролем безліч точок, дають можливість розмежувати доступ за часом, об’єктами, статусом, при цьому ще й дозволяють дистанційно керувати точками доступу.

Дротові або провідні[12] СКД це ті, які для функціонування потребують прокладання екранованого кабелю, в деяких випадках просто двопровідна вита пара. Їх можна відзначити надійністю та стабільністю, оскільки вони мають меншу схильність до перешкод. Радіоканальне з'єднання характеризується передаванням даних задопомогою радіочастотного зв'язку, що в свою чергу полегшує та прискорює встановлення системи.

Біометричні СКД базуються на унікальних фізичних або поведінкових характеристиках особи, для ідентифікації та аутентифікації. Вона складається з таких аспектів, як сканування відбитків пальців, розпізнавання лиця, сканування радужки ока також розпізнавання голосу. На даний момент такий тип захисту набирає все більше популярності, оскільки біометричні дані є унікальними для кожної людини, що робить систему стійкою до шахрайства чи підробки.

При встановленні відповідного обладнання можна отримати не тільки гарант безпеки щодо сторонніх осіб на підприємстві, а й контролювати пересування персоналу. Оскільки на великих підприємствах немає такої змоги власноруч керувати кожним працівником. Ще воно розв'язує безліч завдань таких як:

– моніторинг входу та виходу працівників, підозріла активність, у позаробочий час перебування на об'єкті;

– можливість обмежити доступ до окремих відділів на підприємстві;

– швидке виявлення спроби потрапити на об'єкт невідомих осіб;

– заощадження витрат на утримання спеціального охоронного персоналу, який має вести облік робочого часу;

– можливість визначення осіб, що перебували на підприємстві у разі виникнення надзвичайної ситуації.

Система контролю доступу складається з кількох основних компонентів, які працюють разом для забезпечення безпеки та обмеження доступу до певних зон або ресурсів. Основні з них наведено в табл. 1.4.

Таблиця 1.4 – Складові СКД

Функція	Складові	Приклад
Ідентифікація користувача	Ідентифікатори	Карти, брелки, мобільні додатки, біометричні дані
	Логін та пароль	Ім’я користувача та пароль для доступу до системи
Засоби контролю доступу	Зчитувачі	Карткові зчитувачі, біометричні сканери, NFC або RFID зчитувачі
	Контролери	Пристрої, які приймають сигнали від зчитувачів та вирішують щодо надання або відмову в доступі
Програмне забезпечення та управління доступом	Системи управління	ПЗ для налаштування, моніторингу та керування СКД, включаючи управління користувачами, встановлення прав доступу, реєстрацію подій
Засоби фізичного обмеження доступу	Електронні замки	Замки, що керуються сигналами від контролерів
	Турнікети та шлюзи	Пристрої для обмеження проходу людей у певні зони
	Двері з електронним управлінням	Двері, які можна відчиняти або зачиняти за допомогою СКД
Системи сповіщення та сигналізації	Сигналізація	Звукові та світлові сигнали
	Повідомлення	Сповіщення через SMS, електронну пошту або інші канали

Засоби ідентифікації є важливими для СКД, вони класифікуються за функціональним призначенням пристроїв, функціональними характеристиками і стійкістю до несанкціонованого доступу у табл. 2.1 подано дану інформацію.

Таблиця 2.1 – Класифікація ідентифікаторів доступу

Вид використовуваних ідентифікаційних ознак	Елемент, що лежить в основі принципу використання	Приклад ідентифікаторів
Оптичні	Мітки нанесені на поверхню або розташовані всередині ідентифікатора, що мають різні оптичні характеристики у відбитому або минаючому оптичному випромінюванні	Карти з штих-кодом або топографічні мітки
Електронні контактні	Електронний код, записаний в мікросхемі ідентифікатора	Електронні ключі
Електрнні радіочастотні	Радіоканал, використовуваний для передачі даних	Безконтактні карти доступу
Механічні	Елементи конструкції ідентифікаторів	Механічні ключі з перфораційними отворами

На даний момент найпоширенішими СКД є з використанням пін-коду або картки RFID рисунок 1.6. Для створення карти використовують карт-принтери. В багатьох офісах захист використовується навть при переміщенні між кабінетами. А ще можливість керувати доступом, адміністратор може змінити права картки. Зручна та швидка система.

Рисунок 1.6 – Різномані види СКД

Кожна з вище перелічених систем має свої переваги і недоліки, вибір конкретної залежить від потреб безпеки, бюджету та умов даного об’єкту.

Встановлення відповідного обладнання не тільки гарантуватиме безпеку, але й вирішить важливі інші завдання. Наприклад це також відстежуватиме прихід та відхід працівників, можлово хтось перебуває на об’єкті в позаробочий час. Ще однією функцією є обмеження доступу до деяких приміщень. А ще наявність можливості визначити людей, котрі перебували на місці під час виникнення надзвичайної ситуації.

Ще деяких плюсів можна виокремити те, що багато сучасних СКД можуть вести детальний журнал доступу, який відображає, хто, коли і де отримував доступ. Це дозволяє відстежувати активність і проводити аудит

безпеки. Гнучкість в управлінні доступом полягає у тому, що адміністратори можуть легко змінювати права доступу користувачів згідно з їх поточним статусом або потребами безпеки. Ще оним рішенням є те що в нас завжди не вистачає часу, але автоматизовані СКД можуть значно зменшити час, який витрачається на ручний контроль доступу і перевірку осіб.

Розглянувши багато СКД було виявленно деякі незручності, що притягуються разом з цими системами. Наприклад вартість впровадження та обслуговування СКД може бути дуже дорогим, особливо для великих приміщень або складних систем.

Багато СКД потребують електропостачання для своєї роботи, тому перерви в постачанні електроенергії можуть призвести до втрати контролю над доступом. Або потрібно щоб офіс мав власний енерго ресурс для забезпечення постійності корекного працювання СКД, що знову несе з собою великі затрати.

Якщо не належним чином налаштована або захищена, система СКД може бути вразливою до обхідних шляхів або атак. СКД можуть бути схильні до технічних проблем, таких як збої обладнання або програмного забезпечення, що може призвести до перерв у роботі та втрати безпеки.

Деякі методи ідентифікації, такі як використання біометричних даних, можуть порушувати приватність користувачів, якщо не зберігаються або не захищаються належним чином.

Розглянувши та зваживши всі фактори та те що техноогії не стоять на місці, можна зазначити, що СКД приносять більше користі хоча й несуть деякі незручності.

2 ПРИНЦИПИ РОБОТИ ТА АРХІТЕКТУРА СИСТЕМИ КОНТРОЛЮ ДОСТУПУ НА БЛОКЧЕЙН-ТЕХНОЛОГІЇ

2.1 Архітектура системи

Вибір архітектури при проектуванні інформаційної системи є фундаментальним рішенням для створення. Зважаючи на важливість безпеки та надійності управління цифровими ключами СКД, архітектура для такої системи на блокчейні має бути детально продуманою. Основна мета полягає в тому, щоб забезпечити надійне зберігання і передачу даних доступу, які інтегруються з блокчейн-системою.

Проаналізувавши стандартні СКД я обрала доступ за ключем через картку з RFID.(Рисунок 2.1)

Рисунок 2.1 – Незапрограмована магнітна картка

Використовуються для зберігання інформації про доступ. Ці картки можуть мати низьку коерцитивність (LoCo) або високу коерцитивність (HiCo) в залежності від вимог безпеки.

Потрібно використати спеціальні принтери(Рисунок 2.2) для нанесення графічного дизайну на картку, логотип організації, ім’я прізвище, посада та фотокартка. Для цього створюється макет картки рисунок 2.3.

Рисунок 2.2 – Карт-принтер

Рисунок 2.3 – Макет друку магнітної картки

Рисунок 2.4 – Енкодер для кодування карти

Наступним етапом має бути кодування магнітної картки. Для цього нам потрібен енкодер(Рисунок 2.4). Під час створення користувача ми можемо

записати інформацію, що буде використовуватися для ідентифікації користувача, а також UID в СКД. UID використовується для ідентифікації, зв’язування користувача з відповідними записами в базі даних або блокчейні або просто пошук та звірення інформації. Зазвичай інформація кодується на першій або другої доріжці магнітної смуги.

Генерування криптографічних ключів (приватний та публічний ключ) для користувача потрібне для контролю доступу відправляє запит до блокчейн-системи для перевірки прав доступу користувача. Смарт-контракт перевіряє відповідність UID і публічного ключа, а також права доступу.. Приватний ключ зберігається безпечно на стороні користувача, а публічний ключ буде асоційований з ідентифікатором картки в блокчейн.

Для обмеження доступу потрібен пропускний пункт, зчитувач пропусків, блок живлення і все це в себе включає мережевий контролер табл. 2.1.

Таблиця 2.1 – Складові мережевого контролера та їх функції

Компонент	Функція
Зчитувач карток	Пристрій, який зчитує UID з магнітних карток або інші дані аутентифікації
Контролер доступу	Центральний пристрій, який приймає дані від зчитувача і обробляє їх
Блок живлення	Забезпечує живлення для зчитувача і контролера
Мережевий інтерфейс	Забезпечує зв'язок з блокчейн-системою та іншими частинами мереж
Замок	Фізичний механізм, який відкривається або закривається на основі рішення контролера

Локальні контролери являють собою пристрої, які встановлюються біля дверей або входів і взаємодіють зі зчитувачами магнітних карт. Вони можуть працювати автономно або підключатися до мережі рисунок 2.5.

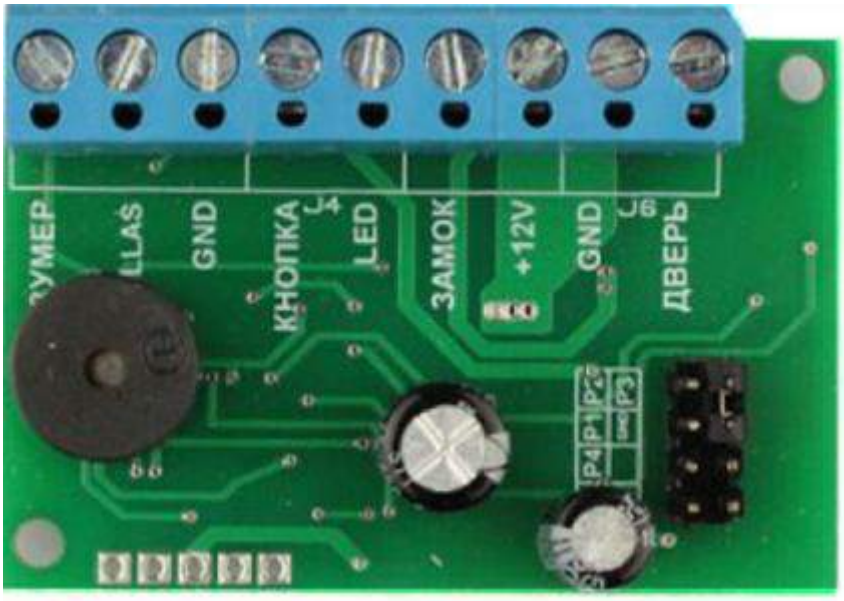


Рисунок 2.5 – Локальний контролер

Блокчейн-інфраструктура, що буде використовуватися в СКД:

– повні вузли, що зберігають повну копію блокчейну і перевіряють всі транзакції. Вони забезпечують безпеку та незмінність даних доступу;

– легкі вузли, що зберігають лише часткову інформацію про блокчейн і взаємодіють з повними вузлами для підтвердження транзакцій;

– смарт-контракти, що визначають правила доступу, зберігаються і виконуються в блокчейні.

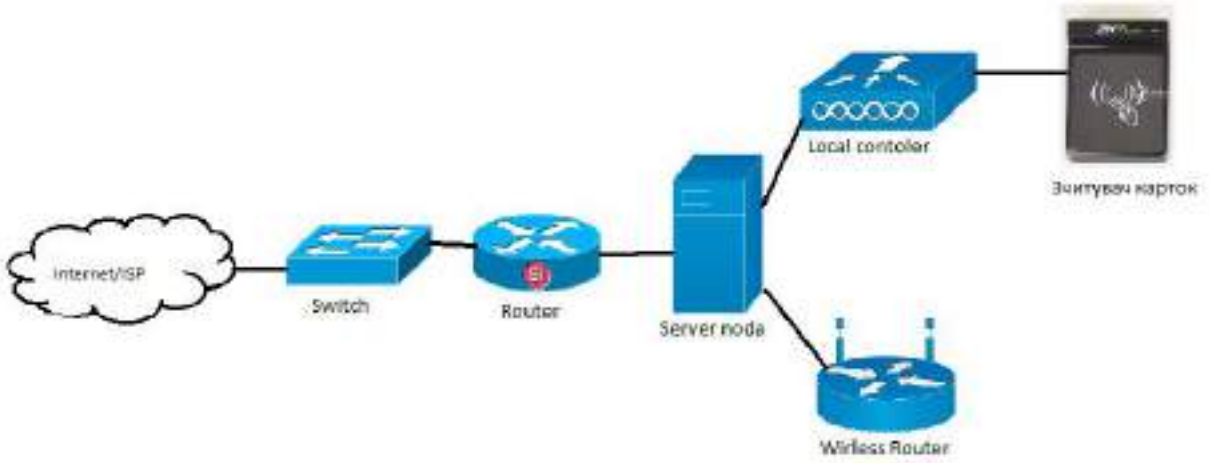


Рисунок 2.6 – Просте схематичне зображення мережі

Програмне забезпечення для управління та моніторингу мережі, включаючи засоби управління конфігураціями, моніторингу продуктивності та виявлення загроз. Мережеві пристрої та інфраструктура є основою для функціонування систем контролю доступу (СКД), особливо коли ці системи інтегровані з блокчейн або іншими мережевими технологіями. Вони забезпечують зв’язок між усіма компонентами системи, від локальних контролерів до центральних серверів або хмарних служб (рисунок 2.6).

Складові мережі:

– Мережеві комутатори, для з’єднання різних пристроїв у локальній мережі (LAN), забезпечуючи передачу даних між ними;

– маршрутизатори, для з'єднання локальних мереж з іншими мережами, зокрема з Інтернетом, і керують маршрутизацією трафіку;

– бездротові точки доступу, для розширення покриття мережі Wi-Fi, забезпечуючи бездротове підключення для пристроїв;

– Модеми, для забезпечення підключення до Інтернету через кабельне, DSL або оптоволоконне з'єднання;

– сервери, для зберігання даних, управління базами даних і виконання додатків, необхідних для роботи СКД;

– мережеві кабелі, для для з'єднання мережевих пристроїв.

Для зручного керування потрібно розробити інтерфейс додатки для адміністраторів. Підсумувавши отримали такий план інтеграції рисунок 2.7.

Proof of Work (PoW) можна використовувати в системах контролю доступу (СКД) для додаткового рівня безпеки та аутентифікації. Ідея полягає в тому, щоб вимагати від клієнта виконання певного обчислювального завдання перед наданням доступу. Це може запобігти атакам на систему, зокрема DoS-атакам, а також забезпечити додаткову перевірку автентичності запитів доступу рисунок 2.8. PoW завдання буде полягати у знаходженні такого значення nonce, яке разом з іншими даними створить хеш із певною кількістю нулів на початку.

Рисунок 2.7 – Функціонування

Цей механізм додає додатковий рівень безпеки, особливо для запобігання автоматизованим атакам, де зловмисники можуть спробувати багаторазові запити на доступ.

PoW вимагає великої кількості електроенергії для виконання обчислень. Це один з основних недоліків цього механізму, оскільки він має значний вплив на довкілля. Процес вирішення задачі є складним, але перевірка правильності розв'язання є легкою. Це забезпечує безпеку, оскільки зловмисникам складно змінити блок, не витрачаючи величезних обчислювальних ресурсів.

PoW є ефективним механізмом для забезпечення децентралізованого консенсусу та безпеки блокчейн-мереж. Однак його енерговитратність стимулює розвиток альтернативних механізмів консенсусу, таких як Proof of Stake (PoS).

```
public PoW(String challenge, int difficulty) {

    this.challenge
    this.difficulty
    challenge;
    difficulty;
    public String mine() {
    int nonce = 0;
    String hash;
    String target = new String(new char[difficulty]).replace('\0', '0');
    do {
    nonce++;
    hash = calculateHash(challenge + nonce);
    }
    while (!hash.substring(difficulty).equals(target));

    System.out.println("Pow Solved: Nonce= nonce, Hash = + hash);
    return Integer.toString(nonce);
  }

public static String applySha256(String input) {
    try {
        MessageDigest digest = MessageDigest.getInstance("SHA-256");
        byte[] hash = digest.digest(input.getBytes("UTF-8"));
        StringBuilder hexString = new StringBuilder();
        for (byte b : hash) {
            hexString.append(String.format("%02x", b));
        }
        return hexString.toString();
    } catch (Exception e) {
        throw new RuntimeException(e);
```

Рисунок 2.8 – PoW приклад коду

Отже, це механізм консенсусу, який використовується в багатьох блокчейн-системах для забезпечення безпеки і надійності мережі. Основна ідея PoW полягає в тому, що учасники мережі (майнери) виконують складні обчислювальні завдання для підтвердження транзакцій і додавання нових блоків до блокчейну.

2.2 Децентралізованість бази даних

Децентралізованість бази даних на блокчейн полягає в тому, що кожен вузол або комп'ютер в мережі блокчейна містить повну або часткову копію бази даних зображено на рисунку 2.9.

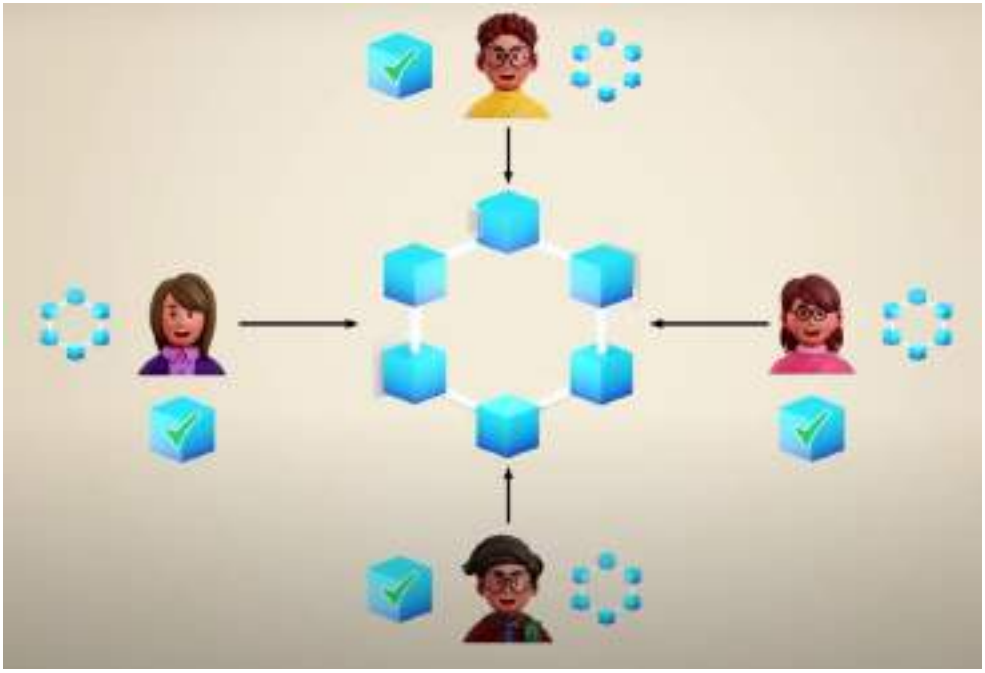


Рисунок 2.9 – Розподільність бази даних

Ланцюжок блоків використовується для журналування подій. Блокчейн може бути використаний для зберігання журналу подій, таких як створення або зміна прав доступу, а також всіх інших важливих подій, пов'язаних з СКД. Це забезпечить невідредаговану та безпечну історію всіх змін в системі.

Усі дані про користувачів, їх права доступу, історію змін і події СКД зберігаються на кожному вузлі мережі блокчейну. Це означає, що немає централізованої бази даних, контроль над якою був би у одного постачальника послуг чи організації. Децентралізованість дозволяє мережі легко масштабуватись, оскільки нові вузли можуть долучатися до мережі, розширюючи її потужності, а також дозволяє кожному учаснику мати повний контроль над власними даними. Децентралізована база даних дозволяє створювати "розумні контракти", які автоматизують багато процесів СКД, забезпечуючи автономну роботу системи без централізованого управління.

Оскільки дані розподілені по всій мережі, це ускладнює спроби змінити чи знищити дані, оскільки це потребуватиме змін в багатьох вузлах одночасно, а не лише в одній централізованій базі даних. Децентралізована база даних СКД на блокчейні є надійною і відмовостійкою. Навіть якщо деякі вузли мережі

виявляться недоступними, інші вузли продовжать працювати і забезпечувати доступ до даних.

Приклад роботи децентралізованої БД табл. 2.1

Таблиця 2.1 – Структура роботи БД

Назва процесу	Функція, що виконує	Наслідок
Розподіл даних	Надсилання копій на кінці кожного вузла	Кожен учасник має повний обсяг даних
Синхронізація даних	При додаванні нових транзакцій, надсилаються дані усім учасникам	Отримання останньої версії бази даних
Консенсунс	Перевірка алгоритмом істиності даних	Надійність та цілісність бази даних
Шифрування та безпека	Користувач отримує унікальний ключ доступу	Безпечне звернення до даних
Автономність	Взаємодія один з одним без посередників	Збереження ресурсів

Надійність та безпека системи контролю доступу (СКД) на блокчейні грають критичну роль, оскільки ці системи відповідають за збереження конфіденційної інформації та управління доступом до ресурсів. Ось кілька ключових аспектів, які впливають на надійність та безпеку СКД на блокчейні.

Безпека мережі блокчейну залежить від алгоритму консенсусу, який визначає, як приймаються нові транзакції та зміни в базі даних. Популярні алгоритми консенсусу, такі як Proof of Work (Підтвердження роботи) або Proof of Stake (Підтвердження ставки), забезпечують високий рівень надійності та безпеки мережі.

Конфіденційні дані, такі як особиста інформація користувачів або ключі доступу, повинні бути шифровані для захисту від несанкціонованого доступу.

Шифрування даних забезпечує конфіденційність інформації, що зберігається в блокчейні.

Система СКД на блокчейні повинна ефективно управляти правами доступу користувачів до різних ресурсів. Смарт-контракти можуть бути використані для автоматизації процесів призначення та зміни прав доступу, забезпечуючи точність та надійність в управлінні доступом.

Журналювання подій забезпечує зберігання історії змін у базі даних СКД на блокчейні дозволяє відстежувати всі дії користувачів та системи. Це допомагає у виявленні та реагуванні на можливі порушення безпеки та аудиті системи.

Аудит та відстеження представляє собою систему СКД на блокчейні може забезпечити можливість аудиту та відстеження всіх дій користувачів і змін у системі. Це робить процес виявлення порушень безпеки більш ефективним та допомагає у забезпеченні високого рівня безпеки.

Забезпечення регулярного резервного копіювання та можливості відновлення даних є важливим аспектом надійності СКД на блокчейні. Це забезпечує здатність швидко відновлювати дані в разі втрати чи пошкодження блокчейну.

Загалом, для забезпечення надійності та безпеки СКД на блокчейні необхідно враховувати цілу низку заходів, включаючи використання ефективних алгоритмів консенсусу, шифрування даних, правильне управління доступом та регулярне аудитування системи.

Публічність та невідмовність є ключовими характеристиками системи контролю доступу на блокчейні. Дані котрі можуть бути промоніторенними іншими вузлами зображені на рисунку 2.10

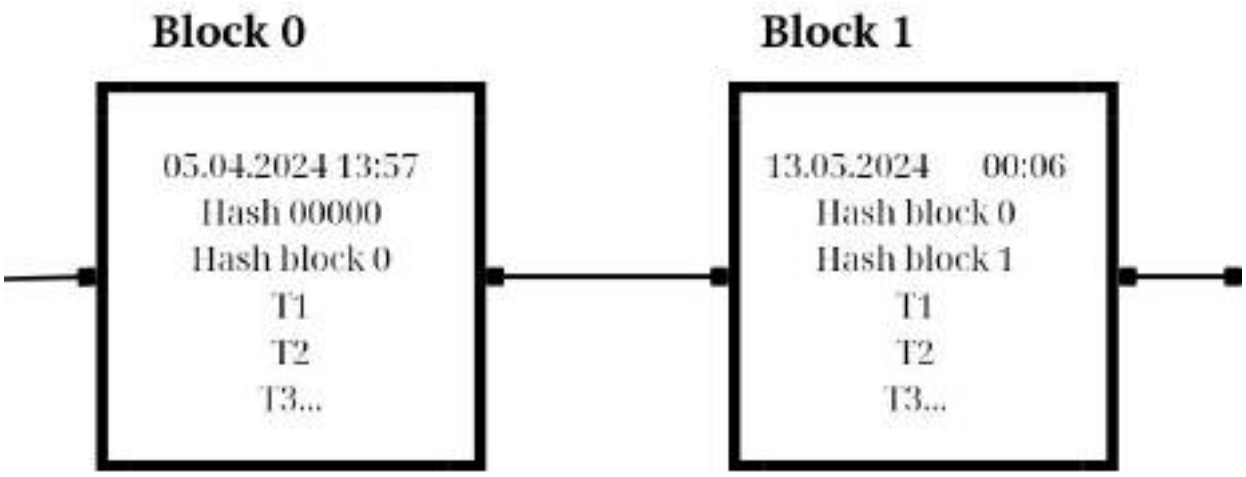


Рисунок 2.10 – Блок

Блокчейн має властивість публічності, що означає, що усі дані та транзакції, які здійснюються в мережі, є відкритими для перегляду всіма учасниками мережі. Це створює високий рівень прозорості та відкритості, оскільки всі дії можуть бути легко перевірені та переглянуті.

Однією з ключових властивостей блокчейну - це невідмовність. Це означає, що одержані дані або транзакції, які вже були додані до блокчейну, не можуть бути змінені або вилучені без консенсусу більшості учасників мережі. Це забезпечує цілісність даних та довіру до інформації, яка зберігається в мережі.

Управління доступом через смарт-контракти, тобто СКД на блокчейні використовує смарт-контракти для автоматизації процесів управління доступом. Це дозволяє забезпечити чітке виконання правил доступу та перевірку дозволів без необхідності централізованого контролю. Приклад реалізації коду на Solidity:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
contract AccessControl {
    address public admin;
```

```
    mapping(address => bool) public authorizedUsers;
    event AccessGranted(address indexed user);
    event AccessRevoked(address indexed user);
    constructor() {
        admin = msg.sender; // Установить администратора как создателя
контракта
    }
    modifier onlyAdmin() {
        require(msg.sender == admin, "Only admin can perform this action");
        _;
    }
    function grantAccess(address _user) public onlyAdmin {
        authorizedUsers[_user] = true;
        emit AccessGranted(_user);
    }
    function revokeAccess(address _user) public onlyAdmin {
        authorizedUsers[_user] = false;
        emit AccessRevoked(_user);
    }
    function checkAccess(address _user) public view returns (bool) {
        return authorizedUsers[_user];
    }
}
```

Дані та транзакції розподіляються по всій мережі, що робить їх більш стійкими до збоїв чи атак. Публічність блокчейну забезпечує прозорість та можливість аудиту всіх дій у мережі. Це дозволяє легко відстежувати та перевіряти діяльність користувачів та системи СКД. Підсумуючи, публічність та невідмовність є важливими аспектами системи контролю доступу на блокчейні,

оскільки вони забезпечують прозорість, надійність та безпеку управління доступом.

2.3 Вибір платформи для розробки

Розглянувши варіанти для програмування СКД, проаналізувавши роботи створені на блокчейн-технології, я обрала мову програмування Java. Використання Java для написання СКД на блокчейні має кілька важливих переваг. Java є мовою програмування, яка забезпечує високу продуктивність, безпеку і платформну незалежність, що робить її привабливим вибором для розробки складних і надійних систем. На рисунку 2.11 зображено переваги та недоліки Java.

Розглянувши також різні варіанти середовищ, найпопулярнішим виявилось IntelliJ IDEA. Воно є одним з найпотужніших інтегрованих середовищ розробки (IDE) для Java, і вона пропонує багато корисних функцій. IntelliJ IDEA пропонує потужні інструменти для автодоповнення коду, рефакторингу, та налагодження, що значно полегшує розробку складних проектів, таких як блокчейн.

IntelliJ IDEA підтримує різні фреймворки для тестування, такі як JUnit та TestNG, що дозволяє легко створювати та виконувати тести для блокчейн-коду.

Рисунок 2.12 – Автодоповнення

Рисунок 2.11 – Java переваги та недоліки

IDE має інтегрований відлагоджувач, який дозволяє легко знаходити та виправляти помилки у вашому коді. Це особливо корисно для розробки блокчейну, де важливо забезпечити безпеку та правильність роботи коду (рисунок 2.11). Там створено дуже зручний інтерфейс для розробки консольних додатків.

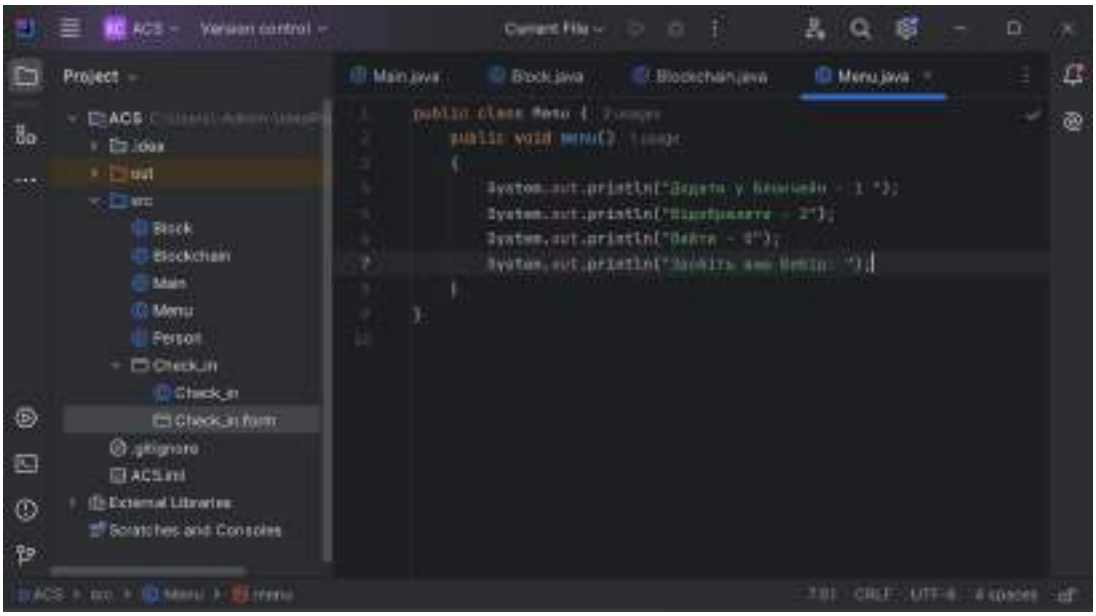


Рисунок 2.11 – Інтерфейс IntelliJ IDEA

Вартість платформи версії IntelliJ IDEA (Ultimate) є комерційною і потребує ліцензії, але існує безкоштовна версія (Community), яка має обмежені можливості. Ця безкоштовна версія інтегрованого середовища розробки (IDE) від компанії JetBrains, призначена для JVM-мов програмування, таких як Java, Kotlin, Groovy, і Scala. Community Edition надає широкий набір інструментів для розробки, тестування та рефакторингу коду.

Основні можливості:

– автозавершення коду(рисунок 2.12);

– аналіз коду(2.13);

– рефакторинг, забезпечує зручне переміщення, перейменування та виділення методу;

– навігація;

– графічний відкладчик;

– інструменти для роботи з базами даних;

– системи контролю версій;

– інструменти для компіляції та збірки проектів.

Рисунок 2.13 – Аналіз помилки

Також для зручного використання СКД потрібна розробка інтерфейсу для адміністратора. Що також підтримується платформою IntelliJ IDEA. Для роботи з інтерфейсами у Java використовуються різні бібліотеки та інструменти, такі як Swing, JavaFX, а також різні фреймворки для веб-розробки, наприклад, Spring MVC. IntelliJ IDEA надає підтримку для всіх цих технологій (рисунок 2.14).

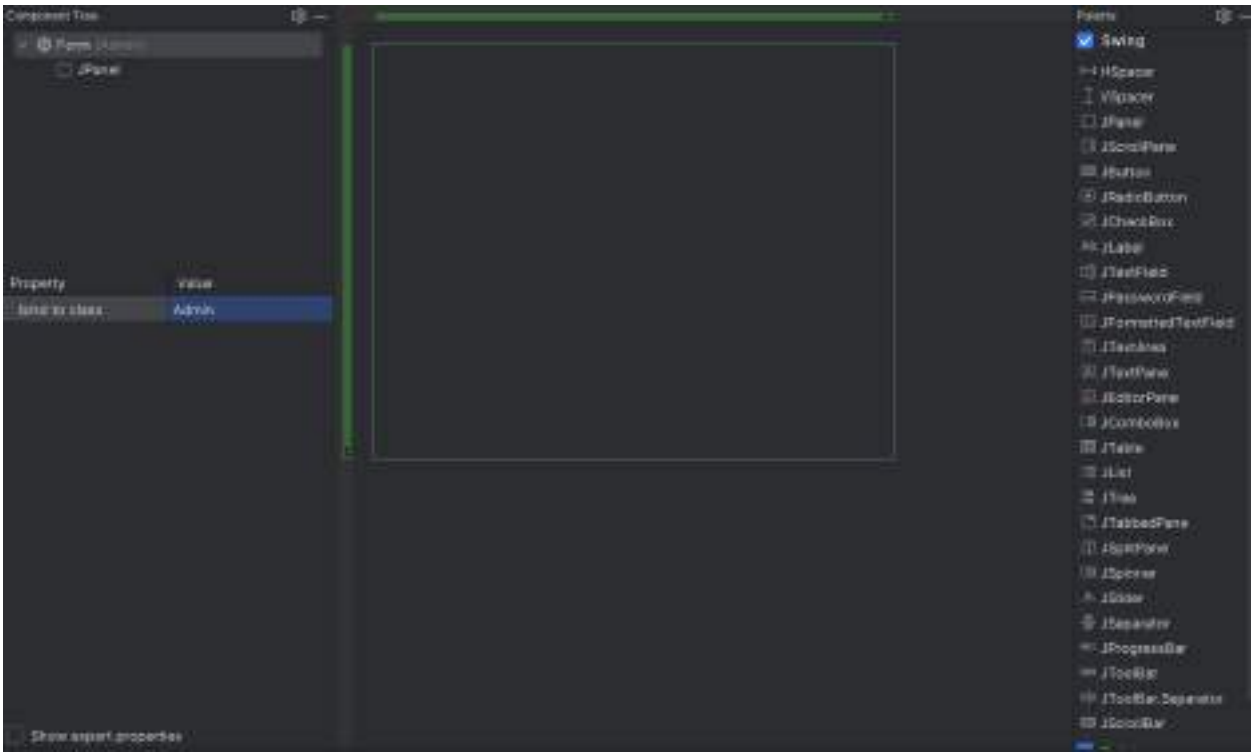


Рисунок 2.14 – Розробка інтерфейсу

Редактор дизайну: IntelliJ IDEA має вбудований редактор дизайну, який дозволяє вам візуально створювати інтерфейси безпосередньо в середовищі розробки.

З огляду на всі переваги, IntelliJ IDEA є відмінним вибором для розробки блокчейну на Java. Вона надає всі необхідні інструменти для ефективної розробки, налагодження, тестування та розгортання ваших блокчейн-додатків. Community Edition ідеально підходить для розробки системи контролю доступу на блокчейн-технології.

3 РОЗРОБКА СИСТЕМИ ТА ТЕСТУВАННЯ

3.1 Розробка блокчейну

Розробка блокчейну на Java включає кілька етапів, які охоплюють всі основні компоненти блокчейн-системи. Для початку роботи необхідно завантажити та встановити Java Development Kit з офіційного сайту Oracle та середовище розробки IntelliJ IDEA. Для початку створимо проект під назвою “Blockchain” процес показано на рисунку 3.1.

Рисунок 3.1 – Створення проекту

Наступним кроком буде реалізація класу «Block». Для цього будуть створені певні типи змінних та задані їм характеристики.

Тадлиця 3.1 – Клас «Block»

Ім’я	Тип даних	Ідентифікатор доступу	Опис
hash	String	public	Зберігатимуть у собі дані хешу поточного блоку
previousHash	String	public	Зберігатимуть хеш попереднього блоку
data	String	private	Дані про транзакції
timeStamp	LocalDataTime	private	Час створення блоку
nonce	Integer	private	Ідентифікатор виконання PoW

В даному класі потрібно ще створити конструктор блоку (рисунок 3.2), функцію для розрахунку хешу блоку, також функцію майнінгу блоку.

```
public Block(String data, String previousHash) {
    this.data = data;
    this.previousHash = previousHash;
    this.timeStamp = new LocalDateTime.now();
    this.hash = calculateHash(); // Генерація хешу при створенні блоку
}
```

Рисунок 3.2 – Конструктор блоку

Функція calculateHash() це функція створення хешу для поточного блоку. Я використали шифрування SSH-256, що була реалізована наступним алгоритмом на рисунку 3.3

```
public static String applySha256(String input) {
    try {
        MessageDigest digest = MessageDigest.getInstance("SHA-256");
        byte[] hash = digest.digest(input.getBytes("UTF-8"));
        StringBuilder hexString = new StringBuilder();
        for (byte b : hash) {
            hexString.append(String.format("%02x", b));
        }
        return hexString.toString();
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
```

Рисунок 3.3 – Алгоритм шифрування

Функція майнинг потрібна для того щоб додавати блоки у блокчейн, також виконує функцію забезпечення безпеки та консенсусу в розподіленій мережі. У контексті блокчейну на Java, ми використовуємо алгоритм PoW (рисунок 3.4).

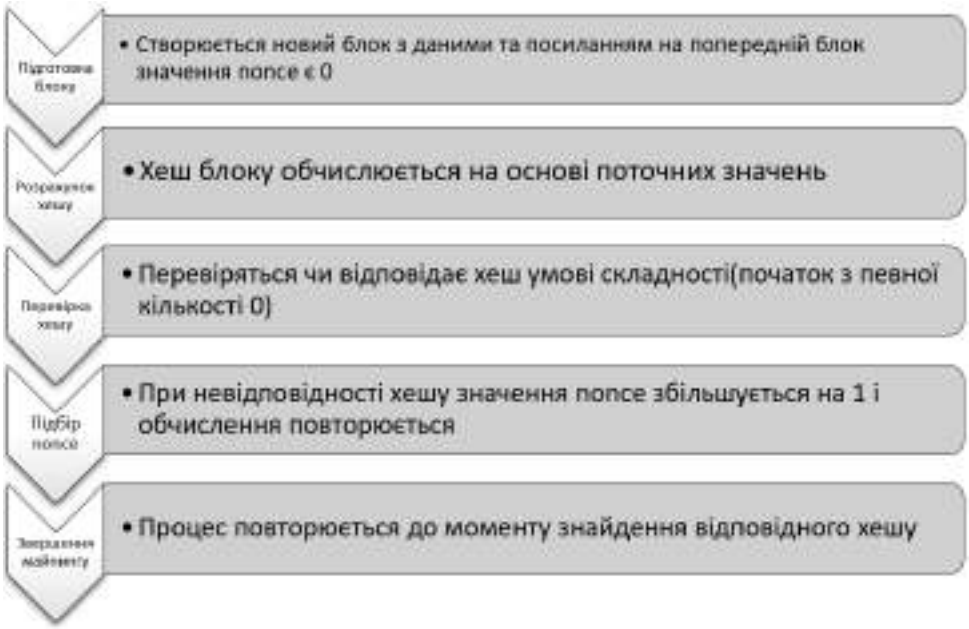


Рисунок 3.4 – Алгоритм виконання PoW

Для створення ланцюжка блоків потрібно створити ArrayList <Block>, що представляє собою список , в якому зберігаються блоки в порядку їх створення. Для зручного використання створимо новий клас під назвою Blockchain , де і будуть реалізовані необхідні функції.

Перший метод AddBlock додавання блоку до ланцюжка. Перед тим в методі викликати функцію майнингу та вбудованим методом blockchain.add(newBlock).

Наступним кроком буде перевірка істинності блоку, втілена наступною частинкою коду на рисунку 3.5.

Інструменти для аналізу інтерфейсів: IntelliJ IDEA має вбудовані інструменти для аналізу інтерфейсів, такі як валідація макетів і перевірка доступності. Механізм, за допомогою якого вузли узгоджуються щодо валідності нових транзакцій та блоків.

```
public static boolean isChainValid() {
        Block currentBlock;
        Block previousBlock;
        String hashTarget = new String(new char[difficulty]).replace('\0', '0');

        for (int i = 1; i < blockchain.size(); i++) {
            currentBlock = blockchain.get(i);
            previousBlock = blockchain.get(i - 1);

            // Порівняння поточного хешу і розрахованого
            if (!currentBlock.hash.equals(currentBlock.calculateHash())) {
                System.out.println("Current Hashes not equal");
                return false;
            }
            // Порівняння попереднього хешу і хешу попереднього блоку
            if (!previousBlock.hash.equals(currentBlock.previousHash)) {
                System.out.println("Previous Hashes not equal");
                return false;
            }
            // Перевірка, чи розв'язаний блок
            if (!currentBlock.hash.substring(0, difficulty).equals(hashTarget)) {
                System.out.println("This block hasn't been mined");
                return false;
            }
        }
        return true;
    }
```

Рисунок 3.5 – Валідація блоку

Для тестування у консолі було реалізовано ще метод виведення:

```
public void printBlock()
    {
        for (int i = 0; i < chain.size(); i++) {
            System.out.println("Block is: " + chain.get(i).getData());
            System.out.println("Hash is: " + chain.get(i).getHash());
            System.out.println("Privious hash is: " +
chain.get(i).getPreviousHash());
            System.out.println("Time is: " + chain.get(i).getTimeStamp());
        }
    }
```

Для доступу до змінних приватного типу потрібно реалізувати функції гетерів та сетерів за принципом на рисунку 3.6.

```
public String getData()
{
    return data;
}

public void setData(String data)
{
    this.data = data;
}
```

Рисунок 3.6 – Реалізація Get та Set функцій доступу

Також врахувавши що конструктор не передбачений для першого блоку, то потрібно його створити вручну, оскільки він не буде містити хеш попереднього блоку:

```
private Block genBlock()
  {
    Block genesis = new Block("First Block of my Blockchain", LocalDateTime.now());
    genesis.setPreviousHash(null);
    genesis.calculateHash();
    return genesis;
  }
```

Наступним кроком буде тустування системи у консолі. Вона буде реалізована у класі main. Для додавання блоків нам потрібно створити об’єкт класу Blockchain, щоб за допомогою нього використовувати методи та функції, котрі знаходяться у даному класі. Код для тестування:

```
Blockchain tcpc = new Blockchain();
    Block block = new Block("Data of block 2", LocalDateTime.now());
    tcpc.AddBlock(block);
    block = new Block( "Data of block 3", LocalDateTime.now());
    tcpc.AddBlock(block);
    tcpc.printBlock();
```

Результат роботи показано на рисунку 3.7.

```
Block is: First Block of my Blockchain
Hash is: 6e512a6a870716c31fbf6f70ff798a58b67c16d60e3b20efb56c56d74161c605
Privious hash is: null
Time is: 2024-05-30T08:42:28.027
Block is: Data of block 2
Hash is: 9ea46cea02f1aebeb1d27ecf83d012b23afb8839bedc1ad6d186739c73a059a9
Privious hash is: 6e512a6a870716c31fbf6f70ff798a58b67c16d60e3b20efb56c56d74161c605
Time is: 2024-05-30T08:42:28.061
```

Рисунок 3.7 – Тестування блокчейну

Провівши ще декілька тестувань з розробленим меню для додавання та видалення блоків в циклі також в консольному варіанті за допомогою циклу while з булевою змінною. Blockchain є важливою складовою будь-якої блокчейн-системи, яка дозволяє керувати створенням, додаванням та перевіркою блоків у ланцюжку. Майнінг є критичним компонентом блокчейн-систем, забезпечуючи безпеку та децентралізовану синхронізацію даних. У нашому прикладі Java ми реалізували базовий алгоритм Proof of Work, який включає створення та перевірку блоків, а також динамічний підбір значення nonce для досягнення заданої складності.

Тестове меню в консолі зображено на рисунку 3.8

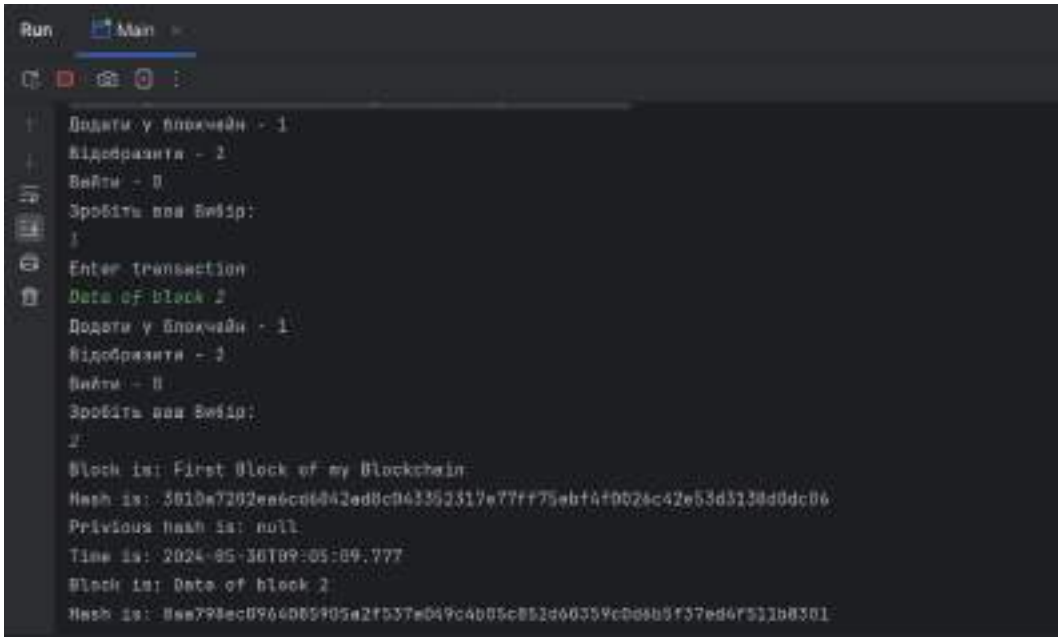


Рисунок 3.8 – Меню консольної системи

Тест було пройдено успішно, тому дану систему буде інтегровано у інтерфейсну прогрмаою за допомогою WebSocket для забезпечення двостороннього зв'язку між вашим блокчейном та інтерфейсною платформою. БЧ може висилати оновлення стану через WebSocket, а інтерфейсна платформа може відправляти команди та отримувати дані через цей канал.

3.2 Розробка смарт контрактів

Написання смарт-контрактів на власному блокчейні, створеному на Java, потребує розробки додаткової інфраструктури, що підтримує виконання і верифікацію цих контрактів. Смарт-контракти можна розглядати як програмний код, який виконується на блокчейні для автоматизації угод та умов.

Смарт-контракти мають бути окремими об'єктами або наборами інструкцій, які зберігаються та виконуються в блокчейні. Смарт-контракт як код**,** тобто контракти зберігаються в блокчейні та можуть виконуватися вузлами мережі. Механізм виконання**,** це б то платформа повинна мати можливість

виконувати код смарт-контрактів та взаємодіяти з блокчейном. Контракти можуть зберігати стан (дані) у блокчейні.

Для початку можна створити інтерфейс для смарт-контрактів, який забезпечить базові методи, що мають бути реалізовані всіма контрактами.

Приклад коду:

```
public interface SmartContract {
    void deploy(); // Метод для розгортання контракту
    void execute(String function, String... params); // Метод для виконання функції контракту
    String query(String function, String... params); // Метод для запиту даних з контракту
}
```

Для реалізації базового смарт-контракту було використано створений базовий клас. Наступним кроком написання, зображено на рисунку 3.9.

Блокчейн-система має підтримувати розгортання та виконання смарт-контрактів. Це можна зробити шляхом додавання нових транзакцій у блокчейн, які зберігають та виконують смарт-контракти. Блокчейн вже було створено тому просто прописуємо методи:

```
// Метод для розгортання контракту
    public void deployContract(SmartContract contract) {
        contracts.put(contract.getClass().getName(), contract);
        contract.deploy();
    }
    // Метод для виконання функцій контракту
    public void executeContract(String contractName, String function, String... params) {
        SmartContract contract = contracts.get(contractName);
```

```
if (contract != null) {
    contract.execute(function, params);
} else {
    System.out.println("Contract not found");
}
```

```
import java.util.HashMap;
import java.util.Map;

public class Blockchain {
    private Map<String, SmartContract> contracts = new HashMap<>();

    // Метод для розгортання контракту
    public void deployContract(SmartContract contract) {
        contracts.put(contract.getClass().getName(), contract);
        contract.deploy();
    }

    // Метод для виконання функцій контракту
    public void executeContract(String contractName, String function, String params) {
        SmartContract contract = contracts.get(contractName);
        if (contract != null) {
            contract.execute(function, params);
        } else {
            System.out.println("Contract not found");
        }
    }

    // Метод для запитів до контракту
    public String queryContract(String contractName, String function, String params) {
        SmartContract contract = contracts.get(contractName);
        if (contract != null) {
            return contract.query(function, params);
        } else {
            return "Contract not found";
        }
    }
    public static void main(String[] args) {
        Blockchain blockchain = new Blockchain();

        // Створення та розгортання контракту
        AccessControlContract accContract = new AccessControlContract("accContract1");
        blockchain.deployContract(accContract);

        // Виконання функцій контракту
        blockchain.executeContract(AccessControlContract.class.getName(), "registerKey", "key1", "owner1");
        blockchain.executeContract(AccessControlContract.class.getName(), "revokeKey", "key1");

        // Запити до контракту
        String access = blockchain.queryContract(AccessControlContract.class.getName(), "checkAccess", "key1");
        System.out.println("Access check: " + access);
    }
}
```

Рисунок 3.9 - Самрт-контракт

Виконання смарт-контракту на блокчейні відбувається у кілька етапів, які можуть трохи відрізнятися залежно від конкретної блокчейн-платформи та мови програмування, в якій він написаний. Однак загальна суть процесу залишається подібною, як на рисунку 3.10.

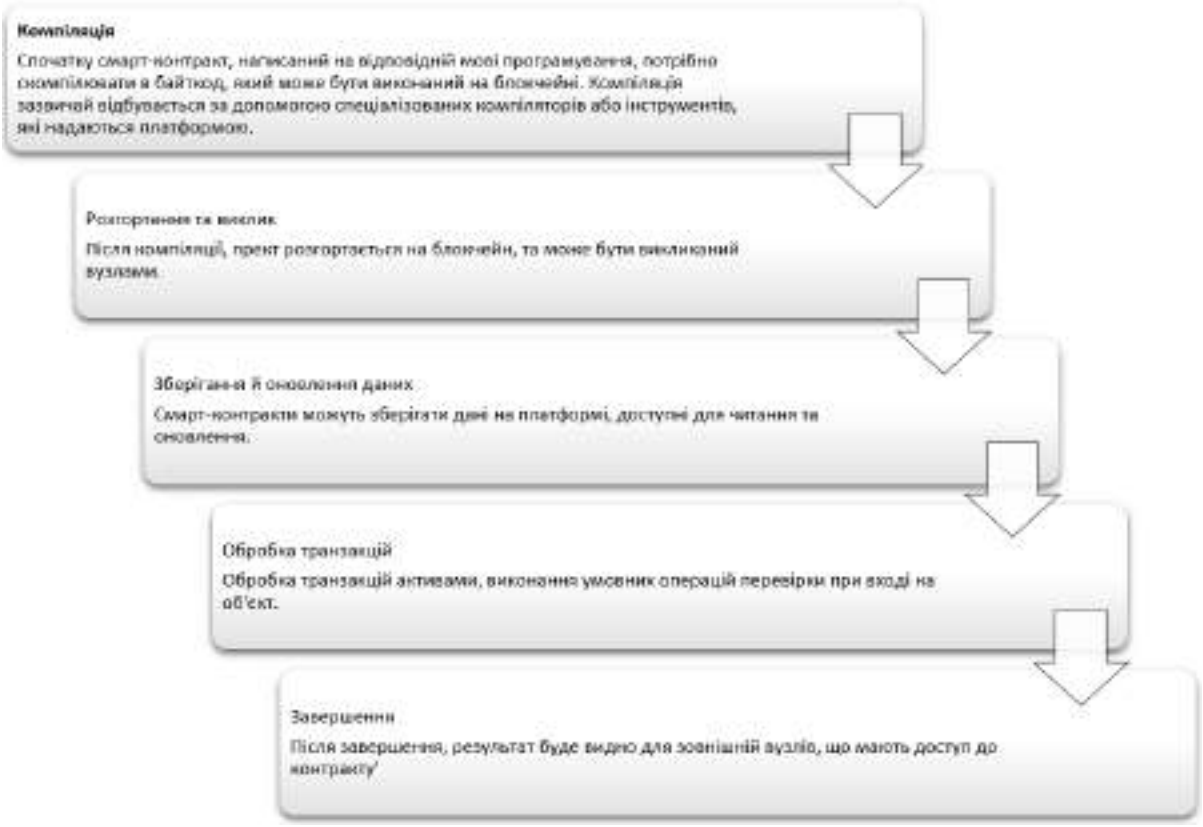


Рисунок 3.10 – Схема роботи смарт-контракту

Ці етапи стануть основою для виконання будь-якого смарт-контракту на блокчейні.

Наступним кроком було написання конкретних смарт-контрактів для перевірки доступу за унікальним ключем:

```
private void registerKey(String keyId, String owner) {
    setState(keyId, owner + ":true");
    System.out.println("Key registered: " + keyId);
}
```

```
    private void revokeKey(String keyId) {
        String currentState = getState(keyId);
        if (currentState != null) {
            setState(keyId, currentState.split(":")[0] + ":false");
            System.out.println("Key revoked: " + keyId);
        }
    }
    private boolean checkAccess(String keyId) {
        String currentState = getState(keyId);
        return currentState != null && currentState.endsWith(":true");
    }
}
```

Таким чином було впроваджено основні функції роботи додатку та створенні платформи для тестування.

3.3 Розробка інтерфейсу

Першим кроком є визначення функціональності, яку повинен мати інтерфейс адміністратора. Основні функції можуть включати:

- керування користувачами (додавання, видалення, оновлення інформації);
- видача та скасування доступу;
- моніторинг транзакцій та подій;
- налаштування системи;
- перегляд журналу подій.

Для початку була запрограмована платформа для входу у систему і створений один ключ на програмному рівні, щоб завжди мати доступ.

Приклад програмного коду для реалізації головного класу:

```
public class Main extends Application {
    public void start(Stage primaryStage) throws Exception {
        FXMLLoader loader = new FXMLLoader(getClass().getResource("/admin_dashboard.fxml"));
        primaryStage.setTitle("Admin Dashboard");
        primaryStage.setScene(new Scene(loader.load()));
        primaryStage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```

Для логіну також створено контролер обробки подій, в якому реалізовано виключення при несанкціонованому доступі. Наступним кроком є налагодження взаємодії з блокчейном, створивши клас для виконання транзакцій. В контролері задано виклики методів з Blockchain для обробки подій, таким чином була звершена інтеграція. Приклад на рисунку 3.11

```
import javafx.fxml.FXML;
import javafx.scene.control.TableView;
import javafx.collections.ObservableList;
import javafx.collections.FXCollections;
public class Menu {

    public class AdminDashboardController {
    private TableView<User> userTable;
    private BlockchainService blockchainService = new BlockchainService();
}
```

Рисунок 3.11 – Приклад коду інтеграції

Наступними кроками було зв’язування інших платформ, програмування належного функціонування додатку. Також для тестування системи пропуску

було створено тестовий зчитувач карти доступу. Також при видалені доступу просто потрібно оновити дані користувача при яких старий ключ анулюється (рисунок 3.12).

```
private void handleUpdateUser() {
    // Логіка для оновлення інформації про користувача
    User selectedUser = userTable.getSelectionModel().getSelectedItem();
    if (selectedUser != null) {
        selectedUser.setUsername("newUsername");
        blockchainService.updateUser(selectedUser);
        refreshUserTable();
    }
```

Рисунок 3.12 – Приклад коду оновлення даних

Реалізація платформ була здійснена з вище описаними принципами з'єднання систем між собою. Як висновок можна сказати, що впровадження блокчейн-технологій у СКД та створення адміністративного інтерфейсу на Java дозволить підвищити безпеку, прозорість та надійність системи. Чітке планування та ретельне тестування допоможуть успішно реалізувати проект і забезпечити його ефективну роботу.

3.4 Тестування та оцінка ефнективності

Тестування охопило всі функції СКД, включаючи додавання, видалення та оновлення користувачів, керування правами доступу, моніторинг транзакцій, роботу з журналами подій та іншими адміністративними завданнями.

Особливу увагу було приділено тестуванню безпеки системи. Це включає перевірку механізмів аутентифікації та авторизації, захист від несанкціонованого доступу, перевірку цілісності даних та виявлення вразливостей.

Важливим етапом перевірки стало те, як система взаємодіє з блокчейном, забезпечуючи правильність виконання транзакцій, обробку смарт-контрактів, синхронізацію даних між різними вузлами мережі та збереження даних у розподіленій мережі.

Проведено тестування в використані адміністративного інтерфейсу, щоб забезпечити його зручність та інтуїтивність для кінцевих користувачів.

Перевірка сумісності системи з різними конфігураціями апаратного та програмного забезпечення, щоб забезпечити її правильну роботу у різних середовищах. Перевірте взаємодію між різними компонентами системи, такими як блокчейн-мережа, бази даних та інтерфейс користувача, щоб забезпечити їх коректну роботу.

Було використано реальні або наближені до реальних дані для тестування, щоб переконатися, що система працює належним чином у реальних умовах.

Тестування функцій СКД на блокчейні є критичним для забезпечення її успішного впровадження та експлуатації. Повне, ретельне та систематичне тестування допоможе виявити та виправити потенційні проблеми, забезпечити високу надійність, безпеку та ефективність системи, а також задоволення потреб кінцевих користувачів.

Інтерфейс програми було розроблено таким чином:

- Вхід користувача (рисунок 3.13);
- голове меню;
- платформа адміністратора (рисунок 3.14);
- дочірні платформи керування користувачами (рисунок 3.15 – 3.16).

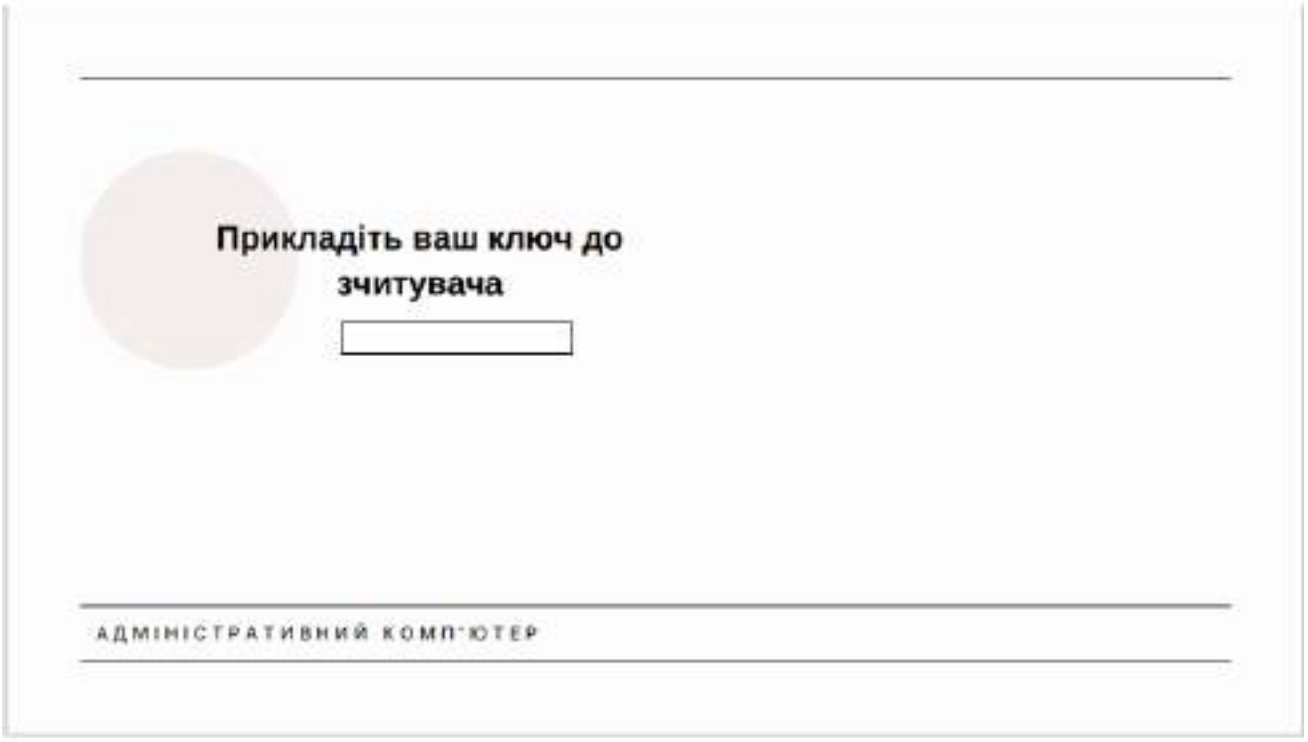


Рисунок 3.13 – Переше вікно

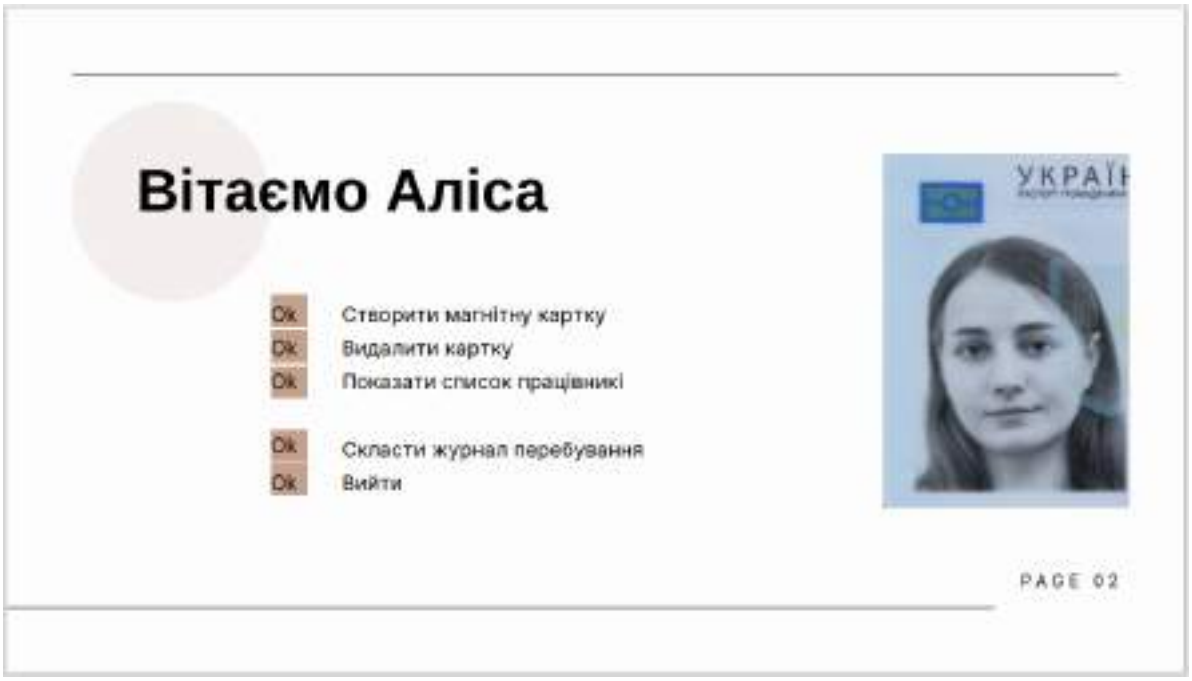


Рисунок 3.14 – Вхід адміністратора

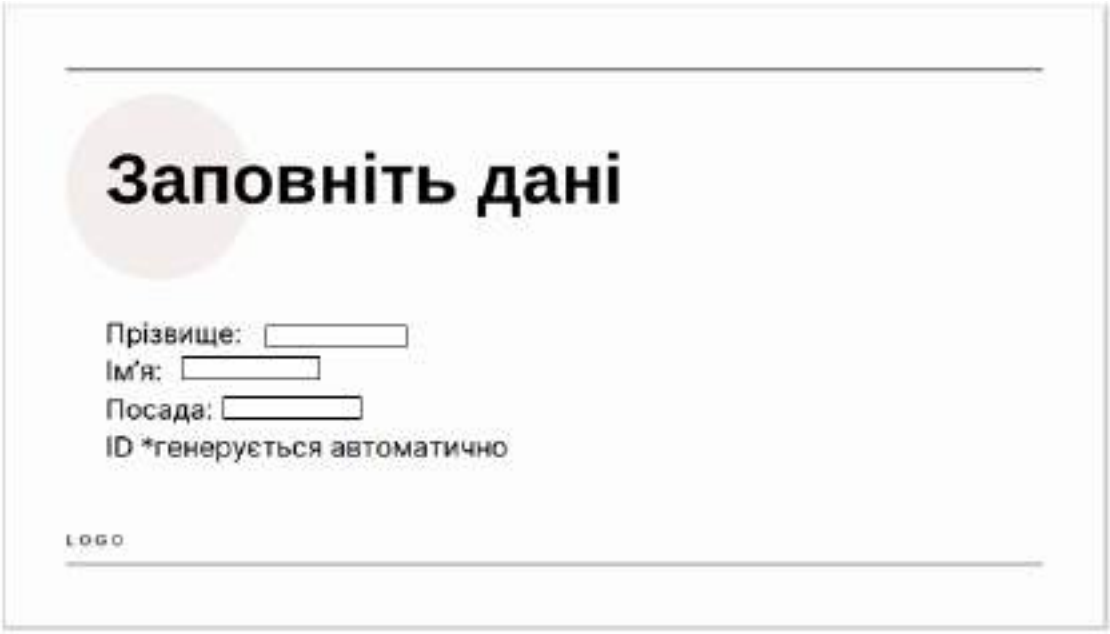


Рисунок 3.15 – Додавання працівника

Програмування мовою Java в IntelliJ IDEA є зручним і ефективним завдяки широкому спектру можливостей, які це середовище розробки пропонує. Також на програмному рівні створено супер користувача для керування адміністративними платформами.

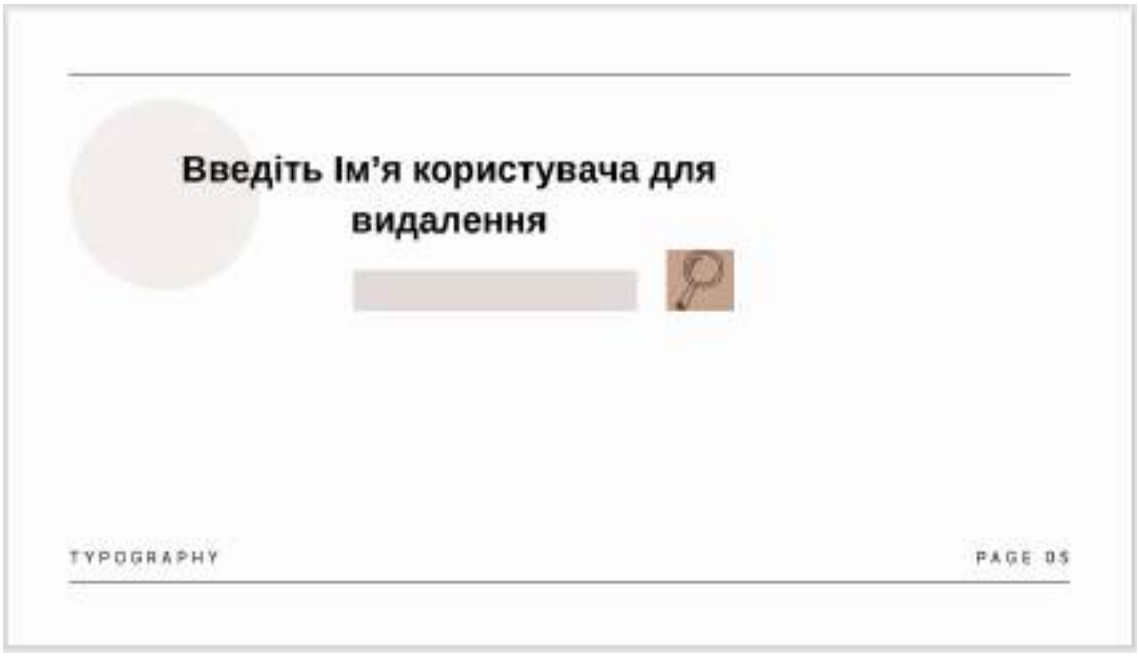


Рисунок 3.16 – Видалення даних за пошуком

Основні результати тестування:

– високий рівень безпеки забезпечується завдяки використанню криптографічних методів;

– смарт-контракти успішно запобігли несанкціонованому доступу та маніпуляціям з даними;

– всі дії користувачів були прозорими та відстежуваними блокчейн забезпечив незмінність журналу доступу, що полегшило аудит;

– система показала високу стійкість до відмов завдяки розподіленій природі блокчейну;

– відсутність єдиного пункту відмови значно підвищила надійність;

– автоматизація процесів через смарт-контракти значно зменшила час на аутентифікацію та авторизацію користувачів;

– система успішно обробляла великий обсяг запитів без значного зниження продуктивності;

Переваги впровадження:

– незмінність даних в блокчейні запобігає будь-яким маніпуляціям з історією доступу;

– криптографічні методи забезпечують захист від несанкціонованого доступу;

– автоматизація за допомогою смарт-контрактів зменшує потребу в ручному управлінні доступом.

– менші витрати на адміністрування та обслуговування системи.

– високий рівень прозорості операцій підвищує довіру користувачів до системи.

– можливість незалежного аудиту сприяє дотриманню нормативних вимог;

– децентралізована природа блокчейну підвищує стійкість системи до кібератак та збоїв.

Деякі публічні блокчейни можуть стикатися з проблемами масштабованості при великій кількості транзакцій. Необхідність балансування між децентралізацією та продуктивністю. Можливість зниження витрат при використанні приватних або гібридних блокчейнів. Впровадження та обслуговування блокчейн-системи вимагає спеціалізованих технічних знань.

Необхідність навчання персоналу для роботи з новими технологіями. Потреба в інтеграції з існуючими IT-системами може створювати додаткові виклики. Важливість забезпечення сумісності та безперебійної роботи. Графіки результатів тестування. Демонстровано ефективність, безпеку та продуктивність системи в різних сценаріях. Порівняння з традиційними системами контролю доступу.

ВИСНОВКИ

У кваліфікаційній роботі бакалавра здійснено дослідження та аналіз можливостей блокчейн-технологій. Їх використання у сфері безпеки даних, контролю доступу, було реалізовано в тестовій формі, оскільки не було потрібного обладнання. Пройшовши складні етапи розробки, я здобула дуже багато корисних навичок та знань в зовсім новій для мене сфері. В результаті я покращила свої навики програмування на Java, та спробувала уже існуючі тестові платформи блокчейну.

Використання даної технології приносить багато позитивних наслідків таких як:

- підвищення безпеки;
- надійність та цілісність даних;
- спрощення процесів;
- підвищення ефективності;
- зменшення витрат.

Технологія блокчейну може забезпечити високий рівень безпеки, оскільки інформація, збережена в блоках, захищена за допомогою криптографії та розподіленої мережі. Це може допомогти уникнути зламів та несанкціонованого доступу до системи контролю доступу. Використання блокчейну може спростити процеси аутентифікації та авторизації, оскільки дозволяє автоматизувати багато операцій та використовувати розумні контракти для автоматичного виконання правил доступу.

Блокчейн забезпечує надійність та цілісність даних, оскільки будь-які зміни в інформації записуються в блоки та розподіляються по всій мережі. Це забезпечує можливість перевірки автентичності та непідробленості даних. Ефективність полягає у тому, що оскільки дозволяє швидко та безпечно виконувати транзакції та обмін даними між різними сторонами.

Хоча впровадження блокчейну може потребувати певних витрат на початкову розробку та інтеграцію, в довгостроковій перспективі воно може зменшити витрати на управління та підтримку систем контролю доступу.

Загалом, впровадження технології блокчейну в системи контролю доступу може принести значні переваги в плані безпеки, ефективності та надійності цих систем, але вимагає обдуманого підходу та врахування специфічних потреб та характеристик конкретного проекту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Юдін О.К. Аналіз та кластфікація систем контролю та управління доступом на підприємстві. О. К. Юдін, О. М. Весельська // Наукоємні технології. – 2018. 220-221 с.

2. Що таке блокчейн? URL: https://blog.whitebit.com/uk/what-is-blockchain-technology (дата звернення: 13.03.2024).

3. BUILD YOUR OWN BLOCKCHAIN. URL: http://ecomunsing.com/build-your-own-blockchain (дата звернення 13.03.2024).

4. Кудь О.В., Кучерявенко М.С., Смичок Є.А. Цифрові активи та їх економіко-правове регулювання у світлі розвитку технології блокчейн монографія / Олександр Кудь, Микола Кучерявенко, Євген Смичок. – Харків : Право, 2019. 14-17 с.

5. Binance Academy. Історія Blockchain. URL: https://academy.binance.com/uk/start-here (дата звернення: 09.03.2024).

6. Blockchain World Wire. URL: https://www.ibm.com/blockchain/solutions/world-wire (дата звернення: 21.03.2024).

7. Blockchain for government and public services. URL: https://www.eublockchainforum.eu/reports (дата звернення: 20.03.2024).

8. Implementing a Simple Blockchain in Java. URL: https://www.baeldung.com/java-blockchain (дата звернення: 26.04.2024).

9. Top 3 blockchain libraries for java developers/ URL: https://javarush.com/en/groups/posts/en.320.top-3-blockchain-libraries-for-java-developers (дата звернення: 23.04.2024).

10. Blockchain in Java. URL: https://www.javatpoint.com/blockchain-java (дата звернення: 26.04.2024).

11. Васильков Ю.В. Інформаційні системи та їх безпека/ А.В. Васильков, М.: Форум, 2015. 565 с.

12. Ворона В.А. Системи контролю та управління доступом м / В. А. Ворона, В. А. Тихонов. — М.: Форум, 2012. 14-15 с.

13. Інформаційна безпека: види загроз та методи їх усунення. URL: https://datami.ua/informatsijna-bezpeka-vidi-zagroz-i-metodi-yih-usunennya/ (дата звернення: 13.04.2024).

14. Java Persistence API. URL: https://ela.kpi.ua/server/api/core/bitstreams/e0a01813-fe13-43cb-a494-476d104f921b/content (дата звернення: 10.04.2024).

15. Meylan P. A. Blockchains Will Change the Way the World Votes. URL: https://www.csis.org/analysis/blockchains-will-change-way-world-votes. (дата звернення: 12.04.2024).

16. Равал С. Децентрализованные приложения. Технология Blockchain в действии. СПб.: Питер, 2017. 240 с.

17. Чернишов Д. У майбутньому Україна переведе всю цифрову державну інформацію на блокчейн-платформу. – URL: http://www.pravove-pole.info/novini/u-majbutnomu-ukraina-perevede-vsjucyfrovu-derzhavnu-informaciju-na-blokchejnplatformu-denys-chernyshov/ (дата звернення: 10.02.2024).

18. Васильєв О. В., Німкович А. І. Впровадження фінтех і блокчейну як інфраструктури ринку цінних паперів / О. В. Васильєв, А. І. Німкович // Управління розвитком. – 2018. – № 1. – С. 30–35.

19. Карчева Г. Т., Карчева І. Я. Інноваційні блокчейн технології як фактор підвищення ефективності фінансової сфери та економіки / Г. Т. Карчева, І. Я. Карчева // Наук. праці Науково-дослідного фінансового інституту. – 2017. – Вип. 4. – С. 39–42.

20. Турнікет роторний STAR-TS URL: https://smartel.ua/ua/product/turniket-rotornyy-star-ts/ (дата звернення: 10.05.2024).

21. Шлагбаум механічний вертикальний МАРО ШМВ-4 URL: https://rozetka.com.ua/ua/290640173/p290640173/ (дата звернення: 10.05.2024).

22. Безсонова А. О. Застосування систем контролю доступу для різних типів підприємств / А. О. Безсонова, О. Д. Василенко. // Всеукраїнська науковопрактична конференція студентів, аспірантів та молодих вчених. – 2023. – С. 117–120.

23. Васильков В. Г. Організація виробництва / В. Г. Васильков. – Київ: Київський національний економічний ун-т, 2003. – 522 с

24. Що таке СКУД?. Ohrana.ua. URL: https://ohrana.ua/uk/stati-i-obzory/chto-takoe-skud.html (дата звернення: 05.05.2024).

25. Introduction to Java. URL: https://www.w3schools.com/java/java_intro.asp (дата звернення 14.04);

26. Mastering Blockchain" by Imran Bashirю. URL: https://drukarnia.com.ua/articles/chitayemo-mastering-blockchain-DPMXO (дата звернення: 13.04.2024).

27. "Java Blockchain Programming" by Narayan Prusty. URL: https://www.infoq.com/articles/review-building-blockchain-projects/ (дата звернення: 15.04.2024).

28. IntelliJ IDEA Community URL: https://www.jetbrains.com/idea/download/?section=windows (дата звернення: 15.03.2024).

29. Ethereum Whitepaper. URL: https://ethereum.org/en/whitepaper/ (дата звернення: 20.03.2024).

30. Merkle Tree in Blockchain: What is it, How does it work and Benefits. URL: https://www.simplilearn.com/tutorials/blockchain-tutorial/merkle-tree-in (дата звернення: 20.03.2024).

ДОДАТКИ А

(Обов’язковий)

Код програмного забезпечення

DataBlock

```
import java.util.Date;
public class Block {
public String hash;
public String previousHash;
private String data; // Дані нашого блоку, наприклад, транзакції
private long timeStamp; // Час створення блоку
private int nonce;
// Конструктор блоку
public Block(String data, String previousHash) {
this.data = data;
this.previousHash = previousHash;
this.timeStamp = new Date().getTime();
this.hash = calculateHash(); // Генерація хешу при створенні
блоку
}
// Розрахунок хешу блоку
public String calculateHash() {
String input = previousHash + Long.toString(timeStamp) +
Integer.toString(nonce) + data;
return StringUtil.applySha256(input);
}
public void mineBlock(int difficulty) {
String target = new String(new
char[difficulty]).replace('\0', '0');
while (!hash.substring(0, difficulty).equals(target)) {
```

```
nonce++;
hash = calculateHash();
}
System.out.println("Block Mined!!! : " + hash);
}
```

// Застосування SHA-256 до вхідного рядка і повернення результату

```
public static String applySha256(String input) {
try {
MessageDigest digest = MessageDigest.getInstance("SHA-256");
byte[] hash = digest.digest(input.getBytes("UTF-8"));
StringBuilder hexString = new StringBuilder(); // Створення
хешу в шістнадцятковому форматі
for (byte elem : hash) {
String hex = Integer.toHexString(0xff & elem);
if (hex.length() == 1) hexString.append('0');
hexString.append(hex);
}
return hexString.toString();
}
catch (Exception e) {
throw new RuntimeException(e);
import java.util.ArrayList;
public class Blockchain {
public static ArrayList<Block> blockchain = new ArrayList<>();
public static int difficulty = 5;

public static void main(String[] args) {
// Додавання блоків до ланцюга
blockchain.add(new Block("First block", "0"));
```

```
System.out.println("Trying to Mine block 1... ");
blockchain.get(0).mineBlock(difficulty)

blockchain.add(new Block("Second block",
blockchain.get(blockchain.size() - 1).hash));
System.out.println("Trying to Mine block 2... ");
blockchain.get(1).mineBlock(difficulty);

blockchain.add(new Block("Third block",
blockchain.get(blockchain.size() - 1).hash));
System.out.println("Trying to Mine block 3... ");
blockchain.get(2).mineBlock(difficulty);
System.out.println("\nBlockchain is Valid: " +
isChainValid());
}
// Перевірка валідності ланцюга блоків
public static boolean isChainValid() {
Block currentBlock;
Block previousBlock;
String hashTarget = new String(new
char[difficulty]).replace('\0', '0');
for (int i = 1; i < blockchain.size(); i++) {
currentBlock = blockchain.get(i);
previousBlock = blockchain.get(i - 1);
// Порівняння поточного хешу і розрахованого
if
(!currentBlock.hash.equals(currentBlock.calculateHash())) {
System.out.println("Current Hashes not equal");
return false;
}.
```

```
// Порівняння попереднього хешу і хешу попереднього блоку
if (!previousBlock.hash.equals(currentBlock.previousHash)) {
System.out.println("Previous Hashes not equal");
return false;
}
// Перевірка, чи розв'язаний блок
if(!currentBlock.hash.substring(0, difficulty).equals(hashTarget))
{
System.out.println("This block hasn't been mined");
return false;
}
return true;
}
import javafx.fxml.FXML;
import javafx.scene.control.TableView;
import javafx.collections.ObservableList;
import javafx.collections.FXCollections;
public class Menu {
public class AdminDashboardController {
private TableView<User> userTable;
private BlockchainService blockchainService = new
BlockchainService();
}

private void handleUpdateUser() {
// Логіка для оновлення інформації про користувача
User selectedUser =
userTable.getSelectionModel().getSelectedItem();
if (selectedUser != null) {
selectedUser.setUsername("newUsername");
```

```
blockchainService.updateUser(selectedUser);
refreshUserTable();
}

import java.util.HashMap;
import java.util.Map;
public class Blockchain {
private Map<String, SmartContract> contracts = new
HashMap<>();
// Метод для розгортання контракту
public void deployContract(SmartContract contract) {
contracts.put(contract.getClass().getName(),
contract);
contract.deploy();
}
// Метод для виконання функцій контракту
public void executeContract(String contractName, String
function, String params) {
SmartContract contract = contracts.get(contractName);
if (contract != null) {
contract.execute(function, params);
} else {
System.out.println("Contract not found");
}
// Метод для запитів до контракту
public String queryContract(String contractName, String
function, String params) {
SmartContract contract = contracts.get(contractName);
if (contract != null) {
return contract.query(function, params);
```

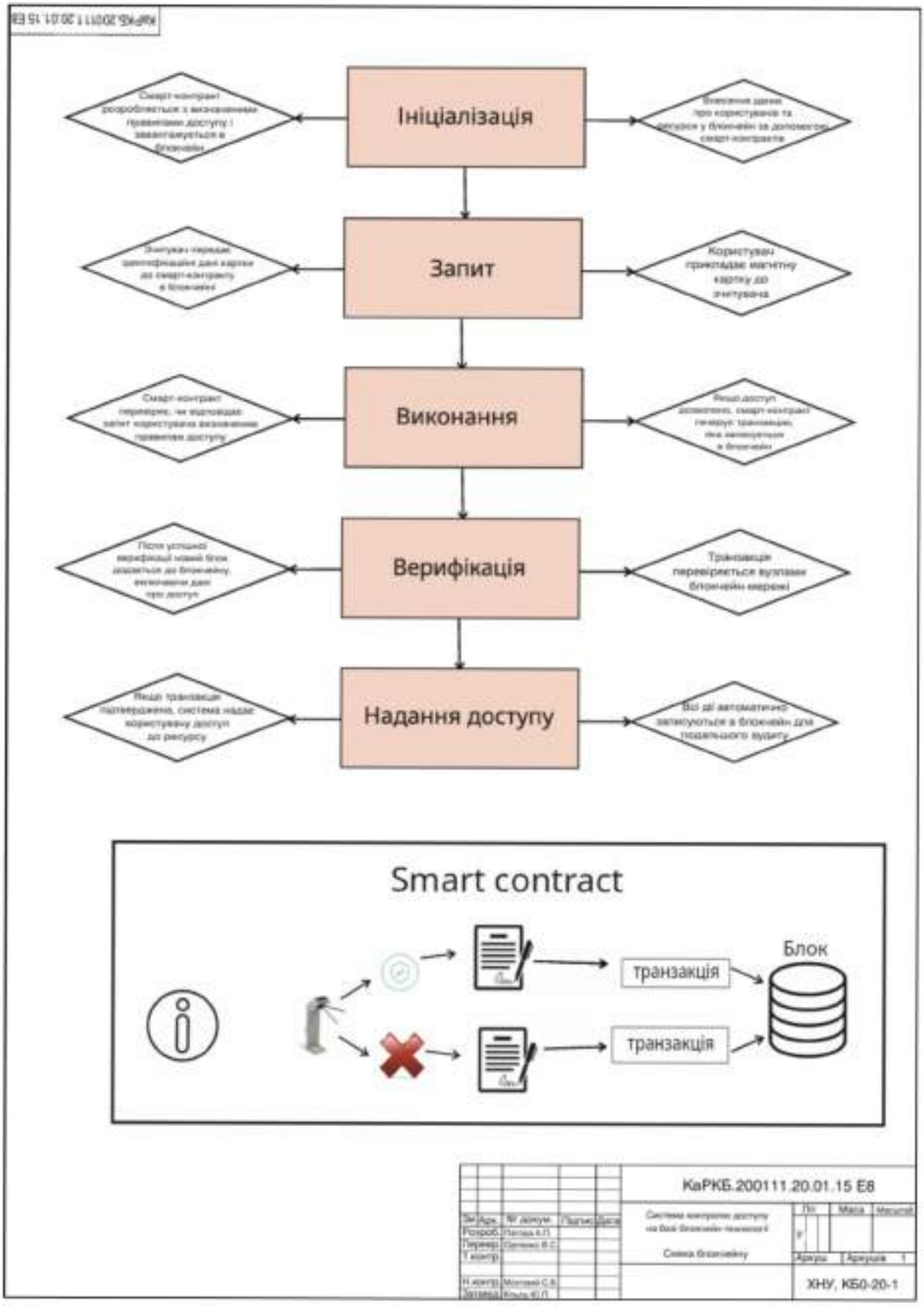
```
} else {
return "Contract not found";
}
}
public static void main(String[] args) {
Blockchain blockchain = new Blockchain();
// Створення та розгортання контракту
AccessControlContract accContract = new
AccessControlContract("accContract1");
blockchain.deployContract(accContract);
// Виконання функцій контракту
blockchain.executeContract(AccessControlContract.class.getName(),
"registerKey", "key1", "owner1");
blockchain.executeContract(AccessControlContract.class.getName(),
"revokeKey", "key1");
public PoW(String challenge, int difficulty) {
this.challenge
this.difficulty
challenge;
difficulty;
public String mine() {
int nonce = 0;
String hash;
String target = new String(new
char[difficulty]).replace('\0', '0');
do {
nonce++;
hash = calculateHash(challenge + nonce);}
while (!hash.substring(difficulty).equals(target));
System.out.println("Pow Solved: Nonce= nonce, Hash = +
```

```
hash);
return Integer.toString(nonce);}
public static String applySha256(String input) {
try {
MessageDigest digest = MessageDigest.getInstance("SHA-256");
byte[] hash = digest.digest(input.getBytes("UTF-8"));
StringBuilder hexString = new StringBuilder();
for (byte b : hash) {
hexString.append(String.format("%02x", b));
}
return hexString.toString();
} catch (Exception e) {
throw new RuntimeException(e);
}
// Порівняння поточного хешу і розрахованого
If (!currentBlock.hash.equals(currentBlock.calculateHash())) {
System.out.println("Current Hashes not equal")
```

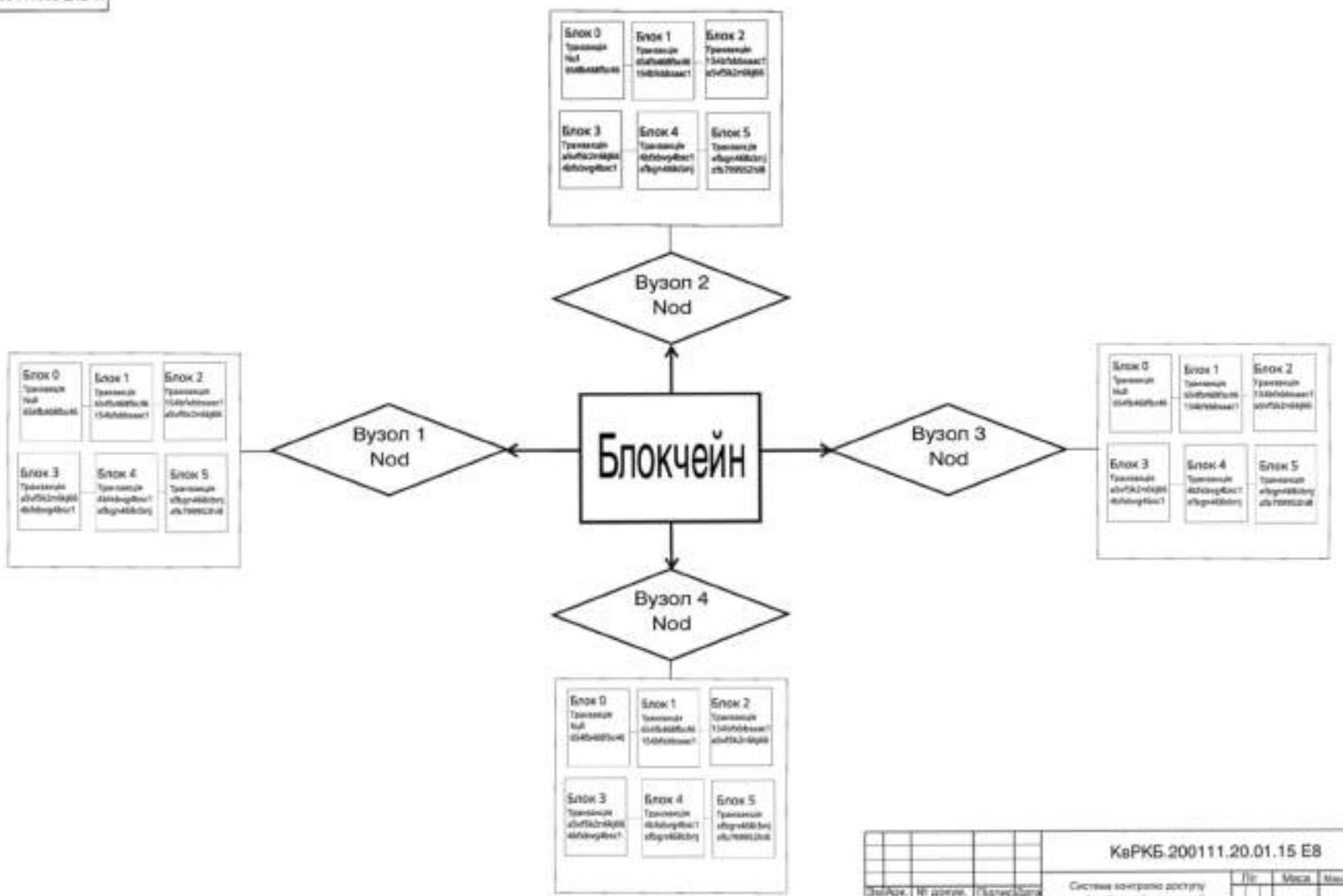
ДОДАТОК Б

(обов’язковий)

Копії графічної частини



КвРКБ.200111.20.01.15 Е8

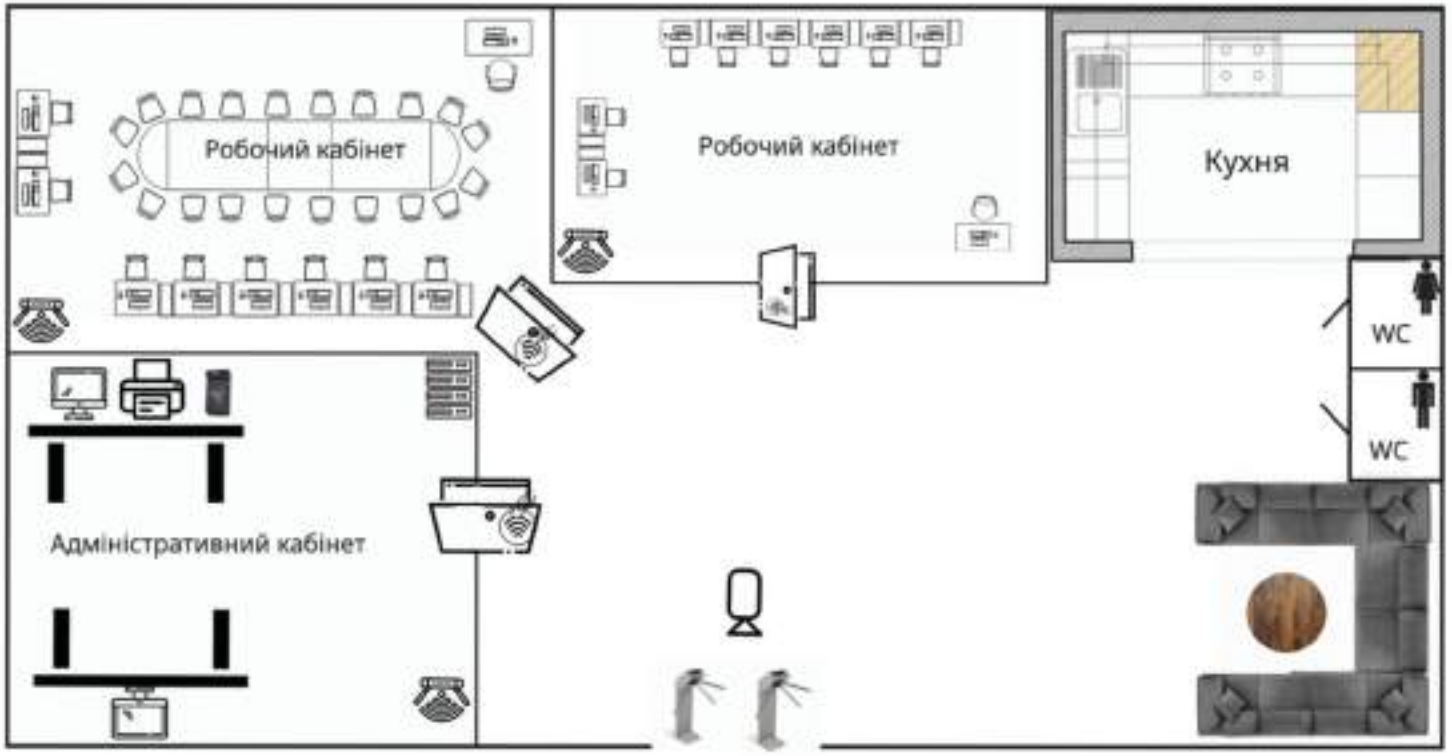


				КвРКБ.200111.20.01.15 Е8
Зм.	Арк.	№ докум.	Підпис	Дата
Розроб.		Пятова А.П.		Система контролю доступу на базі блокчейн-технології
Перевір.		Орлянко В.С.		
Т.контр.				Схема блокчейну
Н.контр.		Молчанов С.В.		
Затверд.		Клюєв Ю.П.		

Літ.	Маса	Масштаб
У		
Аркуш	Аркушів 1	

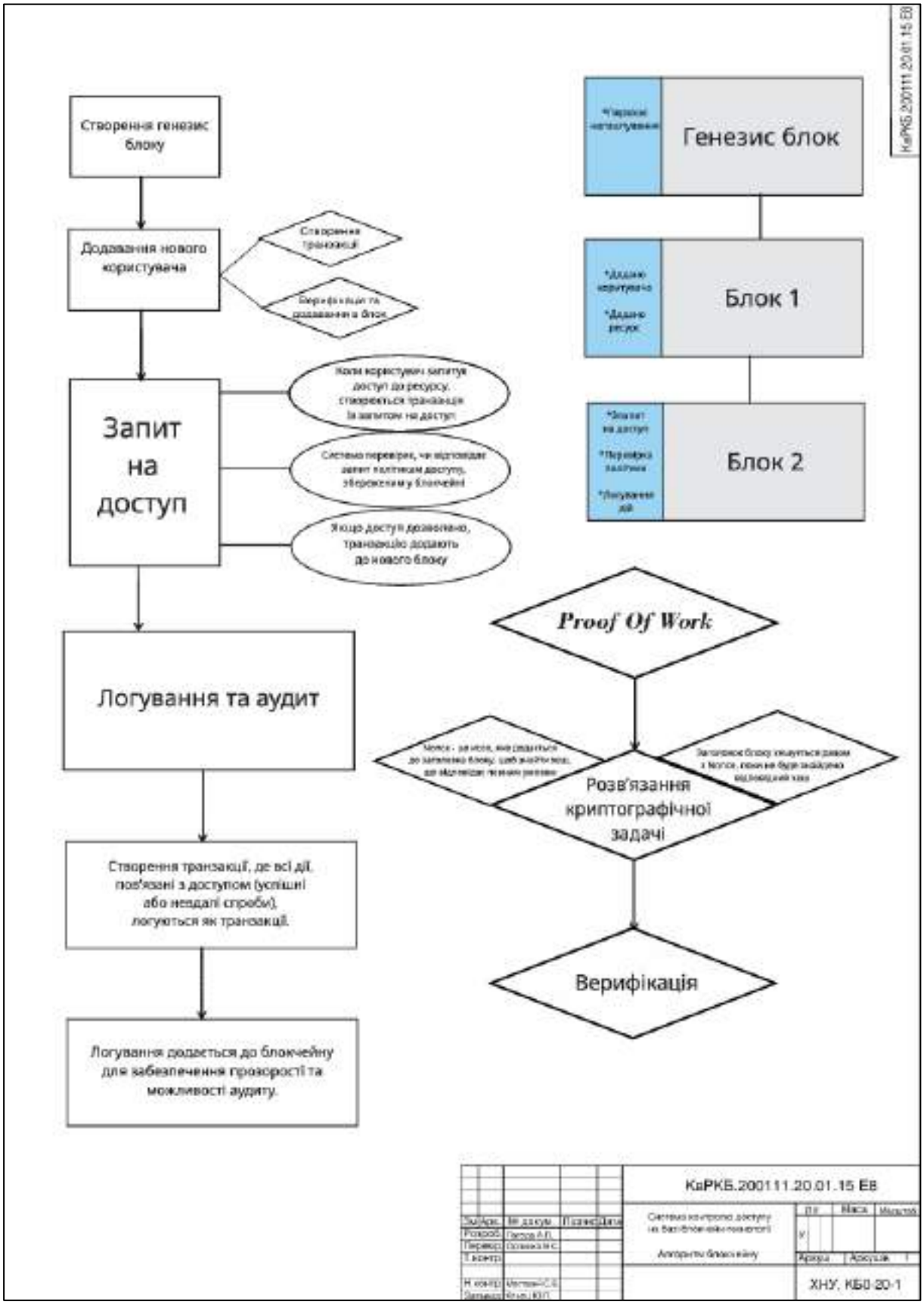
ХНУ, КБО-20-1

КаРКБ.200111.20.01.15 Е8



- персональний комп'ютер
- мережевий роутер
- турнікет
- двері з замком та зчитувачем
- сервер для збереження аудиту подій

					КаРКБ.200111.20.01.15 Е8		
Зм.	Арк.	№ докум.	Підпис	Дата	Система контролю доступу на базі біометричних технологій	Літ.	Маса	Масштаб
Розроб.		Пилипів А.П.				У		
Перевір.		Одинець В.О.			Архітектура СКД на підприємстві			
Т. контр.						Аркуш	Аркушів 1	
Н. контр.		Молодик С.В.				ХНУ, КБО-20-1		
Затверд.		Ільюк Ю.П.						


КвРКБ.200111.20.01.15 ЕВ
Створення генезис блоку
Додавання нового користувача
Створення транзакції
Верифікація та додавання в блок
Запит на доступ
Коли користувач запитує доступ до ресурсу, створюється транзакція із запитом на доступ
Система перевіряє, чи відповідає запит політикам доступу, збереженим у блокчейні
Якщо доступ дозволено, транзакцію додають до нового блоку
Логування та аудит
Створення транзакції, де всі дії, пов'язані з доступом (успішні або невдалі спроби), логуються як транзакції.
Логування додається до блокчейну для забезпечення прозорості та можливості аудиту.
Генезис блок
Блок 1
Блок 2
Proof Of Work
Розв'язання криптографічної задачі
Верифікація
КвРКБ.200111.20.01.15 ЕВ
Система контролю доступу на базі блокчейн-технології
Алгоритм блокчейну
ХНУ, КБ0-20-1

Завідувачу кафедри кібербезпеки
к.т.н., доц. Кльоцу Ю.П.
Лагоди Аліси Павлівни
ПІБ здобувача вищої освіти

Студентки ФІТ, 4 курсу, групи КБ-20-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 31.08.2023, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлена. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщена та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

6.06.2024
дата

підпис

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 16%**

	Документ		Сумарний збіг по Базі Даних	
ID: 132632 Назва: Система контролю доступу на базі блокчейн-технології Додано в БД: 2024-06-26 Автора: Лагода А.П. Керівники: Орленко В.С. Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	51951	511	911 (2%)	13 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

UNICHECK
by Turnitin

Ім'я користувача:
Кафедра кібербезпеки

ID перевірки:
1016388853

Дата перевірки:
26.06.2024 10:51:27 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
26.06.2024 10:59:01 EEST

ID користувача:
100008300

Назва документа: **Лагода_антиплагіат (1)**

Кількість сторінок: **57** **Кількість слів:** **7607** **Кількість символів:** **58266** **Розмір файлу:** **3.79 MB** ID **файлу:** **1016201106**

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

6.76%
Схожість

Найбільша схожість: **1.35%** **з Інтернет-джерелом** (https://essuir.sumdu.edu.ua/handle/123456789/85921)

6.06% Джерела з Інтернету	397	Сторінка 59
1.45% Джерела з Бібліотеки	47	Сторінка 60

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	4
Підозріле форматування	14 сторінок

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система контролю доступу на базі блокчейн-технології

Автор: Лагода Аліса Павлівна

Спеціальність: 125 – Кібербезпека

Освітня програма: освітньо-професійна

Науковий керівник: Орленко Вікторія Сергіївна, канд. техн. наук, доцентка

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Оригінальність тексту роботи за результатами перевірки системою Unichek складає 93,24%, оригінальність тексту роботи за результатами перевірки системою Anti-Plagiarism v-15.257 складає 99%.

Згідно з Положенням про систему забезпечення академічної доброчесності у ХНУ (https://khmnu.edu.ua/wp-content/uploads/normatyvni-dokumenty/polozhennya/pro-systemu-zabezpechennya-akademichnoyi-dobrochesnosti.pdf, Додаток В) кваліфікаційна робота, виконана за , освітньо-професійною програмою, кількісні показники рівня унікальності тексту у відсотках до загального обсягу матеріалу в якій складає 75-100 %, визнається роботою з високою унікальністю тексту: «Текст вважається унікальним і не потребує додаткових дій щодо запобігання неправомірним запозиченням».

Керівник роботи _____ Вікторія ОРЛЕНКО

Завідувач кафедри кібербезпеки _____ Юрій КЛЬОЦ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «бакалавр»

Студент Лагода Аліса Павлівна

Тема Система контролю доступу на базі блокчейн технології

Спеціальність 125 – Кібербезпека

Обсяг кваліфікаційної роботи освітньо-кваліфікаційного рівня «бакалавр»:

кількість листів креслень 5 ; кількість сторінок записки 64 .

1. Короткий зміст роботи та прийнятих рішень У кваліфікаційній роботі була розроблена система контролю доступу на базі блокчейн-технології для підприємства. Захист даних реалізовується за допомогою розподіленої бази даних, що забезпечує технологія блокчейн. У процесі проєктування були розроблені такі компоненти: система контролю доступу, модель загроз, система тдля ттестування. При цьому також надані рекомендації щодо роботи та впровадження системи, а також зазначено мінімальне необхідне обладнання для використання.

2. Висновок про відповідність кваліфікаційної роботи завданню У кваліфікаційній роботі було виконано поставлене завдання як у теоретичній, так і в практичній частині.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі бакалаврської роботи чітко формулюється вага даної системи для суспільства, оскільки захист даних та контроль доступу є ключовими елементами у цифровому світі. Завдання роботи включають аналіз існуючих систем, визначення переваг та недоліків блокчейн-рішень, а також розробку та тестування власної моделі. У першому розділі розглядаються поняття про блокчейн та існуючі системи побудовані на її основі, системи контролю доступу. Наступні розділи присвячені розробці системи контролю доступу на основі бдлокчейн-технології із такою системою доступу як магнітний ключ. Також було зазначено мінімальний склад системи. Автор провів тестування моделі, що підтвердили її працездатність.

4. Позитивні сторони роботи Кваліфікаційна робота має практичну цінність. Вона полягає у розробці системи контролю доступу на базі блокчейн-технології, що забезпечує захист інформації та виявляє стійкість до несанкціонованого доступу. Завдяки цьому підприємство є захищеним від витоку інформації та вторгнення зловмисників. Студент продемонстрував розуміння заданої теми, вміння аналізувати інформацію та використовувати засвоєнні знання на практиці структура роботи логічно побудована матеріал подано зрозуміло.

5. Негативні сторони роботи Система є фінансово затратною, що не дає змоги впровадження у маленькі підприємства. Не було складене порівняння щодо систем, які не пов'язані з блокчейном. Також варто було би детальніше розглянути економічні аспекти. На програмному рівні можна було використати вже існуючі блокчейни в тестовому варіанті.

6. Оцінка графічного оформлення та пояснювальної записки роботи В цілому, графічне оформлення відповідає вимогам системи та присутня також тестова платформа для сканування, а пояснювальна записка відповідає нормам оформлення.

7. Відгук про роботу в цілому Кваліфікаційна робота відповідає нормам щоб отримати позитивну оцінку. Усі розділи роботи мають логічну послідовність, що сприяє зрозумінню викладеного матеріалу в рамках теми роботи. Графічний матеріал надає змогу побачити і також об'єднає інформацію у зручний формат. В роботі наявна доцільність та ефективність прийнятих рішень для досягнення мети

8. Інші зауваження Переліку використаних джерел лише 30 ресурсів, недостатня кількість посилань використаних у тексті. Відсутній детальний опис проведення тестування на стійкість до несанкціонованого доступу.

9. Оцінка кваліфікаційної роботи: Враховуючи всі позитивні та негативні сторони представленої кваліфікаційної роботи, можна зробити висновок, що вона заслуговує оцінки «добре»

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи) Бармак Олександр Володимирович, д.т.н., професор, завідувач кафедри комп'ютерних наук.

«19» червня 2024. _____ (підпис)