Вінницький національний технічний університет

Факультет менеджменту та інформаційної безпеки

Кафедра менеджменту та безпеки інформаційних систем

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Захищена система спільного використання автомобілів на основі смарт контрактів з можливістю анонімізації даних користувачів для унеможливлення їхнього відслідковування»

Виконав: ст. 2-го курсу, групи КІТС-22 мз
спеціальності 125– Кібербезпека
Освітня програма – Кібербезпека
інформаційних технологій та систем
(шифр і назва напряму підготовки, спеціальності)

_____Саврацький В.М._____
(прізвище та ініціали)

Керівник: д.т.н., проф. каф. МБІС
_____Яремчук Ю.Є._____
(прізвище та ініціали)
«____» _____ 2024 р.

Опонент: к.т.н., доцент, каф. ОТ
_____Колесник І.С._____
(прізвище та ініціали)
«____» _____ 2024 р.

Допущено до захисту
Голова секції УБ кафедри МБІС

_____Юрій ЯРЕМЧУК
“_____” _____2024 р.

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

Рівень вищої освіти II-й (магістерський)
Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітньо-професійна програма – Кібербезпека інформаційних технологій та систем

ЗАТВЕРДЖУЮ
Голова секції УБ, кафедра МБІС
_____ **Юрій ЯРЕМЧУК**
“ ” _____ 2024 р.

З А В Д А Н Н Я
на магістерську кваліфікаційну роботу студенту
Саврацький Володимир Михайлович
(прізвище, ім`я, по-батькові)

1. Тема роботи:

«Захищена система спільного використання автомобілів на основі смарт контрактів з можливістю анонімізації даних користувачів для унеможливлення їхнього відслідковування»
Керівник роботи_: Яремчук Ю.Є. д.т.н., професор
(прізвище, ім`я, по-батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від “ 11” березня 2024 року № 81
2. Строк подання студентом роботи_за тиждень до захисту.__
3. Вихідні дані до роботи:
Стандарти, електронні джерела, підручники та наукові статті по темі, які стосуються теми магістерської кваліфікаційної роботи.
4. Зміст текстової частини:

У вступі викладено актуальність теми та сформульовано мету роботи. У першому розділі розглянуто теоретичні засади створення захищеної системи спільного використання автомобілів на основі смарт контрактів, включаючи основні принципи та можливості застосування смарт контрактів, механізми анонімізації даних користувачів, та застосування криптографічних методів для забезпечення безпеки даних. У другому розділі розроблено захищену систему для спільного використання автомобілів на основі смарт контрактів з можливістю анонімізації даних користувачів для унеможливлення їхнього відслідковування. У третьому розділі здійснено програмну реалізацію розробленої системи, включаючи вибір мови програмування, реалізацію системи смарт контрактів.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов`язкових креслень)

У дугому розділі магістерської кваліфікаційної роботи наведено 4 рисунка, у третьому розділі – 15 рисунків

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основна частина			
I	Яремчук Ю.Є. д.т.н., проф. каф. МБІС		
II	Яремчук Ю.Є. д.т.н., проф. каф. МБІС		
III	Яремчук Ю.Є. д.т.н., проф. каф. МБІС		
Економічна частина			
IV	Причепа І.В., к.е.н., доц. каф. ЕПВМ		

7.Дата видачі завдання _____2024 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи		Примітка
1.	Визначення напрямку магістерської роботи, формулювання теми	1.03.2024	15.03.2024	
2.	Аналіз предметної області обраної теми	15.03.2024	1.04.2024	
3.	Розробка роботи	1.04.2024	10.04.2024	
4.	Написання магістерської роботи на основі розробленої теми	10.04.2024	20.04.2024	
5.	Передзахист магістерської кваліфікаційної роботи	20.04.2024	10.05.2024	
6.	Виправлення, уточнення, корегування магістерської кваліфікаційної роботи	10.05.2024	4.06.2024	
7.	Захист магістерської кваліфікаційної роботи	10.06.2024	10.06.2024	

Студент _____ Саврацький В.М.
(підпис)

Керівник роботи _____ Яремчук Ю.Є.
(підпис)

АНОТАЦІЯ

УДК 004.56.5(043.2)

Саврацький В.М. Захищена система спільного використання автомобілів на основі смарт контрактів з можливістю анонімізації даних користувачів для унеможливлення їхнього відслідковування

Магістерська кваліфікаційна робота зі спеціальності 125 – «Кібербезпека», освітня програма «Кібербезпека інформаційних технологій та систем». Вінниця: ВНТУ, 2024. 113 с.

На укр.мові. Бібліогр.: 39 назв; рис.: 21; табл. 15

У магістерській кваліфікаційній роботі розглянуто теоретичні засади створення таких систем, включаючи основні принципи та можливості застосування смарт контрактів, механізми анонімізації даних користувачів, та застосування криптографічних методів для забезпечення безпеки даних.

Перша частина роботи містить аналіз популярних систем спільного використання автомобілів та висновки з постановкою задачі. Друга частина присвячена розробці захищеної системи, де детально розглянуто особливості створення такої системи, розробка загального алгоритму роботи системи та алгоритму обміну даними, а також алгоритми реєстрації та авторизації користувачів, і надання доступу до автомобіля.

Третя частина роботи охоплює програмну реалізацію розробленої системи, включаючи вибір мови програмування, реалізацію системи смарт контрактів та блокчейну, а також системи анонімізації користувачів та API для спільного використання автомобілів. Окремо проведено тестування реалізованої системи.

У четвертій частині роботи розглянуто економічні аспекти, зокрема, проведення комерційного та технологічного аудиту науково-технічної розробки.

Ключові слова: смарт контракти, анонімізації, спільний доступ.

ABSTRACT

Savratskiy V.M. Secure car sharing system based on smart contracts with the possibility of anonymizing user data to prevent their tracking

Master's qualification thesis on specialty 125 - "Cybersecurity", educational program "Cybersecurity of information technologies and systems". Vinnytsia: VNTU, 2024. 113 p.

In the Ukrainian language. Bibliography: 39 titles; Fig.: 21; table 15.

The master's qualification work examines the theoretical foundations of creating such systems, including the basic principles and possibilities of using smart contracts, mechanisms for anonymizing user data, and the use of cryptographic methods to ensure data security.

The first part of the work contains an analysis of popular car sharing systems and conclusions with a problem statement. The second part is devoted to the development of a secure system, which details the features of creating such a system, the development of a general system operation algorithm and data exchange algorithm, as well as algorithms for user registration and authorization, and providing access to the car.

The third part of the work covers the software implementation of the developed system, including the choice of programming language, the implementation of the smart contract system and blockchain, as well as user anonymization systems and car sharing APIs. Testing of the implemented system was carried out separately.

In the fourth part of the work, economic aspects are considered, in particular, conducting a commercial and technological audit of scientific and technical development, calculating costs for conducting scientific and research work, calculating the economic efficiency of scientific and technical development in case of its possible commercialization, as well as the effectiveness of investments and their payback period .

Keywords: smart contracts, anonymization, shared access.

ЗМІСТ

ВСТУП

Актуальність. У сучасному світі зростає популярність концепції спільного використання автомобілів як альтернативного способу транспортування, що дозволяє зменшити витрати на транспорт, знизити вплив на навколишнє середовище та підвищити мобільність населення. Водночас існують серйозні виклики, пов'язані з конфіденційністю та безпекою особистих даних користувачів таких систем. Існуючі платформи спільного використання автомобілів часто не забезпечують достатнього рівня захисту приватності і можуть стати об'єктом атак кіберзлочинців.

Використання технологій блокчейну та смарт контрактів відкриває нові можливості для створення захищених і прозорих систем, які забезпечують анонімізацію даних користувачів та унеможливлюють їх відслідковування. Впровадження таких систем може допомогти збільшити довіру користувачів до платформ спільного використання автомобілів і сприяти їхньому подальшому розвитку.

З цієї причини актуальним є розвиток і дослідження нових методів забезпечення конфіденційності та безпеки даних користувачів в системах спільного використання автомобілів, що базуються на технологіях блокчейну та смарт контрактів. Розробка інноваційних підходів до цих проблем є важливою для підтримки сталого розвитку міст і підвищення якості життя громадян.

Метою і задачею дослідження даної роботи є розробка захищеної системи спільного використання автомобілів на основі смарт контрактів з можливістю анонімізації даних користувачів для унеможливлення їхнього відслідковування.

Задачами дослідження є:

– вивчення основних принципів та можливостей застосування смарт контрактів.

– аналіз існуючих механізмів анонімізації даних користувачів та криптографічних методів для забезпечення безпеки даних.

– проведення аналізу популярних систем спільного використання автомобілів.

– розробка загального алгоритму роботи системи та системи обміну даними.

– створення алгоритмів реєстрації та авторизації користувачів, а також алгоритму надання доступу до автомобіля.

– програмна реалізація системи, включаючи смарт контракти, блокчейн та системи анонімізації даних користувачів.

– тестування реалізованої системи та оцінка її ефективності.

– проведення економічного обґрунтування розробки, включаючи оцінку комерційного потенціалу та економічної ефективності.

Об'єктом дослідження є системи спільного використання автомобілів.

Предметом дослідження є методи забезпечення конфіденційності та безпеки даних користувачів в системах спільного використання автомобілів на основі смарт контрактів.

Новизна роботи полягає в розробці інноваційної системи спільного використання автомобілів, яка базується на смарт контрактах і використовує передові методи анонімізації та криптографії для забезпечення конфіденційності та безпеки даних користувачів. Запропонований підхід дозволяє унеможливити відслідковування користувачів, що є важливим фактором для збереження їхньої приватності.

Практична цінність роботи полягає у створенні захищеної системи спільного використання автомобілів, яка може бути впроваджена в реальні умови для підвищення ефективності та безпеки таких систем. Розроблені алгоритми та програмне забезпечення можуть бути використані для подальшого розвитку технологій спільного використання автомобілів, що забезпечує високу конфіденційність та захист даних користувачів.

1 ТЕОРЕТИЧНІ ЗАСАДИ СТВОРЕННЯ ЗАХИЩЕНОЇ СИСТЕМИ СПІЛЬНОГО ВИКОРИСТАННЯ АВТОМОБІЛІВ НА ОСНОВІ СМАРТ КОНТРАКТІВ

У першому розділі магістерської кваліфікаційної роботи буде проведено аналіз та дослідження теоретичних основ створення захищеної системи спільного використання автомобілів на основі смарт-контрактів, включаючи їхні можливості та застосування, механізми анонімізації даних та застосування криптографічних методів для забезпечення безпеки.

Результати аналізу будуть узагальнені в висновках, і буде поставлена задача для подальших досліджень.

1.1 Смарт-контракти: основні принципи та можливості застосування

Смарт-контракти – це програмні коди, які автоматизовано виконують, перевіряють або виконують угоди в рамках блокчейн-платформи. Вони є основою для реалізації різноманітних децентралізованих додатків, які не вимагають централізованого контролю. Основні принципи та можливості смарт-контрактів включають [1]:

– децентралізація. Смарт-контракти працюють на блокчейн-платформах, що означає, що вони виконуються на всіх вузлах мережі, а не в одній централізованій системі. Це забезпечує високу стійкість до втручання та надійність угод;

– надійність та безпека. Завдяки криптографії та протоколам консенсусу, смарт-контракти забезпечують високий рівень безпеки. Інформація, збережена в смарт-контрактах, надійно захищена від маніпуляцій та змін;

– автоматизація. Смарт-контракти автоматично виконують угоди при виконанні певних умов, без необхідності посередників або додаткових дій користувачів;

– прозорість. Всі операції, які виконуються в смарт-контракті, записуються в блокчейні, що забезпечує високий рівень прозорості та відкритості для всіх учасників системи;

– ефективність витрат: Використання смарт-контрактів дозволяє уникнути витрат на посередників та зменшити витрати на проведення угод.

Поняття смарт-контрактів виникло в рамках розробки блокчейн-технологій, зокрема на основі платформи Ethereum. Смарт-контракти є цифровими програмами, які автоматично виконують умови угод, встановлені між сторонами, без потреби в довіреній третій стороні. Основна їхній принцип полягає в тому, що вони дозволяють автоматизувати та зафіксувати угоди між учасниками, не втручаючись при цьому в діяльність третьої сторони [2].

Призначення смарт-контрактів можна розглядати у різноманітних сферах індустрії та діяльності, а саме:

– цифрові фінансові угоди. Смарт-контракти можуть використовуватися для автоматизованого виконання фінансових угод, таких як передача грошей, мікрокредитування, страхування тощо. Наприклад, смарт-контракт може автоматично виконати оплату після отримання певної послуги або товару;

– управління власністю. Смарт-контракти можуть бути застосовані для визначення прав власності на цифрові активи, такі як криптовалюти, токени, нерухомість або інтелектуальна власність. Вони дозволяють автоматизувати процеси передачі власності та управління правами;

– системи управління ресурсами. У контексті спільного використання автомобілів або інших ресурсів, смарт-контракти можуть контролювати доступ до ресурсів, автоматизувати процеси бронювання, оплати та використання, забезпечуючи ефективне управління;

– електронні голосування та урядові послуги. Смарт-контракти можуть використовуватися для забезпечення безпечного та прозорого проведення електронних голосувань або для реалізації різних урядових послуг, забезпечуючи автоматизацію та недоторканність у процесі прийняття рішень.

Принципи роботи смарт-контрактів на основі блокчейн-технологій можна розглядати з декількох ключових аспектів:

– безпека та недоступність для змін;

Блокчейн – це децентралізована система, в якій дані зберігаються на різних вузлах мережі в блоках. Кожен блок містить криптографічний хеш попереднього блоку, що утворює ланцюжок блоків.

Інформація про смарт-контракти зберігається в блоках разом з іншими даними. Ці блоки захищені криптографічними підписами, які гарантують їх цілісність та недопускають внесення будь-яких змін без відповідного авторизованого підпису.

– децентралізація;

Дані смарт-контрактів зберігаються на всіх вузлах мережі блокчейн, що робить їх незалежними від будь-якого централізованого органу чи сервера. Це гарантує, що немає єдиного точки виходу для атак та недоступності.

– розподіленість;

Інформація про смарт-контракти розподіляється між всіма вузлами мережі. Це забезпечує відмовостійкість, оскільки навіть якщо деякі вузли мережі вийдуть з ладу, інші продовжать забезпечувати доступ до даних і виконання угод [3].

– криптографічна безпека;

Кожен блок містить криптографічний хеш попереднього блоку та дані, що забезпечує високий рівень безпеки та невідмовності. Це означає, що будь-які спроби модифікувати існуючі блоки або додати нові будуть виявлені як недійсні.

– прозорість та верифікація.

Блокчейн є прозорою системою, оскільки дані доступні для перегляду всіма учасниками мережі. Це робить смарт-контракти відкритими та перевіряємими для всіх зацікавлених сторін.

Приклад простої схеми роботи смарт-контракту (рис. 1.1).

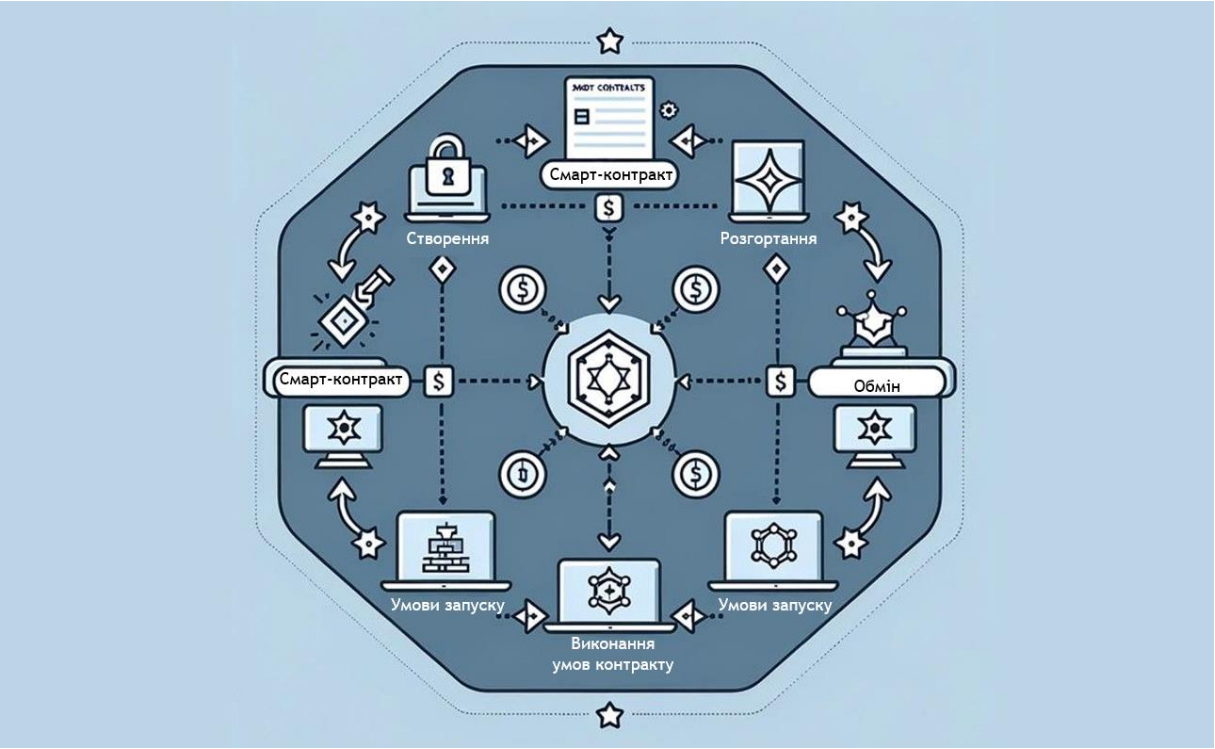


Рисунок 1.1 – Схема роботи смарт контракту

На цій схемі представлено загальний принцип взаємодії з смарт-контрактом:

– ініціалізація контракту. Створення смарт-контракту у блокчейні з визначеними умовами та параметрами;

– виклик функції. Учасники системи (наприклад, користувачі або інші контракти) викликають функції контракту;

– виконання логіки. Смарт-контракт перевіряє умови та параметри, виконує визначену логіку та здійснює необхідні дії;

– оновлення стану. Зміна стану контракту після виконання функції, яка може впливати на подальші дії в системі або зміну стану учасників.

Ця схема демонструє загальний процес взаємодії з смарт-контрактом та його роботу в рамках блокчейн-системи [3].

Отже, смарт-контракти, які базуються на блокчейн-технологіях, забезпечують високий рівень безпеки, надійності та прозорості у виконанні угод, роблячи їх ідеальним інструментом для автоматизації та захисту у різних сферах діяльності.

Використання смарт-контрактів має численні переваги у порівнянні з традиційними методами укладення угод та виконання операцій. Детальний огляд переваг:

– автоматизація;

Смарт-контракти дозволяють автоматизувати виконання угод без потреби у втручанні сторін або довіреної третьої сторони. Це робить процес укладання та виконання угод більш ефективним та швидким.

– безпека;

Інформація про смарт-контракти зберігається на блокчейні, що забезпечує високий рівень безпеки та недоступність для змін. Криптографічні підписи та хеші гарантують цілісність та автентичність даних.

– надійність;

Смарт-контракти виконуються безпосередньо в мережі блокчейн, що робить їх надійними та невідмовними. Вони працюють відповідно до заданих умов та логіки, не підділяючись до емоційних впливів або змін настрою.

– прозорість;

Блокчейн забезпечує прозорість та відкритий доступ до всіх транзакцій та операцій, здійснених за допомогою смарт-контрактів. Це робить угоди та операції більш прозорими та перевіряемими для всіх сторін.

– низькі витрати;

Використання смарт-контрактів може призвести до зниження витрат на посередників, операційні витрати та адміністративні витрати. Це робить укладання та виконання угод економічно вигідними [4].

– глобальність;

Смарт-контракти можуть бути використані у будь-якій частині світу, оскільки їхня виконавча логіка розміщується на блокчейні, доступному для всіх учасників мережі. Це дозволяє створювати глобальні рішення та угоди без обмежень щодо місця проживання або географічних меж.

– підвищена ефективність;

Автоматизація виконання угод та операцій за допомогою смарт-контрактів дозволяє зменшити час, зусилля та ресурси, потрібні для виконання таких дій. Це дозволяє підвищити ефективність бізнес-процесів та зосередитися на стратегічних завданнях.

Врахування цих переваг дозволяє користувачам максимально використовувати потенціал смарт-контрактів для автоматизації, захисту та оптимізації їхніх угод та операцій [4].

Смарт-контракти – це нова технологія з великим потенціалом для автоматизації, захисту та оптимізації угод та операцій. Їх децентралізований, надійний, прозорий та ефективний характер робить їх ідеальним інструментом для різних сфер діяльності.

1.2 Використання смарт контрактів у системі спільного використання автомобілів

Використання смарт-контрактів у системі спільного використання автомобілів може бути дуже корисним, оскільки дозволяє автоматизувати та упростити багато аспектів цієї сфери.

Автоматизація процесів у системі спільного використання автомобілів за допомогою смарт-контрактів дозволяє забезпечити ефективне та безпечне управління та використанням автопарку.

Смарт-контракти автоматизують процес бронювання, підтвердження та оплати оренди автомобіля. Користувачі можуть взаємодіяти з системою через мобільний додаток або веб-платформу, вибираючи автомобіль, дату та час оренди. Після вибору смарт-контракт автоматично перевіряє доступність автомобіля, резервує його для користувача, підтверджує умови та узгоджує оплату.

Смарт-контракти автоматизують списання коштів з рахунку орендаря та перерахування їх орендодавцю. Після завершення оренди смарт-контракт автоматично розраховує вартість користування автомобілем відповідно до

встановлених умов (час, відстань тощо) та автоматично здійснює оплату за використання.

Смарт-контракти автоматизують процес відкриття та закриття дверей автомобіля за допомогою електронних ключів. Після бронювання та оплати користувач отримує в електронному вигляді ключ доступу до автомобіля через мобільний додаток. Смарт-контракт автоматично керує системою доступу автомобіля та надає доступ лише у визначені часові і просторові рамки.

Смарт-контракти можуть автоматизувати процес оформлення та обробки страхових випадків. При кожному бронюванні автомобіля смарт-контракт може перевіряти статус страхування, а також автоматично оформляти страховий поліс на період користування автомобілем. У разі страхового випадку смарт-контракт може автоматично ініціювати процес розгляду та вирішення страхової вимоги [5].

Застосування смарт-контрактів для автоматизації цих процесів у системі спільного використання автомобілів дозволяє покращити зручність, безпеку та ефективність управління автопарком для як власників, так і користувачів.

Система спільного використання автомобілів на основі смарт-контрактів потребує чіткої та безпечної системи управління доступом, щоб гарантувати, що лише авторизовані користувачі можуть отримати доступ до автомобіля. Кілька способів, якими смарт-контракти можуть використовуватися для управління доступом:

Ідентифікація користувачів:

– перевірка особистості. Смарт-контракти можуть використовуватися для перевірки особистості користувачів, наприклад, за допомогою їхніх паспортів, водійських посвідчень або інших документів, що посвідчують особу;

– біометрична аутентифікація. Смарт-контракти можуть використовувати біометричні дані, такі як відбитки пальців або сканування обличчя, для аутентифікації користувачів.

Надання доступу:

– електронні ключі. Смарт-контракти можуть генерувати та видавати електронні ключі, які дозволяють користувачам відкривати двері та запускати двигун автомобіля;

– геозонування. Смарт-контракти можуть використовуватися для обмеження зони, в якій може використовуватися автомобіль. Це може бути корисно для запобігання крадіжкам або для забезпечення того, щоб автомобіль використовувався лише в певному районі;

– час використання. Смарт-контракти можуть використовуватися для встановлення чітких часових рамок, протягом яких користувач може використовувати автомобіль.

Моніторинг та контроль:

– відстеження місцезнаходження. Смарт-контракти можуть використовуватися для відстеження місцезнаходження автомобіля. Це може бути корисно для запобігання крадіжкам або для відстеження того, де використовується автомобіль;

– поведінка водія. Смарт-контракти можуть використовуватися для моніторингу поведінки водія, наприклад, дотримання правил дорожнього руху. Це може бути корисно для оцінки ризиків та для страхування.

Блокування:

– у разі несанкціонованого доступу. Смарт-контракти можуть використовуватися для блокування автомобіля у разі несанкціонованого доступу;

– у разі порушення умов. Смарт-контракти можуть використовуватися для блокування автомобіля у разі порушення користувачем умов оренди.

Переваги використання смарт-контрактів для управління доступом:

– підвищення безпеки. Смарт-контракти eliminyють ризики шахрайства та несанкціонованого доступу;

– підвищення прозорості. всі умови доступу до автомобіля записані в смарт-контракті та доступні для публічного аудиту;

– збільшення ефективності. смарт-контракти автоматизують рутинні процеси, пов'язані з управлінням доступом.

Виклики використання смарт-контрактів для управління доступом:

– складність розробки. смарт-контракти потребують знань програмування та розуміння блокчейн-технологій;

– юридична невизначеність. правовий статус смарт-контрактів у багатьох юрисдикціях залишається невизначеним;

– технічні обмеження. смарт-контракти мають обмеження щодо складності та обсягу даних.

Ці функції сприяють покращенню безпеки та контролю за користуванням автомобілями у системі спільного використання, забезпечуючи більшу надійність та ефективність в управлінні доступом [9].

Система спільного використання автомобілів на основі смарт-контрактів може використовувати дані для:

Пробіг:

– смарт-контракти можуть автоматично фіксувати пробіг автомобіля. Це може бути корисно для:

Оплати: розрахунку вартості оренди;

Страхування: оцінки ризиків;

Технічного обслуговування: планування та відстеження.

Стан:

– смарт-контракти можуть використовуватися для фіксації стану автомобіля перед та після оренди. Це може бути корисно для:

Виявлення пошкоджень: визначення відповідальності за пошкодження;

Технічного обслуговування: відстеження стану та планування ремонту;

Поведінка водія:

– смарт-контракти можуть використовуватися для моніторингу поведінки водія, наприклад, дотримання правил дорожнього руху. Це може бути корисно для:

Оцінки ризиків: визначення безпечних водіїв;

Страхування: розрахунку страхових премій;

Стимулювання безпечного водіння: винагородження безпечних водіїв.

Переваги фіксації даних:

– підвищення прозорості. всі дані про використання автомобіля доступні для публічного аудиту;

– збільшення ефективності. дані можуть використовуватися для покращення роботи системи;

– підвищення безпеки. дані можуть використовуватися для виявлення та запобігання шахрайству.

Фіксація даних може зробити систему спільного використання автомобілів більш прозорою, ефективною та безпечною.

Після аналізу теоретичних аспектів застосування смарт-контрактів у системі спільного використання автомобілів та їх можливостей, розглянемо приклад схеми (рис. 1.2) роботи смарт-контракту.

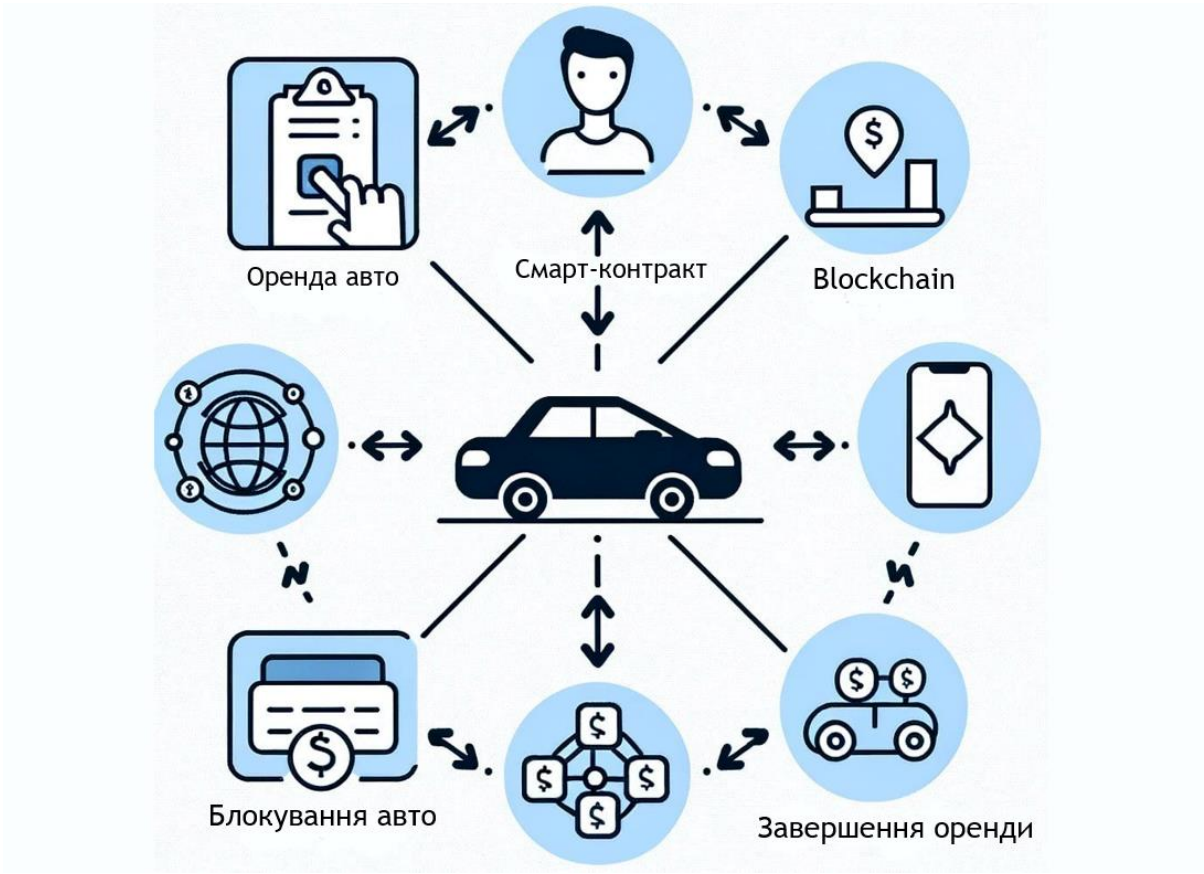


Рисунок 1.2 – Приклад схеми роботи смарт-контракту у системі спільного використання автомобілів

На схемі представлено загальний принцип взаємодії учасників системи та смарт-контракту [11]:

– користувачі реєструються та взаємодіють з системою через веб-інтерфейс або мобільний додаток;

– після реєстрації, користувачі можуть переглядати доступні автомобілі та резервувати їх;

– після вибору автомобіля, користувачі встановлюють параметри оренди (тривалість, маршрут тощо) та підтверджують оплату;

– смарт-контракт автоматично перевіряє умови оренди та виконує відповідні дії: розблоковує доступ до автомобіля на зазначений час, списує кошти з рахунку користувача тощо;

– після закінчення оренди, користувачі повертають автомобіль, а смарт-контракт автоматично завершує оренду та блокує доступ до автомобіля.

Ця схема демонструє основні кроки та взаємодію учасників у системі спільного використання автомобілів на основі смарт-контрактів. Цей приклад може бути використаний як основа для подальшого розвитку та налаштування системи відповідно до конкретних вимог та потреб користувачів.

В цілому, смарт-контракти мають значний потенціал для покращення системи спільного використання автомобілів, роблячи її більш зручною, безпечною, прозорою та ефективною [12].

Завдяки цим крокам смарт-контракти можуть революціонізувати систему спільного використання автомобілів, роблячи її більш доступною та зручною для всіх.

1.3 Механізми анонімізації даних користувачів для забезпечення конфіденційності

Захист конфіденційності даних користувачів є важливою складовою будь-якої системи спільного використання автомобілів. Анонімізація даних

користувачів може допомогти захистити їхню конфіденційність та мінімізувати ризики зловживання даними [13].

Псевдонімізація є ефективним методом анонімізації даних, який полягає у заміні реальних імен або інших ідентифікаторів на випадково згенеровані або зашифровані значення, що надаються користувачам у якості альтернативи. Розглянемо детальніше процес псевдонімізації та його переваги:

– генерація псевдонімів;

Першим кроком у процесі псевдонімізації є генерація або створення псевдонімів для кожного індивідуального користувача. Ці псевдоніми можуть бути випадково згенерованими унікальними ідентифікаторами або зашифрованими значеннями, які не можна легко пов'язати з реальною особою.

– заміна реальних ідентифікаторів;

Після того, як псевдоніми згенеровані, вони замінюють реальні імена, адреси, номери телефонів та інші особисті дані користувачів у системі. Це означає, що при збереженні або передачі даних застосовуються лише псевдоніми, а не реальні особисті дані.

– збереження безпеки даних;

Використання псевдонімів дозволяє зберігати особисті дані користувачів у безпечній формі. Навіть якщо зловмисники отримають доступ до бази даних, вони не зможуть отримати реальні особисті дані, оскільки вони будуть зашифровані або представлені у вигляді псевдонімів.

– збереження конфіденційності;

Однією з ключових переваг псевдонімізації є збереження конфіденційності даних користувачів. Псевдоніми не пов'язані з реальними особами, тому неможливо встановити ідентичність користувача на основі цих даних, що додає додатковий шар захисту особистої інформації.

– можливість використання для аналізу.

Помимо анонімізації, псевдоніми можуть залишати дані в придатному для аналізу стані. Наприклад, у випадку дослідницьких або статистичних проектів, де

потрібно аналізувати шаблони чи зв'язки між користувачами, псевдонімізація дозволяє використовувати ці дані без порушення конфіденційності особистої інформації.

Використання псевдонімів є ефективним способом забезпечення анонімізації та конфіденційності даних користувачів у системі спільного використання автомобілів, зменшуючи ризик неправомірного доступу до особистої інформації.

Після застосування псевдонімів для анонімізації особистих даних, можна також застосувати шифрування для додаткового забезпечення конфіденційності. Шифрування даних перед їх зберіганням або передачею дозволяє захистити їх від несанкціонованого доступу та перегляду [15].

Шифрування даних – це процес перетворення звичайного тексту (наприклад, особистих даних користувачів) у криптографічно захищений текст, який може бути розшифрований лише за допомогою відповідного ключа. У контексті системи спільного використання автомобілів за допомогою смарт-контрактів, шифрування даних є ключовим механізмом для забезпечення конфіденційності особистої інформації користувачів. Детальний розгляд процесу шифрування даних:

– вибір криптографічного алгоритму;

Перед шифруванням даних необхідно вибрати відповідний криптографічний алгоритм. Зазвичай для цього використовуються сучасні алгоритми шифрування, такі як AES (Advanced Encryption Standard), RSA (Rivest-Shamir-Adleman) або ECC (Elliptic Curve Cryptography), які відомі своєю високою стійкістю та безпекою.

– генерація ключа шифрування;

Для шифрування та розшифрування даних потрібно використовувати ключ, який може бути відомим лише авторизованим особам. Генерація безпечного ключа є важливим етапом процесу. Зазвичай для генерації ключів використовуються криптографічні генератори випадкових чисел [19].

– шифрування даних;

Після вибору алгоритму та генерації ключа шифрування звичайний текст (наприклад, особисті дані користувача) перетворюється у криптографічно

захищений шифрований текст за допомогою вибраного алгоритму та згенерованого ключа. Шифрування забезпечує захист даних від несанкціонованого доступу та перегляду.

– збереження або передача зашифрованих даних;

Після шифрування дані можуть бути збережені в блокчейні або передані через мережу. Шифровані дані залишаються безпечними та конфіденційними, оскільки для їх розшифрування потрібний відповідний ключ.

– розшифрування даних.

Для отримання оригінальних даних потрібно розшифрувати шифрований текст за допомогою того ж самого алгоритму та ключа, який використовувався для шифрування. Цей процес може бути виконаний лише авторизованими особами, які мають доступ до відповідного ключа [20].

Шифрування даних є важливим механізмом анонімізації для захисту особистої інформації користувачів у системі спільного використання автомобілів, забезпечуючи конфіденційність та безпеку даних під час їх збереження та передачі.

Після застосування шифрування для забезпечення безпеки даних можна перейти до обговорення конфіденційності транзакцій. Шифрування даних перед передачею допомагає зберегти конфіденційність транзакцій та особистої інформації користувачів, що підвищує рівень довіри до системи спільного використання автомобілів на основі смарт-контрактів.

Конфіденційність транзакцій в системі спільного використання автомобілів на основі смарт-контрактів є важливим аспектом, оскільки вона забезпечує приватність і конфіденційність усіх фінансових операцій. Розглянемо детальніше механізми, які можуть бути використані для забезпечення конфіденційності транзакцій:

– приватні блокчейни;

Використання приватних блокчейнів є одним із способів забезпечення конфіденційності транзакцій. У приватному блокчейні доступ до записів обмежується лише певним групам учасників, що дозволяє зберігати інформацію

про транзакції конфіденційною. Тільки авторизовані сторони мають доступ до даних про транзакції, збережених у блокчейні.

– технології забезпечення конфіденційності;

Деякі криптовалютні системи використовують технології, які спеціально розроблені для забезпечення конфіденційності транзакцій. Наприклад, техніка змішування монет дозволяє змішувати вхідні та вихідні транзакції, що ускладнює відстеження шляху грошей і забезпечує більшу анонімність.

– шифрування та підписи;

Для забезпечення конфіденційності транзакцій може бути використане шифрування даних та цифрові підписи. Шифрування даних під час транзакції забезпечує конфіденційність інформації, тоді як цифрові підписи дозволяють перевірити автентичність транзакції без розкриття особистих даних.

– анонімні адреси.

Деякі криптовалютні системи дозволяють створювати анонімні адреси для здійснення транзакцій. Ці адреси не пов'язані з реальними іменами чи ідентифікаторами користувачів і дозволяють зберігати конфіденційність та анонімність під час проведення операцій [21].

Застосування цих механізмів дозволяє забезпечити конфіденційність та анонімність транзакцій у системі спільного використання автомобілів на основі смарт-контрактів. Це робить систему більш привабливою для користувачів, які цінують свою приватність та конфіденційність своїх фінансових операцій.

Обмеження доступу до особистих даних є ключовим аспектом забезпечення конфіденційності та безпеки в системі спільного використання автомобілів на основі смарт-контрактів. Розглянемо детальніше, як це може бути реалізовано за допомогою смарт-контрактів:

– встановлення прав доступу;

Смарт-контракти можуть встановлювати правила доступу до особистих даних, що забезпечують, що лише авторизовані користувачі мають доступ до цих

даних. Це може включати встановлення рівнів доступу або надання спеціальних дозволів для конкретних дій або операцій.

– необхідна інформація;

Смарт-контракти можуть програмуватися таким чином, щоб передавати лише необхідну інформацію для виконання певних операцій. Наприклад, якщо користувач орендує автомобіль, смарт-контракт може передати лише інформацію про час оренди та місце забору автомобіля, залишаючи інші особисті дані в зашифрованому або анонімізованому вигляді [22].

– шифрування даних;

Особисті дані можуть бути шифрованими перед збереженням у блокчейні чи передачею через мережу. Це забезпечує, що навіть якщо дані отримані незаконно, вони будуть зашифрованими і не доступними для розгляду без відповідного ключа.

– автоматичне видалення даних;

Смарт-контракти можуть включати механізми автоматичного видалення особистих даних після завершення певних операцій або після закінчення терміну зберігання. Це допомагає мінімізувати ризик зберігання зайвої особистої інформації та підвищує рівень конфіденційності даних.

– аудит доступу.

Смарт-контракти можуть реєструвати всі запити на доступ до даних, що дозволяє проводити аудит і виявляти недозволений доступ. Це дозволяє вчасно реагувати на можливі порушення безпеки та неправомірний доступ до особистих даних користувачів.

Застосування цих механізмів дозволяє забезпечити обмеження доступу до особистих даних користувачів у системі спільного використання автомобілів, забезпечуючи конфіденційність та безпеку даних.

Анонімізація даних користувачів та забезпечення конфіденційності є важливими аспектами системи спільного використання автомобілів на основі смарт-контрактів.

Застосування цих механізмів робить систему спільного використання автомобілів більш привабливою для користувачів, які цінують свою приватність та конфіденційність.

Завдяки цьому система спільного використання автомобілів на основі смарт-контрактів може стати безпечним та зручним способом подорожувати, не жертвуючи приватністю.

1.4 Застосування криптографічних методів для забезпечення безпеки даних

Криптографія відіграє ключову роль у забезпеченні безпеки даних у системі спільного використання автомобілів на основі смарт-контрактів. Вона використовується для захисту даних від несанкціонованого доступу, крадіжки, модифікації та інших загроз.

Шифрування даних є одним з найважливіших і ефективних методів забезпечення конфіденційності інформації в сучасних інформаційних системах. Розглянемо детальніше основні аспекти шифрування даних:

– принцип роботи;

Шифрування даних – це процес перетворення звичайного тексту (відомого як «відкритий текст») у незрозумілий для незаконного користувача вигляд (відомий як «шифртекст») за допомогою спеціального математичного алгоритму та ключа шифрування [23].

– види шифрування;

Існують два основних види шифрування: симетричне та асиметричне. У симетричному шифруванні використовується один ключ для як шифрування, так і розшифрування даних (наприклад, AES). У випадку асиметричного шифрування використовуються два ключі: публічний та приватний (наприклад, RSA).

– процес шифрування;

Під час шифрування даних, відкритий текст перетворюється у шифртекст за допомогою обраного алгоритму та ключа шифрування. Шифртекст може бути

переданий по відкритій мережі або збережений у системі без ризику незаконного доступу.

– ключі шифрування;

Ключ шифрування – це секретна інформація, яка використовується для шифрування даних. Для симетричного шифрування, цей ключ обов'язково потрібно захистити і зберігати в безпечному місці. Для асиметричного шифрування, приватний ключ слід захищати, а публічний ключ може бути розповсюджений.

– алгоритми шифрування;

Для забезпечення найвищого рівня безпеки можна використовувати різні алгоритми шифрування, такі як AES, RSA, DES (Data Encryption Standard), і т.д. Кожен з них має свої особливості та застосується у відповідності до потреб безпеки системи.

– шифрування в реальному часі.

Шифрування може застосовуватися у реальному часі під час передачі даних через мережу, так що навіть якщо дані перехоплені, вони будуть незрозумілими для зловмисників.

Загалом, шифрування даних є важливим і ефективним методом забезпечення конфіденційності інформації в сучасних інформаційних системах. Воно використовується для захисту особистих даних, фінансової інформації, комерційних та державних секретів, і грає ключову роль у забезпеченні безпеки даних у системі спільного використання автомобілів на основі смарт-контрактів [25].

Цифровий підпис є ключовим елементом в системах забезпечення безпеки даних і використовується для підтвердження автентичності, цілісності та невідмінності даних. Давайте розглянемо детальніше основні аспекти цифрового підпису:

– принцип роботи;

Цифровий підпис – це спеціальний криптографічний засіб, який використовується для підпису даних. Він працює на основі асиметричного криптографічного алгоритму, де користувач має два ключі: приватний та публічний. Приватний ключ використовується для створення підпису, а публічний ключ – для перевірки цього підпису.

– генерація підпису;

Користувач, який хоче підписати певні дані, використовує свій приватний ключ для створення унікального цифрового підпису цих даних. Цей підпис гарантує, що дані не були змінені після підпису та підтверджує автентичність походження даних.

– перевірка підпису;

Інші користувачі, які отримують ці дані, можуть перевірити цифровий підпис, використовуючи публічний ключ власника підпису. Підпис перевіряється на відповідність інформації, що підписується, і якщо підпис є валідним, це підтверджує автентичність та цілісність даних.

– використання у системі спільного використання автомобілів;

У контексті системи спільного використання автомобілів, цифровий підпис може бути використаний для підтвердження автентичності та цілісності даних про оренду автомобіля, операції платежів, угод тощо. Наприклад, при оформленні контракту на оренду автомобіля створюється цифровий підпис, який забезпечує підтвердження даних і узгодження між сторонами.

– захист від подальших змін.

Цифровий підпис також надає захист від подальших змін даних після підпису. Якщо дані будуть змінені, то перевірка підпису не буде успішною, що підтвердить, що дані були недозволено модифіковані.

Загалом, цифровий підпис грає ключову роль у забезпеченні безпеки даних та підтвердженні їхньої автентичності в різних системах, включаючи систему спільного використання автомобілів на основі смарт-контрактів.

Хеш-функції є важливим інструментом криптографії, який використовується для створення унікальних хеш-кодів або відбитків даних. Детальніше розглянемо основні аспекти хеш-функцій:

– принцип роботи;

Хеш-функція приймає на вхід послідовність даних будь-якої довжини і генерує унікальний вихід, відомий як хеш-код або хеш-значення. Цей вихідний хеш зазвичай має фіксовану довжину, незалежну від довжини вхідних даних.

– унікальність хешу;

Одне з основних властивостей хеш-функцій полягає у тому, що для різних вхідних даних генерується унікальний хеш-код. Це означає, що малі зміни в вихідних даних призводять до значних змін у вихідному хеші.

– цілісність даних;

Хеш-функції використовуються для перевірки цілісності даних. Якщо навіть невелика зміна буде внесена в вихідні дані, це призведе до повного змінення хеш-значення. Таким чином, хеш може служити індикатором будь-якої недозволеної модифікації даних.

– швидкодія;

Хеш-функції зазвичай є швидкодійними і ефективними в обчисленнях. Це дозволяє використовувати їх для обробки великих обсягів даних без значного збільшення часу обчислення.

– використання в криптографії;

Хеш-функції використовуються в різних криптографічних застосунках, таких як підписи повідомлень, захист від підробки, зберігання паролів, перевірка цілісності даних тощо.

– алгоритми хешування.

Існує багато різних алгоритмів хешування, таких як MD5, SHA-1, SHA-256, які мають різні рівні безпеки та швидкодії. Більш безпечні алгоритми, наприклад, SHA-256, зазвичай рекомендуються для застосування в криптографічних цілях.

Хеш-функції грають важливу роль у забезпеченні безпеки даних, вони застосовуються в різних галузях, від криптографії до зберігання паролів, і вони є необхідним елементом для забезпечення цілісності та безпеки в системі спільного використання автомобілів на основі смарт-контрактів.

Протоколи аутентифікації є важливими складовими систем безпеки, що дозволяють перевірити і підтвердити ідентичність користувачів або систем у мережі. Розглянемо детальніше основні аспекти протоколів аутентифікації:

TLS (Transport Layer Security);

TLS є криптографічним протоколом, який забезпечує безпеку комунікації в мережі шляхом захисту протоколів передачі даних на транспортному рівні. Він забезпечує конфіденційність, цілісність даних та аутентифікацію між клієнтом та сервером. TLS використовує сертифікати для підтвердження ідентичності сторін.

HTTPS (Hypertext Transfer Protocol Secure);

HTTPS є захищеною версією протоколу HTTP, яка використовує TLS або SSL (Secure Sockets Layer) для захисту комунікації між клієнтом та сервером. Вона забезпечує шифрування даних під час передачі через мережу, а також перевірку автентичності сервера за допомогою сертифікатів.

Kerberos;

Kerberos є протоколом аутентифікації, який використовується для безпечної аутентифікації користувачів у розподілених комп'ютерних мережах. Він базується на взаємодії між клієнтом, сервером аутентифікації та службою керування ключами (Key Distribution Center – KDC) для надання та перевірки керуючих даних доступу [26].

OAuth;

OAuth – це відкритий протокол аутентифікації, який дозволяє користувачам давати обмежений доступ до своїх ресурсів іншим додаткам без передачі свого пароля. Він широко використовується для авторизації користувачів на веб-сервісах та додатках.

SAML (Security Assertion Markup Language);

SAML є стандартом для обміну даними аутентифікації та авторизації між безпечними стосунками. Він використовує XML для передачі захищених аутентифікаційних токенів між ідентифікованим (заявником) та послугою (системою, яка здійснює доступ до ресурсів).

OpenID Connect.

OpenID Connect – це протокол аутентифікації, який побудований на основі протоколу OAuth 2.0 і надає стандартизований спосіб авторизації для користувачів. Він дозволяє користувачам використовувати один і той же акаунт для доступу до різних ресурсів та додатків.

Протоколи аутентифікації є ключовими для забезпечення безпеки та конфіденційності в мережах та додатках. Вони дозволяють ефективно перевіряти і підтверджувати ідентичність користувачів та забезпечувати безпеку в обміні даними.

Застосування цих криптографічних методів допомагає забезпечити безпеку даних у системі спільного використання автомобілів, зберігаючи конфіденційність, цілісність та доступність інформації для авторизованих користувачів.

Криптографія є потужним інструментом для забезпечення безпеки даних у системі спільного використання автомобілів на основі смарт-контрактів. Завдяки правильному застосуванню криптографії, система може стати більш надійною та зручною для користувачів.

1.5 Аналіз популярних систем спільного використання автомобілів

Аналіз популярних систем спільного використання автомобілів є важливим кроком для розуміння поточного стану цього ринку та ідентифікації ключових особливостей, переваг і недоліків цих систем.

Розглянемо деякі з найпопулярніших систем спільного використання автомобілів і проведемо їх аналіз.

CarBlock – це децентралізована платформа спільного користування автомобілями, побудована на блокчейні NEO. Вона прагне революціонізувати традиційну модель каршерінгу, пропонуючи користувачам низку переваг.

Особливості CarBlock:

– відсутність посередників. CarBlock дозволяє власникам автомобілів напряму здавати їх в оренду користувачам, без участі посередників, таких як традиційні компанії каршерінгу. Це дозволяє знизити транзакційні витрати та пропонувати більш конкурентоспроможні ціни;

– використання криптовалюти CBC. Для оренди автомобілів користувачі використовують власну криптовалюту CarBlock (CBC). Це забезпечує безпечні та швидкі транзакції, а також потенціал для отримання винагород учасникам платформи;

– система репутації. CarBlock використовує систему репутації, щоб заохочувати відповідальну поведінку як орендарів, так і орендодавців. Користувачі отримують рейтинги на основі їхньої історії оренди, що допомагає підвищити довіру на платформі;

– децентралізоване управління (DAO). CarBlock керується спільнотою користувачів через децентралізовану автономну організацію (DAO). Учасники DAO голосують за ключові рішення щодо розвитку платформи, такі як внесення змін до правил або розподіл винагород.

Переваги CarBlock:

– прозорість та безпека. Блокчейн забезпечує прозорість усіх транзакцій та зберігання даних, що підвищує довіру між учасниками;

– економія коштів. Відсутність посередників дозволяє знизити транзакційні витрати та пропонувати більш конкурентоспроможні ціни на оренду;

– гнучкість. Власники автомобілів можуть встановлювати власні ціни та умови оренди, а користувачі можуть вибрати автомобіль, який найкраще відповідає їхнім потребам;

– спільнота та стимули. Система репутації та DAO сприяють розвитку спільноти відповідальних користувачів та стимулюють активну участь.

Недоліки CarBlock:

– необхідність володіння криптовалютою CBC. Користувачам потрібна криптовалюта CBC для оренди автомобілів. Це може стати бар'єром для користувачів, які не знайомі з криптовалютою;

– обмежена доступність. Наразі CarBlock може мати меншу базу користувачів та меншу кількість доступних автомобілів порівняно з традиційними платформами каршерінгу;

– технічні складнощі. Використання блокчейну та криптовалюти може бути складним для деяких користувачів, які не мають технічних знань.

CarBlock пропонує інноваційну модель каршерінгу з потенціалом підвищення прозорості, зниження витрат та надання більшого контролю користувачам. Однак платформа ще знаходиться на ранній стадії розвитку, і їй потрібно буде подолати певні бар'єри для залучення ширшої аудиторії.

WiBLE – це децентралізована платформа спільного користування автомобілями, побудована на блокчейні AION. Вона пропонує альтернативу традиційним компаніям каршерінгу, надаючи користувачам низку переваг завдяки технології блокчейн та смарт-контрактів [25].

Особливості WiBLE:

– відсутність посередників: WiBLE дозволяє власникам автомобілів напряму здавати їх в оренду користувачам, без участі посередників. Це може знизити транзакційні витрати та пропонувати більш конкурентоспроможні ціни;

– використання криптовалюти AION: Для оренди автомобілів користувачі використовують криптовалюту AION. WiBLE може запровадити власну токенову систему для додаткових функцій та винагород;

– інтеграція з IoT: WiBLE може інтегруватися з пристроями Інтернету речей (IoT) для автоматичного доступу до автомобілів. Це дозволяє орендарям уникати відвідування офісів або зустрічей з орендодавцями;

– система лояльності: WiBLE може запровадити програму лояльності, яка винагородить активних користувачів знижками, пільгами або іншими перевагами.

Переваги WiBLE:

– зручність та швидкість. Смарт-контракти можуть автоматизувати процес оренди, роблячи його швидшим та зручнішим для користувачів;

– вигідні винагороди. Система лояльності може заохочувати користувачів частіше користуватися платформою;

– автоматизований доступ. Інтеграція з IoT дозволяє користувачам отримувати доступ до автомобілів без додаткових зусиль;

– прозорість та безпека. Блокчейн забезпечує прозорість транзакцій та зберігання даних, підвищуючи довіру між учасниками.

Недоліки WiBLE:

– необхідність володіння криптовалютою AION. Користувачам потрібна криптовалюта AION для оренди автомобілів. Це може бути бар'єром для користувачів, які не знайомі з криптовалютою;

– обмежена доступність. Наразі WiBLE може мати меншу базу користувачів та меншу кількість доступних автомобілів порівняно з традиційними платформами каршерінгу;

– залежність від AION. Платформа залежить від стійкості та масштабованості блокчейну AION. Якщо AION має проблеми, це може вплинути на роботу WiBLE.

WiBLE пропонує інноваційний підхід до каршерінгу, використовуючи смарт-контракти та блокчейн для підвищення зручності, безпеки та потенційно зниження витрат. Однак, як і інші децентралізовані платформи, WiBLE стикається з проблемами обмеженої доступності та необхідності використання криптовалюти.

WeGo – децентралізована платформа спільного користування автомобілями, побудована на блокчейні EOS. Вона відрізняється від інших подібних платформ тим, що пропонує можливість створювати локальні ринки каршерінгу.

Особливості WeGo:

– створення локальних ринків. WeGo дозволяє спільнотам створювати та керувати власними локальними ринками спільного користування автомобілями. Це дає змогу адаптувати умови оренди до місцевих потреб та реалій;

– використання криптовалюти EOS. Для оренди автомобілів користувачі використовують криптовалюту EOS. Локальні ринки можуть також випускати власні токени для додаткових функцій та винагород;

– децентралізоване управління (DAO). Кожен локальний ринок WeGo може керуватися спільнотою користувачів через DAO. Учасники голосують за ключові рішення, такі як встановлення цін, умови оренди та розподіл доходів;

– відкрита та прозора система. Завдяки блокчейну всі транзакції та дані зберігаються відкрито та прозоро. Це підвищує довіру між учасниками платформи.

Переваги WeGo:

– гнучкість та автономія. Локальні ринки можуть встановлювати власні правила та умови, що дозволяє краще відповідати місцевим потребам;

– спільнота та спільне управління. Модель DAO сприяє розвитку спільноти відповідальних користувачів та залучає їх до управління платформою;

– відкритість та прозорість. Блокчейн забезпечує простежуваність усієї інформації, підвищуючи довіру між учасниками.

– потенціал для інновацій. Локальні ринки можуть експериментувати з різними моделями оренди та винагородження.

Недоліки WeGo:

– необхідність володіння криптовалютою EOS. Користувачам потрібна криптовалюта EOS для оренди автомобілів. Це може бути бар'єром для користувачів, які не знайомі з криптовалютою;

– складність у налаштуванні. Створення та управління локальним ринком може бути складним завданням, що вимагає технічних знань та зусиль спільноти;

– обмежена доступність. Наразі WeGo може мати меншу базу користувачів та меншу кількість доступних автомобілів порівняно з традиційними платформами каршерінгу;

– технічні бар'єри. Користувачам потрібен певний рівень технічних знань для участі в DAO та взаємодії з платформою на основі блокчейну.

WeGo пропонує унікальний підхід до децентралізованого каршерінгу, надаючи спільнотам можливість створювати власні локальні ринки. Ця модель може бути корисною для регіонів з особливими потребами або для заохочення спільного користування автомобілями на місцевому рівні. Однак складність налаштування та технічні бар'єри можуть обмежити її поширення.

ShareRing – це децентралізована платформа спільного користування ресурсами, побудована на блокчейні Ethereum. Вона пропонує користувачам можливість ділитися не лише автомобілями, але й іншими ресурсами, такими як будинки, речі та послуги.

– різноманіття ресурсів. ShareRing дозволяє ділитися широким спектром ресурсів, роблячи платформу універсальною для різних потреб;

– глобальна платформа. ShareRing не має географічних обмежень, що дозволяє користувачам з усього світу ділитися ресурсами;

– система репутації. ShareRing використовує систему репутації, щоб заохочувати відповідальну поведінку учасників платформи;

– децентралізоване управління (DAO). ShareRing керується спільнотою користувачів через DAO. Учасники голосують за ключові рішення щодо розвитку платформи, такі як внесення змін до правил або розподіл винагород.

Переваги ShareRing:

– різноманіття. Платформа пропонує широкий спектр ресурсів для спільного користування, що робить її корисною для різних потреб;

– глобальний доступ. ShareRing не обмежений географічно, що дає доступ до більшої кількості ресурсів;

– спільнота та стимули. Система репутації та DAO сприяють розвитку відповідальної спільноти та стимулюють активну участь;

– прозорість та безпека. Блокчейн забезпечує прозорість та безпечність усіх транзакцій.

Недоліки ShareRing:

– необхідність володіння криптовалютою SHR. Користувачам потрібна криптовалюта SHR для оренди та здачі в оренду ресурсів. Це може бути бар'єром для користувачів, які не знайомі з криптовалютою;

– рання стадія розвитку. ShareRing все ще знаходиться на ранній стадії розвитку, тому деякі функції можуть бути недоступні або нестабільні;

– складність юзабіліті. Інтерфейс платформи може бути складним для користувачів, які не мають технічних знань;

– незначна поширеність. ShareRing має меншу базу користувачів порівняно з традиційними платформами спільного користування.

ShareRing пропонує інноваційний підхід до спільного користування ресурсами, поєднуючи різноманіття, доступність та децентралізоване управління. Однак платформа все ще знаходиться на ранній стадії розвитку, їй потрібно подолати бар'єри, пов'язані з криптовалютою та юзабіліті, щоб досягти ширшого прийняття [27].

Для аналізу компаній, які можуть використовувати смарт-контракти для систем спільного використання автомобілів, розглянемо кілька ключових факторів:

– анонімізація та захист даних;

Компанії, які високо цінують приватність та безпеку особистих даних своїх користувачів, можуть бути зацікавлені в використанні смарт-контрактів для забезпечення анонімізації даних. Вони можуть шукати рішення, які дозволяють зберігати дані в безпечному та анонімному форматі, щоб зменшити ризики порушення приватності та зловживання доступом до інформації.

– використання різноманітної валюти для оплати;

Компанії, які мають міжнародні клієнти або операції у країнах з різними валютами, можуть бути зацікавлені в розвитку систем, які дозволяють оплату послуг у різних валютах за допомогою смарт-контрактів. Це може полегшити операції та зменшити витрати на конвертацію валют для користувачів.

– використання приватних блокчейнів;

Компанії, які мають обмежену кількість учасників та потребують високого рівня конфіденційності та безпеки, можуть використовувати приватні блокчейни для реалізації своїх смарт-контрактів. Це дозволяє обмежити доступ до даних лише авторизованим сторонам та забезпечити відповідність з правовими вимогами.

– конфіденційність транзакцій.

Компанії, які шукають способи забезпечення конфіденційності та анонімності транзакцій, можуть використовувати смарт-контракти з механізмами анонімізації та шифрування даних. Це дозволяє уникнути розголошення конфіденційної інформації та забезпечити безпеку платежів.

Для більш зручного порівняння ключових факторів компаній, які можуть використовувати смарт-контракти для систем спільного використання автомобілів, можна створити таблицю порівняння. У цій таблиці (табл. 1.1) кожній компанії призначено оцінку щодо кожного з ключових факторів, таких як анонімізація, захист даних, різноманітність валют, використання приватних блокчейнів та конфіденційність транзакцій.

Таблиця 1.1 – порівняння ключових факторів компаній

Критерій	Компанія				
	CarBlock	WiBLE	WeGo	ShareRing	Данна розробка
Анонімізація та захист даних	Часткова	Часткова	Часткова	Часткова	Повна
Використання різноманітної валюти для оплати	Немає	Немає	Немає	Так	Так
Використання приватних блокчейнів	Немає	Немає	Немає	Немає	Так

Продовження таблиці 1.1

Конфіденційність транзакцій	Немає	Немає	Немає	Немає	Так

У рамках даного дослідження було проведено аналіз компаній, які використовують смарт-контракти для систем спільного використання автомобілів. На основі проведеного аналізу можна зробити наступні висновки:

Усі досліджені компанії, такі як CarBlock, WiBLE, WeGo та ShareRing, пропонують лише частковий рівень анонімізації та захисту даних. Це може вказувати на те, що вони можуть не забезпечувати достатньо високий рівень конфіденційності для користувачів. У той час, дана розробка пропонує повний захист та анонімізацію, що може бути важливим для користувачів з погляду конфіденційності та безпеки даних.

Тільки дана розробка та ShareRing пропонують можливість використання різних валют для оплати, що може бути зручно для користувачів, особливо якщо вони мають рахунки в різних валютах або використовують криптовалюту.

Данна розробка є єдиним учасником, який використовує приватний блокчейн. Це може забезпечити більший рівень конфіденційності та контролю над даними користувачів порівняно з іншими компаніями, які використовують загальнодоступні блокчейни.

Тільки дана розробка пропонує конфіденційність транзакцій, що може бути важливим аспектом для користувачів, які прагнуть зберегти конфіденційність своїх фінансових операцій [26].

Отже, дана розробка вирізняється серед інших компаній, пропонуючи повний захист та анонімізацію даних, можливість використання різних валют для оплати, використання приватних блокчейнів та забезпечення конфіденційності транзакцій. Ці аспекти можуть зробити дана розробку привабливою для користувачів, які прагнуть зберегти приватність та безпеку своїх даних і фінансових операцій.

1.6 Висновки та постановка задачі

У рамках даної роботи були розглянуті теоретичні засади створення захищеної системи спільного використання автомобілів на основі смарт-контрактів. Вивчення смарт-контрактів дозволило з'ясувати їхні основні принципи та можливості застосування в системах спільного використання автотранспорту. Також було проаналізовано механізми анонімізації даних користувачів та застосування криптографічних методів для забезпечення безпеки даних у таких системах.

Проведено аналіз популярних систем спільного використання автомобілів для виявлення потенційних проблем та можливостей для вдосконалення. На основі отриманих результатів можна сформулювати постановку задачі для подальших досліджень та розробки нової системи спільного використання автотранспорту на основі смарт-контрактів з використанням механізмів анонімізації та криптографічних методів для забезпечення безпеки даних.

У ході огляду теоретичного матеріалу сформульовані наступні завдання:

– розробити систему обміну даними між модулями системи;

– розробити загальний алгоритм роботи системи;

– розробити алгоритм реєстрації та авторизації користувачів з анонімізацією та шифруванням їхніх даних;

– розробити алгоритм роботи смарт-контракту та надання доступу до автомобілю.

Ці завдання спрямовані на створення ефективної та безпечної системи спільного використання автомобілів, яка забезпечить користувачам високий рівень конфіденційності та безпеки в їхніх транзакціях та взаємодії з системою.

2 РОЗРОБКА ЗАХИЩЕНОЇ СИСТЕМИ ДЛЯ СПІЛЬНОГО ВИКОРИСТАННЯ АВТОМОБІЛІВ НА ОСНОВІ СМАРТ КОНТРАКТІВ З МОЖЛИВІСТЮ АНОНІМІЗАЦІЇ ДАНИХ КОРИСТУВАЧІВ ДЛЯ УНЕМОЖЛИВЛЕННЯ ЇХНЬОГО ВІДСЛІДКОВУВАННЯ

В даному розділі буде здійснена розробка захищеної системи для спільного використання автомобілів на основі смарт контрактів з можливістю анонімізації даних користувачів для унеможливлення їхнього відслідковування, будуть розроблені алгоритми авторизації, реєстрації користувачів, також буде розроблено алгоритм надання доступу автомобіля та фіксування даних за допомогою смарт контрактів, також буде розроблена система обміну даними між модулями.

2.1 Особливості створення захищеної системи для спільного використання автомобілів на основі смарт контрактів з можливістю анонімізації даних користувачів для унеможливлення їхнього відслідковування

Розробка захищеної системи для спільного використання автомобілів на основі смарт-контрактів, з можливістю анонімізації даних користувачів для унеможливлення їхнього відслідковування, є важливим кроком у напрямку забезпечення приватності та безпеки в економіці спільного користування. Цей підхід вимагає інтеграції різних технологічних інновацій, включаючи блокчейн, смарт-контракти, технології анонімізації та захищені методи аутентифікації. Ось декілька ключових аспектів, які потрібно врахувати при створенні такої системи [28]:

1. використання блокчейна і смарт-контрактів: блокчейн забезпечує децентралізовану та незмінну платформу для запису транзакцій, що дозволяє зберігати дані безпечно та прозоро. Смарт-контракти автоматизують виконання угод на основі попередньо заданих умов, забезпечуючи надійність та ефективність процесів без необхідності посередників.

2. анонімізація даних: для забезпечення конфіденційності користувачів важливо впровадити механізми анонімізації та псевдонімізації даних. Це може включати використання технік, як-от змішування транзакцій або нульові знання доказів, для того, щоб забезпечити учасникам можливість здійснювати транзакції без розкриття своєї справжньої ідентичності.

3. безпека та приватність: необхідно розробити комплексні механізми захисту, які забезпечують безпеку системи від зовнішніх та внутрішніх загроз. Це включає захист від маніпуляцій з даними, атак на конфіденційність та безпеку інформації. Важливо також забезпечити захист від можливих вразливостей у смарт-контрактах

4. управління ключами та ідентифікація: забезпечення безпечного управління ключами є критично важливим для запобігання несанкціонованому доступу до даних. Водночас, система повинна включати ефективні методи ідентифікації та аутентифікації користувачів, які не порушують їхню приватність.

5. інтеграція з існуючими системами: для забезпечення широкого прийняття системи, важливо забезпечити її сумісність та інтеграцію з існуючими платформами для спільного використання автомобілів, платіжними системами та іншими сервісами.

Використання блокчейна і смарт-контрактів у системі для спільного використання автомобілів пропонує революційний підхід до управління транзакціями та взаємодій між учасниками. Ці технології не лише забезпечують високий рівень безпеки та прозорості, але й автоматизують виконання домовленостей, значно знижуючи потребу в посередниках. Ось детальніше про кожну з цих технологій та їхнє застосування:

Блокчейн — це розподілена база даних, яка забезпечує незмінність записів завдяки криптографічному ланцюгу блоків. Кожен блок містить пакет транзакцій, які підтверджені мережею через процес консенсусу. Це забезпечує високу ступінь безпеки та надійність, оскільки модифікувати або видалити інформацію без відома всієї мережі стає неможливим.

Застосування в системі спільного використання автомобілів:

– децентралізоване зберігання даних: дозволяє зберігати дані про користувачів, транзакції та стан автомобілів у безпечному та незмінному вигляді.

– прозорість: усі учасники мають доступ до інформації про транзакції та статус автомобілів, забезпечуючи прозорість угод.

– надійність: висока стійкість до збоїв та атак, оскільки немає єдиного центрального вузла, який можна було б скомпрометувати.

Смарт-контракти — це програми, які автоматично виконують умови контрактів на блокчейні, коли виконані певні умови. Це знижує потребу в посередниках, оскільки код гарантує виконання умов без зовнішнього втручання.

Застосування в системі спільного використання автомобілів:

– автоматизація угод: автоматичне розблокування доступу до автомобіля після виконання умов оплати без потреби вручну перевіряти кожну транзакцію.

– гнучкість умов: можливість програмування різноманітних умов оренди, включаючи тарифи, час оренди, штрафи за запізнення та інше.

Анонімізація даних є ключовим компонентом у захисті приватності користувачів в системах, які використовують блокчейн, особливо в секторі спільного використання автомобілів. Ця практика включає процеси та технології, які дозволяють приховати особистість користувачів, тим самим забезпечуючи конфіденційність їхніх даних. Нижче наведено кілька ключових підходів до анонімізації даних у таких системах:

Псевдонімізація замінює особисті ідентифікатори інформації на псевдоніми, що не дозволяють безпосередньо ідентифікувати особу без використання додаткової інформації. У контексті блокчейна це може бути реалізовано через використання унікальних криптографічних ключів або адрес, які служать як ідентифікатори для транзакцій, не розкриваючи справжні ідентичності учасників.

Нульові знання докази (Zero-Knowledge Proofs, ZKP) дозволяють одній стороні довести іншій, що вони знають певну інформацію, не розкриваючи самої інформації. У контексті спільного використання автомобілів, це може бути використано для підтвердження віку, водійського стажу або інших вимог без необхідності розкриття особистих даних [29].

Змішування транзакцій є методом, який дозволяє об'єднати кілька транзакцій від різних користувачів у одну, з метою ускладнення відстеження окремих платежів до конкретних осіб. Це знижує ймовірність того, що транзакції можна буде прив'язати до конкретних користувачів [30].

Конфіденційні транзакції дозволяють приховувати суму транзакції, що пересилається через блокчейн. Це може бути реалізовано за допомогою криптографічних методів, таких як педерсенівські зобов'язання, які дозволяють учасникам підтвердити, що транзакція не порушує збалансованості активів без розкриття фактичної суми.

2.2 Розробка загального алгоритму роботи ситеми та ситеми обміму даними

Для розробки загального алгоритму роботи системи і системи обміну даними, ми можемо виокремити наступні ключові елементи:

1. система керування автомобілями – це центральний компонент, що керує доступом до автомобілів, відстеженням їхнього стану та управлінням розкладами використання.
2. інфраструктура керування автомобілями – фізичні та програмні засоби, необхідні для підтримки системи керування автомобілями, включаючи датчики на автомобілях, мережу зв'язку та сервери.
3. база даних з публічними даними – база даних, яка містить інформацію, доступну для всіх користувачів системи, наприклад, доступність автомобілів.
4. база даних з приватними даними – база даних, яка зберігає конфіденційну інформацію, таку як особисті дані користувачів та історію поїздок.
5. користувач – особа, яка використовує систему для оренди та управління автомобілем.
6. система спільного використання автомобілів – платформа, яка дозволяє користувачам знаходити, резервувати та оплачувати використання автомобілів.

7. система анонімізації та шифрування даних – інструменти та методології, які забезпечують захист персональних даних користувачів і їхню анонімність.

8. платформа керування платежами – система для обробки фінансових транзакцій, пов'язаних з орендою автомобілів.

9. блокчейн – технологія розподіленого реєстру, яка використовується для забезпечення безпеки транзакцій та довіри між учасниками.

для розробки загального алгоритму роботи системи можна скористатися наступною послідовністю кроків:

1. користувач використовує систему спільного використання автомобілів для пошуку та резервування доступного автомобіля.

2. дані користувача анонімізуються та шифруються для забезпечення конфіденційності.

3. система керування автомобілями надає доступ до автомобіля після підтвердження бронювання та проведення платежу.

4. інфраструктура керування автомобілями відстежує стан автомобіля під час використання та збирає необхідні дані для публічної бази даних.

5. після завершення використання автомобіля дані про поїздку зберігаються в приватній базі даних.

6. фінансові транзакції обробляються через платформу керування платежами, і дані про транзакції зберігаються в блокчейні для забезпечення прозорості та незмінності.

Створимо схему обміну даними між різними модулями на основі описаних вище елеменітів (рис.2.1).

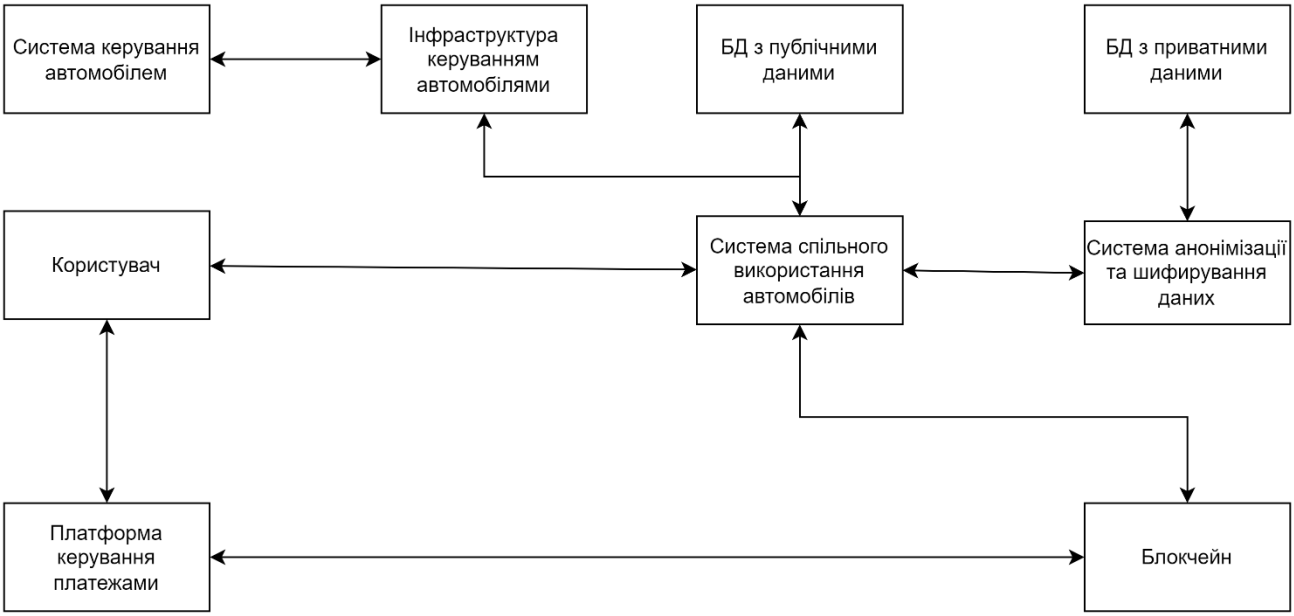


Рисунок 2.1 – Схема обміну даними в системі

Алгоритм роботи захищеної системи спільного використання автомобілів, заснованої на смарт-контрактах із можливістю анонімізації даних користувачів, можна розглянути як послідовність етапів:

Реєстрація користувача:

– користувач подає заявку на реєстрацію через інтерфейс системи.

– система збирає мінімально необхідні дані для встановлення ідентичності.

– здійснюється верифікація користувача з використанням методів біометричної аутентифікації або kyc (know your customer).

– інформація користувача анонімізується та зберігається у вигляді псевдоніма в блокчейні.

Створення смарт-контракту:

– користувач обирає необхідний автомобіль та умови використання через систему.

– на основі вибраних параметрів автоматично генерується смарт-контракт.

– смарт-контракт зберігається у блокчейні і стає активним після підтвердження оплати.

Оплата за використання:

– користувач виконує оплату через інтегровану платіжну систему.

– платіжна система забезпечує анонімізацію транзакцій для унеможливлення відстеження.

– оплата фіксується в блокчейні та активує смарт-контракт для надання доступу до автомобіля.

Доступ до автомобіля:

– смарт-контракт комунікує з системою керування автомобілями для надання доступу до забронюваного автомобіля.

– користувач отримує доступ до автомобіля за допомогою мобільного додатку або rfid-тега.

Використання автомобіля:

– дані про стан та використання автомобіля збираються інфраструктурою у реальному часі.

– анонімізовані дані про поїздку тимчасово зберігаються для адміністративних цілей та можливих страхових випадків.

Завершення використання:

– по завершенню користування автомобілем дані автоматично архівуються та анонімізуються згідно з політикою конфіденційності.

– смарт-контракт завершується та фіксує завершення угоди у блокчейні..

Цей алгоритм може бути адаптований та доповнений відповідно до конкретних вимог та регуляцій. Важливо підкреслити, що кожен крок включає заходи щодо забезпечення безпеки та конфіденційності даних.

2.3 Розробка алгориму реєстрації та авторизації користувачів

В еру цифрових технологій, коли зростання кількості онлайн-сервісів вимагає все більшої уваги до захисту особистої інформації, алгоритми реєстрації та авторизації користувачів набувають стратегічного значення. Ці алгоритми не тільки забезпечують простий і безперешкодний вхід для користувачів у системи,

але й відіграють критичну роль у забезпеченні конфіденційності, інтеграції та безпеки [31].

Реєстрація користувачів є першим кроком у встановленні взаємин між користувачем та сервісом, під час якого збираються і верифікуються особисті дані, створюється унікальний ідентифікатор та ініціюються профілі користувачів. Правильно структурований процес реєстрації може знизити ризик неправомірного доступу та зловживань, а також підвищити довіру користувачів до платформи.

Розробимо алгоритм реєстрації користувачів в системі (рис. 2.2).

1. Початок реєстрації в системі:
   – Користувач ініціює реєстрацію через інтерфейс користувача.
2. Введення користувачем даних для реєстрації:
   – Користувач вводить особисті дані, необхідні для створення облікового запису.
3. Шифрування даних:
   – Введені дані шифруються для забезпечення безпеки під час передачі.
4. Відправлення даних:
   – Зашифровані дані відправляються на сервер системи.
5. Відправлення повідомлення про недостовірність даних (при потребі):
   – Якщо дані не пройшли попередню валідацію на клієнтській стороні, система відправляє повідомлення про недостовірність даних.
6. Отримання даних системою:
   – Сервер системи отримує дані.
7. Дешифрування даних:
   – Сервер системи дешифрує отримані дані.
8. Парсинг та перевірка даних:
   – Сервер аналізує дані та перевіряє їх на валідність та повноту.
9. Перевірка успішна:
   – Якщо дані відповідають усім вимогам, процес продовжується.

Рисунок 2.2 – Алгоритм реєстрації користувачів

10. Створення унікального публічного ідентифікатора користувача:
    - Система генерує унікальний публічний ідентифікатор для користувача.
11. Записування публічних даних в базу даних:
    - Публічні дані користувача записуються у публічну частину бази даних.
12. Шифрування приватних даних:
    - Приватні дані користувача шифруються перед зберіганням.
13. Записування приватних даних в приватну базу даних:
    - Зашифровані приватні дані зберігаються у приватній частині бази даних.
14. Відправлення повідомлення про успішну реєстрацію:
    - Система відправляє користувачу повідомлення про успішну реєстрацію.
15. Кінець реєстрації:Реєстраційний процес завершено.

Авторизація користувачів – це процес, що дозволяє визначити, чи має особа право доступу до певних даних або ресурсів. Алгоритм авторизації забезпечує, що кожен запит на доступ відбувається відповідно до політики безпеки та що всі сесії користувачів можуть бути належно моніторовані та контрольовані. Ефективна авторизація допомагає запобігти несанкціонованому доступу та можливим зламам, важливо з точки зору захисту приватної інформації користувачів та забезпечення безпечного користувацького досвіду.

Розробимо алгоритм авторизації користувачів в системі (рис. 2.3).

1. Початок авторизації в системі:
    - Користувач ініціює процес авторизації в системі.
2. Введення користувачем даних для авторизації:
    - Користувач вводить дані для входу в систему (наприклад, логін та пароль).
3. Шифрування даних:
    - Введені дані шифруються для забезпечення безпеки під час передачі.

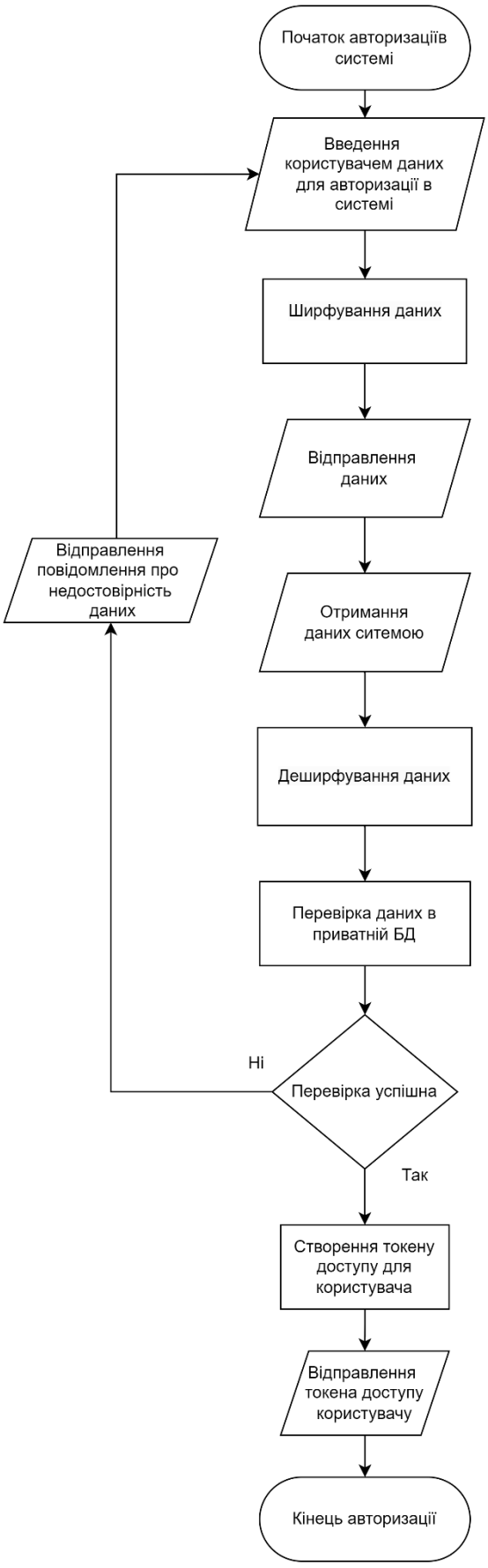


Рисунок 2.3 – Алгоритм авторизації користувачів

4. Відправлення даних:
   – Зашифровані дані відправляються на сервер системи.
5. Відправлення повідомлення про недостовірність даних (при потребі):
   – Якщо дані не відповідають вимогам, відправляється повідомлення про помилку.
6. Отримання даних системою:
   – Сервер отримує дані для авторизації.
7. Дешифрування даних:
   – Сервер дешифрує отримані дані.
8. Перевірка даних в приватній базі даних:
   – Перевіряється відповідність введених даних з даними в приватній базі даних.
9. Перевірка успішна:
   – Якщо дані коректні, доступ дозволено.
10. Створення токену доступу для користувача:
   – Генерується токен, який дозволить користувачу доступ до ресурсів системи.
11. Відправлення токену доступу користувачу:
   – Токен відправляється користувачу.
12. Кінець авторизації:
   – Процес авторизації завершено.

Отже, ретельно розроблені та впроваджені алгоритми реєстрації та авторизації є невід'ємними для створення надійної, безпечної та приватної цифрової екосистеми.

2.4 Розробка алгоритму роботи системи надання доступу до автомобіля

Сучасні ринкові умови та зростаючий попит на спільне використання транспортних засобів вимагають новаторських технологічних рішень, які б спрощували процеси і водночас забезпечували б надійність та безпеку. Відповідь

на ці виклики може бути знайдена в блокчейн технологіях та смарт-контрактах, які разом формують фундамент для автоматизації та оптимізації систем спільного використання автомобілів. Алгоритм надання доступу до автомобіля, заснований на смарт-контрактах, є одним із ключових елементів цієї екосистеми.

Цей алгоритм включає в себе кроки, що забезпечують не лише ефективність взаємодії між користувачем і сервісом, але й гарантують високий рівень безпеки та приватності даних. Від створення платежу до моніторингу використання автомобіля та закриття смарт-контракту — кожен крок оптимізований для забезпечення надійного та зручного доступу до автомобілів. Така система дозволяє користувачам легко бронювати, отримувати доступ та використовувати автомобілі без зайвих формальностей, при цьому знижуючи ризики для постачальників послуг і створюючи надійну облікову запису всіх транзакцій в незмінному блокчейні.

Алгоритм надання доступу до автомобіля (рис .2.4):

1. Початок:
   - користувач ініціює процес надання доступу до автомобіля через систему.
2. Створення платежу користувачем:
   - користувач вводить в систему деталі платежу, включаючи час і дату використання автомобіля.
3. Відправлення платежу до платформи обробки платежів:
   - платіж відправляється до відповідної платіжної системи для обробки.
4. Відправлення повідомлення невдалій платіж (якщо необхідно):
   - якщо платіж не може бути оброблений, користувач отримує повідомлення про помилку.
5. Отримання результату платежа:
   - система отримує підтвердження про успішний або невдалий результат платежа.
6. Платіж успішний:
   - перевірка, чи платіж був успішно оброблений.

7. Створення смарт-контракта з ідентифікатором користувача:

– у разі успішного платежу, створюється смарт-контракт, який містить інформацію про користувача та умови доступу до автомобіля.

8. Внесення даних про автомобіль та час його використання в смарт-контракт:

– деталі про автомобіль та час використання заносяться в смарт-контракт.

9. Внесення запису в блокчейн:

– смарт-контракт реєструється в блокчейні, забезпечуючи незмінність та прозорість угоди.

10. Надання користувачу доступу до автомобіля:

– користувачу надається доступ до автомобіля згідно з умовами смарт-контракту.

11. Моніторинг використання автомобіля:

– система відстежує стан використання автомобіля, наприклад, чи автомобіль використовується у визначений час.

12. Час використання вийшов?:

– перевірка, чи закінчився час використання автомобіля згідно з смарт-контрактом.

13. Завершення оренди:

– якщо час оренди вийшов, процедура завершення оренди ініціюється.

14. Закриття смарт-контракта:

– смарт-контракт закривається, і ця інформація реєструється в блокчейні.

15. Блокування автомобіля:

– автомобіль блокується для подальшого використання до наступної активації через новий смарт-контракт.

16. Кінець:

– процес надання доступу до автомобіля завершено.

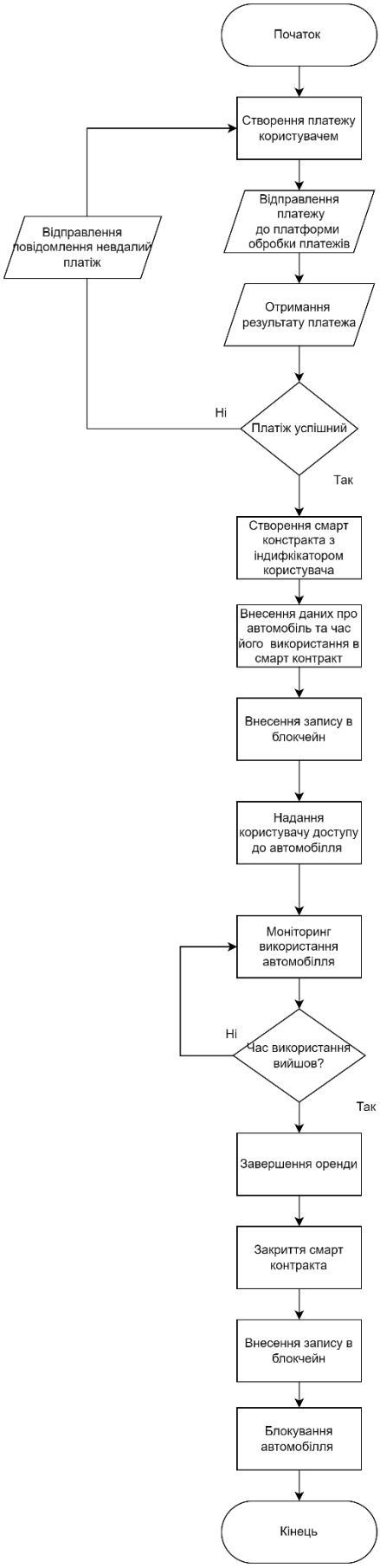


Рисунок 2.4 - Алгоритм надання доступу до автомобіля

Цей алгоритм використовує можливості блокчейну для забезпечення безпечного та прозорого процесу оренди автомобілів. Він дозволяє не тільки спростити процеси оренди та доступу до автомобілів, але й забезпечує вищий рівень безпеки та зменшує можливість шахрайства чи помилок.

2.5 Висновки до розділу.

У ході проведеного дослідження було розроблено та детально описано алгоритми реєстрації користувачів, авторизації та надання доступу до автомобілів в рамках системи спільного використання на основі смарт-контрактів. Ретельно сплановані етапи кожного з процесів відповідають високим стандартам безпеки та приватності, які є актуальними для сучасного цифрового суспільства.

Алгоритми були структуровані таким чином, щоб забезпечити прозору та надійну взаємодію між користувачами та сервісами, водночас гарантуючи анонімність та захист особистих даних. Використання блокчейн-технологій та смарт-контрактів дозволяє створити децентралізовану систему, що підвищує довіру користувачів та знижує ризики пов'язані з централізованим зберіганням даних.

Запропоновані алгоритми можуть бути використані як основа для розробки комплексних програмних рішень для компаній, що надають послуги спільного використання автомобілів, що, в свою чергу, може сприяти збільшенню ефективності галузі, зниженню витрат та підвищенню користувацького досвіду.

Результати дослідження підтверджують, що інтеграція блокчейну та смарт-контрактів у системи спільного використання автомобілів має значний потенціал та може слугувати надійним інструментом для розвитку інноваційних технологічних рішень у сфері транспортних послуг.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

В даному розділі буде здійснено обгрунтування вибіру мови програмуваня для реалізації, після чого буде здійснена програмна реалізації системи смарт контрактів та блокчейну, після чого буде здійснена програмна реалізація системи анонімізації користувачів та програмна реалізціая API для спільного використання автомобілів, після реалізації всіх частин системи буде здійснено иестування реалізованої системи.

3.1 Вибір мови програмуваня для реалізації

цьому підрозділі детально розглянуто вибір мови програмування Python для реалізації захищеної системи спільного використання автомобілів, що базується на смарт-контрактах з можливістю анонімізації даних користувачів. Рішення щодо вибору мови програмування було прийнято на основі аналізу різних критеріїв, що включають простоту вивчення та використання, підтримку блокчейн-технологій та смарт-контрактів, можливості для реалізації криптографічних операцій, наявність активного співтовариства та документації, а також масштабованість і продуктивність [33].

Python був обраний як основна мова програмування з кількох важливих причин. Перш за все, Python вирізняється своєю простотою та читабельністю. Його інтуїтивно зрозумілий синтаксис сприяє швидкому освоєнню та ефективному розробленню програмного забезпечення. Це особливо важливо для швидкого створення та тестування прототипів, що є невід'ємною частиною будь-якого інноваційного проекту. Завдяки своїй простоті, Python є популярним вибором у багатьох освітніх та науково-дослідних установах, що підтверджує його доступність для новачків та досвідчених програмістів [36].

Ще однією значущою перевагою Python є його потужні можливості для інтеграції з технологіями блокчейну та смарт-контрактами. Бібліотека web3.py забезпечує зручний інтерфейс для взаємодії з блокчейном Ethereum, дозволяючи легко створювати, розгортати та взаємодіяти зі смарт-контрактами. Окрім цього,

існують такі інструменти, як eth-brownie та truffle, які можна використовувати для управління контрактами та їх тестування. Це дозволяє розробникам зосередитися на логіці бізнес-процесів, а не на технічних деталях інтеграції.

Python також володіє потужними криптографічними бібліотеками, такими як PyCryptodome та cryptography, які підтримують широкий спектр криптографічних алгоритмів. Це особливо важливо для реалізації функцій анонімізації даних користувачів, таких як шифрування та підписування даних. Ці бібліотеки дозволяють легко реалізувати необхідні криптографічні операції, забезпечуючи високий рівень безпеки даних користувачів.

Активне співтовариство розробників є ще однією вагомою перевагою Python. Велика кількість розробників, що використовують цю мову, створює умови для швидкого вирішення проблем завдяки підтримці з боку спільноти. Крім того, добре написана та широко доступна документація для більшості бібліотек та інструментів Python значно полегшує процес розробки, дозволяючи швидко знаходити необхідну інформацію та приклади використання.

Нарешті, Python забезпечує достатню продуктивність та можливість масштабування для реалізації більшості завдань, пов’язаних з обробкою транзакцій та взаємодією зі смарт-контрактами. Хоча Python може бути повільнішим порівняно з деякими іншими мовами програмування, його продуктивності достатньо для даного проекту. Використання асинхронного програмування та бібліотек, таких як asyncio, дозволяє досягти високої продуктивності та ефективності в обробці запитів, що є критично важливим для систем, які мають забезпечувати швидкий відгук та обробку великої кількості транзакцій у реальному часі [37].

Альтернативи Python, такі як JavaScript (Node.js), Go та Rust, також мають свої переваги. JavaScript (Node.js) вирізняється високою продуктивністю та підтримкою асинхронного програмування, проте має складніший синтаксис та менше криптографічних бібліотек. Go пропонує високу продуктивність та хорошу підтримку паралелізму, але має менше готових бібліотек для роботи з блокчейном. Rust забезпечує високу продуктивність та безпечність пам'яті, але має високий

поріг входу і менш зручна для швидкого прототипування. Після порівняння всіх варіантів Python був обраний найбільш підходящою мовою програмування для даного проекту [39].

Таким чином, мову програмування Python було обрано для реалізації захищеної системи спільного використання автомобілів на основі смарт-контрактів з можливістю анонімізації даних користувачів. Його простота, потужні бібліотеки та активне співтовариство роблять його оптимальним вибором для цього проекту, забезпечуючи швидку розробку, високий рівень безпеки та ефективну взаємодію з блокчейном.

3.2 Програмна реалізації системи смарт контрактів та блокчейну

Для реалізації смарт-контрактів було обрано платформу Ethereum. Мова програмування Solidity використовується для написання смарт-контрактів на цій платформі. Бібліотека web3.py забезпечує інтеграцію з блокчейном Ethereum у Python, що дозволяє взаємодіяти зі смарт-контрактами, викликати їх методи, відправляти транзакції та отримувати дані з блокчейну.

Смарт-контракт забезпечує реєстрацію користувачів, зберігаючи їхню інформацію в блокчейні.

```
pragma solidity ^0.8.0;

contract CarSharing {
   struct User {
      string name;
      string email;
   }

   mapping(address => User) public users;

   function registerUser(string memory _name, string memory _email) public {
      users[msg.sender] = User(_name, _email);
   }
}
```

Функція для бронювання автомобілів дозволяє користувачам бронювати доступні автомобілі.

```
pragma solidity ^0.8.0;

contract CarSharing {
   struct Car {
```

```
    uint id;
    string model;
    bool isAvailable;
  }
  struct Booking {
    address user;
    uint carId;
    uint startTime;
    uint endTime;
  }
  Car[] public cars;
  Booking[] public bookings;
  function addCar(string memory _model) public {
    cars.push(Car(cars.length, _model, true));
  }
  function bookCar(uint _carId, uint _startTime, uint _endTime) public {
    require(cars[_carId].isAvailable, "Car is not available");
    bookings.push(Booking(msg.sender, _carId, _startTime, _endTime));
    cars[_carId].isAvailable = false;
  }
}
```

Функція для здійснення платежів забезпечує транзакції між користувачами та власниками автомобілів

```
pragma solidity ^0.8.0;
contract CarSharing {
  struct Payment {
    address from;
    address to;
    uint amount;
  }
  Payment[] public payments;
  function makePayment(address _to) public payable {
    require(msg.value > 0, "Payment must be greater than 0");
    payments.push(Payment(msg.sender, _to, msg.value));
    payable(_to).transfer(msg.value);
  }
}
```

Для підключення до блокчейну Ethereum використовується бібліотека web3.py.

```
from web3 import Web3
# Підключення до локального вузла Ethereum
web3 = Web3(Web3.HTTPProvider('http://127.0.0.1:8545'))
# Перевірка підключення
print(web3.isConnected())
# ABI та адреса смарт-контракту
abi = '[ABI контракту]'
contract_address = '0xYourContractAddress'
# Ініціалізація контракту
contract = web3.eth.contract(address=contract_address, abi=abi)
```

Для взаємодії зі смарт-контрактами можна викликати їх методи, використовуючи web3.py

```
        tx_hash = contract.functions.registerUser('John Doe', 'john@example.com').transact({'from':
web3.eth.accounts[0]})
        # Очікування підтвердження транзакції
        receipt = web3.eth.waitForTransactionReceipt(tx_hash)
        print("User registered:", receipt)
```

Безпека та анонімність користувачів досягаються шляхом використання криптографічних алгоритмів та технологій блокчейну

```
        from Crypto.Cipher import AES
        from Crypto.Random import get_random_bytes
        def encrypt_data(data, key):
           cipher = AES.new(key, AES.MODE_EAX)
           nonce = cipher.nonce
           ciphertext, tag = cipher.encrypt_and_digest(data.encode('utf-8'))
           return nonce, ciphertext, tag
def decrypt_data(nonce, ciphertext, tag, key):
           cipher = AES.new(key, AES.MODE_EAX, nonce=nonce)
           data = cipher.decrypt_and_verify(ciphertext, tag)
           return data.decode('utf-8')
        key = get_random_bytes(16)
        nonce, ciphertext, tag = encrypt_data('Confidential Data', key)
        print("Encrypted data:", ciphertext)

        decrypted_data = decrypt_data(nonce, ciphertext, tag, key)
        print("Decrypted data:", decrypted_data)
```

Програмна реалізація системи смарт-контрактів та блокчейну включає створення, розгортання та інтеграцію смарт-контрактів з блокчейном Ethereum, забезпечуючи безпеку та анонімність користувачів. Використання мови програмування Solidity для створення смарт-контрактів та бібліотеки web3.py для інтеграції з блокчейном у Python дозволяє зручно та ефективно розробляти систему. Завдяки реалізації безпечних та анонімних транзакцій, система забезпечує високий рівень безпеки та конфіденційності даних користувачів, що є критично важливим для спільного використання автомобілів на основі смарт-контрактів.

3.3 Програмна реалізація системи анонімізації користувачів

онімізація користувачів є критично важливим аспектом для забезпечення конфіденційності та безпеки даних. У цьому контексті використовуються різні криптографічні методи для захисту особистих даних користувачів.

Першим кроком у процесі анонімізації є використання псевдонімів замість реальних імен користувачів. Це допомагає захистити їхню особисту інформацію та

зменшити ризик відстеження. Псевдоніми можуть бути згенеровані випадковим чином або обрані користувачами.

```
import random
import string
def generate_pseudonym(length=10):
    characters = string.ascii_letters + string.digits
    pseudonym = ''.join(random.choice(characters) for _ in range(length))
    return pseudonym
# Приклад використання
pseudonym = generate_pseudonym()
print("Generated pseudonym:", pseudonym)
```

Для забезпечення додаткового рівня безпеки особисті дані користувачів можуть бути зашифровані перед їх зберіганням у блокчейні. Використання асиметричного шифрування дозволяє захистити дані від несанкціонованого доступу. У Python можна використовувати бібліотеку cryptography для реалізації шифрування та дешифрування даних:

```
from cryptography.hazmat.primitives.asymmetric import rsa
from cryptography.hazmat.primitives import serialization
from cryptography.hazmat.primitives.asymmetric import padding
from cryptography.hazmat.primitives import hashes
private_key = rsa.generate_private_key(public_exponent=65537, key_size=2048)
public_key = private_key.public_key()
# Серіалізація ключів
pem_private_key = private_key.private_bytes(
    encoding=serialization.Encoding.PEM,
    format=serialization.PrivateFormat.PKCS8,
    encryption_algorithm=serialization.NoEncryption()
)
pem_public_key = public_key.public_bytes(
    encoding=serialization.Encoding.PEM,
    format=serialization.PublicFormat.SubjectPublicKeyInfo
)
def encrypt_data(data, public_key):
    ciphertext = public_key.encrypt(
        data.encode('utf-8'),
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
```

```
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    return ciphertext
def decrypt_data(ciphertext, private_key):
    decrypted_data = private_key.decrypt(
        ciphertext,
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    return decrypted_data.decode('utf-8')
data = "Sensitive User Data"
ciphertext = encrypt_data(data, public_key)
print("Encrypted data:", ciphertext)
decrypted_data = decrypt_data(ciphertext, private_key)
print("Decrypted data:", decrypted_data)
```

Для забезпечення додаткового рівня анонімності транзакції користувачів можуть бути змішані з іншими транзакціями, що ускладнює їхнє відстеження. Це досягається за допомогою використання змішувальних служб (mixing services) або протоколів, які групують та перемішують транзакції від різних користувачів.

```
import random

def mix_transactions(transactions):
    mixed_transactions = transactions[:]
    random.shuffle(mixed_transactions)
    return mixed_transactions
```

Для забезпечення ще вищого рівня анонімності можна використовувати технологію ZK-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge). ZK-SNARKs дозволяють підтверджувати правдивість твердження без розкриття самого твердження або додаткової інформації. Це дозволяє здійснювати транзакції без розкриття особистих даних користувачів.

Для реалізації ZK-SNARKs у Python можна використовувати бібліотеку py-snarks або інтеграцію з відповідними блокчейн-платформами, які підтримують ZK-SNARKs.

```
pragma solidity ^0.8.0;
import "https://github.com/scipr-lab/libsnark/blob/master/contracts/ZkSnarkVerifier.sol";
contract CarSharingWithZKSnarks {
   // Контракт, який використовує ZK-SNARKs для анонімізації транзакцій
   function verifyTransaction(
      uint[2] memory a,
      uint[2][2] memory b,
      uint[2] memory c,
      uint[1] memory input
   ) public returns (bool) {
      return ZkSnarkVerifier.verifyTx(a, b, c, input);
   }
}
```

Для використання цієї функціональності необхідно підготувати доказ (proof) за допомогою відповідних інструментів та передати його у смарт-контракт для верифікації.

Програмна реалізація системи анонімізації користувачів включає використання псевдонімів, шифрування особистих даних, змішування транзакцій та застосування технології ZK-SNARKs. Ці методи дозволяють забезпечити високий рівень конфіденційності та безпеки даних користувачів у системі спільного використання автомобілів на основі смарт-контрактів. Завдяки цим підходам система може гарантувати захист особистої інформації користувачів та унеможливити їхнє відстеження.

3.4 Програмна реалізціая API для спільного використання автомобілів

API (Application Programming Interface) дозволяє користувачам реєструватися, бронювати автомобілі, здійснювати платежі та переглядати доступні автомобілі через стандартні HTTP-запити.

Для реалізації API обрано фреймворк Flask, який є легким і зручним у використанні веб-фреймворком для Python. Flask дозволяє швидко створювати RESTful API з мінімальними зусиллями. Бібліотека web3.py використовується для взаємодії зі смарт-контрактами на блокчейні Ethereum.

API забезпечує кілька основних кінцевих точок для взаємодії з системою:

– реєстрація користувача: дозволяє користувачам реєструватися в системі.

– отримання списку доступних автомобілів: повертає список автомобілів, які доступні для бронювання.

– бронювання автомобіля: дозволяє користувачам бронювати автомобілі.

– здійснення платежу: забезпечує можливість здійснення платежів між користувачами.

Ініціалізація Flask та підключення до блокчейну.

```
from flask import Flask, request, jsonify
from web3 import Web3
app = Flask(__name__)
# Підключення до локального вузла Ethereum
web3 = Web3(Web3.HTTPProvider('http://127.0.0.1:8545'))
# Перевірка підключення
if not web3.isConnected():
   print("Не вдалося підключитися до Ethereum")
# ABI та адреса смарт-контракту
abi = '[ABI контракту]'
contract_address = '0xYourContractAddress'
# Ініціалізація контракту
contract = web3.eth.contract(address=contract_address, abi=abi)
```

Кінцева точка для реєстрації користувача

```
@app.route('/register', methods=['POST'])
def register_user():
   data = request.json
   name = data['name']
   email = data['email']
   # Виклик методу registerUser смарт-контракту
   tx_hash = contract.functions.registerUser(name, email).transact({'from': web3.eth.accounts[0]})
   receipt = web3.eth.waitForTransactionReceipt(tx_hash)
   return jsonify({"message": "User registered", "receipt": receipt})
```

Кінцева точка для отримання списку доступних автомобілів:

```
@app.route('/cars', methods=['GET'])
def get_available_cars():
   cars = []
   for i in range(contract.functions.getCarsCount().call()):
      car = contract.functions.cars(i).call()
      if car[2]:  # Перевірка, чи автомобіль доступний
         cars.append({"id": car[0], "model": car[1]})
   return jsonify({"available_cars": cars})
```

Кінцева точка для бронювання автомобіля:

```
@app.route('/book', methods=['POST'])
def book_car():
   data = request.json
   car_id = data['car_id']
   start_time = data['start_time']
```

```
    end_time = data['end_time']
    # Виклик методу bookCar смарт-контракту
    tx_hash = contract.functions.bookCar(car_id, start_time, end_time).transact({'from': web3.eth.accounts[0]})
    receipt = web3.eth.waitForTransactionReceipt(tx_hash)

    return jsonify({"message": "Car booked", "receipt": receipt})
```

Кінцева точка для здійснення платежу

```
@app.route('/pay', methods=['POST'])
def make_payment():
    data = request.json
    to_address = data['to']
    amount = data['amount']
    # Вклик методу makePayment смарт-контракту
    tx_hash = contract.functions.makePayment(to_address).transact({
        'from': web3.eth.accounts[0],
        'value': web3.toWei(amount, 'ether')
    })
    receipt = web3.eth.waitForTransactionReceipt(tx_hash)
    return jsonify({"message": "Payment made", "receipt": receipt})
```

Програмна реалізація API для спільного використання автомобілів забезпечує зручний та безпечний спосіб взаємодії користувачів із системою на основі смарт-контрактів. Використання Flask для створення RESTful API дозволяє легко реалізувати функціональність реєстрації, бронювання автомобілів, здійснення платежів та отримання доступної інформації про автомобілі. Інтеграція з блокчейном Ethereum за допомогою бібліотеки web3.py дозволяє забезпечити прозорість та безпеку операцій, що є критично важливим для успішного функціонування системи спільного використання автомобілів.

3.5 Тестування реалізованої системи

Тестування є ключовим етапом для забезпечення надійності, безпеки та коректності функціонування системи.

Для початку роботи користувачу потрібно, щоб він спочатку ввійшов в систему на сторінці входу (рис. 3.1).

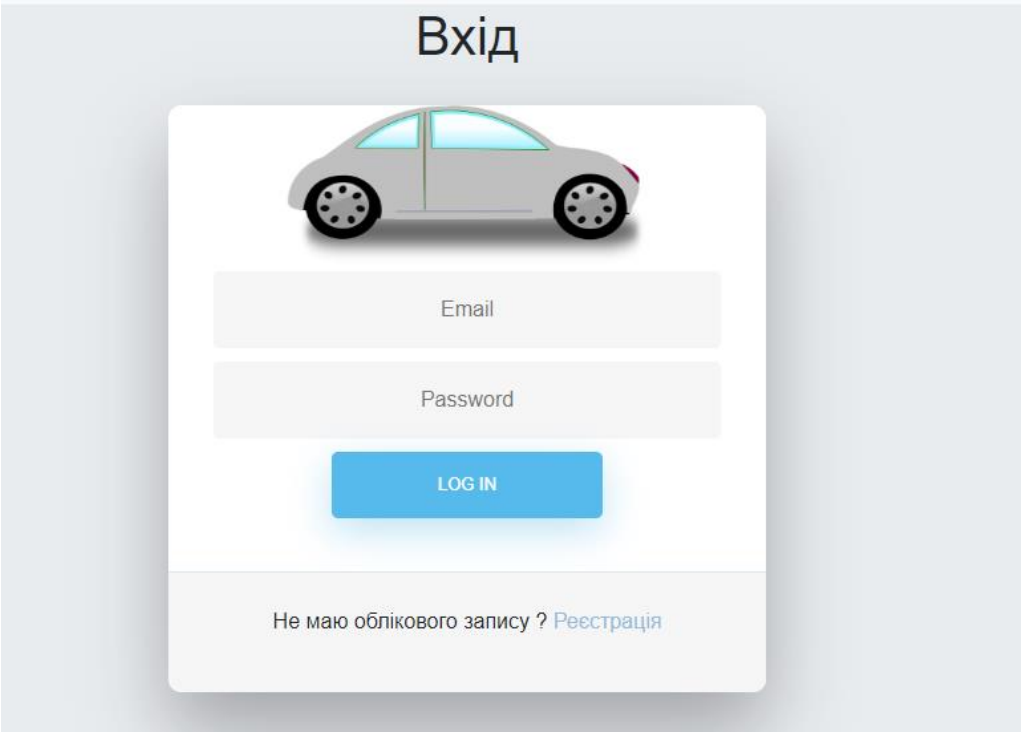


Рисунок 3.1 – Сторінка входу

Якщо в користувача немає облікового запису він може зареєструватися на сторінці реєстрації (рис 3.2).

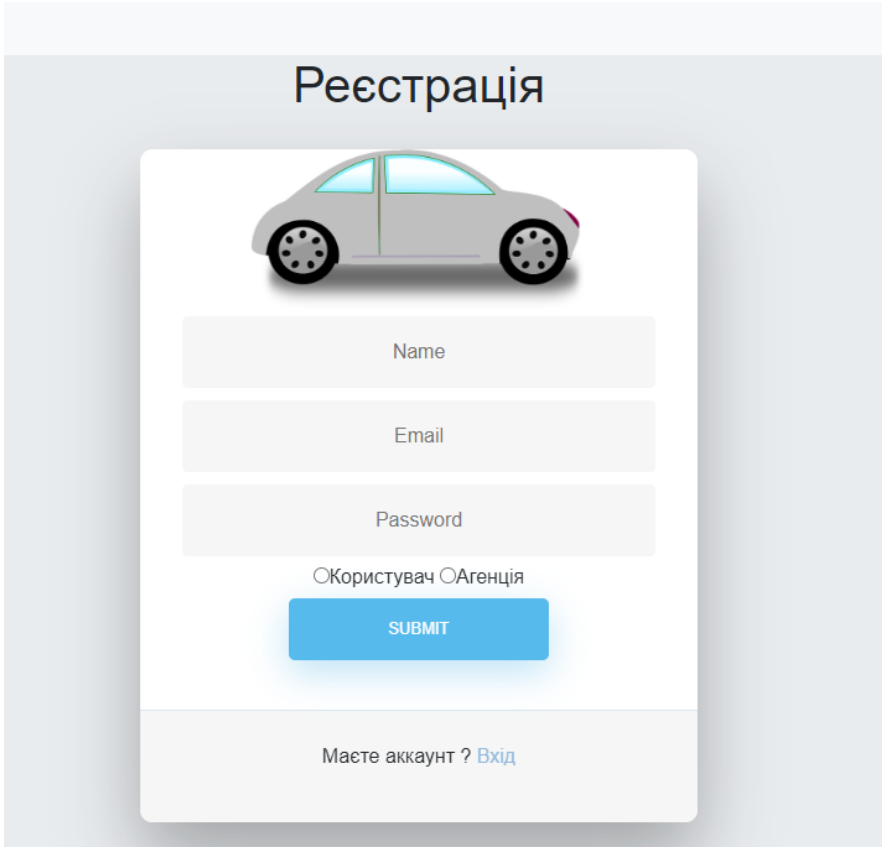


Рисунок 3.2 – Сторінка реєстрації

Після того як користувач зареєструвався в системі і ввійшов в свій аккаунт він зможе побачити головну сторінку з авто які він може орендувати (рис. 3.3).

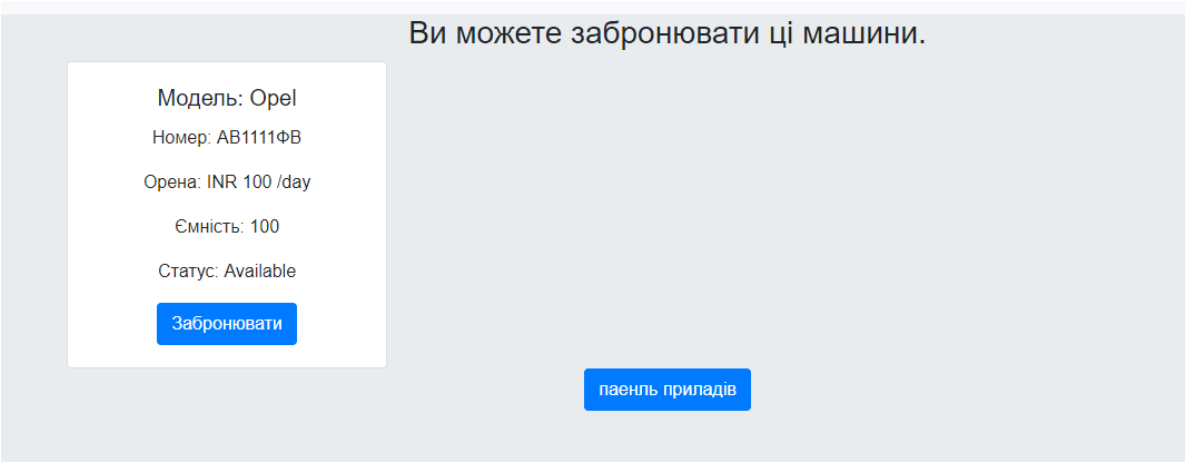


Рисунок 3.3 – Сторінка з авто

На даній сторінці користувач може почити інформацію про автомобілі, які він може забронвати, якщ корристувач нажме кнопку забронювати авто, то йому відкриється сторінка бронювання авто (рис. 3.4).

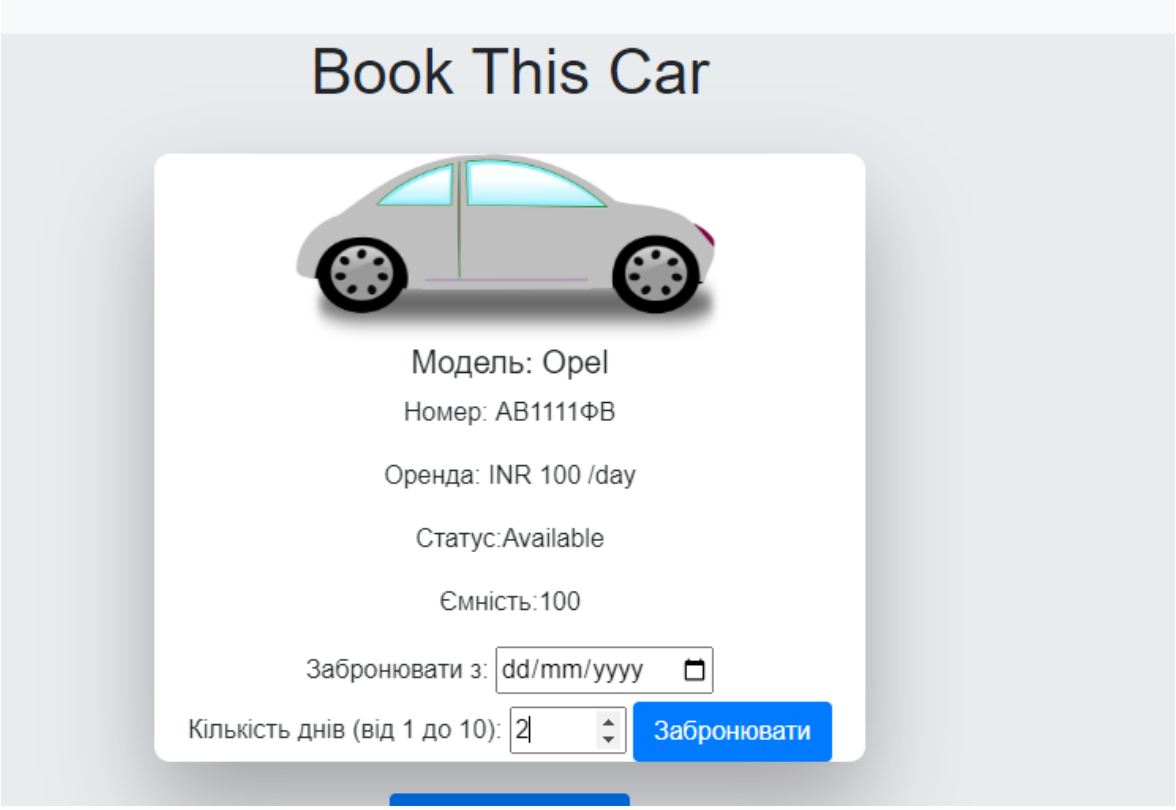


Рисунок 3.4 – Сторінка бронювання авто

На сторінці бронювання авто користувач може ще раз подивитися подробиці про авто та вибарти з якої дати і на скільки днів він хоче забронювати авто.

Після того як користувач забронює авто він зможе побачити це на сторінці його забронюваних автомобілів (рис. 3.5).

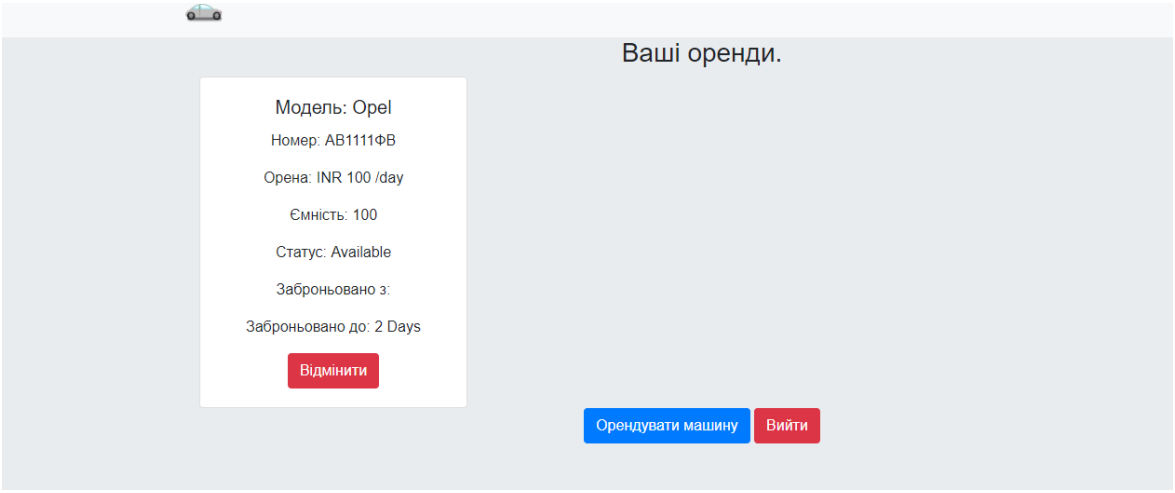


Рисунок 3.5 – Сторінка забронюваних авто користувачем

Розробленимй додаток також дозволяє працювати зі сторони орендодавця.

Арендодавець може побачити всі свої авто на стоірнці його автомобілів (рис. 3.6).

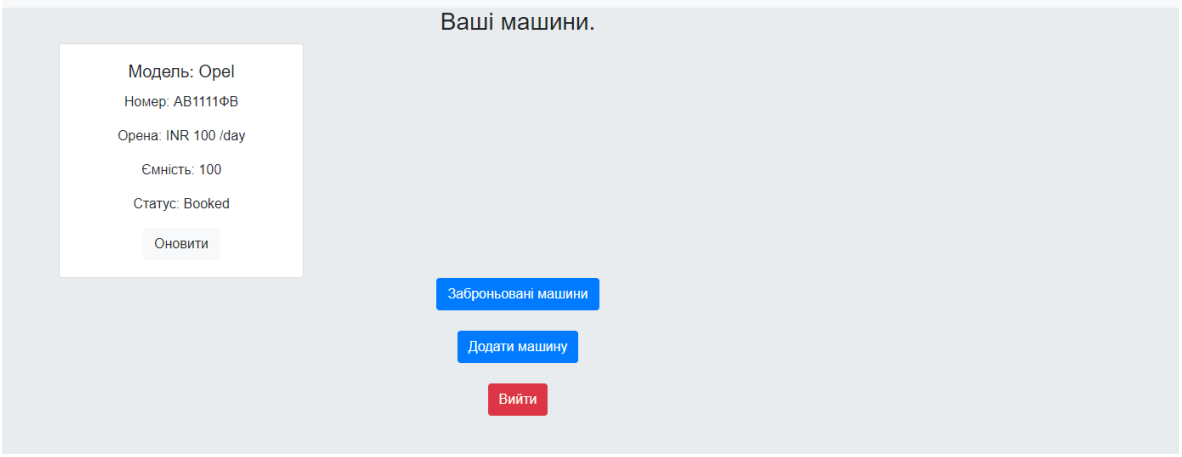


Рисунок 3.6 – Сторінка авто орендодавця

Також орендодавець може побачити список всіх його забронюваних автомобілів на сторінці його забронюваних автомобілів (рис. 3.7).

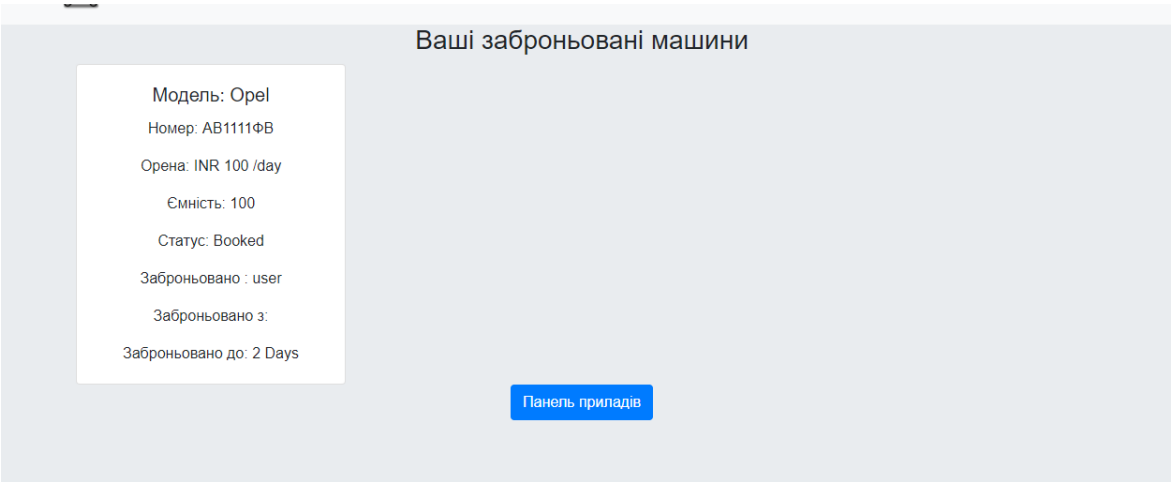


Рисунок 3.7 – Сторінка заброньованих авто орендодавця

Також орендодавець може додавти нові автомобюілі на відповідній сторінці (рис. 3.8).

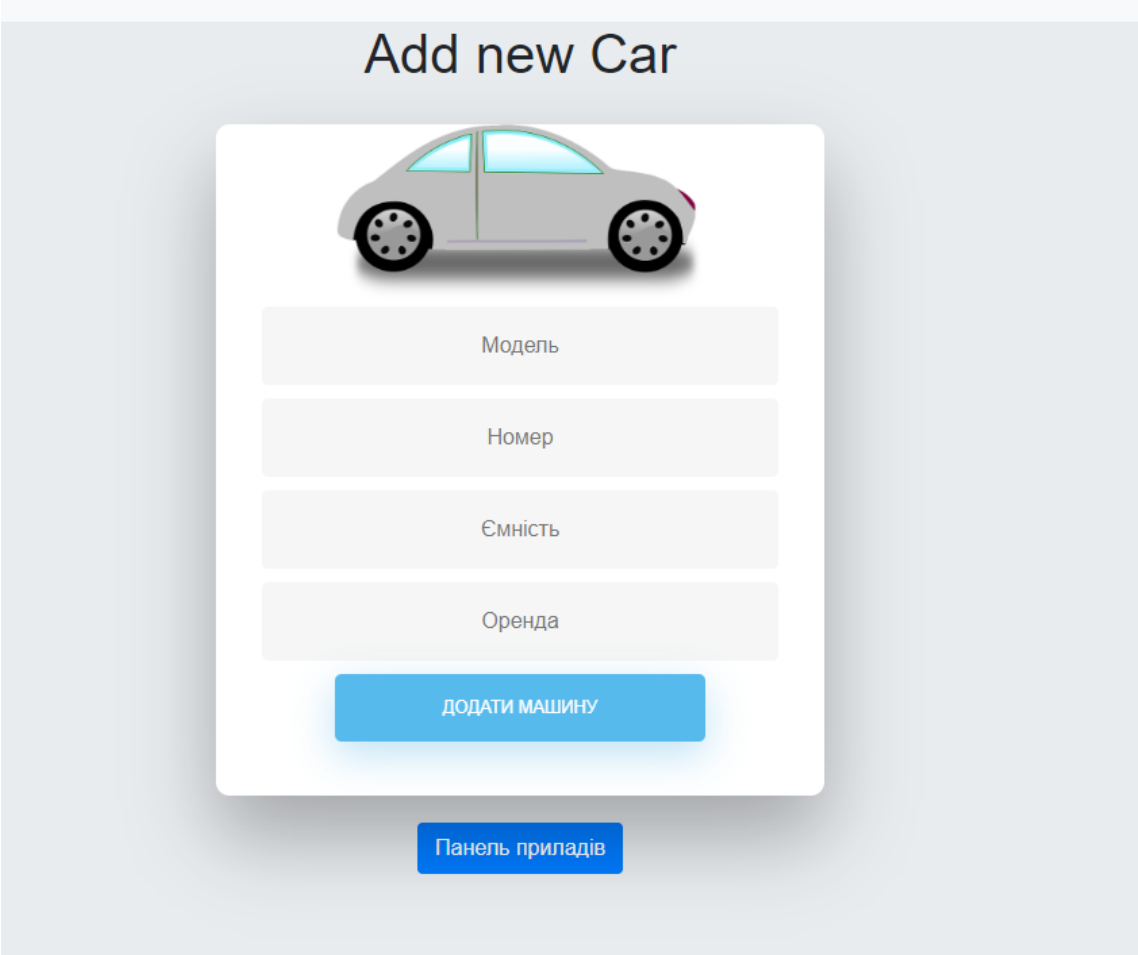


Рисунок 3.8 – Сторінка додавання нового авто

Також власник авто може редагувати дані про свої свої авто, на сторінці оновлення інформації про авто (рис.3.9).

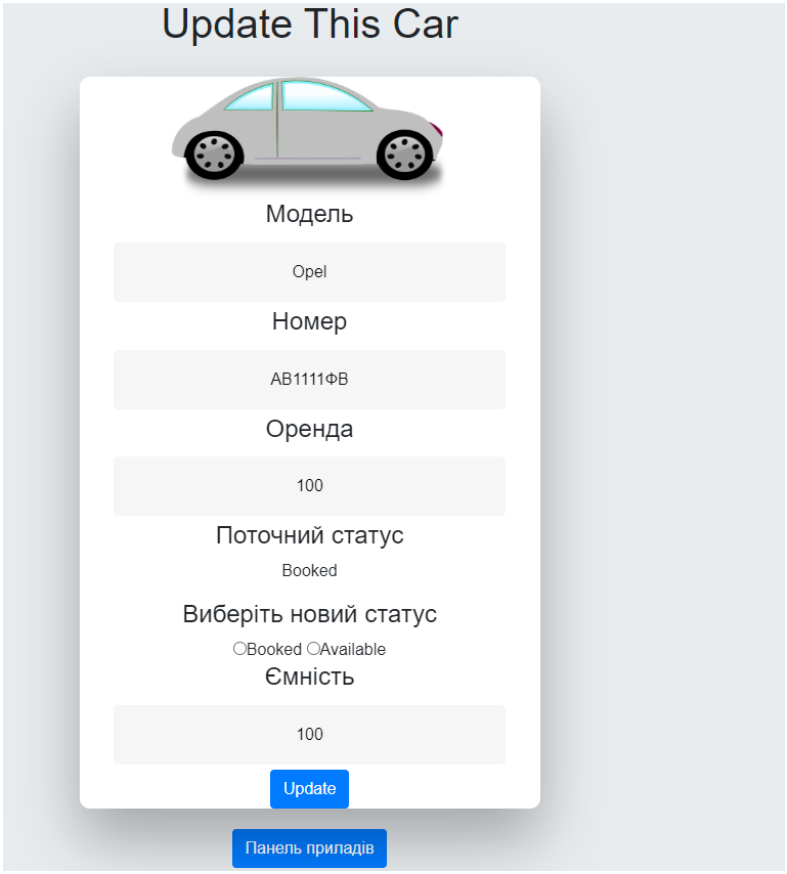


Рисунок 3.9 – Сторінка оновлення даних авто

Перевіримо записи в блокчейні для перевірки чи створився смарт контракт при оренді автомобіля користувачем (рис 3.10).

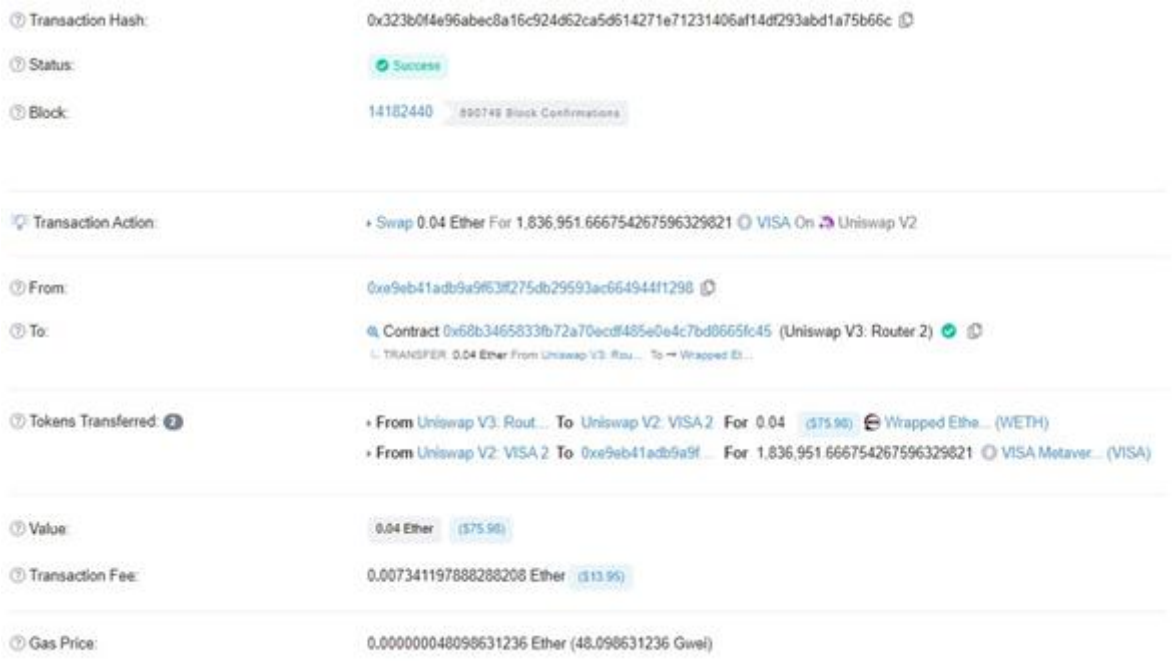


Рисунок 3.10 – Дані смарт контракту

Як можна побачити на сторінці 3.10 контрак створився успішно, що означає про правильність роботи системи.

Тестування реалізованої системи є критично важливим етапом для забезпечення її надійності, безпеки та коректності функціонування.

Завдяки цьому система спільного використання автомобілів на основі смарт-контрактів може функціонувати ефективно та безпечно, забезпечуючи захист особистих даних користувачів та їхню конфіденційність.

3.6 Висновки до розділу

Розроблена система спільного використання автомобілів на основі смарт-контрактів з анонімізацією даних користувачів є ефективним рішенням для забезпечення безпеки, конфіденційності та прозорості в умовах розподіленої системи. Використання сучасних технологій блокчейну та програмування дозволило створити функціональну систему, яка може бути впроваджена для реальних застосувань у сфері спільного використання автомобілів.

Система анонімізації користувачів була реалізована за допомогою методу генерації тимчасових ідентифікаторів, які використовуються для усіх транзакцій та взаємодій у системі. Це дозволяє зберігати конфіденційність даних користувачів та унеможливлювати їхнє відслідковування.

Ця робота демонструє важливість використання сучасних технологій для створення нових інноваційних рішень у сфері транспортної індустрії, які забезпечують зручність користувачів та ефективність управління ресурсами.

4 ЕКОНОМІЧНА ЧАСТИНА

У цьому розділі проведено комплексний аналіз економічного потенціалу розробки, що включає оцінку комерційних можливостей, прогноз витрат на виконання науково-дослідної роботи та впровадження її результатів, а також прогноз комерційних вигод від реалізації розробленого продукту. Окрім того, здійснено розрахунок ефективності вкладених інвестицій та визначено термін їх окупності.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

– проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;

– розраховано витрати на здійснення науково-технічної розробки;

– розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

На основі проведеного аналізу буде зроблено висновок щодо економічної доцільності розробки проекту "Захищена система спільного використання автомобілів на основі смарт-контрактів з можливістю анонімізації даних користувачів для унеможливлення їх відстежування". Цей висновок враховуватиме всі аспекти рентабельності та перспективності впровадження розробки на ринок, включаючи потенційні ризики та очікувані вигоди для інвесторів [17].

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Мета проведення технологічного аудиту полягає в оцінці комерційного потенціалу розробки, що виникла в результаті науково-технічної діяльності.

У рамках магістерської кваліфікаційної роботи була створена система спільного використання автомобілів, яка базується на смарт-контрактах та має можливість анонімізації даних користувачів для запобігання їхнього відслідковування.

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в	Технічні та споживчі влас-тивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї

Продовження таблиці 4.1

9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь- які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
Технічна здійсненність концепції	5	4	5
Ринкові переваги (наявність аналогів)	4	5	4
Ринкові переваги (ціна продукту)	5	3	5
Ринкові переваги (технічні властивості)	5	5	4
Ринкові переваги (експлуатаційні витрати)	4	4	3
Ринкові перспективи (розмір ринку)	5	3	5
Ринкові перспективи (конкуренція)	3	5	3
Практична здійсненність (наявність фахівців)	3	4	3
Практична здійсненність (наявність фінансів)	3	4	5
Практична здійсненність (необхідність нових матеріалів)	3	3	3
Практична здійсненність (термін реалізації)	3	4	3
Практична здійсненність (розробка документів)	2	3	4
Сума балів	*45*	*47*	*42*
Середньоарифметична сума балів $СБ_с$	*44,6*		

На основі аналізу даних, представлених у таблиці 4.2, ми здійснимо оцінку науково-технічного рівня та комерційного потенціалу розробки. Крім того, використаємо рекомендації, запропоновані у таблиці 4.3, для уточнення висновку.

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41…48	Високий
31…40	Вище середнього
21…30	Середній
11…20	Нижче середнього
0…10	Низький

В результаті нашої аналізу теми "Захищена система спільного використання автомобілів на основі смарт контрактів з можливістю анонімізації даних

користувачів для унеможливлення їхнього відслідковування" було визначено, що рівень комерційного потенціалу розробки склав 44,6 балів. Згідно з рекомендаціями, поданими у таблиці 4.3, такий результат свідчить про високу комерційну важливість проведених досліджень.

4.2 Розрахунок витрат на проведення науково-дослідної роботи

Прогнозування витрат на реалізацію науково-дослідної, дослідно-конструкторської та конструкторсько-технічної роботи включає три ключові етапи, що уважно розглядають різні аспекти витрат та впливають на всі етапи виконання проекту.

На першому етапі проводиться визначення витрат, які напряму пов'язані зі зусиллями та ресурсами, витраченими виконавцями цього розділу роботи. Сюди включаються витрати на оплату праці, навчання та інші витрати, які безпосередньо пов'язані з виконанням цієї конкретної роботи [17].

Другий етап передбачає розрахунок загальних витрат на виконання всієї роботи. Це включає витрати на матеріали, обладнання, послуги та інші загальні витрати, що стосуються всього проекту.

Третій етап охоплює прогнозування загальних витрат на впровадження результатів даної роботи. Це включає витрати на впровадження розробок, рекламу, підготовку персоналу та інші витрати, пов'язані з реалізацією отриманих результатів.

Витрати на основну заробітну плату дослідників ($З_о$) розраховують відповідно до посадових окладів працівників, за формулою:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \tag{4.1}$$

де k – кількість посад дослідників, залучених до процесу досліджень;

$М_{пі}$ – місячний посадовий оклад конкретного дослідника, грн;

t_i – кількість днів роботи конкретного дослідника;

$T_р$ – середня кількість робочих днів в місяці, $T_р$ = 21…23 дні.

Проведені розрахунки потрібно звести до таблиці.

$$З_о = \frac{36\,000 * 50}{21} = 85\,714{,}3\ (грн.)$$

Таблиця 4.4 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	36 000,00	1 714,28	50	85 714,3
Розробник	32 000,00	1 523,8	44	67 047,6
Програмний архітектор	25 000,00	1 190,47	15	17 857,14
			Всього	170 619,04

$$З_о = 170\,619{,}04\ (грн.)$$

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт розраховують за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i ,$$

(4.2)

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника на виконання певної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_р \cdot t_{зм}},$$

(4.3)

де M_M – розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати (залежно від діючого законодавства), грн;

$К_і$ – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду;

$К_с$ – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об’єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

$Т_р$ – середня кількість робочих днів в місяці, приблизно $Т_р$ = 21…23 дні;

$t_{зм}$ – тривалість зміни, год.

$М_М$ = 8000 грн.

$$С_1 = \frac{8000 * 1{,}1 * 1{,}65}{21 * 8} = 82{,}5 \text{ (грн.)}$$

$$З_{р1} = 82{,}5 * 4 = 330 \text{ (грн.)}$$

Проведені розрахунки потрібно звести до таблиці.

Таблиця 4.5 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Встановлення комп'ютерного обладнання	4	2	1,1	82,5	330
Організація робочого простору дослідника	3	2	1,1	82,5	247,5
Імплементація програмного забезпечення	3	5	1,7	127,5	385,5
Перевірка системи	2	2	1,1	82,5	160
				Всього	1123,0

$$З_р = 1123{,}0 \text{ (грн.)}$$

Додаткова заробітна плата розраховується як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \quad (4.3)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати.

$$З_{дод} = (170\ 619{,}04 + 1123{,}0) * \frac{12}{100} = 20\ 609{,}04\ (грн.)$$

Нарахування на заробітну плату дослідників та робітників розраховується як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%}, \quad (4.4)$$

де $Н_{зп}$ – норма нарахування на заробітну плату.

$$З_н = (170\ 619{,}04 + 1123{,}0 + 20\ 609{,}04) * \frac{22}{100} = 42\ 317{,}23\ (грн.)$$

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби й предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за прямим призначенням згідно з нормами їх витрачання, а також витрачені придбані напівфабрикати, що підлягають монтажу або виготовленню й додатковій обробці в цій організації, чи дослідні зразки, що виготовляються виробниками за документацією наукової організації [17].

Витрати на матеріали (М) у вартісному вираженні розраховуються окремо для кожного виду матеріалів за формулою:

$$М = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} В_j \cdot Ц_{вj}, \quad (4.5)$$

де $Н_j$ – норма витрат матеріалу j-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу j-го найменування, грн/кг;

$К_j$ – коефіцієнт транспортних витрат, (Кj = 1,1 … 1,15);

$В_j$ – маса відходів j-го найменування, кг;

$Ц_{вj}$ – вартість відходів j-го найменування, грн/кг.

Проведені розрахунки потрібно звести до таблиці.

$$M_1 = 223 * 2 * 1{,}1 = 490{,}6 \text{ (грн.)}$$

Таблиця 4.8 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за од, грн	Норма витрат, од	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Папір для принтера	223	2	0	0	490,6
Офісні ручки	19,50	6	0	0	128,7
Папір для нотаток	109	1	0	0	119,9
Коробка для зберігання паперів	52	1	0	0	57,2
Тонер ColorWay HP LJ P1005/1006 (TH-1005) black	130	1	0	0	143
				Всього	939,4

$$M = 939{,}4 \text{ (грн.)}$$

Витрати на комплектуючі ($К_в$), необхідні для проведення науково-дослідної роботи за темою «Захищена система спільного використання автомобілів на основі смарт контрактів з можливістю анонімізації даних користувачів для унеможливлення їхнього відслідковування», не передбачені. У цьому проекті не планується використання додаткових комплектуючих, оскільки всі необхідні ресурси для проведення досліджень вже є в наявності [17].

До статті «Спецустаткування для наукових (експериментальних) робіт» включаються витрати на виготовлення та придбання спеціалізованого обладнання, необхідного для проведення досліджень. Сюди також входять витрати на проектування, виготовлення, транспортування, монтаж та встановлення цього обладнання. У даній науково-дослідній роботі, присвяченій «Захищена система спільного використання автомобілів на основі смарт контрактів з можливістю анонімізації даних користувачів для унеможливлення їхнього відслідковування», витрати на спецустаткування не передбачені, оскільки всі необхідні засоби вже є в наявності або вони не потребуються для реалізації проекту [17].

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

До балансової вартості програмного забезпечення входять витрати на його інсталяцію, тому ці витрати беруться додатково в розмірі 10…12% від вартості програмного забезпечення.

Балансову вартість програмного забезпечення розраховують за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \tag{4.6}$$

де $Ц_{iпрг}$ – ціна придбання одиниці програмного засобу цього виду, грн;

$С_{пргі}$. – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$К_і$ – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($К_і$ = 1,10…1,12);

k – кількість найменувань програмних засобів.

Отримані результати необхідно звести до таблиці.

$$В_{прг} = 8\,540 * 2 * 1{,}1 = 18\,788 \text{ (грн.)}$$

Таблиця 4.9 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
WebStorm	2	8540	18 788
DataGrip	2	10 200	22 440
OFFICE 2021 PRO PLUS	1	5 100	5 610
		Всього	46 838

$$В_{прг} = 46\,838 \text{ (грн.)}$$

До статті «Амортизація обладнання, програмних засобів та приміщень» відносять амортизаційні відрахування по кожному виду обладнання, устаткування

та інших приладів і пристроїв, а також програмного забезпечення для проведення науково-дослідної роботи, за його наявності в дослідній організації або на підприємстві.

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо можуть бути розраховані з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (4.7)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

Проведені розрахунки необхідно звести до таблиці.

$$A_{обл} = \frac{36\,999 * 2}{3 * 12} = 2055{,}5 \text{ (грн.)}$$

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Ноутбук ASUS TUF Gaming A15 FA506NC-HN026	36 999	3	2	2055,5
Ноутбук HP Pavilion	26 999	3	2	1499,94
МФУ HP Laser 135a	7 899	5	2	263,3
Офісне приміщення	67 992	20	2	566,6
			Всього	4 385,34

$$А_{обл} = 4\,385{,}34\ (грн.)$$

До статті «Паливо та енергія для науково-виробничих цілей» належать витрати на придбання у сторонніх підприємств, установ і організацій будь-якого палива, що витрачається з технологічною метою на проведення досліджень. Стаття формується у разі виконання енергоємних наукових досліджень за методом прямого внесення витрат і досягає значної питомої ваги у собівартості досліджень [18].

Витрати на силову електроенергію ($В_е$) розраховують за формулою:

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{впі}}{\eta_i}, \quad (4.8)$$

де W_{yi} – встановлена потужність обладнання на певному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

Це – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії);

$К_{впі}$ – коефіцієнт, що враховує використання потужності, $К_{впі}$ <1;

η_i – коефіцієнт корисної дії обладнання, η_i<1.

Проведені розрахунки необхідно звести до таблиці.

$$В_е = \frac{0{,}6 * 352 * 7{,}50 * 0{,}95}{0{,}97} = 1551{,}34\ (грн.)$$

Таблиця 4.11 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Ноутбук ASUS TUF Gaming A15 FA506NC-HN026	0,6	352	1551,34
Ноутбук HP Pavilion	0,55	400	1615,97
МФУ HP Laser 135a	0,15	3	2,20
Офісне приміщення	0,2	400	587,6
		Всього	3757,11

$$В_е = 3\,757{,}11\ (грн.)$$

До статті «Службові відрядження» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень [18].

Витрати за статтею «Службові відрядження» розраховуються як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{св} = (З_о + З_р) \cdot \frac{Н_{св}}{100\%}, \tag{4.9}$$

де $Н_{св}$ – норма нарахування за статтею «Службові відрядження»;

$Н_{св}$ = 20%.

$$В_{св} = (170\ 619{,}04 + 1123{,}0) * \frac{20}{100} = 34\ 348{,}4\ (грн.)$$

До статті «Витрати на роботи, які виконують сторонні підприємства, установи і організації» належать витрати на проведення досліджень, що не можуть бути виконані штатними працівниками або наявним обладнанням організації, а виконуються на договірній основі іншими підприємствами, установами і організаціями незалежно від форм власності та позаштатними працівниками [18].

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуються як 30…45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{сп} = (З_о + З_р) \cdot \frac{Н_{сп}}{100\%}, \tag{4.9}$$

де $Н_{сп}$ – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації».

$Н_{сп}$ = 45%.

$$В_{сп} = (170\ 619{,}04 + 1123{,}0) * \frac{45}{100} = 77\ 283{,}9\ (грн.)$$

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50…100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \tag{4.10}$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати».

$Н_{ів}$ = 50%.

$$I_в = (170\ 619{,}04 + 1123{,}0) * \frac{55}{100} = 91\ 889{,}16\ (грн.)$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100…150% від суми основної заробітної плати дослідників та робітників за формулою:

$$В_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \tag{4.11}$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$Н_{нзв}$ = 100%

$$В_{нзв} = (170\ 619{,}04 + 1123{,}0) * \frac{100}{100} = 171\ 742{,}04\ (грн.)$$

Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$B_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + В_{спец} + В_{прг} + А_{обл} + В_е + В_{св} + В_{сп} + І_в + В_{нзв}$$

(4.12)

$$\text{В}_{\text{заг}} = 170\,619{,}04 + 1123{,}0 + 20\,48{,}54 + 42\,317{,}23 + 939{,}4 + 46\,838 + 4\,385{,}34 + 3\,757{,}11 + 34\,348{,}4 + 77\,283{,}9 + 91\,889{,}16 + 171\,742{,}04 = 647\,291{,}16\ (\text{грн.})$$

Загальні витрати ЗВ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{B_{заг}}{\eta},$$

(4.13)

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,7.

$$\text{ЗВ} = \frac{647\,291{,}16}{0{,}7} = 924\,701{,}7\ (\text{грн.})$$

4.3 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

У ринкових умовах загальним позитивним результатом, який потенційний інвестор може отримати від впровадження науково-технічної розробки, є збільшення чистого прибутку. Результати дослідження за темою «Захищена система спільного використання автомобілів на основі смарт контрактів з можливістю анонімізації даних користувачів для унеможливлення їхнього відслідковування» передбачають комерціалізацію протягом трьох років. У цьому випадку майбутній економічний ефект буде базуватися на таких даних:

– ΔN – збільшення кількості споживачів продукту у періоди, що аналізуються, завдяки покращенню його характеристик:

Перший рік – 1000 користувачів;

Другий рік – 1500 користувачів;

Третій рік – 1100 користувачів.

– N – кількість споживачів, які використовували аналогічний продукт до впровадження результатів нової науково-технічної розробки, становила 3000 користувачів.

– $Ц_б$ – вартість програмного продукту до впровадження результатів розробки складала 12 000 грн.

– ±ΔЦо – зміна вартості програмного продукту після впровадження результатів науково-технічної розробки складе 2000 грн.

Таким чином, запровадження інноваційної розробки призведе до збільшення кількості користувачів та вартості продукту, що позитивно вплине на чистий прибуток потенційного інвестора.

Можливе збільшення чистого прибутку у потенційного інвестора для кожного з трьох років, протягом яких очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, розраховується за формулою:

$$\Delta П_i = (\pm \Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (4.14)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2024 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0{,}8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту. Приймемо $\rho = 30\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\Delta П_1 = (2000 \times 3000 + 12\,000 \times 1000) \times 0{,}83 \times 0{,}3 \times \left(1 - \frac{0{,}18}{100}\right)$$

$$= 4\,473\,932{,}4 \text{ (грн.)}$$

Збільшення чистого прибутку 2-го року:

$$\Delta П_2 = (2000 \times 3000 + 12\,000 \times (1000 + 1500)) \times 0{,}83 \times 0{,}3 \times \left(1 - \frac{0{,}18}{100}\right)$$
$$= 8\,947\,864{,}8\ (\text{грн.})$$

Збільшення чистого прибутку 3-го року:

$$\Delta П_2 = (2000 \times 3000 + 12\,000 \times (1000 + 1500 + 1100)) \times 0{,}83 \times 0{,}3$$
$$\times \left(1 - \frac{0{,}18}{100}\right) = 12\,227\,748{,}56\ (\text{грн.})$$

Отже, згідно з розрахунками, комерційна вигода від впровадження розробки буде значною, як і очікувалося, і виявиться у суттєвому зростанні чистого прибутку підприємства.

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Приведена вартість збільшення всіх чистих прибутків *ПП*, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \tag{4.15}$$

де ΔПі – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

Т – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ =0,05…0,15;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$ПП = \frac{4\,473\,932{,}4}{(1+0{,}2)^1} + \frac{8\,947\,864{,}8}{(1+0{,}2)^2} + \frac{12\,227\,748{,}56}{(1+0{,}2)^3} = 17\,018\,315{,}37\ (\text{грн.})$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{інв} \cdot ЗВ,$$

(4.16)

де інв k – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай інв k =2...5, але може бути і більшим;

ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$\text{PV} = 924\,701{,}7 * 3 = 2\,774\,105{,}1\ (\text{грн.})$$

Тоді абсолютний економічний ефект $E_{абс}$ або чистий приведений дохід (NPV, Net Present Value) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV,$$

(4.16)

де ПП – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн;

PV – теперішня вартість початкових інвестицій, грн.

$$\text{E}_{\text{абс}} = 17\,018\,315{,}37 - 2\,774\,105{,}1 = 14\,244\,210{,}27\ (\text{грн.})$$

Оскільки Еабс > 0, доведено, що проведення наукових досліджень для розробки програмного продукту та його подальше впровадження принесуть прибуток. Це підтверджує доцільність проведення таких досліджень. Проте, навіть попри цю обґрунтованість, це ще не означає автоматичної зацікавленості інвесторів у фінансуванні даної програми. Важливо також враховувати додаткові фактори, такі як ринкові умови, конкурентоспроможність продукту, ризики та потенційні вигоди, які можуть вплинути на рішення інвесторів [18].

Внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розраховується за формулою:

$$E_в = \sqrt[T_ж]{1 + \frac{E_{абс}}{PV}} - 1, \tag{4.17}$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, грн;

PV – теперішня вартість початкових інвестицій, грн;

$T_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, роки.

$$E_в = \sqrt[3]{1 + \frac{14\,244\,210{,}27}{2\,774\,105{,}1}} - 1 = 0{,}83$$

Далі визначають бар'єрну ставку дисконтування мін τ , тобто мінімальну внутрішню економічну дохідність інвестицій, нижче якої кошти у впровадження науково-технічної розробки та її комерціалізацію вкладатися не будуть [18].

Мінімальна внутрішня економічна дохідність вкладених інвестицій мін τ визначається за формулою:

$$\tau_{мін} = d + f, \tag{4.18}$$

де d – середньозважена ставка за депозитними операціями в комерційних банках;

f – показник, що характеризує ризикованість вкладення інвестицій; зазвичай величина f =0,05...0,5, але може бути і значно вищою

$$\tau_{min} = 0{,}2 + 0{,}3 = 0{,}5$$

Оскільки $E_в$ = 82% > τ_{min} = 50%, то у інвестора є потенційна зацікавленість у фінансуванні даної наукової розробки.

Далі розраховуємо період окупності інвестицій $T_{ок}$ (DPP, Discounted Payback Period), які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}, \tag{4.19}$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$$\text{Т}_{\text{ок}} = \frac{1}{0{,}83} = 1{,}2 \text{ (рік)}$$

З огляду на те, що період повернення інвестицій у реалізацію наукового проекту становить менше трьох років, можна зробити висновок, що фінансування нової розробки є виправданим. Крім того, короткий термін окупності підвищує привабливість проекту для інвесторів, оскільки він знижує фінансові ризики та дозволяє швидше отримати прибуток. Це також свідчить про ефективність та перспективність проекту в умовах сучасного ринку.

4.5 Висновки до розділу

У цьому розділі проведено оцінку комерційного потенціалу розробки "Захищена система спільного використання автомобілів на основі смарт-контрактів з можливістю анонімізації даних користувачів для унеможливлення їхнього відстежування".

З розрахунків витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи видно, що загальні витрати на розробку складають 924 701,7 грн.

Термін окупності вкладених інвестицій у реалізацію проекту становить 1,2 роки, що також підтверджує доцільність фінансування нової розробки. З урахуванням отриманих економічних показників можна вважати, що запропонована розробка програмного засобу має високий комерційний потенціал і, таким чином, є обґрунтованою для подальшого впровадження.

ВИСНОВКИ

У рамках дипломної роботи була проведена розробка захищеної системи спільного використання автомобілів на основі смарт контрактів з можливістю анонімізації даних користувачів для унеможливлення їх відслідковування. Головною метою роботи було створення системи, яка забезпечує високий рівень конфіденційності та безпеки даних, що є критичним для створення довіри до системи спільного використання автомобілів.

Результати дослідження показали, що використання смарт контрактів та технологій блокчейну є ефективним способом управління доступом до автомобілів. Це дозволяє забезпечити прозорість і надійність системи, а також знизити витрати на посередників. Розроблений підхід до анонімізації даних користувачів відповідає сучасним вимогам до захисту приватності і може бути використаний для подальшого розвитку подібних систем.

Економічне обґрунтування показало, що розроблена система має значний комерційний потенціал. Прогнозується, що вона може бути привабливою для інвесторів, що сприятиме її подальшому впровадженню та розвитку на ринку.

Отримані результати можуть бути використані для подальшого вдосконалення систем спільного використання автомобілів та розширення їхнього застосування в реальних умовах. Враховуючи зростаючий інтерес до блокчейн технологій та захисту даних, дослідження в цій області має великий потенціал для подальшого розвитку і впровадження.

Ця робота не лише внесла вагомий внесок у теорію і практику захисту персональних даних в системах спільного використання автомобілів, але й відкрила двері для нових досліджень у цій важливій галузі інформаційної безпеки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Authentication - Django REST framework. *Home - Django REST framework*. URL: https://www.django-rest-framework.org/api-guide/authentication/.

2. Jena B. K. AES Encryption: Secure Data with Advanced Encryption Standard. *Simplilearn.com*. URL: https://www.simplilearn.com/tutorials/cryptography-tutorial/aes-encryption.

3. Magnusson A. Token-based Authentication: Everything You Need to Know. *StrongDM: Your Partner in Zero Trust Privileged Access*. URL: https://www.strongdm.com/blog/token-based-authentication.

4. What Is an Authentication Token? | Fortinet. *Fortinet*. URL: https://www.fortinet.com/resources/cyberglossary/authentication-token.

5. What Is Token-Based Authentication? | Okta. *Employee and Customer Identity Solutions | Okta*. URL: https://www.okta.com/identity-101/what-is-token-based-authentication/.

6. What is token-based authentication?. *Stack Overflow*. URL: https://stackoverflow.com/questions/1592534/what-is-token-based-authentication.

7. Cloudflare Application Services | Security and Performance. Cloudflare. URL: https://www.cloudflare.com/application-services/?utm_source=google&utm_medium=cpc&utm_campaign=DG_EMEig7BqZ7VtZrN6wUzk1Nhp6nkaAgwEEALw_wcB&gclsrc=aw.ds).

8. Кібербезпека. URL: https://www.span.eu/ua/рішення-та-послуги/сервіси-з-безпеки/кібербезпека/?gad_source=1&gclid=Cj0KCQiA67CrBhC1ARIsACKAa8REXQS8JOs2ZaaOW6g5OyaeVWM8ksz3M6viwjX_pclBefnGsQgibfMaAgYHEALw_wcB.

9. Contributors to Wikimedia projects. Information security - Wikipedia. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Information_security

10. Imperva. URL: https://www.imperva.com/learn/data-security/information-security-infosec/.

11. What is information security (infosec)?. Cisco. URL: https://www.cisco.com/c/en/us/products/security/what-is-information-security-infosec.html.

12. What is information security (infosec)? Goals, types and applications. Exabeam. URL: https://www.exabeam.com/explainers/information-security/information-security-goals-types-and-applications/.

13. Yasar K., Wright G., Teravainen T. What is information security (infosec)? – techtarget definition. Security. URL: https://www.techtarget.com/searchsecurity/definition/information-security-infosec.

14. What is information security? | IBM. IBM in Deutschland, Österreich und der Schweiz | IBM. URL: https://www.ibm.com/topics/information-security.

15. What is information security? - geeksforgeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/what-is-information-security/.

16. INFOSEC - Glossary | CSRC. NIST Computer Security Resource Center | CSRC. URL: https://csrc.nist.gov/glossary/term/INFOSEC.

17. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. Вінниця : ВНТУ, 2021. 42 с.

18. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа. Вінниця : ВНТУ, 2016. 113 с.

19. What is authorization? - examples and definition - auth0. *Auth0*. URL: https://auth0.com/intro-to-iam/what-is-authorization .

20. Academy B. Seed Phrase | Binance Academy. Binance Academy. URL: https://academy.binance.com/en/glossary/seed-phrase.

21. What is authorization? - examples and definition - auth0. *Auth0*. URL: https://auth0.com/intro-to-iam/what-is-authorization .

22. Aholi ICT limited - cyber security services company in bangladesh. Aholi ICT Limited - Cyber Security Services Company in Bangladesh. URL: https://www.aholiict.com/usbsecurity.php

23. What Is a USB Security Key?. USB Memory Direct. URL: https://www.usbmemorydirect.com/blog/what-is-a-usb-security-key/

24. Eban Escott. Adopting a AAA approach to software security. URL: https://workingmouse.com.au/software-development/adopting-a-aaa-approach-to-software-security/

25. Suzanne Humphries. What is a USB security key, and should you use one?. URL: https://www.reviewgeek.com/63448/what-is-a-usb-security-key-and-should-you-use-one/

26. What is authentication, authorization, and accounting (AAA)?. URL: https://www.fortinet.com/resources/cyberglossary/aaa-security.

27. Visual studio code. Visual Studio Code. URL: https://code.visualstudio.com/

28. Редактор коду visual studio code. Habr. URL: https://habr.com/ua/articles/490754 (дата звернення: 25.05.2023).

29. JavaScript. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript

30. What is javascript? A basic introduction to JS for beginners. Hostinger Tutorials. URL: https://www.hostinger.com/tutorials/what-is-javascript).

31. Visual studio code. Visual Studio Code. URL: https://code.visualstudio.com/).

32. Редактор коду visual studio code. Habr. URL: https://habr.com/ua/articles/490754

33. JavaScript. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript.

34. Electron. Electron. URL: https://www.electronjs.org/ .

35. Node.js. Node.js. URL: https://nodejs.org/uk .

36. The Modern JavaScript. URL: https://javascript.info/ .

37. Sufiyan T. What is node.js: a comprehensive guide. *Simplilearn.com*. URL: https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs).

38. Megida D. What is javascript? A definition of the JS programming language. freeCodeCamp.org. URL: https://www.freecodecamp.org/news/what-is-javascript-definition-of-js/

39. What is javascript? A basic introduction to JS for beginners. Hostinger Tutorials. URL: https://www.hostinger.com/tutorials/what-is-javascript).

ДОДАТКИ

Додаток А. Технічне завдання

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

ЗАТВЕРДЖУЮ
Голова секції "Управління інформаційною безпекою" кафедри МБІС
д.т.н., професор
_____ Юрій ЯРЕМЧУК
"____" _____2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до магістерської кваліфікаційної роботи на тему:
Захищена система спільного використання автомобілів на основі смарт контрактів з можливістю анонімізації даних користувачів для унеможливлення їхнього відслідковування
08-72.МКР.000.00.000.ТЗ

Керівник магістерської кваліфікаційної роботи
д.т.н., професор каф. МБІС
_____ Яремчук Ю.Є.__

Вінниця – 2024 р.

1. Найменування та область застосування

Захищена система спільного використання автомобілів на основі смарт контрактів з можливістю анонімізації даних користувачів для унеможливлення їхнього відслідковування

2.Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ № 81 від 11. 03. 2024 р.

3. Мета та призначення розробки

3.1 Мета розробки: Реалізація захищеної системи спільного використання автомобілів на основі смарт контрактів з можливістю анонімізації даних користувачів для унеможливлення їхнього відслідковування

3.2 Призначення: захист системи спільного використання автомобілів

4. Джерела розробки

4.1. Ахрамович В. М. Ідентифікація й аутентифікація, керування доступом // Сучасний захист інформації. – 2016. №4.– С. 47-51.

4.2. Бурячок В.Л. Політика інформаційної безпеки: підручник. / В.Л.Бурячок, Р.В.Грищук, В.О.Хорошко / За заг. ред. докт. техн. наук, проф. В.О. Хорошка. – К.: ПВП «Задруга», 2014. – 222 с.

4.3. Єсін В.І. Безпека інформаційних систем і технологій / В.І.Єсін, О.О. Кузнецов, Л.С. Сорока. – Харків: ХНУ імені В.Н. Каразіна, 2013. – 632 с.

4.4. ZakariaOmar, ZangooeiToomaj, MohdAfiziMohdShukran. Enhancing Mixing Recognition-Based and Recall-Based Approach in Graphical Password Scheme. IJACT, Vol. 4, No. 15, pp. 189-197, 2012.

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Програмний засіб повинен мати зручний, легкий у використанні інтерфейс користувача;

5.1.2 Реалізація не повинна вимагати спеціальних ліцензійних програмних додатків;

5.1.3 Програмний засіб повинен виконувати процес автентифікації користувачів у системі.

5.2 Вимоги до надійності:

5.2.1 Програмний засіб повинен працювати без помилок, у випадку виникнення критичних ситуацій необхідно передбачити виведення відповідних повідомлень;

5.2.2 Бази даних повинні бути налаштовані на автоматичне створення резервних копій;

5.2.3 Програмний засіб повинен виконувати свої функції.

5.3 Вимоги до складу і параметрів технічних засобів:

– процесор – Pentium 1500 МГц і подібні до них;

– оперативна пам’ять – не менше 512 Mb;

– середовище функціонування – операційна система сімейство Windows;

– вимоги до техніки безпеки при роботі з програмою повинні відповідати існуючим вимогам та стандартам з техніки безпеки при користуванні комп’ютерною технікою.

6. Вимоги до програмної документації

6.1 Обов'язкова поетапна інструкція для майбутніх користувачів, наведена у пункті 3.4

7. Вимоги до технічного захисту інформації

7.1 Необхідно забезпечити захист розроблюваного програмного засобу від несанкціонованого використання.

7.2 Неможливість отримання доступу незареєстрованих користувачів до інформаційних ресурсів.

8. Техніко-економічні показники

8.1 Цінність результатів використання даного проекту повинна перевищувати витрати на його реалізацію.

8.2 Має бути реалізований таким чином, щоб підходити для використання широкого загалу.

9. Стадії та етапи розробки

№	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи		Примітка
1.	Визначення напрямку магістерської роботи, формулювання теми	1.03.2024	15.03.2024	
2.	Аналіз предметної області обраної теми	15.03.2024	1.04.2024	
3.	Розробка роботи	1.04.2024	10.04.2024	
4.	Написання магістерської роботи на основі розробленої теми	10.04.2024	20.04.2024	
5.	Передзахист магістерської кваліфікаційної роботи	20.04.2024	10.05.2024	
6.	Виправлення, уточнення, корегування магістерської кваліфікаційної роботи	10.05.2024	4.06.2024	
7.	Захист магістерської кваліфікаційної роботи	10.06.2024	10.06.2024	

10. Порядок контролю та прийому

10.1 До приймання магістерської кваліфікаційної роботи надається:

– ПЗ до магістерської кваліфікаційної роботи;
– програмний додаток;
– презентація;
– відзив керівника роботи;
– відзив опонента

Технічне завдання до виконання прийняв _____ Саврацький В.М.

Додаток Б. Лістинг програми

```
pragma solidity ^0.8.0;

contract CarSharing {
   struct User {
      string name;
      string email;
   }

   mapping(address => User) public users;

   function registerUser(string memory _name, string memory _email) public {
      users[msg.sender] = User(_name, _email);
   }
}
pragma solidity ^0.8.0;

contract CarSharing {
   struct Car {
      uint id;
      string model;
      bool isAvailable;
   }
   struct Booking {
      address user;
      uint carId;
      uint startTime;
      uint endTime;
   }
   Car[] public cars;
   Booking[] public bookings;
   function addCar(string memory _model) public {
      cars.push(Car(cars.length, _model, true));
   }
   function bookCar(uint _carId, uint _startTime, uint _endTime) public {
      require(cars[_carId].isAvailable, "Car is not available");
      bookings.push(Booking(msg.sender, _carId, _startTime, _endTime));
      cars[_carId].isAvailable = false;
   }
}

pragma solidity ^0.8.0;
contract CarSharing {
   struct Payment {
      address from;
      address to;
      uint amount;
   }
   Payment[] public payments;
   function makePayment(address _to) public payable {
      require(msg.value > 0, "Payment must be greater than 0");
      payments.push(Payment(msg.sender, _to, msg.value));
      payable(_to).transfer(msg.value);
   }
}
from web3 import Web3
# Підключення до локального вузла Ethereum
```

```
web3 = Web3(Web3.HTTPProvider('http://127.0.0.1:8545'))
# Перевірка підключення
print(web3.isConnected())
# ABI та адреса смарт-контракту
abi = '[ABI контракту]'
contract_address = '0xYourContractAddress'
# Ініціалізація контракту
contract = web3.eth.contract(address=contract_address, abi=abi)
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes
def encrypt_data(data, key):
    cipher = AES.new(key, AES.MODE_EAX)
    nonce = cipher.nonce
    ciphertext, tag = cipher.encrypt_and_digest(data.encode('utf-8'))
    return nonce, ciphertext, tag
def decrypt_data(nonce, ciphertext, tag, key):
    cipher = AES.new(key, AES.MODE_EAX, nonce=nonce)
    data = cipher.decrypt_and_verify(ciphertext, tag)
    return data.decode('utf-8')
key = get_random_bytes(16)
nonce, ciphertext, tag = encrypt_data('Confidential Data', key)
print("Encrypted data:", ciphertext)

decrypted_data = decrypt_data(nonce, ciphertext, tag, key)
print("Decrypted data:", decrypted_data)
import random

import string

def generate_pseudonym(length=10):

    characters = string.ascii_letters + string.digits

    pseudonym = ''.join(random.choice(characters) for _ in range(length))

    return pseudonym

# Приклад використання

pseudonym = generate_pseudonym()

print("Generated pseudonym:", pseudonym)

from cryptography.hazmat.primitives.asymmetric import rsa

from cryptography.hazmat.primitives import serialization

from cryptography.hazmat.primitives.asymmetric import padding

from cryptography.hazmat.primitives import hashes

private_key = rsa.generate_private_key(public_exponent=65537, key_size=2048)

public_key = private_key.public_key()

# Серіалізація ключів

pem_private_key = private_key.private_bytes(

    encoding=serialization.Encoding.PEM,

    format=serialization.PrivateFormat.PKCS8,

    encryption_algorithm=serialization.NoEncryption()

)

pem_public_key = public_key.public_bytes(
```

```
    encoding=serialization.Encoding.PEM,
    format=serialization.PublicFormat.SubjectPublicKeyInfo
)
def encrypt_data(data, public_key):
    ciphertext = public_key.encrypt(
        data.encode('utf-8'),
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    return ciphertext
def decrypt_data(ciphertext, private_key):
    decrypted_data = private_key.decrypt(
        ciphertext,
        padding.OAEP(
            mgf=padding.MGF1(algorithm=hashes.SHA256()),
            algorithm=hashes.SHA256(),
            label=None
        )
    )
    return decrypted_data.decode('utf-8')
data = "Sensitive User Data"
ciphertext = encrypt_data(data, public_key)
print("Encrypted data:", ciphertext)
decrypted_data = decrypt_data(ciphertext, private_key)
print("Decrypted data:", decrypted_data)
from flask import Flask, request, jsonify
from web3 import Web3
app = Flask(__name__)
# Підключення до локального вузла Ethereum
web3 = Web3(Web3.HTTPProvider('http://127.0.0.1:8545'))
# Перевірка підключення
if not web3.isConnected():
    print("Не вдалося підключитися до Ethereum")
# ABI та адреса смарт-контракту
abi = '[ABI контракту]'
contract_address = '0xYourContractAddress'
# Ініціалізація контракту
contract = web3.eth.contract(address=contract_address, abi=abi)
@app.route('/register', methods=['POST'])
def register_user():
    data = request.json
```

```
        name = data['name']
        email = data['email']
        # Виклик методу registerUser смарт-контракту
        tx_hash = contract.functions.registerUser(name, email).transact({'from':
web3.eth.accounts[0]})
        receipt = web3.eth.waitForTransactionReceipt(tx_hash)
        return jsonify({"message": "User registered", "receipt": receipt})
    @app.route('/cars', methods=['GET'])
    def get_available_cars():
        cars = []
        for i in range(contract.functions.getCarsCount().call()):
            car = contract.functions.cars(i).call()
            if car[2]:  # Перевірка, чи автомобіль доступний
                cars.append({"id": car[0], "model": car[1]})
        return jsonify({"available_cars": cars})
    @app.route('/book', methods=['POST'])
    def book_car():
        data = request.json
        car_id = data['car_id']
        start_time = data['start_time']
        end_time = data['end_time']
        # Виклик методу bookCar смарт-контракту
        tx_hash = contract.functions.bookCar(car_id, start_time, end_time).transact({'from':
web3.eth.accounts[0]})
        receipt = web3.eth.waitForTransactionReceipt(tx_hash)

        return jsonify({"message": "Car booked", "receipt": receipt})
    @app.route('/pay', methods=['POST'])
    def make_payment():
        data = request.json
        to_address = data['to']
        amount = data['amount']
        # Вклик методу makePayment смарт-контракту
        tx_hash = contract.functions.makePayment(to_address).transact({
            'from': web3.eth.accounts[0],
            'value': web3.toWei(amount, 'ether')
        })
        receipt = web3.eth.waitForTransactionReceipt(tx_hash)
        return jsonify({"message": "Payment made", "receipt": receipt})
```

Додаток В. Ілюстративний матеріал

ЗАХИЩЕНА СИСТЕМА СПІЛЬНОГО ВИКОРИСТАННЯ АВТОМОБІЛІВ НА ОСНОВІ СМАРТ КОНТРАКТІВ З МОЖЛИВІСТЮ АНОНІМІЗАЦІЇ ДАНИХ КОРИСТУВАЧІВ ДЛЯ УНЕМОЖЛИВЛЕННЯ ЇХНЬОГО ВІДСЛІДКОВУВАННЯ

ВИКОНАВ: СТ. ГРУПИ КІТС-22МЗ САВРАЦЬКИЙ В.М.

КЕРІВНИК: Д.Т.Н., ПРОФ. КАФ. МБІС ЯРЕМЧУК Ю.Є.

ВСТУП

- У СУЧАСНОМУ СВІТІ ЗРОСТАЄ ПОПУЛЯРНІСТЬ КОНЦЕПЦІЇ СПІЛЬНОГО ВИКОРИСТАННЯ АВТОМОБІЛІВ ЯК АЛЬТЕРНАТИВНОГО СПОСОБУ ТРАНСПОРТУВАННЯ, ЩО ДОЗВОЛЯЄ ЗМЕНШИТИ ВИТРАТИ НА ТРАНСПОРТ, ЗНИЗИТИ ВПЛИВ НА НАВКОЛИШНЄ СЕРЕДОВИЩЕ ТА ПІДВИЩИТИ МОБІЛЬНІСТЬ НАСЕЛЕННЯ. ВОДНОЧАС ІСНУЮТЬ СЕРЙОЗНІ ВИКЛИКИ, ПОВ'ЯЗАНІ З КОНФІДЕНЦІЙНІСТЮ ТА БЕЗПЕКОЮ ОСОБИСТИХ ДАНИХ КОРИСТУВАЧІВ ТАКИХ СИСТЕМ. ІСНУЮЧІ ПЛАТФОРМИ СПІЛЬНОГО ВИКОРИСТАННЯ АВТОМОБІЛІВ ЧАСТО НЕ ЗАБЕЗПЕЧУЮТЬ ДОСТАТНЬОГО РІВНЯ ЗАХИСТУ ПРИВАТНОСТІ І МОЖУТЬ СТАТИ ОБ'ЄКТОМ АТАК КІБЕРЗЛОЧИНЦІВ.

АКИУАЛЬНІСТЬ ТА МЕТА РОБОТИ

- **АКТУАЛЬНІСТЬ.** ВИКОРИСТАННЯ ТЕХНОЛОГІЙ БЛОКЧЕЙНУ ТА СМАРТ КОНТРАКТІВ ВІДКРИВАЄ НОВІ МОЖЛИВОСТІ ДЛЯ СТВОРЕННЯ ЗАХИЩЕНИХ І ПРОЗОРИХ СИСТЕМ, ЯКІ ЗАБЕЗПЕЧУЮТЬ АНОНІМІЗАЦІЮ ДАНИХ КОРИСТУВАЧІВ ТА УНЕМОЖЛИВЛЮЮТЬ ЇХ ВІДСЛІДКОВУВАННЯ. ВПРОВАДЖЕННЯ ТАКИХ СИСТЕМ МОЖЕ ДОПОМОГТИ ЗБІЛЬШИТИ ДОВІРУ КОРИСТУВАЧІВ ДО ПЛАТФОРМ СПІЛЬНОГО ВИКОРИСТАННЯ АВТОМОБІЛІВ І СПРИЯТИ ЇХНЬОМУ ПОДАЛЬШОМУ РОЗВИТКУ. З ЦІЄЇ ПРИЧИНИ АКТУАЛЬНИМ Є РОЗВИТОК І ДОСЛІДЖЕННЯ НОВИХ МЕТОДІВ ЗАБЕЗПЕЧЕННЯ КОНФІДЕНЦІЙНОСТІ ТА БЕЗПЕКИ ДАНИХ КОРИСТУВАЧІВ В СИСТЕМАХ СПІЛЬНОГО ВИКОРИСТАННЯ АВТОМОБІЛІВ, ЩО БАЗУЮТЬСЯ НА ТЕХНОЛОГІЯХ БЛОКЧЕЙНУ ТА СМАРТ КОНТРАКТІВ. РОЗРОБКА ІННОВАЦІЙНИХ ПІДХОДІВ ДО ЦИХ ПРОБЛЕМ Є ВАЖЛИВОЮ ДЛЯ ПІДТРИМКИ СТАЛОГО РОЗВИТКУ МІСТ І ПІДВИЩЕННЯ ЯКОСТІ ЖИТТЯ ГРОМАДЯН
- **МЕТОЮ І ЗАДАЧЕЮ ДАНОЇ РОБОТИ** Є РОЗРОБКА ЗАХИЩЕНОЇ СИСТЕМИ СПІЛЬНОГО ВИКОРИСТАННЯ АВТОМОБІЛІВ НА ОСНОВІ СМАРТ КОНТРАКТІВ З МОЖЛИВІСТЮ АНОНІМІЗАЦІЇ ДАНИХ КОРИСТУВАЧІВ ДЛЯ УНЕМОЖЛИВЛЕННЯ ЇХНЬОГО ВІДСЛІДКОВУВАННЯ.

СМАРТ-КОНТРАКТИ

- СМАРТ-КОНТРАКТИ – ЦЕ ПРОГРАМНІ КОДИ, ЯКІ АВТОМАТИЗОВАНО ВИКОНУЮТЬ, ПЕРЕВІРЯЮТЬ АБО ВИКОНУЮТЬ УГОДИ В РАМКАХ БЛОКЧЕЙН-ПЛАТФОРМИ. ВОНИ Є ОСНОВОЮ ДЛЯ РЕАЛІЗАЦІЇ РІЗНОМАНІТНИХ ДЕЦЕНТРАЛІЗОВАНИХ ДОДАТКІВ, ЯКІ НЕ ВИМАГАЮТЬ ЦЕНТРАЛІЗОВАНОГО КОНТРОЛЮ.

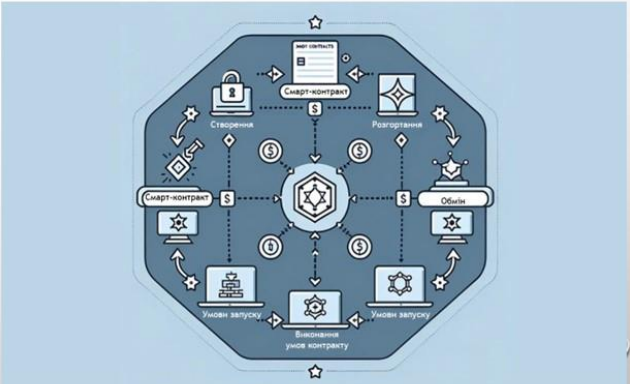
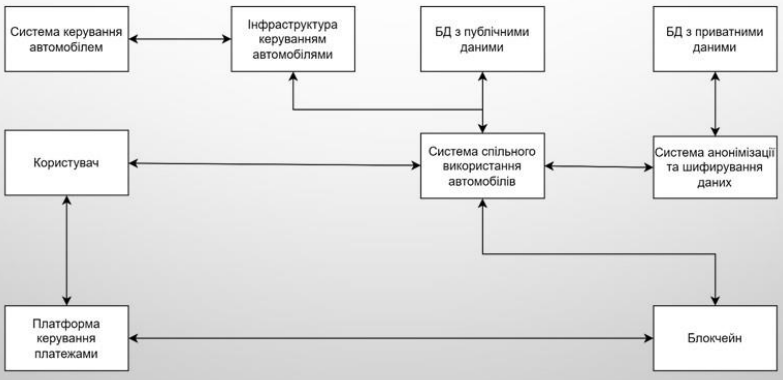
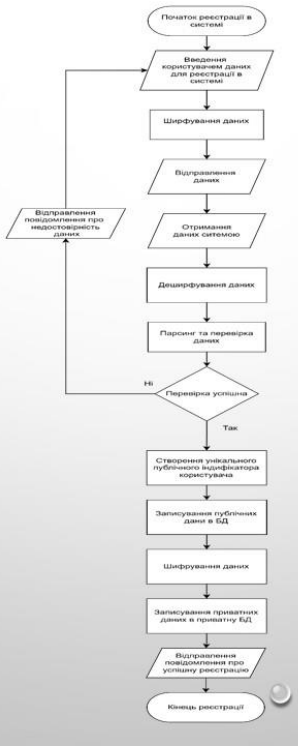


СХЕМА РОБОТИ СМАРТ-КОНТРАКТУ У СИСТЕМІ СПІЛЬНОГО ВИКОРИСТАННЯ АВТОМОБІЛІВ

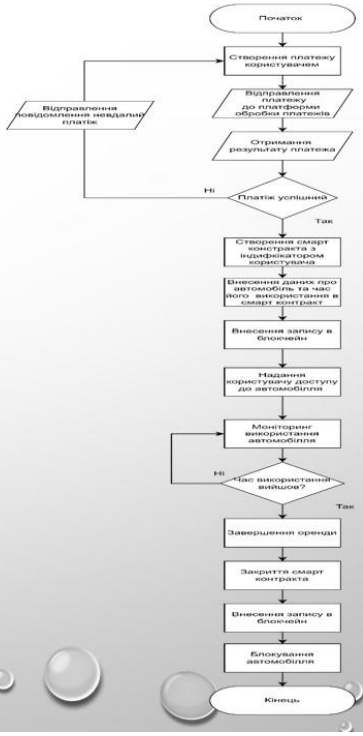
РОЗРОБЛЕНА СХЕМА ОБМІНУ ДАНИМИ В СИСТЕМІ



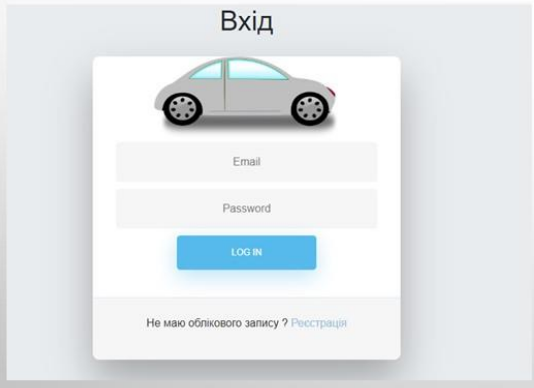
АЛГОРИТМИ АВТОРИЗАЦІЇ ТА РЕЄСТРАЦІЇ КОРИСТУВАЧІВ

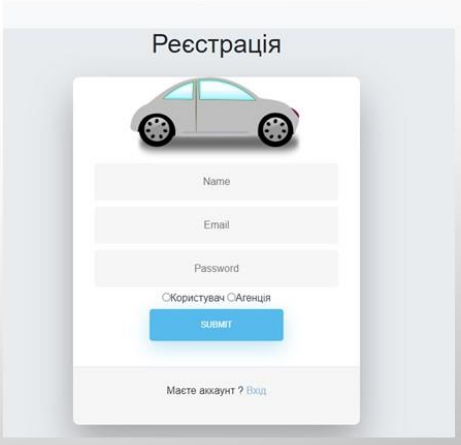


АЛГОРИТМ НАДАННЯ ДОСТУПУ ДО АВТОМОБІЛЯ

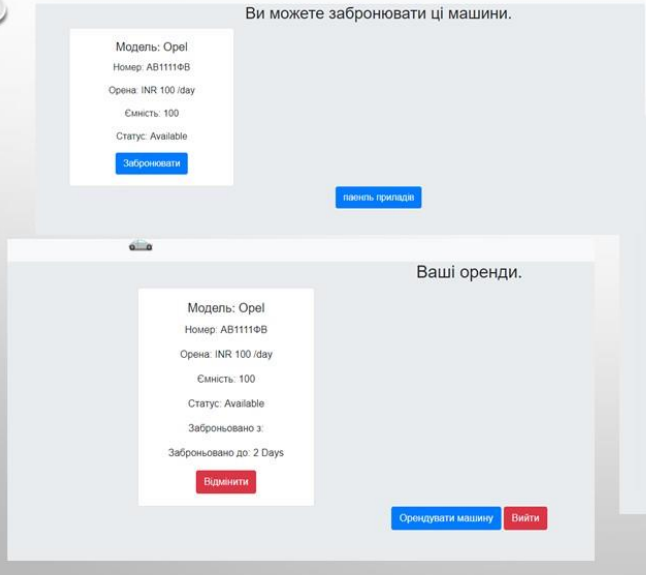


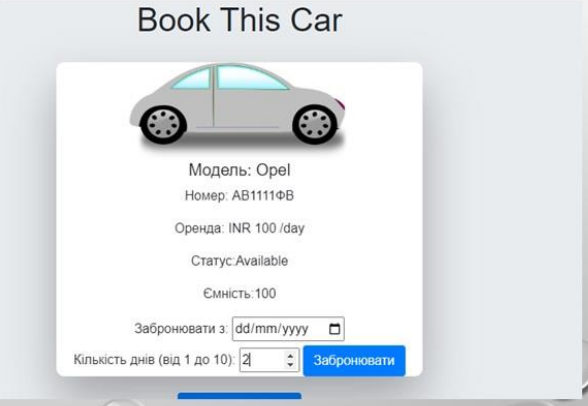
ІНТЕРФЕЙС ДОДАТКУ



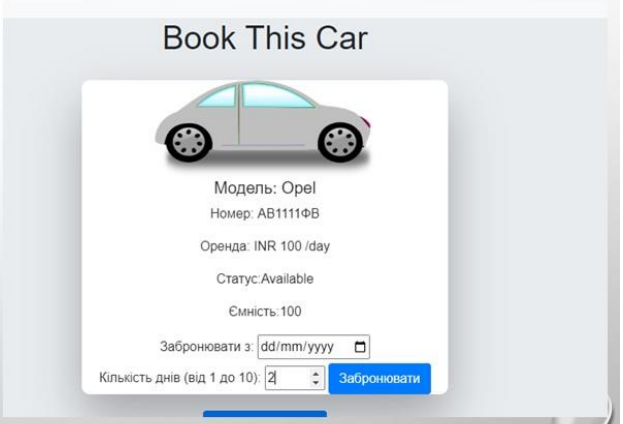


ІНТЕРФЕЙС ДОДАТКУ

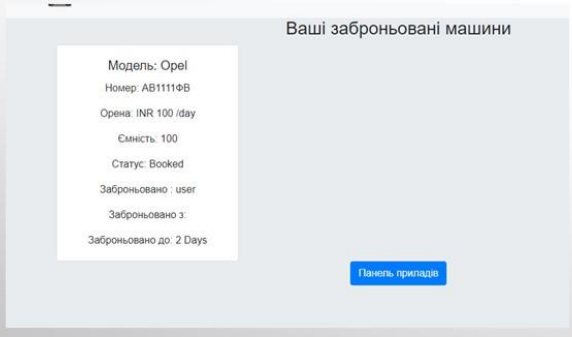


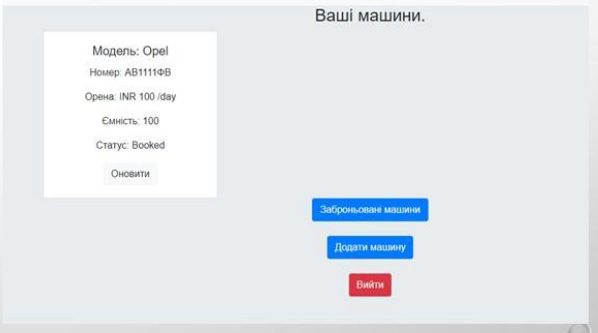


ІНТЕРФЕЙС ДОДАТКУ

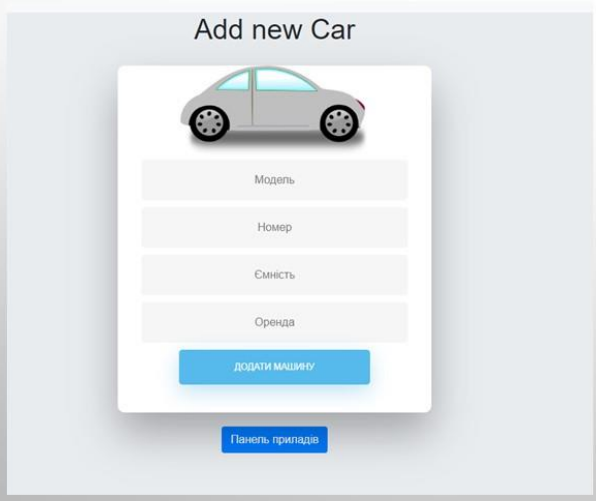


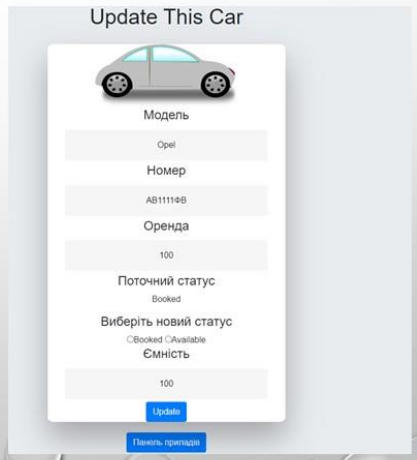
ІНТЕРФЕЙС ДОДАТКУ



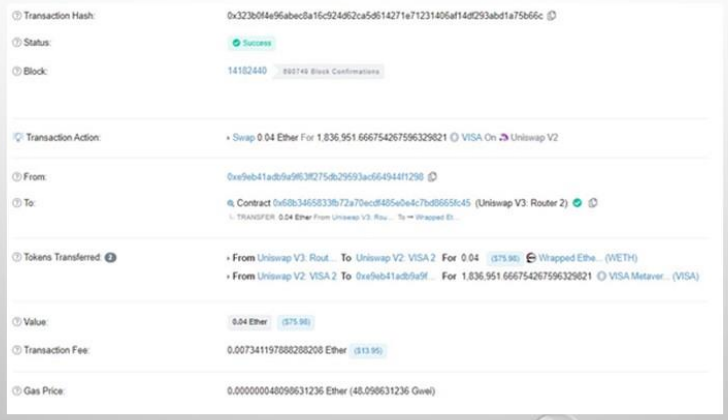


ІНТЕРФЕЙС ДОДАТКУ





ДАНІ СМАРТ КОНТРАКТУ



ВИСНОВКИ

- РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ ПОКАЗАЛИ, ЩО ВИКОРИСТАННЯ СМАРТ КОНТРАКТІВ ТА ТЕХНОЛОГІЙ БЛОКЧЕЙНУ Є ЕФЕКТИВНИМ СПОСОБОМ УПРАВЛІННЯ ДОСТУПОМ ДО АВТОМОБІЛІВ. ЦЕ ДОЗВОЛЯЄ ЗАБЕЗПЕЧИТИ ПРОЗОРІСТЬ І НАДІЙНІСТЬ СИСТЕМИ, А ТАКОЖ ЗНИЗИТИ ВИТРАТИ НА ПОСЕРЕДНИКІВ. РОЗРОБЛЕНИЙ ПІДХІД ДО АНОНІМІЗАЦІЇ ДАНИХ КОРИСТУВАЧІВ ВІДПОВІДАЄ СУЧАСНИМ ВИМОГАМ ДО ЗАХИСТУ ПРИВАТНОСТІ І МОЖЕ БУТИ ВИКОРИСТАНИЙ ДЛЯ ПОДАЛЬШОГО РОЗВИТКУ ПОДІБНИХ СИСТЕМ.
- ОТРИМАНІ РЕЗУЛЬТАТИ МОЖУТЬ БУТИ ВИКОРИСТАНІ ДЛЯ ПОДАЛЬШОГО ВДОСКОНАЛЕННЯ СИСТЕМ СПІЛЬНОГО ВИКОРИСТАННЯ АВТОМОБІЛІВ ТА РОЗШИРЕННЯ ЇХНЬОГО ЗАСТОСУВАННЯ В РЕАЛЬНИХ УМОВАХ. ВРАХОВУЮЧИ ЗРОСТАЮЧИЙ ІНТЕРЕС ДО БЛОКЧЕЙН ТЕХНОЛОГІЙ ТА ЗАХИСТУ ДАНИХ, ДОСЛІДЖЕННЯ В ЦІЙ ОБЛАСТІ МАЄ ВЕЛИКИЙ ПОТЕНЦІАЛ ДЛЯ ПОДАЛЬШОГО РОЗВИТКУ І ВПРОВАДЖЕННЯ.

ДЯКУЮ ЗА УВАГУ!

Додаток Г. Протокол перевірки на антиплагіат

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Захищена система спільного використання автомобілів на основі смарт контрактів з можливістю анонімізації даних користувачів для унеможливлення їхнього відслідковування

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ: Кафедра менеджменту та безпеки інформаційних систем
Факультет менеджменту та інформаційної безпеки
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 97 % Схожість 3 %

Аналіз звіту подібності (відмітити потрібне):

1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.

3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ (підпис) Коваль Н.П. (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ (підпис) Саврацький В.М. (прізвище, ініціали)

Керівник роботи _____ (підпис) Яремчук Ю.Є. (прізвище, ініціали)