Вінницький національний технічний університет
Факультет інформаційних технологій та комп’ютерної інженерії
Кафедра комп’ютерних наук

Пояснювальна записка
до магістерської кваліфікаційної роботи

на тему: «Інформаційна технологія управління криптовалютними активами»

Виконав: студент 2 курсу,
групи 1КН-18м
спеціальності 122 «Комп'ютерні науки»
Марков Д.Е.
Керівник: к.т.н., доцент. каф. КН, Сілагін О. В.
Рецензент:

Вінниця, 2019 рік

ЗАТВЕРДЖУЮ

Завідувач кафедри КН

д.т.н., проф.. Яровий А.А.

_____

(підпис)

“____” _____2019 року

ЗАВДАННЯ

на магістерську кваліфікаційну роботу на здобуття кваліфікації магістра зі спеціальності: 122 – «Комп'ютерні науки»

_____

08-22.МКР.010.18.000.ПЗ

Магістранта групи 1КН-18м Марков Дмитро Едуардович

Тема магістерської кваліфікаційної роботи: «Інформаційна технологія управління криптовалютними активами»

Вхідні дані: Мова програмування: об’єктно-орієнтована, середовище розробки: підтримує роботу з об`єктно-орієнтованими мовами; Операційна система Android та iOS. Реляційна база даних з потужністю до 1000 записів, підтримка стандарту ODBS.

Короткий зміст частин магістерської кваліфікаційної роботи:

1. Графічна: структурна схема функціонування інформаційної технологія управління криптовалютними активами, діаграма класів інформаційної технологія управління криптовалютними активами, схема алгоритму купівлі криптовалюти, схема ієрархічної генерації ключів, приклад роботи додатку.

2. Текстова (пояснювальна записка): вступ, обґрунтування доцільності розробки інформаційної технології управління криптовалютними активами, моделювання інформаційної технології управління криптовалютними активами, математична модель ієрархічної генерації ключів, програмна реалізація інформаційна технологія управління криптовалютними активами, економічна частина, висновки, перелік використаних джерел, додатки.

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	Кінець	
1	Обґрунтування доцільності розробки інформаційної технології управління криптовалютними активами.			Аналітичний огляд літературних джерел, задачі досліджень, розділ 1 ПЗ
2	Моделювання інформаційної технології управління криптовалютними активами			Математичні моделі, розділ 2
3	Програмна реалізація інформаційної технології управління криптовалютними активами			розділ 3
4	Підготовка економічної частини			розділ 4
5	Апробація та/або впровадження результатів дослідження			тези доповідей/акт впровадження
6	Оформлення пояснювальної записки, графічного матеріалу та презентації			Пояснювальна записка, графічний матеріал, презентація

Консультанти з окремих розділів магістерської кваліфікаційної роботи

1. Науковий керівник _____ канд. техн. наук, доц., доц. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)

“____”_____20___р. О.В. Сілагін.
ініціали та прізвище

2. Економічна частина _____ канд. екон. наук, доц. каф. ЕПВМ
(підпис) наук. ступінь, вченезвання (посада)

“____”_____20___р. М.В. Бальзан
ініціали та прізвище

Дата попереднього захисту роботи “____”_____20___р.

Рецензент _____ доц., доцент кафедри ПЗ.
(підпис) наук. Ступінь, вчене звання (посада)

В.В. Войтко

Завдання видав науковий керівник _____ канд. техн. наук, доц., доц. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)

“____”_____20___р. О. В. Сілагін
ініціали т апрізвище

Завдання отримав магістрант _____ Д.Е. Марков
(підпис) ініціали та прізвище

“____”_____20___р.

АНОТАЦІЯ

Магістерська робота присвячена розробці інформаційної технології управління криповалютними активами. Були розглянуто і визначено основні переваги і недоліки технології Blockchain. Було проаналізовано програмне і апартне забезпечення для управління криптовалютними активами, на основі яких було сформовано вимоги до програмного модуля. Розроблено загальну схему алгоритму роботи даної інформаційної технології. Створений програмний додаток написаний на мові програмування JS, програмне забезпечення характеризується зручністю та зрозумілістю інтерфейсу, швидкістю та точністю опрацювання даних, що забезпечує всі вимоги користувача щодо організації процесу.

ABSTRACT

The master's thesis is devoted to the development of information technology for managing crypto-currency assets. The main advantages and disadvantages of Blockchain technology were considered and identified. The software and hardware for managing cryptocurrency assets were analyzed, on the basis of which the requirements for the software module were formed. The general scheme of algorithm of work of this information technology is developed. Created software application written in the programming language JS, the software is characterized by the convenience and clarity of the interface, speed and accuracy of data processing, which provides all the requirements of the user regarding the organization of the process.

ЗМІСТ

ВСТУП

Актуальність теми дослідження. Одна з великих проблем людського суспільства - відсутність довіри. Люди не довіряють бізнесу, бізнес не довіряє своїм партнерам і людям, люди не довіряють один одному. І для цього, на жаль, є підстави. Для вирішення проблеми довіри створені інститути посередників. Нотаріус посвідчує заповіт, і це служить захистом від його подальшої підробки. Банк гарантує магазину оплату за випущеної їм для вас пластиковій картці, гарантуючи, що на вашому рахунку достатньо грошей.

Однак з'явилася інша проблема - проблема довіри до посередника. Так, довіра до паперу, завіреної нотаріусом, вище, але хто може гарантувати, що він встоїть перед хабаром і не виправить заповіт заднім числом? Хто гарантує, що у банку не відкличуть ліцензію і послані через нього гроші не пропадуть?

Нові технології тільки покращували процес взаємодії, не зазіхаючи на сам інститут посередників. Так, платіжку в банк тепер потрібно підписувати електронним підписом, а не ставити на роздрукований бланк платіжного доручення підпис і печатку. Так, це надійніше. Будь-яку друк можна замовити по Інтернету, а підпис підробити.

Довіра підвищилася, але проблема до кінця не вирішилася. У засвідчувального центру можуть бути всі потрібні сертифікати, але це не вбереже від людського фактора. Ваш підпис зможе просто скопіювати адміністратор. І чи є гарантія, що співробітник, що засвідчує, не видасть ваш електронний підпис зловмисникові по фальшивому або недійсним паспортом або за підробленою довіреністю? Крім того, посередники коштують грошей. Так що цілком зрозуміло бажання на них заощадити. Наприклад, за допомогою технології блокчейн.

Блокчейн - це децентралізована база даних, заснована на одноранговій (p2p) мережі, загальному реєстрі і криптографії публічного і приватного ключа. Увійшовши в блокчейн-мережу, користувач підключається до інших комп'ютерів мережі для того, щоб обмінюватися з ними даними: блоками і записами. Отримавши

нові дані, кожен користувач перевіряє їх коректність, і, переконавшись у достовірності, зберігає їх у себе, а також передає коректні дані далі по мережі.

Учасники мережі діляться на дві групи: звичайні користувачі, які створюють нові записи, і Майнер, які створюють блоки. Звичайні користувачі створюють і поширюють через мережу записи, наприклад, про грошові перекази або про передачу прав власності. Майнер збирають записи, перевіряють їх і записують в блоки, а потім розсилають ці блоки по мережі. Після чого звичайні користувачі отримують блоки і зберігають їх у себе, щоб можна було коректно створювати свої і достовірно перевіряти чужі нові записи.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Мета роботи – підвищення швидкості роботи системи, а також розширення функціональних можливостей по управлінню криптовалютними активами.

Об'єктом дослідження є процес управління криптовалютними активами на базі технології blockchain**.**

Предметом дослідження є інформаційні технології, моделі, алгоритми та програмні засоби в області управління криптовалютними активами.

Для досягнення мети, що була описана раніше, необхідно виконати ряд задач:

- обґрунтування доцільності розробки інформаційної технології управління криптовалютними активами;
- моделювання інформаційної технології управління криптовалютними активами;
- програмна реалізація інформаційної технології управління криптовалютними активами;

- тестування і аналіз результатів роботи інформаційної технології управління криптовалютними активами;
- виконати задачі економічної частини;

Методи дослідження. У роботі використані наступні методи наукових досліджень: криптографічні хеш функції, ієрархічна генерація ключів, технологія Blockchain, алгоритми консенсусу блокчейн мереж, методи криптографічного шифрування, методи розподілених обчислень, об'єктно-орієнтованого програмування для автоматизації розрахунків.

Наукова новизна одержаних результатів полягає в наступному:

− удосконалено інформаційну технологію управління криптовалютними активами, за рахунок застосуванням некастодіального підходу для зберігання криптовалюти та використання мнемонічної фрази, що підвищує швидкість роботи із криптовалютними активами.

− вперше запропоновано модель купівлі і продажу криптовалюти не вимагаючи верифікації користувача, за рахунок використання p2p переводів, що підвищує швидкість роботи із криптовалютними активами.

Практичне значення одержаних результатів полягає у наступному:

1. Розроблено алгоритми підвищення швидкості інформаційної технології управління криптовалютними активами на основі некастодіального підходу зберігання криптовалюти і застосування мнемонічної фрази.

2. Розроблено програмне забезпечення для зберігання, відправки, продажу, купівлі і обміну криптовалют.

3. Програма і алгоритми розроблені в даній роботі впровадженні на підприємстві ФОП « Груша В.В.».

Розроблені алгоритми можуть бути впроваджені в начальний процес у якості лекції на тему «Блокчейн-технологія» дисципліни «GPGPU-технології».

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими, та збіжністю

результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок магістранта. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно.

Апробація результатів роботи. Результати роботи були апробовані на конференціях «Наукові відкриття та фундаментальні наукові дослідження: світовий досвід 2019», «Міжнародна НПК ІОН-2018» та «Молодь в науці: дослідження, проблеми, перспективи (МН-2020)», опубліковані в збірниках наукових праць. Програми і алгоритми розроблені в даній роботі плануються до впровадження на підприємстві ФОП «Груша В.В.».

Публікації. За результатами досліджень опубліковано тези доповіді науково-технічної конференції [1,2,3].

1 ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ УПРАВЛІННЯ КРИПТОВАЛЮТНИМИ АКТИВАМИ

1.1 Аналіз предметної області управління криптовалютними активами

Блокчейн - це тип розподіленої бази даних, яка зберігає записи цифрових транзакцій. Замість того, щоб мати центрального адміністратора, як традиційні бази даних (банки, уряд і бухгалтерія), вона має мережу тиражованих баз даних, синхронізовану через Інтернет і яку видно всім користувачам в мережі.

Коли цифрова угода здійснюються в блокчейн, вона групується в криптографічно захищеному блоці з іншими угодами, які відбулися в останні декілька хвилин і розсилається по всій мережі.

Підтверджений блок транзакцій потім датується і додається до ланцюга в лінійному, хронологічному порядку. Нові блоки перевірених транзакцій пов'язані з більш старими блоками, утворюють ланцюжок блоків, які показують кожну транзакцію, досягнуту в історії цього блокчейну[1].

Існує два види ланцюжка:

- Публічний Blockchain - відкрита, доповнювальна база даних. Такий вид блокчейна використовується в криптовалютіBitcoin. Кожен учасник може записувати і читати дані.
- Приватний блокчейн має обмеження по запису / читання даних. Можуть встановлюватися пріоритетні вузли. Підвид PrivateBlockchain - ексклюзивний блокчейн. В такому ланцюжку встановлюється група осіб, що займається обробкою транзакцій.

Підбиваючи попередні підсумки, перерахуємо ключові особливості Blockchain:

- Децентралізація - в ланцюжку немає сервера. Кожен учасник - це і є сервер. Він підтримує роботу всього блокчейна;
- Прозорість - інформація про транзакції, контрактах і так далі зберігається у відкритому доступі. При цьому ці дані неможливо змінити;

- Теоретична необмеженість - теоретично блокчейн можна доповнювати записами до нескінченності. Тому його часто порівнюють з суперкомп'ютером;
- Надійність - для запису нових даних необхідний консенсус вузлів блокчейна (рисунок 1.1). Це дозволяє фільтрувати операції і записувати тільки легітимні транзакції. Здійснити підміну хеша майже нереально[2].

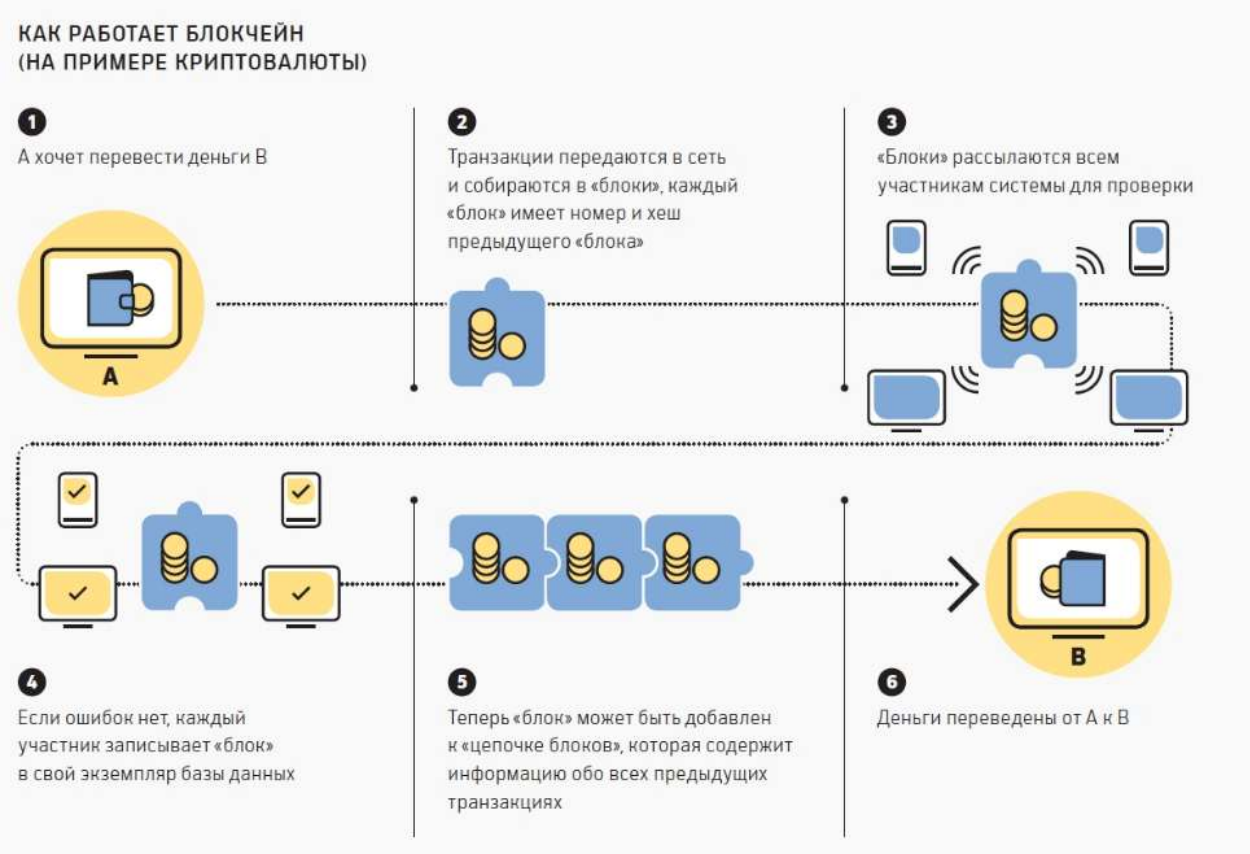


Рисунок 1.1 – Приклад роботи блокчейну

Конструкція усієї системи – це послідовність блоків(ланцюжок), а не замкнуте коло чи щось інше. Кожен з блоків містить масив певних даних і всі блоки пов'язані між собою. Тобто, новий «масив» може бути створений тільки після того, як закритий старий масив (рисунок 1.2).

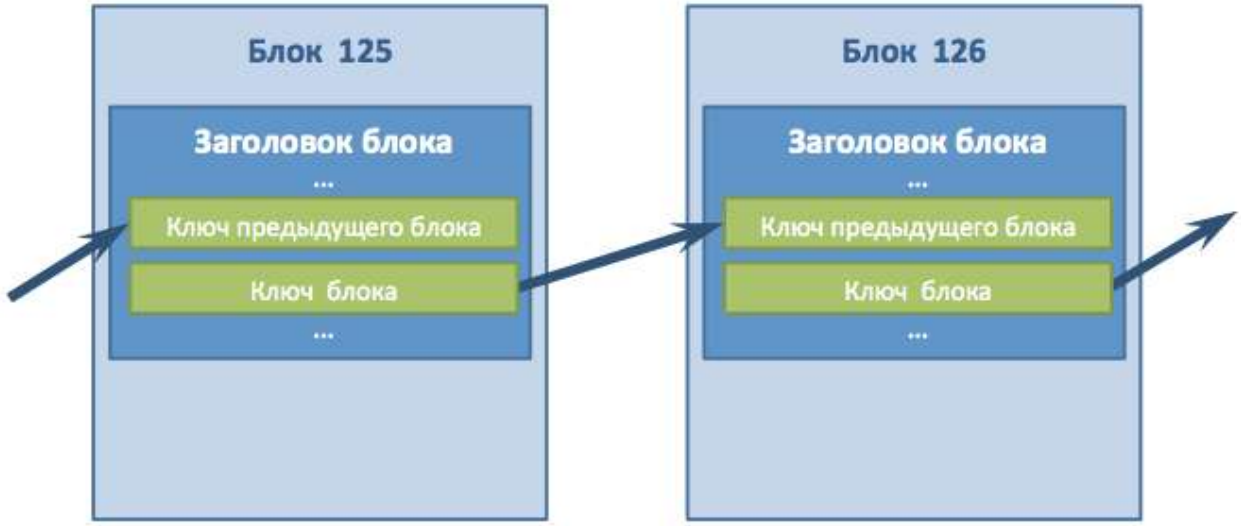


Рисунок 1.2 – Послідовність блоків

Блок транзакцій - спеціальна структура для запису групи транзакцій в блокчейні. Транзакція вважається завершеною і достовірної («підтвердженої»), коли перевірені її формат і підписи, і коли сама транзакція об'єднана в групу з декількома іншими і записана в спеціальну структуру - блок. Вміст блоків може бути перевірено, так як кожен блок містить інформацію про попередньому блоці. Всі блоки збудовані в один ланцюжок, який містить інформацію про всі вчинені коли-небудь операції в базі. Найперший блок в ланцюжку - первинний блок (англ. Genesisblock) - розглядається як окремий випадок, так як у нього відсутній батьківський блок.

Блок складається з заголовка і списку транзакцій. Заголовок блоку включає в себе свій хеш, хеш попереднього блоку, хеші транзакцій і додаткову службову інформацію. В системі Bitcoin першою транзакцією в блоці завжди вказується отримання комісії, яка стане нагородою користувачеві за створений блок. Далі йде список транзакцій, сформований з черги транзакцій, ще не записаних в попередні блоки. Критерій відбору з черги задає майнер самостійно. Це не обов'язково повинна бути хронологія за часом. Наприклад, можуть включатися тільки операції з високою комісією або за участю заданого списку адрес. Для транзакцій в блоці використовується деревоподібна хешування, аналогічне формування хеш-суми для файлу в протоколі BitTorrent. Транзакції, крім нарахування комісії за створення блоку, містять всередині параметра input посилання на транзакцію з попереднім

станом даних (в системі Bitcoin, наприклад, дається посилання на ту транзакцію, по якій були отримані біткоїни, що витрачаються). Операції з передачі майнеру комісії за створення блоку не мають «вхідних» транзакцій, тому в даному параметрі може вказуватися будь-яка інформація (для них це поле зветься англ. Coinbaseparameter)[3].

Створений блок буде прийнятий іншими користувачами, якщо числове значення хешу заголовка дорівнює або менше певного цільового числа, величина якого періодично коригується. Так як результат хешування функції SHA-256 вважається незворотнім, на даний момент немає алгоритму отримання бажаного результату, крім випадкового перебору. Якщо хеш не задовольняє умові, то в заголовку змінюється параметр nonce і хеш перераховується. Зазвичай потрібна велика кількість перерахунків. Коли варіант знайдений, вузол розсилає отриманий блок іншим підключеним вузлів, які перевіряють блок. Якщо помилок немає, то блок вважається доданим в ланцюжок і наступний блок повинен включити в себе його хеш.

Доступ до Blockchain відбувається з використанням спеціальних ключів, які гарантують надійність всієї мережі. Він є у кожного користувача. Ключ являє собою набір криптографічних записів. Він абсолютно унікальний, що гарантує неможливість підміни даних і хакерських атак. Щоб це здійснити, хакерам потрібно отримати доступ до всіх комп'ютерів мережі[3].

Механізми, що забезпечують дієздатність і надійність блокчейна - це алгоритми ProofofWork або PoW, виконаної роботи, і ProofofStake або PoS, підтвердження частки. Завдяки ним в блокчейне досягається консенсус.

Алгоритм ProofofWork застосовується в блокчейнеBitcoin. Механізм його роботи схожий зі звітністю в офісі. Співробітники регулярно складають звіти для перевірки, що підтверджують факт того, що вони виконали певне завдання. Без цього вони не отримають зарплату, так як не підтвердили факт виконаної роботи.

Якщо в ProofofWork на перший план виходить обчислювальна потужність, то в ProofofStake роль відіграє баланс гаманця. Здійснення і підтвердження транзакцій відбуватиметься без активної участі обчислювальної техніки, а завдяки активним монетам на гаманцях. В ідеалі, всі власники криптовалюти на блокчейні з PoS

виступатимуть в ролі інвесторів. Роль майнінгу відійде на другий план. Однак у алгоритму є істотні недоліки - можливо проведення дублюючих транзакцій[4].

1.2. Аналіз програмного і апаратного забезпечення для зберігання і управління криптовалютними активами

У звичайному світі люди звикли зберігати гроші в гаманці, і якщо провести аналогію між паперовими грошима і криптовалютами, то у них виявиться багато спільного. Це означає, що і криптовалюта теж повинна зберігається в гаманці. Однак, якщо звичайний гаманець може мати лише кілька параметрів і призначений тільки для зберігання грошей, то ось в криптовалютних гаманцях все дещо складніше. За типом зберіганнякриптовалюти їх можна розділити на «гарячі» і «холодні»; за типом зберігання приватних ключів - на «кастодіальні» і «не кастодіальні», а за типом інсталяції - на локальні, мобільні, апаратні, браузерні і паперові (так-так, саме паперові).

Криптовалютний гаманець - це додаток, за допомогою якого можна зберігати криптовалюту. Хоча і фізично він ніде не зберігається - користувачам просто дають дані, які забезпечують доступ до свого рахунку. Ці дані, в залежності від типу гаманця, можуть являти собою стандартну пару «емейл + пароль», приватний ключ або seed-фразу. Основним завданням гаманця є зберігання, а також можливість відправляти і отримувати криптовалюти від інших людей[5].

Відмінність між гарячим і холодним гаманцем полягає в тому, що гарячий гаманець працює при підключенні до інтернету, а холодний може працювати і без. Гарячі електронні гаманці менш захищені, так як існує ризик крадіжки ваших персональних даних через інтернет, проте, при цьому вони більш затребувані серед користувачів. Холодні електронні гаманці ж застосовуються для "холодного зберігання" криптовалюти, тому вони більш безпечні.

Суть кастодіальних гаманців в тому, що вони НЕ дають доступу до свого приватному ключу, а просто зберігають його на своєму централізованому сервері. Найчастіше таке рішення надають криптовалютні біржі. Плюс такого рішення в тому,

що можна відновити доступ до облікового запису через пошту, якщо пароль був загублений. Мінус – обліковий запис може бути заморожений в разі якогось втручання, а для розморожування користувача можуть попросити пройти процедуру KYC. Також користувач може втратити гроші під час хакерських атак, що останнім часом є дуже популярним подією.

Некастодіальні електронні гаманці працюють навпаки - вони надають повний контроль над своїми приватними ключами, не використовуючи сервер. Величезним плюсом такого рішення є те, що кошти належать тільки користувачу. Ніхто інший не зможе ними заволодіти без його seed-фрази. Однак, в цьому полягає і мінус такого гаманця, так як, якщо seed-фразабуде втрачена, то доступ до гаманця вже ніяк не вийде повернути.

Локальний (десктопний) гаманець - це програма, яка встановлюється на стаціонарний комп'ютер або ноутбук. Даний вид гаманців є одним з найскладніших для користувачів, але при цьому володіє найкращими показниками з безпеки і анонімності. Потрібно відзначити, що найчастіше їх використовують досвідченні користувачі або компанії, які проводять розробки на блокчейні. Десктопні електронні гаманці можна розділити на 2 вида:

- Товстий гаманець –в даному випадку, мається на увазі завантаження на комп'ютер повної копіїблокчейна. За фактом товстий гаманець криптовалют - це повна нода мережі, яка не тільки дозволяє вам керувати своїм рахунком, але і підтримує роботу блокчейна. З огляду на, що блокчейнтого ж біткоїна займає вже близько 250Гб, то для роботи гаманця відповідно потрібно високопродуктивне «залізо»;
- Тонкий гаманець –на відміну від товстого гаманця, займає на комп'ютері всього кілька мегабайт пам'яті і встановлюється за пару хвилин. Це програма-клієнт, для роботи якої не потрібно завантажувати на комп'ютер увесь блокчейн. Він дозволяє створювати адреси криптовалют і виконувати транзакції. З блокчейном тонкий гаманець взаємодіє не безпосередньо, як товсті гаманці, а через сервер розробників програми. Тому вони вважаються менш захищеними, але зате набагато зручніше у використанні.

Апаратний гаманець криптовалют це окремий пристрій, що на вигляд нагадує «флешку». Такий блокчейн гаманець служить для «холодного» зберігання криптовалют і підключається до інтернету тільки тоді, коли потрібно зробити транзакцію. Апаратні гаманці надають зручний доступ до блокчейну з високим ступенем захисту, так як приватні ключі зберігаються тільки в пам'яті самого пристрою. Незважаючи на їх вартість - від 60 до 100 доларів, вони дозволяють здійснювати транзакції таким чином, що хакери не можуть до них дістатися. При втраті такого гаманця ніхто крім вас не зможе нічого зробити із засобами, при цьому ви з легкістю зможете відновити до них доступ через seed-фразу на новий пристрій. Тому по співвідношенню надійності і зручності використання апаратні гаманці займають майже лідируючі позиції.

Web-гаманці або браузерні– це досить простий тип гаманців для використання, він не вимагає від користувача якихось особливих знань в криптовалютах, більш того має низку переваг:

- Користуватися гаманцем можна на різних пристроях, незалежно від вашого місця знаходження, головне, щоб був вільний доступ до Інтернету;
- Немає необхідності в скачуванні всіх блоків мережі, що економить багато часу і вільного дискового простору;
- У більшості, подібні сервіси пропонують своїм користувачам додаткові зручності, такі як відсутність комісії на перекази між користувачами, відправка монет іншим його користувачам на адресу електронної пошти або номер телефону;

Однак, ви повинні пам'ятати, що такі гаманці мають «кастодіальних» рішення. При використанні Web-гаманця доступ до коштів має і сторонній сервіс. Тому, їх збереження залежить вже не тільки від самого користувача. При зломі такого ресурсу монети користувачів найімовірніше будуть вкрадені.

Мобільні гаманці криптовалют – цегаманці криптовалют, що можна встановити на мобільні пристрої (смартфони, планшети). Потрібно відзначити, що вони увібрали в себе всі кращі якості від перерахованих вище видів гаманців. Адже вони можуть бути «не кастодіальними», досить анонімними і при цьому надають доступ до

криптовалют в будь-якій точці світу, де є інтернет. Так як це окремий додаток, то найчастіше розробники наділяють його ще й корисними додатковими функціями.Що стосується безпеки, то мобільні гаманці займають «золоту середину», так як крім звичайного PIN-коду можуть мати прив'язку до відбитку пальця або FACE ID (зазвичай налаштовується користувачем за бажанням)[5].

1.3 Аналіз програм аналогів

1) Metamask

MetaMask - це розширення або мобільний додаток, який дозволяє працювати з блокчейномEthereum. Це дозволяє запускати EthereumdApps безпосередньо у вашому браузері без запуску повного вузла Ethereum.

MetaMask включає в себе безпечний ідентифікаційний сейф, який надає користувальницький інтерфейс для керування криптовалютними активами та підпису транзакцій у блокчейні.

Ви можете встановити додаток MetaMask у Chrome, Firefox, Opera та новий браузер Brave. Головне вікно програми представлено на рисунку 1.3.

2) Ledger Wallet Bitcoin & Altcoins / Ethereum / Ripple

Ця програма є менеджером облікового запису, де ви можете надсилати та отримувати транзакції Bitcoin та перевіряти свій баланс, аналогічно функціонує з Ethereum і Ripple.

Для кожної транзакції відправки ваша заявка попросить вас підтвердити вашу транзакцію. Головне вікно програми для Bitcoin & Altcoins зображено на рисунку 1.4, для Ethereum – на рисунку 1.5, Ripple - 1.6:

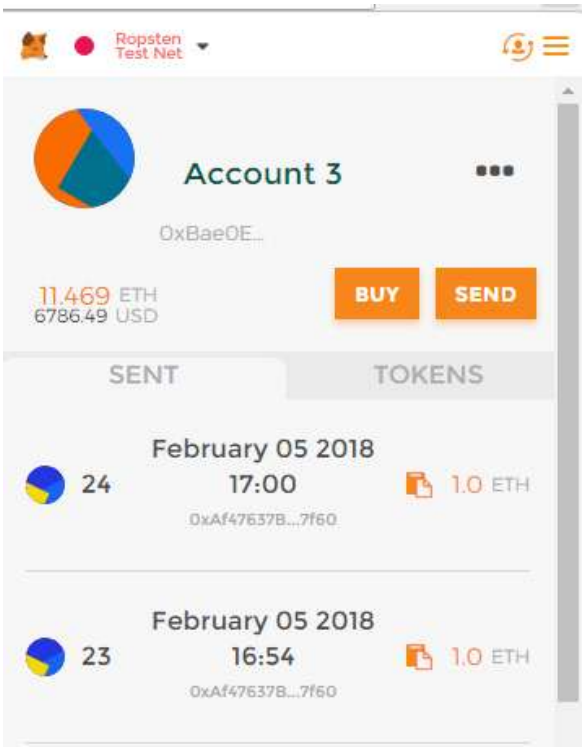


Рисунок 1.3 – Головне вікно браузерного розширення MetaMask

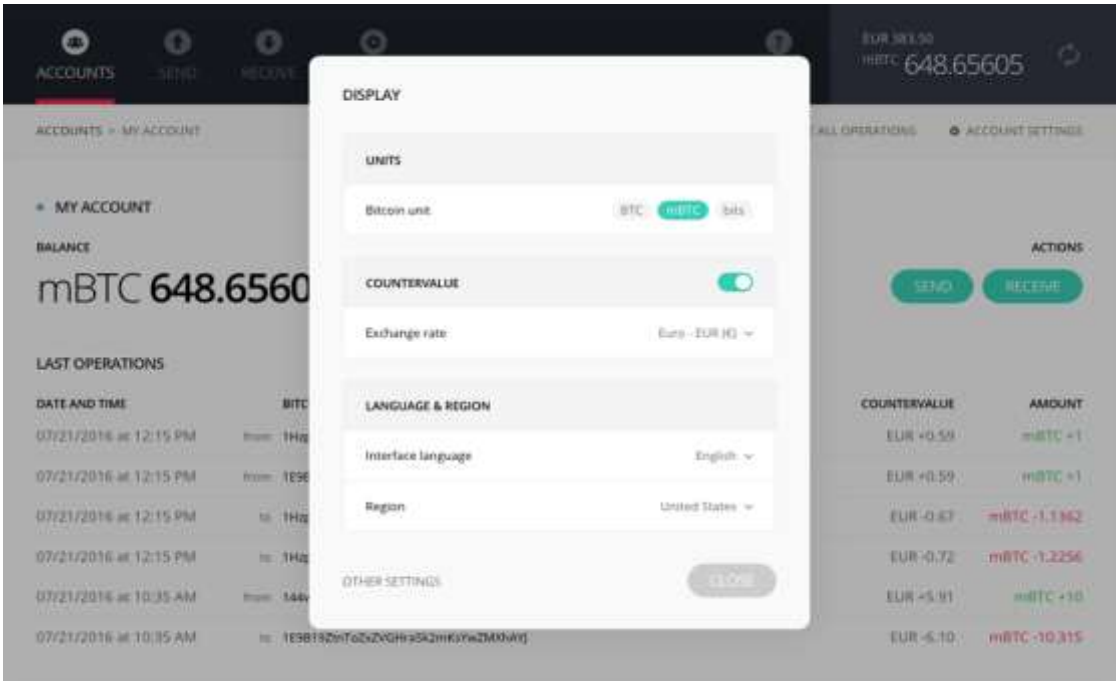


Рисунок 1.4 – Головне вікно десктопного додатку Ledger Wallet Bitcoin and Altcoins

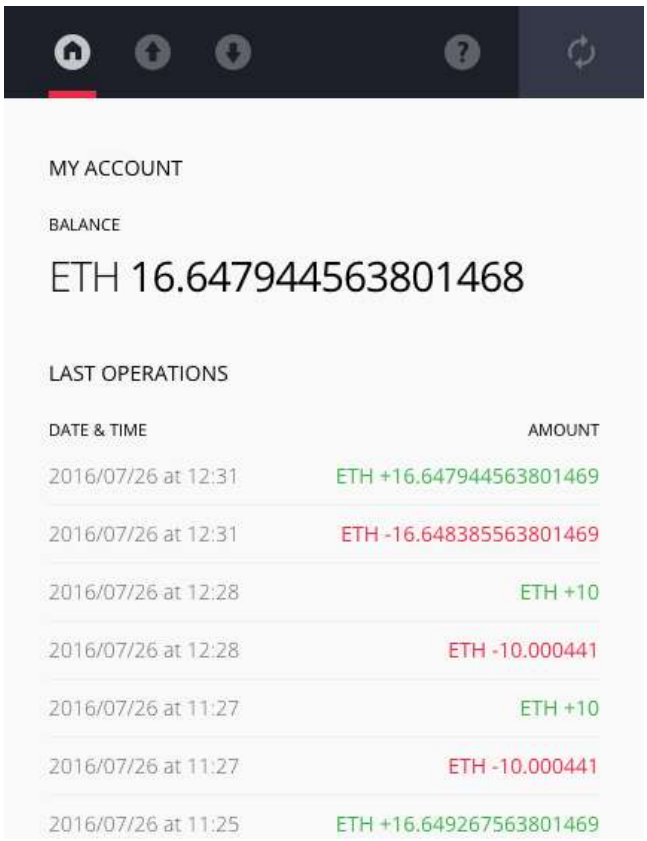


Рисунок 1.5 – Головне вікно мобільного додатку Ledger Wallet Ethereum

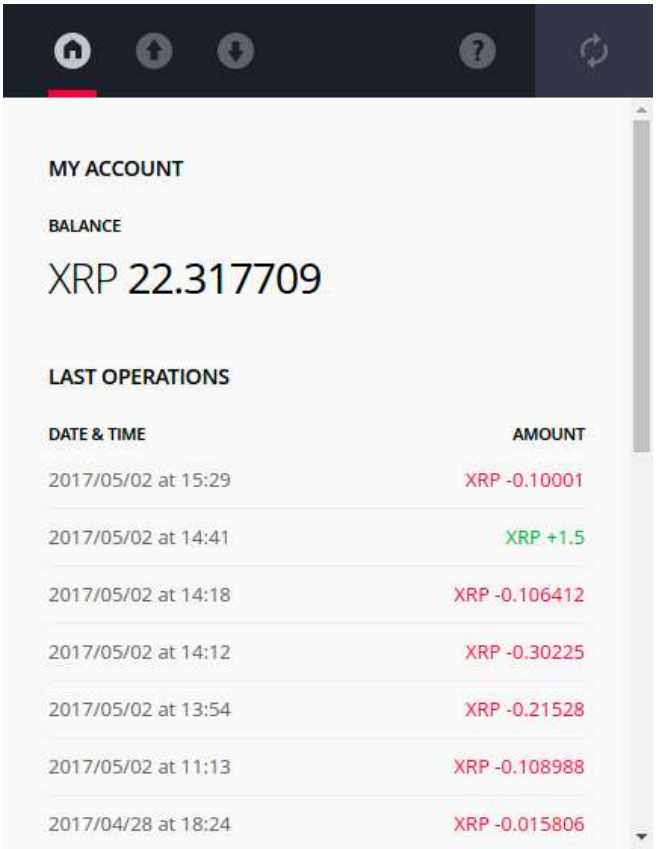


Рисунок 1.6 – Головне вікно мобільного додатку Ledger Wallet Ripple

3) Copay

Copay - це універсальний гаманець для Bitcoin від Bitpay, доступний на пристроях iOS, Android, Windows Phone, Linux, Max OS X та Windows. Оскільки Copay доступний на декількох платформах, легко використовувати один і той же гаманець або облікові записи на кількох пристроях.

Простий, чистий користувальницький інтерфейс Copay робить його гарним вибором для нових користувачів Bitcoin. Copay також є хорошим варіантом для підприємств через функцію спільного облікового запису, яка вимагає певної кількості користувачів на підпис кожної транзакції. Наприклад, два співзасновники могли б створити 2 з 2 гаманці, де обидва повинні будуть підписати кожну транзакцію. Робочі вікна програми зображено на рисунку 1.7.

4) ToastWallet

ToastWallet - це один із найбільш зручних гаманців, який використовується для зберігання RippleXRP. Процес зберігання Ripple в цьому гаманці досить простий, щоб кожен міг з легкістю ним користуватися.

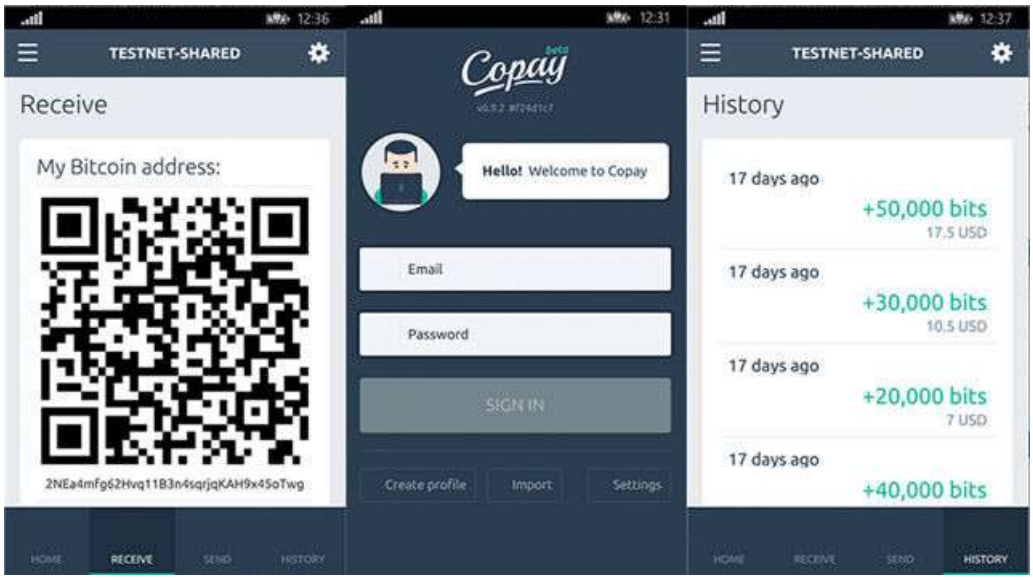


Рисунок 1.7 – Робочі вікна додатку Copay

Він пропонує велику безпеку для перерахованих коштів. Сама перша особливість цього гаманця полягає в тому, що він доступний майже для всіх платформ. Користувачі можуть використовувати його незважаючи на те, чи

використовують вони Windows, Android і навіть iOS. Окремі програми цього кошика доступні для кожної платформи.

Наступна перевага використання ToastWallet полягає в тому, що його легко використовувати. Користувачі не повинні виконувати різні кроки, щоб зберігати XRP, як у інших гаманцях.

Ще однією перевагою використання ToastWallet є те, що він не приймає плату за транзакцію та абсолютно безкоштовно для всіх користувачів. У вас лише 20XRP резерв для обробки ваших транзакцій. Робочі вікна цього додатку зображено на рисунку 1.8.

5) Exodus

Exodus - це мультивалютний десктопний гаманець, який підтримує 8 криптовалютних валют (включаючи Litecoin, Bitcoin, Ethereum). Він має дуже інтерактивний користувальницький інтерфейс і підтримується активним керівництвом групи JPRichardson та DanielCastagnoli. Він також надає своїм користувачам систему відновлення одним клацанням, щоб відновити гаманецьдостатньо фрази відтворення з 12 слів. Він навіть має вбудований ShapeShift-конвертер для любителів мульти-криптовалюти. Ви можете використовувати цю функцію, щоб обміняти будь-яку вашу валюту на іншу.

Exodus доступний на Mac, Linux та Windows. Головне вікно цього додатку зображено на рисунку 1.9:

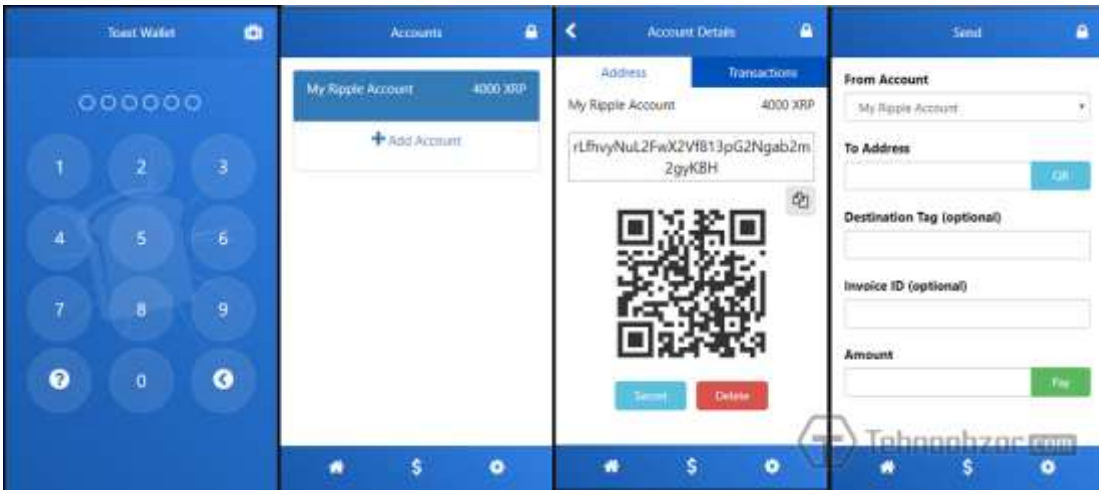


Рисунок 1.8 – Робочі вікна додатку Toast Wallet

Рисунок 1.9 – Головне вікно додатку Exodus

6) Paytomat Wallet

Paytomat Wallet дозволяє переводити кріптовалюти іншим особам як вручну, так і автоматично. Так, в першому випадку користувач самостійно вказує ім'я одержувача в системі EOS і суму, після чого підключається за допомогою програми до блокчейну - і транзакція встає в чергу. У другому випадку всі дані для транзакції користувач отримує через QR-код, наданий йому продавцем. Додаток самостійно створює транзакцію на зазначену суму, підключається до блокчейн — і платіж проходить, а користувач отримує тільки звітні дані про оплату.

На сьогодні Paytomat Wallet підтримує кілька десятків криптовалют. Серед них- Bitcoin, Ethereum, Litecoin, Dash, NET, Coin, Waves, TRON, EOS та інші, що працюють на базі EOS, ERC 20 і BIP-2. Також є підтримка NEW Mosaics, що є рідкістю на ринку.

Проте всі ці плюси, ставляться під сумнів, коли мова йде про купівлі/продаж/обмін криптовалют. Оскільки вони чи не реалізовані, чи працюють не коректно або повільно, що і є його найбільшим недоліком.

Головне вікно додатку Paytomat Wallet зображено на рисунку 1.10:

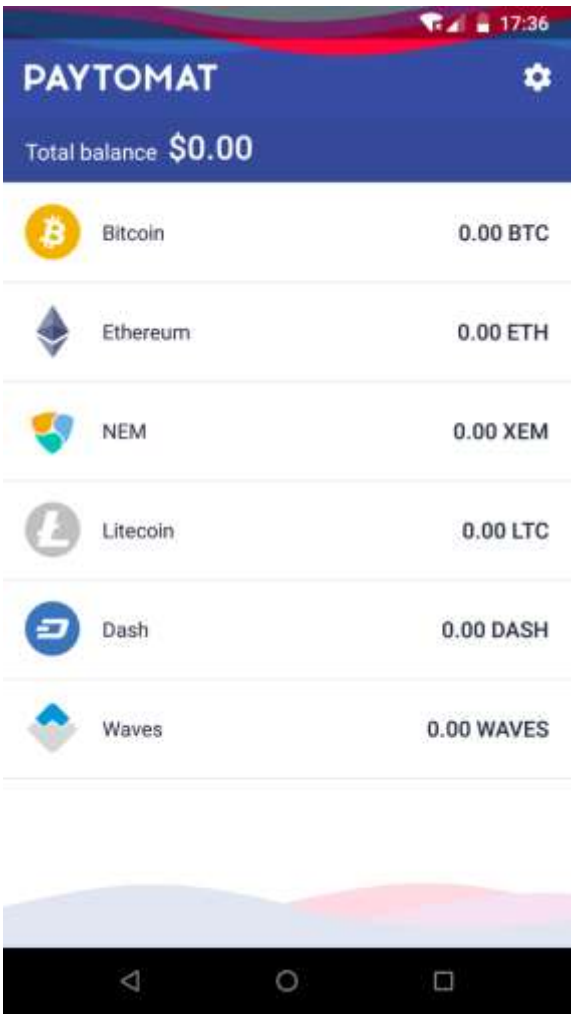


Рисунок 1.10 – Головне вікно додатку Paytomat Wallet

1.4 Постановка задачі на розробку нової технології

Підставою для розробки технічного завдання є завдання з дипломного проектування.

Відповідно до вищенаведених задач та методів їх розв’язання, було прийнято рішення розробити інформаційну технологію управління криптовалютними активами, призначену для підвищення швидкості управління криптовалютами та розширення її функціональних можливостей .

Експлуатація розробленої платформи передбачається шляхом впровадження її користувачам, котрі планують зберігати фінанси у криптовалютах, використовувати криптовалюти для проведення платежів, для трейдінгу та інше.

Дана система повинна виконувати наступні функції:

- створення гаманців під різні криптовалюти;
- можливість зберігання, отримання і надсилання криптовалют;
- можливість купівлі і продажі криптовалюти;
- мати високий рівень захисту;
- мати можливість імпорту в інші пристрої;

Вимоги по архітектурі:

Модуль, що розробляється, повинен бути масштабованим і гнучким, зберігати незалежність від джерела даних. Це повинно бути досягнуто за рахунок використання технології blockchain, некастодіального підходу, а також гнучкої мови програмування.

Вимоги до інтерфейсу:

Програмна частина повинна володіти дружнім, інтуїтивно зрозумілим користувачеві графічним інтерфейсом, що дозволяє здійснювати в повній мірі всі функціональні можливості модуля.

Вимоги до апаратних і програмних ресурсів:

Передбачається робота додатку на мобільному пристрої. Засоби, необхідні для роботи програми, не повинні створювати конфліктних ситуацій при організації інформаційних потоків з пристроєм користувача.

Вимоги до програмного забезпечення системи:

Інформаційна технологія управління криптовалютними активами повинна працювати на мобільних пристроях з операційними системами iOS і Android.

Усі вищенаведені складові при поєднанні забезпечують повнофункціональне навантаження проекту. Технічне завдання розміщене у додатку А.

1.5 Висновки

У результаті аналізу сучасного стану управління фінансами було з’ясовано, що ця задача є дуже важливою на сьогоднішній день та її технічні рішення використовуються у багатьох сферах людської діяльності. Зокрема в роботі фінансових систем, банків і т.д.

Проаналізовано існуючі задачі по створенню систем управління криптовалютними активами.

Існуючі аналоги по управлінню криптовалютними активами мають ряд недоліків, такі як недостатній рівень захисту, надання власних коштів користувача стороннім сервісам, відсутність мобільної версії, недостатній функціонал і інші.

Було визначено, що обраний метод захисту криптовалютних активів має високий рівень криптостійкості і шкидкості, але з’являється складність роботи з некастодіальним підходом**.** Для розробки програмного продукту сформовано технічне завдання, яке враховує всі вимоги по інформаційної технології управління криптовалютними активами.

2. МОДЕЛЮВАННЯІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ УПРАВЛІННЯ КРИПТОВАЛЮТНИМИ АКТИВАМИ

2.1 Аналіз алгоритмів консенсусу

Алгоритм консенсусу може визначатися як механізм, за допомогою якого блокчейн мережу досягає консенсусу. Публічні (децентралізовані) блокчейни побудовані як розподілені системи, і оскільки вони не покладаються на центральні органи, розподілені вузли повинні узгоджувати валідацію транзакції. Саме тут вступають в силу алгоритми консенсусу. Вони запевняють, що дотримуються правила протоколу, і гарантують, що всі транзакції відбуваються довіреною способом, тому монети можуть бути витрачені тільки один раз.

Терміни алгоритм і протокол часто використовуються як взаємозамінні, але це не одне і те ж. Простіше кажучи, ми можемо визначити протокол як первинні правила блокчейна, а алгоритм як механізм, за допомогою якого вони будуть виконуватися[6].

Крім використання в фінансових системах, технологія blockchain може застосовуватися до широкого кола підприємств, і може бути придатна для різних випадків використання. Але незалежно від контексту, блокчейн мережу буде побудована над протоколом, який визначить, як повинна працювати система, тому всі різні частини системи і всі учасники мережі повинні будуть слідувати базовим правилам протоколу.

У той час як протокол визначає правила, алгоритм повідомляє системі, які заходи необхідно вжити для дотримання цих правил і отримання бажаних результатів. Наприклад, алгоритмом консенсусу блокчейн, є те, що визначає валідацію транзакцій і блоків.

Існує кілька типів алгоритмів консенсусу. У кожного є свої переваги і недоліки, при спробі збалансування безпеки, з функціональністю і масштабованість. Отже, розглянемо їх:

- Proof-of-Work (PoW):

Ймовірно завдяки Біткоїну, алгоритм консенсусу PoW є найбільш відомим способом підтвердження транзакцій. Основна думка полягає в тому, щоб вузли блокчейн мережі, що підтверджують транзакції проробляли досить складну обчислювальну роботу (прорахунок алгоритму), результат роботи якого був би легко і швидко перевіряємий іншими вузлами мережі.

Перший вузол, який повністю провів всі необхідні обчислення - отримує винагороду від блокчейн мережі. Всі вузли змагаються між собою (нарощуючи ємність обчислювальних ресурсів), щоб опинитися тим самим, першим вузлом, який отримав винагороду.

Основним недоліком цього алгоритму є безглузді енергетичні витрати - велика кількість вузлів виробляють обчислення, але в реальності тільки один (перший) проводить успішну роботу і отримує винагороду[7].

- Proof-of-Stake (PoS):

Один з найпопулярніших алгоритмів консенсусу в блокчейн мережах. У цьому алгоритмі творцем наступного блоку в ланцюжку блоків вибирається вузол, який володіє більшим балансом - кількістю ресурсів, наприклад монет в криптовалюті. За саме створення блоку, вузол винагороду не отримує. Винагорода виплачується за проведення транзакції.

Можливі варіанти вибору вузла:

- випадковим чином з найбільш "багатих" вузлів;
- випадковим чином з найбільшстарихвузлів.

Основними перевагами, є:

- істотне зниження споживання електроенергії (щодо PoW методу);
- для створення атаки Double-spending, необхідно сконцентрувати більше 50% від загальної кількості всієї валюти, що буде коштувати величезних статків. У тому випадку, якщо ж атакуючий все ж зможе сконцентрувати таку кількість коштів, він своїми діями порушить баланс і сам більше постраждає від своєї ж атаки.

Основними недоліками, є:

- мотивація, в концентрації коштів, що може призводити до централізації мережі[7].

- Delegated Proof-of-Stake (DPoS):

Один з різновидів алгоритму консенсусу Proof-Of-Stake, в якому блоки підписують обрані представники. Власники найбільших балансів вибирають своїх представників, кожен з яких отримує право підписувати блоки в блокчейн мережі. Кожен представник, що володіє одним або більше відсотками від всіх голосів потрапляє до ради. З сформованого "ради директорів" вибирається (по-колу) наступний представник, який і підпише наступний блок. У тому випадку, якщо з якої-небудь причини представник пропустив свою чергу в підписанні, він позбавляється делегованих голосів і залишає "рада директорів", після чого на його місце обирається наступний найбільш підходящий кандидат.

Власники балансів делегуючи свої голоси, аж ніяк не втрачають над ними контролю, так як в будь-який момент можу їх відкликати у свого представника.
Основними перевагами алгоритму DPoS є:

- власники балансів мають можливість делегувати свої голоси (при цьому не передаючи сам баланс),
- власники балансів мають можливість отримати додатковий дохід від їх володіння,
- мінімізація витрат на підтримку блокчейн мережі. На відміну від класичного PoS, знижується кількість "непотрібної роботи" при виборі наступного голосуючого[7].

- Leased Proof-of-Stake (LPoS):

Як зрозуміло з назви, LPoS - ще одна модифікація алгоритму Proof-of-Stake. На даний момент він підтримується тільки платформою Waves. В рамках цього алгоритму, будь-який користувач має можливість передавати свій баланс в оренду вузлам-майнерам, а за це вузли-майнери діляться частиною прибутку з користувачами. Таким чином, даний алгоритм консенсусу дозволяє отримати дохід від майнінговой діяльності, не ведучи самого майнінгу.

- Proof-of-Capacity (PoC):

Іноді ще званий Proof-of-Space (PoSpace), ще один алгоритм консенсусу. Було знайдено лише одну з відносно великих мереж, що підтримує даний алгоритм, це Burstcoin. PoC працює за наступним принципом:

- кожен майнер обчислює досить великий обсяг даних, який записується на дискову підсистему (жорсткий диск, хмарні системи зберігання) вузла. Такий, початковий набір даних в PoC називається "ділянку".
- для кожного нового блоку в блокчейні, майнер читає невеликий набір даних (1/4096, що приблизно становить 0.024%) від свого загального збереженого обсягу і повертає результат(дедлайн), що дорівнює часу в секундах, котрий пройшов з моменту створення останнього блоку, після якого майнер зможе створити новий блок.
- майнер, який отримав мінімальний час дедлайну, підписує блок і отримує винагороду за транзакції.

Таким чином обчислювальні ресурси необхідні майнеру для цієї роботи обмежені часом, який необхідний для читання файлів з дискової підсистеми. Саме цей фактор дозволяє проводити майнінг з досить високою енергоефективністю. Майнери змагаються між собою за розміри збережених даних, на відміну від швидкості роботи обладнання, яке є визначним в майнінгу побудованому на PoW[8].

- Proof-of-Importance (PoI):

Алгоритм консенсусу використовуваний блокчейн платформою NEM. Значимість кожного користувача в мережі NEM визначається, як кількість коштів наявних у нього на балансі і кількість проведених транзакцій з / на його гаманець. На відміну від більш звичного PoS, який враховує тільки баланс наявних коштів у користувача, PoI враховує як кількість коштів, так і активність користувача в блокчейн мережі. Такий підхід залучає користувачів не просто тримати кошти у себе на рахунку, а й активно використовувати їх[8].

- Proof-of-Activity (PoA):

Опис алгоритму опубліковано в 2014 році, як потенційно нового і більш надійного алгоритму для біткоїна, проте інформації про його імплементацію немає. Автори алгоритму PoA спробували об'єднати два найбільш популярних алгоритми,

такі як Proof-of-Work і Proof-of-Stake, з метою збільшення рівня захисту від потенційно можливих атак (51% attack, Denial-of-Serviceattacks (DoS)). Принцип роботи алгоритму описаний нижче:

- кожен майнерблокчейн мережі намагається згенерувати заголовок порожнього блоку, який включає в себе хеш попереднього блоку, публічний адресу майнера, індекс поточного блоку в блокчейні і nonce.
- після генерації заголовка порожнього блоку, що відповідає поточним вимогам складності, вузол розсилає цей заголовок в блокчейн мережу.
- всі вузли мережі розглядають заголовок такого блоку, як дані отримані від псевдовипадкових власників. Використовуючи хеш розісланого заголовка блоку і хеш попереднього блоку + N пресетів з використанням алгоритму follow-the-satoshi вибираються стейкхолдери.
- кожен стейкхолдер, що знаходиться онлайн, перевіряє отриманий, порожній заголовок блоку на його коректність. Під час перевірки, кожен, хто отримав заголовок, перевіряє: чи є він одним з перших N-1 стейкхолдерів "щасливчиків" цього блоку і в цьому випадку підписує заголовок порожнього блоку своїм секретним ключем і відправляє його в блокчейн мережу.
- Коли N-й стейкхолдер бачить, що він повинен стати підписантом цього блоку, він, на додаток до заголовку порожнього блоку, додає блок з включеними транзакціями (кількість включаються транзакцій він вибирає сам), всі підписи N-1 від інших стейкхолдерів і підписує блок .
- Стейкхолдер N розсилає новий, підготовлений блок. Вузли отримують цей блок, переконуються в його законності і додають цей блок в блокчейн.
- Премія за транзакції, яку отримав N-стейкхолдер, розподіляється між майнером і N стейкхолдерами "щасливчиками"[8].

- Proof-of-Authority (PoAuthority):

PoA алгоритм консенсусу стоїть дещо осібно від інших алгоритмів, так як для своєї роботи йому не потрібно мати взагалі будь-якого майнінгу, як у випадку з PoW або PoS. У блокчейн мережі, що базується на PoAuthority, всі транзакції і блоки перевіряються за допомогою схвалених аккаунтів(валідаторів). Проведення

транзакцій і створення блоків, проходить в автоматичному режимі за допомогою обчислювальних потужностейвалідатора.

Позитивним моментом даного алгоритму є:

- відсутність майнінгу і як наслідок, істотне зниження витрат на його обслуговування.

Негативний момент використання даного алгоритму:

- як зрозуміло з самого опису - ключовими особами, є валідатори, що призводить до централізації. Ймовірно в деяких випадках, в приватних мережах і за допомогою повністю (на скільки це можливо) довірених аккаунтів це має сенс[8].

- Proof-of-Burn (PoB):

Ще один цікавий тип алгоритму консенсусу Proof-of-Burn. При його використанні майнер відправляє монети на випадкову адресу згенерованого хешу, витратити кошти з цієї адреси практично неможливо, так як ймовірність підібрати до нього ключі близиться до нуля. За таке спалювання монет, майнер отримує постійний шанс знайти PoB блок і отримати за нього нагороду. Шанси на майнінг збільшуються при збільшенні кількості спалених монет. Економічно цей процес спалювання монет можна уявити як покупка бурової установки для майнінгу. Природно такий алгоритм має сенс використовувати тільки на пізніх етапах існування тієї чи іншої криптовалюти, тоді коли є що "спалювати". Цікавою думкою є те, що цей метод також добре підходить для трансферукриптовалют з "старих" в "нові". Наприклад "стара" криптовалюта знаходиться у фінальній точки свого майнінгу, ми можемо використовувати метод PoB тоді, коли для того, щоб отримати "нову" криптовалюта, нам необхідно спалити "стару".

Даний алгоритм використовується на платформі Slimcoin[9].

2.2Аналіз інформаційної безпеки

Дві основні криптографічні концепції, які стоять за технологією блокчейн - це хешування і цифровий підпис.

Хешування - це процес перетворення довільного обсягу даних за певним алгоритмом в рядок фіксованого розміру, яка називається хешем.

В даний час існує безліч загальнодоступних алгоритмів хешування. Суть їх зводиться до того, що на вході функція хешування отримує довільний обсяг біт, робить над цими даними деякі обчислення і повертає певну кількість біт. Наприклад, 256 біт[10].

Як блокчейн використовує хеш-кодування? Хеши використовуються для відображення поточного стану ланцюга. Вхідними даними тут є повний стан блокчейна, тобто всі транзакції, які були здійснені до сих пір, і результуючий хеш представляє собою поточний стан блокчейна.

Хеш тут використовується для того, щоб будь-який учасник міг переконатися, що стан блокчейна залишається незмінним.

Перший хеш розраховується для першого блоку, так званого Genesis-блоку. Послідовність транзакцій, записаних всередині блоку, використовується для обчислення хешу Genesis-блоку.

Для кожного нового блоку, який генерується згодом, в якості вхідних даних для визначення його власного хешу, використовуються хеш попереднього блоку і інформація про його власних транзакціях.

Саме так і утворюється ланцюжок блоків або blockchain. Кожен хеш нового блоку вказує на хеш попереднього.

Така система хешування гарантує, що жодна транзакція в історії не може бути змінена, оскільки якщо хоч одна частинка транзакцій поміняється, то так само зміниться і хеш блоку, в якому вона записана, і відповідно зміняться хеші наступних блоків[11].

Цифрові підписи, так само як і справжні підписи, - це один із способів довести, що хтось є тим, за кого він себе видає. З урахуванням того, що в цій ситуації використовується криптографія і математика, це значно безпечніше, ніж звичайні рукописні підписи, які можна легко підробити.

В асиметричних системах шифрування користувачі генерують за певним алгоритмом так звані пари ключів, кожна з яких є відкритим ключем (publickey) і закритим ключем (privatekey).

Відкритий і закритий ключі пов'язані один з одним за допомогою деяких математичних відносин. Відкритий ключ призначений для поширення публічно. Він служить в якості адреси для прийому повідомлень від інших користувачів.

Закритий ключ зберігають в секреті. Він використовується в якості цифрового підпису для повідомлень, відправлених іншим користувачам. Підпис включається в повідомлення, таким чином, одержувач може ідентифікувати відправника за допомогою його відкритого ключа.

За допомогою цього способу одержувач може переконатися, що саме відправник і послав це повідомлення. Створення пари ключів аналогічно створенню облікового запису в блокчейні, але без необхідності реєстрації де-небудь[12].

Більш того, кожна транзакція, що виконується в блокчейні, підписується цифровим підписом відправника, використовуючи його закритий ключ. Цей підпис гарантує, що тільки власник облікового запису може перевести з неї гроші.

Розглянемо який алгоритм хешування використовується в основних криптовалютах(тих, що використовуються в нашому додатку).

1) SHA-256

Цей алгоритм хешування використовується в Bitcoin, та у його форкуBitcoinCash.

SHA-256 являє собою односпрямовану функцію для створення цифрових відбитків фіксованої довжини (256 біт, 32 байт) з вхідних даних розміром до 2,31 ексабайт (2^{64} біт) і є окремим випадком алгоритму з сімейства криптографічних алгоритмів SHA-2 (SecureHashAlgorithmVersion 2) опублікованим АНБ США в 2002 році[11].

Хеш-функції сімейства SHA-2 побудовані на основі структури Меркле - Дамгарда.

Оригінал тексту після доповнення розбивається на блоки, кожен блок - на 16 слів. Алгоритм пропускає кожен блок повідомлення через цикл з 64 ітераціями. На

кожній ітерації 2 слова перетворюються, функцію перетворення задають інші слова. Результати обробки кожного блоку складаються, сума є значенням хеш-функції. Так як ініціалізація внутрішнього стану проводиться результатом обробки попереднього блоку, то немає можливості обробляти блоки паралельно. Графічне представлення однієї ітерації обробки блоку даних представлено на рисунку 2.1.

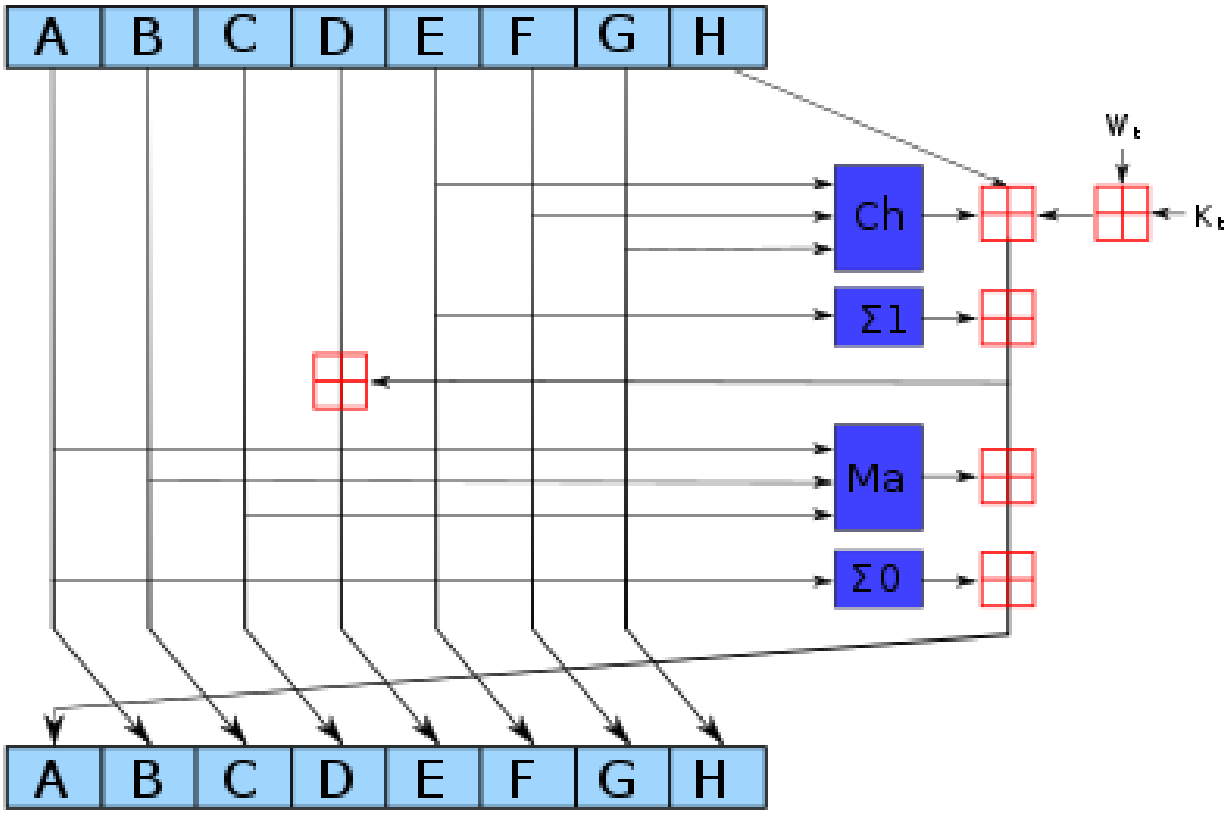


Рисунок 2.1 – Графічне представлення однієї ітерації

2) Ethash

Алгоритм хешування Ethereum перший час носив ім'я DaggerHashimoto і дуже нагадував собою алгоритм Scrypt. Однак сам його механізм хешування володів істотною відмінністю від попередника, так як створював цілий граф (дерево з великою кількістю відгалужень) послідовних вузлів. Ця система ще більш ускладнювала процес розшифровки і робила практично неможливим випадковий підбір значень. Надалі в алгоритм DaggerHashimoto були внесені деякі поліпшення і проведено ребрендинг, в результаті якого він був перейменований в Ethash[12].

Тепер опис алгоритму Ethereum виглядає як хешування метаданих останнього блоку системи, для якого використовується спеціальний код під назвою Nonce. Сам Nonce являє собою звичайне двійкове число, що задає унікальне значення хешу. Тепер випадковий підбір правильного значення стає ще більш проблематичним, ніж в попередній версії алгоритму: фактично підбір хешу тепер можливий лише методичним перебором всіх можливих варіантів. Графічне представлення роботи алгоритму представлено на рисунку 2.2.

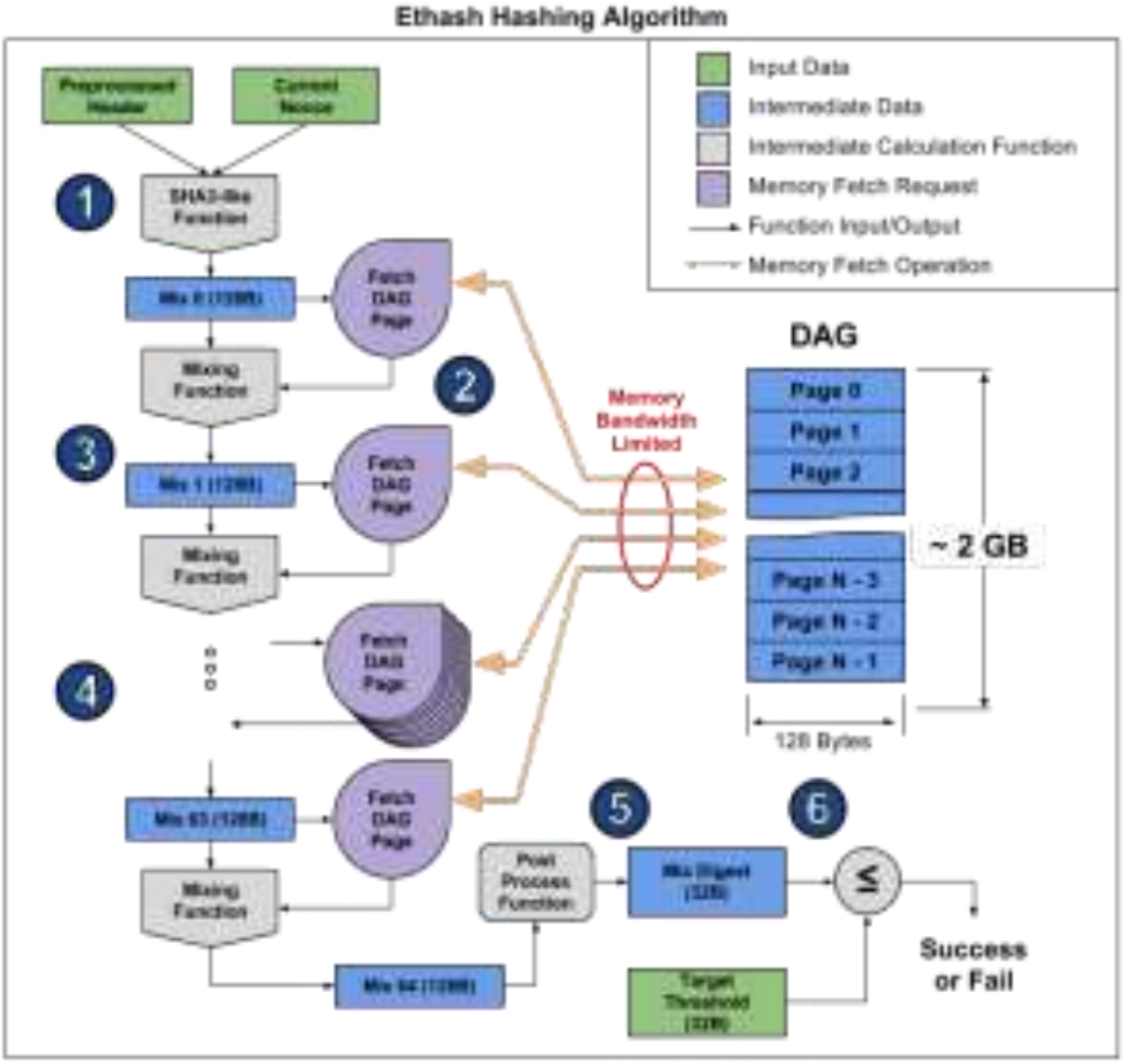


Рисунок 2.2 – графічне представлення роботи алгоритму

3) Scrypt

Використовується в Litecoin.

Функція хешування Scrypt спеціально розроблялася з метою ускладнити апаратні реалізації шляхом збільшення кількості ресурсів, необхідних для обчислення. Тому такої концентрації майнінг-ресурсів як в біткоїнів не повинно статися, і він залишиться децентралізованим.

За своєю суттю, Scrypt-майнінг не сильно відрізняється від біткойн-майнінг. На вхід подається блок даних, до нього застосовується хеш-функція, на виході ми намагаємося отримати «гарний хеш». Ось тільки сама хеш-функція набагато складніше в обчисленні. Даний алгоритм використовує більш значна кількість оперативної пам'яті (пам'яті з довільним доступом), ніж SHA-256. Пам'ять в Scrypt використовується для зберігання великої вектора псевдовипадкових бітових послідовностей, що генеруються в самому початку алгоритму. Після створення вектора його елементи запитуються в псевдовипадковому порядку і комбінуються один з одним для отримання підсумкового ключа[11].

Так як алгоритм генерації вектора відомий, в принципі можлива реалізація scrypt, що не вимагає особливо багато пам'яті, а вираховувати кожен елемент в момент звернення. Однак обчислення елемента щодо складно, і в процесі роботи функції scrypt кожен елемент зчитується багато разів. У Scrypt закладений такий баланс між пам'яттю і часом, що реалізації, які не використовують пам'ять, виходять занадто повільними. Графічне представлення роботи алгоритму представлено на рисунку 2.3.

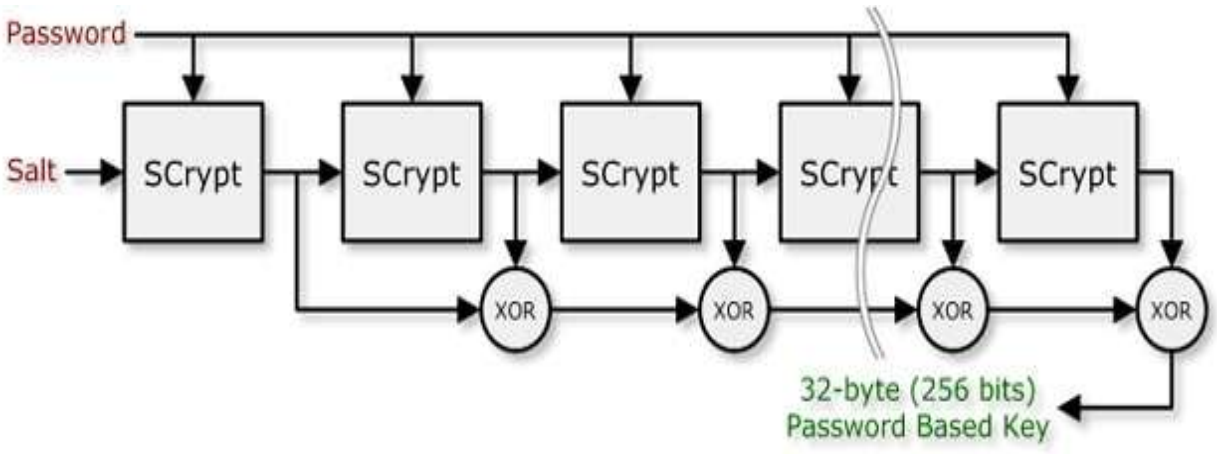


Рисунок 2.3 – Графічне представлення роботи алгоритму

2.3 Проектування структури і розробка алгоритмів інформаційної технології управління криптовалютними активами

На даний момент, не існує технології, що дозволяє в повній мірі проводити операції відправки, обміну криптовалюти, продажу і купівлі в декілька кліків, по найкращому курсу з існуючих, не переглядаючи усі обмінники, зберігати криптовалюту так, щоб до неї ніхто, крім користувача, не мав доступу. Тому, багато людей шукають засіб, котрий допоможе їм проводити фінансові операції з крипто валютами досить швидко і зберігати їх надійно, не даючи біржам доступ до своїх коштів, що збереже їх від можливості втрати через недосконалість захисту біржі.

Також, для користувача буде зручним мати лише один гаманець для усіх криптовалют, щоб не витрачати час на зміну їх для проведення певної операції. Тобто, потрібно надати користувачам основні криптовалюти і токени у використання, з можливістю доповнення списку.

На сьогодні, найбільш популярні криптовалюти з найбільшою капіталізацією це Bitcoin, Ethereum, Ripple, Bitcoin Cash і Litecoin повинні бути додані до платформи зараз, або в найближчому майбутньому.

Bitcoin (BTC) - це цифрова валюта, яку створили, щоб вирішити всі проблеми онлайн платежів. Криптовалюта Bitcoin (BTC) - це повністю децентралізована цифрова одиниця, яку по праву називають грошима в мережі інтернет. Під словом «децентралізована», ми маємо на увазі, що мережа bitcoin не регулюється і не контролюється якимось органом. Підтримують роботу мережі, так звані ноди, завдяки яким перевіряються і підтверджуються операції всередині мережі. Децентралізація мережі - це основна мета криптовалюти bitcoin[9].

Технічні особливості криптовалюти Bitcoin (BTC):

- Рік заснування - 2009 року;
- Метод захисту - Proof of Work;
- Алгоритм - SHA 256;
- Торговий тікер - BTC;
- Максимальна емісія - 21 000 000 BTC.

Bitcoin був створений в 2009 році, невідомим програмістом під ніком Сатоши Накамото. У код bitcoin були закладені три основні ідеї:

- Bitcoin не повинен ніким регулюватися;
- Його емісія не повинна бути нескінченна;
- Вартість монети безпосередньо залежить від попиту на неї.

Для взаємодії з blockchain Bitcoin доцільно використовувати Insight-API від Bitpay з підключенням до insight.bitpay.com. Insight-API – це REST і web socket API для Bitcore Node. Цей API забезпечує наш додаток усіма необхідними функціями для взаємодії з bitcoin. Наприклад, «/insight-api/tx/[:txid]» дозволить отримати усю необхідну інформацію про певну транзакцію за її txid, «/insight-api/rawtx/[:rawid]» - отримати 16-ве представлення даних транзакції за її rawid, «/insight-api/addr/[:addr][?noTxList=1][&from=&to=]» - отримати інформацію про усі транзакції по заданій адресі addr і коригувати діапазон пошуку(можна вивести лише останні або найперші транзакції по адресі), «/insight-api/addr/[:addr]/balance» - отримати баланс за заданою адресою addr, «/insight-api/addr/[:addr]/utxo» - отримати utxo за заданою адресою addr, «/insight-api/utils/estimatefee[?nbBlocks=2]» - оцінити комісію на заданому блоці, «/insight-api/tx/send» - надіслати транзакції до блокчейну.

Ripple - криптовалюта, створена для здійснення моментальних грошових переказів і обміну активів. Система позиціонується як універсальний хаб для розрахунків між банками, корпораціями та приватними користувачами. Ripple може оперувати будь-якими величинами, які мають власну цінність: національними валютами і криптовалютою, дорогоцінними металами, фізичними активами і навіть бонусними балами авіакомпаній.

Ripple - одна з найстаріших криптовалют: поточна версія протоколу була представлена в 2012 році. Для підтвердження угод система використовує консенсусний реєстр (consensus ledger), контроль записів в якому здійснюють великі вузли-валідатори. Власниками вузлів зазвичай виступають банки, що використовують протокол для розрахунків з партнерами.

Паливом для роботи Ripple служить токен XRP. 100 мільярдів монет були створені на початковому етапі існування протоколу, і це число є межею емісії

криптовалюти. Кошти, що залишилися, заморожені за допомогою смарт-контрактів, їх розблокування відбуватиметься по мірі зростання попиту на криптовалюту.

Так як випуск граничного числа токенов XRP стався на ранньому етапі існування протоколу, в системі неможливо здійснювати видобуток монет шляхом майнінгу або форджінга. Нагорода за перевірку транзакцій також відсутня: комісії за вчинення переводів безповоротно «згорають», зменшуючи кількість токенів в обороті. Банки, що використовують протокол для взаємних розрахунків, зацікавлені в стабільній роботі системи, так що саме вони найчастіше є власниками нод[10].

Для взаємодії з blockchain Ripple доцільно використовувати Data API V2 Tool з підключення до вузда ripple.com. Data API V2 забезпечує доступ до інформації про зміни в Ledger XRP, включаючи історію транзакцій та оброблені аналітичні дані. Ця інформація зберігається у спеціальній базі даних, яка звільняє випущені сервери для зменшення кількості історичних версій журналу. API Data v2 також виступає як джерело даних для програм, таких як XRP Charts та ripple.com. Цей API забезпечує наш додаток усіма необхідними функціями для взаємодії з Ripple. Наприклад, «getAccount» надає інформацію про створення певного аккаунта, «getAccountBalance» - інформацію про баланс аккаунта у різних валютах, «getAccountTransactionHistory» - дозволяє отримати інформацію про усі транзакції по заданій адресі, «getTransaction» - дозволить отримати усю необхідну інформацію про певну транзакцію за її хешом, «getTransactionCost» - надасть ціну за виконання одної транзакції.

Litecoin - форк Bitcoin, пірінгова електронна платіжна система, яка використовує однойменну криптовалюта.

Litecoin є другим після Namecoin форком Bitcoin і має лише невеликі відмінності від нього. Litecoin можуть використовуватися для обміну на Bitcoin або загальноприйняті гроші в обмінниках, а також для електронної оплати товарів або послуг у продавців, готових їх приймати.

Адреси Litecoin складаються з 33 символів і починаються з літери L або цифри 3.За перебування нового блоку в мережі встановлена нагорода, спочатку рівна 50 LTC і зменшується вдвічі за кожні 840000 блоків. Складність обчислення Litecoin

підбирається таким чином, щоб, в середньому, один блок генерувався 2,5 хвилини, що в 4 рази швидше, ніж Bitcoin, це дозволяє швидше отримувати підтвердження транзакцій. Емісія Litecoin алгоритмічно обмежена. Максимальна кількість litecoin'ів, яка увійде в обіг, перевищує максимальне число bitcoin'ів в 4 рази (84 млн. проти 21)[9].

Bitcoin Cash - це створена в результаті хардфорка Bitcoin криптовалюта, яка має покращені технічні характеристики. Bitcoin Cash відрізняється від свого попередника більш досконалою моделлю роботи. По-перше, пропускна здатність мережі Bitcoin Cash, так як її розмір блоку більше в 4 рази - 8 Мб. По-друге, мережу Bitcoin Cash можна по праву вважати більш децентралізованою, так як його видобуток можливий за допомогою звичайних відео карт, що дає можливість більшій кількості користувачів заробляти на майнінгу BCH[10].

Основні характеристики даної монети:

- Біржовий тікер або скорочення Bitcoin Cash - BCH або BCC;
- Для майнінгу використовується алгоритм SHA-256;
- Блоки мають розмір 8 MB;
- За транзакції потрібно платити комісію, але вона невелика. В середньому плата становить 0.07 USD;
- Середній час на формування блоку - 15 хвилин, але зазвичай блок можна сформувати і за десять хвилин;
- У кожної транзакції Bitcoin Cash є свій унікальний цифровий підпис, який можна перевірити;
- Хешрейт мережі 1.996 Ehash / s;
- Коли з'явився Bitcoin Cash, нагорода за блок становила 12.5 BCH.

Серед додатків, що працють з blockchain поширена клієнт-серверна архітектура, в рамках якої весь обмін даними здійснюється між клієнтом, сервером і блокчейном. Для того щоб виконувати функції API різних блокчейнів, нам потрібно буде підключитися до відповідних вузлів. При виклику цих функцій ми матимемо можливість отримати інформацію про транзакцію, а також зміну балансу на адресі.

Щоб надати користувачам нашого мобільного додатку можливість швидкої купівлі, продажу чи обміну криптовалют і фіатних коштів по найкращому курсу ми будемо використовувати сервер, який буде опрацьовувати запити і перенаправляти їх до одного з підключених провайдерів. Перевірка балансів, виконання транзакцій по відправленню криптовалюти буде напряму в мобільному додатку. Структурна схема інформаційної технології управління криптовалютними активами зображено на рис.2.4.

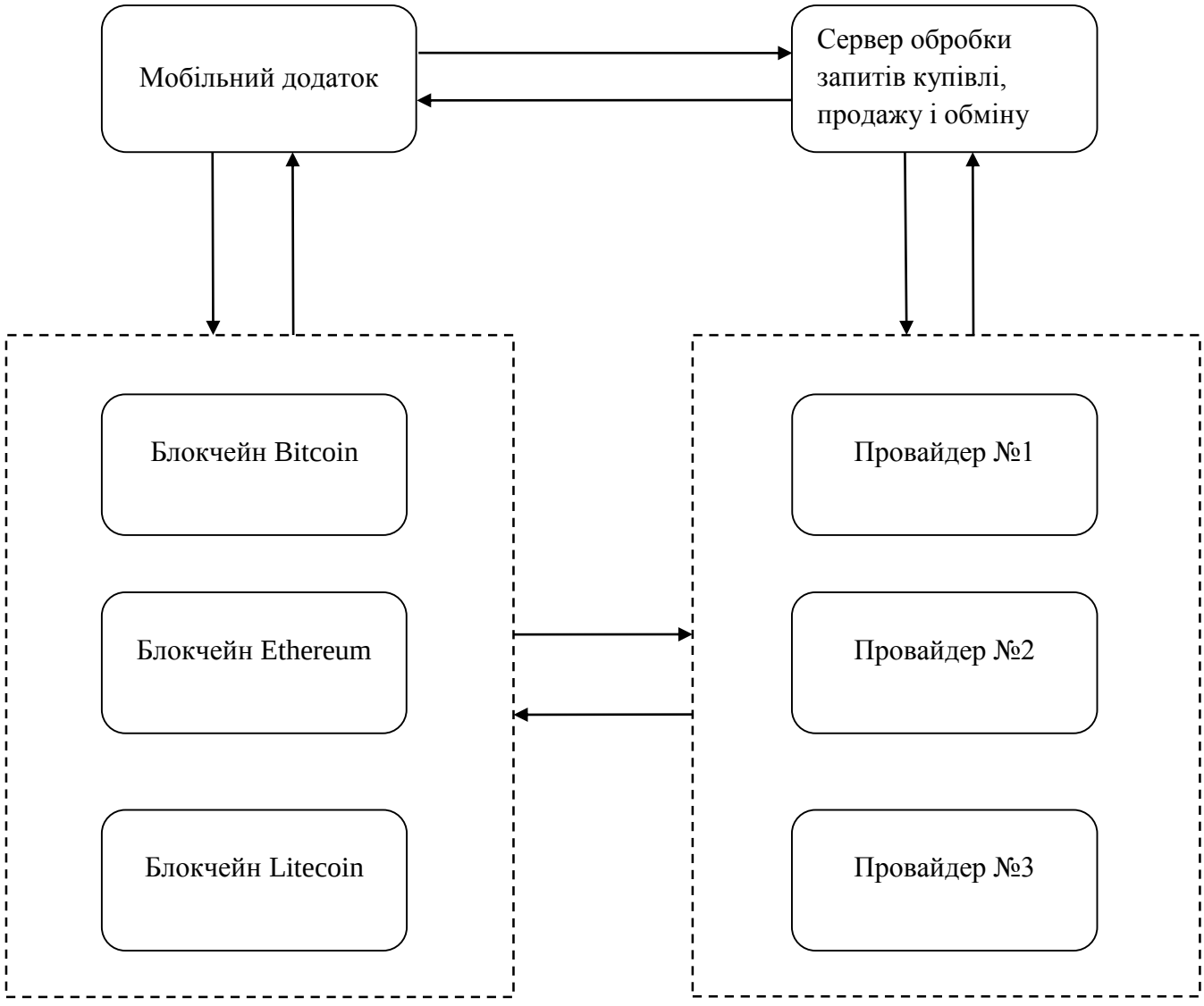


Рисунок 2.4 – Структурна схема функціонування інформаційної технології управління криптовалютними активами

Для опису структури клієнтської частини API проекту використаємо діаграму UML. Зокрема: діаграму класів.

Діаграма класів вказує типи класів системи і різного роду зв'язки, що виникають між ними. На цих діаграмах вказуються також їхні атрибути, операції (методи) та обмеження, що накладаються на зв'язки між класами (рис.2.5) .

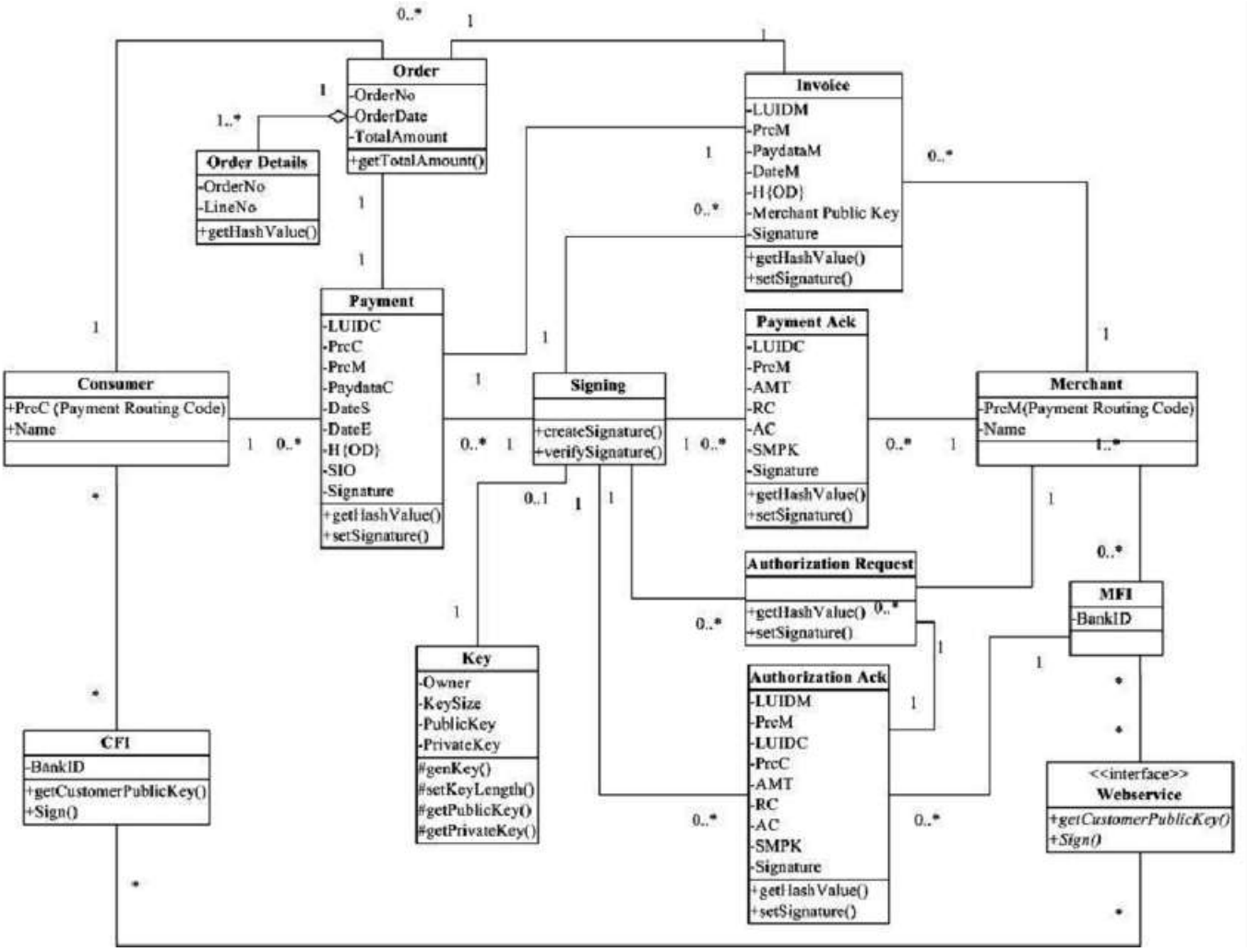


Рисунок 2.5 – Діаграма класів інформаційної технології управління криптовалютними активами

Найважливіша і найперша річ, яку потрібно зробити, щоб потрапити у світ криптовалют – це отримати їх собі на рахунок. Одним, і найпоширенішим, варіантом є купівля криптовалюти. Її можна купити на біржі, через обмінник, або напряму. Виключивши найнебезпечніший варіант, вибір суттєво зменшується. Мінусом купівлі на біржі є те, що криптовалюта «осяде» на адресі, що видасть Вам біржка,

проте цю адресу не можна використовувати для роботою з блокчейном, тому що вона є локальною для біржі. А отже, потрібно буде вкотре заплатити комісію біржі і вивести на іншу адресу, що також потребує часу. Обмінник, в свою чергу, відправляє кошти на адресу блокчейн мережі, проте щоб знайти обмінник з нормальним курсом Вам потребується багато часу. Запропонована інформаційна технологія уникає всіх цих мінусів і проводить операцію купівлі найшвидше, по найкращому курсу та напряму на адресу блокчейн мережі. Схема алгоритму купівлі криптовалюти зображено на рис 2.6

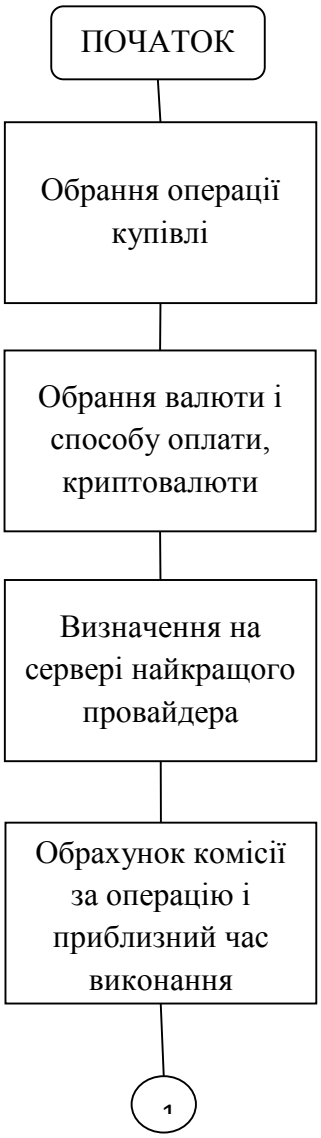
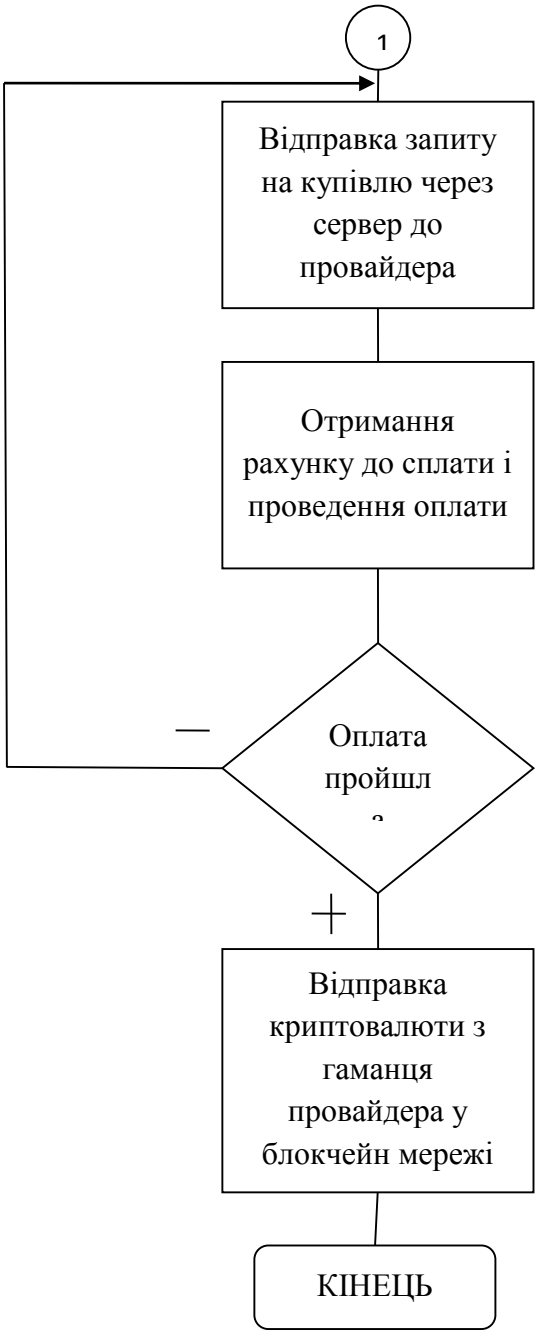


Рисунок 2.6 – Схема алгоритму купівлі інформаційної технологія управління криптовалютними активами

Продовження рисунку 2.6



Зобразимо процес проведення фінансових операцій, а саме купівлі, продажу чи обміну через діаграму послідовності інформаційної технології, що зображено на рисунку 2.7.

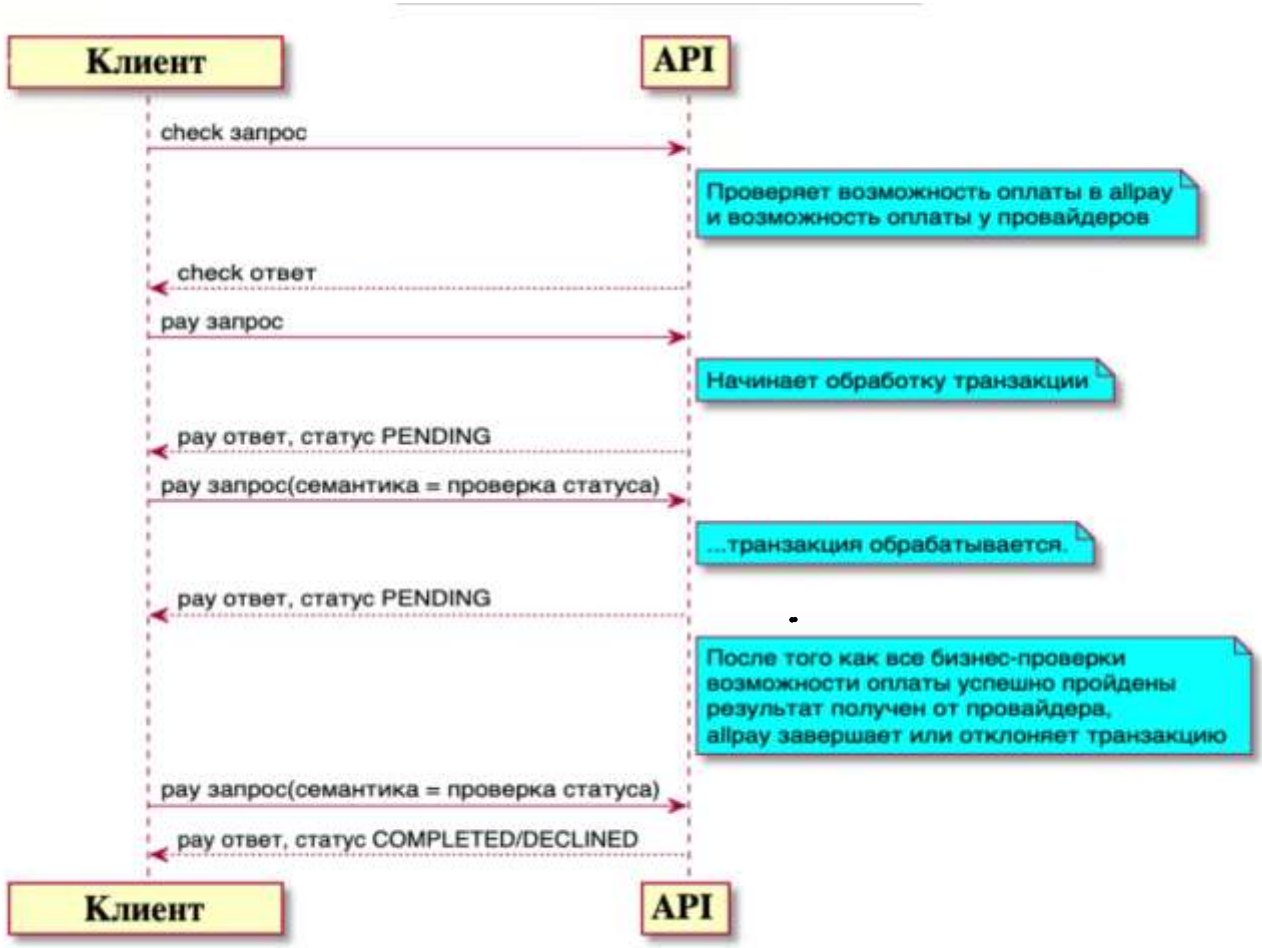


Рисунок 2.7 – Діаграма послідовності інформаційної технології

2.4 Математична модель ієрархічної генерації ключів інформаційної технології управління криптовалютними активами

В першу чергу визначимо, що таке детерміністичний гаманець (deterministic wallet). Коли мова йде про генерацію ключів, часто вживається слово "гаманець", тому що в контексті криптовалют володіння особистим ключем є доказом володіння монетами, а в даному випадку гаманець і ключ мають схожий зміст.

Детерміністичний гаманець - це гаманець, в якому усі використані особисті ключі були породжені з одного загального для всіх ключів секрету. Особливість полягає в тому, що є можливість з одного секрету породити скільки-завгодно пар ключів для електронного підпису. Можна використовувати нові адреси для кожного, хто входить платежу і здачі.

Зручно, що ключі такого гаманця можна легко перенести на інший пристрій, зробити резервну копію і потім відновити, тому що фактично потрібно резервувати тільки один основний секрет. Крім того, всі породжені з основного секрету особисті ключі, один з одним ніяк не пов'язані. Не можна простежити і зв'язок між породженими адресами (визначити що всі вони належать одному користувачеві), а маючи породжений особистий ключ, не можна відновити загальний секрет[12].

Розлянемо кодування основного секрету, що зображено в таблиці 2.1. Тут є певний стандартизований підхід, який був визначений в BIP39. Це так зване CheckEncoding кодування основного секрету в мнемонічну фразу - набір слів, який легко записати на папір і при необхідності запам'ятати. При введенні є можливість перевірити контрольну суму, тобто виявити помилку, якщо така є, з досить великою ймовірністю.

$$Checksum = Entropy/32$$

$$Words = (Entropy + Checksum)/11$$

Таблиця 2.1 – кодування основного секрету BIP39

Entropy	Checksum	Data	Words
128	4	132	12
160	5	165	15
192	6	198	18
224	7	231	21
256	8	264	24

Фактично є основний секрет (Entropy) - дані, з яких розгортаються всі особисті ключі гаманця. Цей секрет може мати різну довжину. Що стосується контрольної суми: на кожні 32 біта Entropy доводиться 1 біт контрольної суми, тобто Checksum за формулою розраховується, як довжина Entropy в бітах, розділена на 32.

Entropy конкатенуються з контрольною сумою, яка розраховується, як подвійний хеш SHA-256 (SHA-2 на довжині 256 біт), після чого відрізається необхідну кількість бітів. Конкатеновані дані переводяться в іншу систему числення:

з двійковій в систему числення за основою 2048 (як бачимо, 2048 - це). І якщо скласти довжину бітів Entropy і контрольної суми, то вийде число, кратне 11-ти. Таким чином, ми отримуємо кількість слів у вихідний мнемонічною фразі[12].

Фактично дані "нарізаються" частинами по 11 біт. Є словник, що складається з 2048 слів, до яких застосовані певні вимоги. За замовчуванням мову словника англійська, але може використовуватися будь-яка. Слова не повинні перевищувати певну довжину (зазвичай межа до 7 символів). Всі вони повинні бути закодовані в UTF-8 з певною нормалізацією всіх символів. Обов'язковою є унікальність кожного слова по перших чотирьох символів.

Перші чотири символи однозначно визначають слово в словнику, а решта символи використовуються, щоб завершити це слово до зручної форми для читання, запам'ятовування і т. Д. Таким чином кожен фрагмент даних, що складається з 11-ти біт, отримує однозначну відповідність у вигляді слова з словника. Якщо Entropy вашого секрету складає 256 біт, то дані для кодування складуть 264 біта, а ваша мнемонічна фраза буде містити 24 слова. Це основний підхід до кодування секрету гаманців в BIP39, який застосовується на практиці частіше за все.

Для того щоб в майбутньому робити резервну копію і використовувати її, вам пропонується виписати цю фразу на зовнішній носій. Найкраще підійде папір, яку ви будете зберігати в надійному місці. Так ви можете відновити повний доступ до всіх своїх ключам.

Детерміністичні гаманці бувають двох типів. Розглянемо основні їх відмінності.

Перший з них найпростіший. Основний секрет тут конкатенуються з індексом дочірнього ключа, який ми хочемо отримати, після чого конкатеновані дані хешуються. Найчастіше це відбувається за допомогою хеш-функції SHA-256[13].

До другого типу відносяться ієрархічні детерміністичні гаманці (hierarchicaldeterministicwallets, HD wallets), принципи яких визначені в BIP32, і є дуже поширеним підходом до ієрархічної генерації ключів.

Розглянемо відмінності цих типів гаманців на рис 2.8.

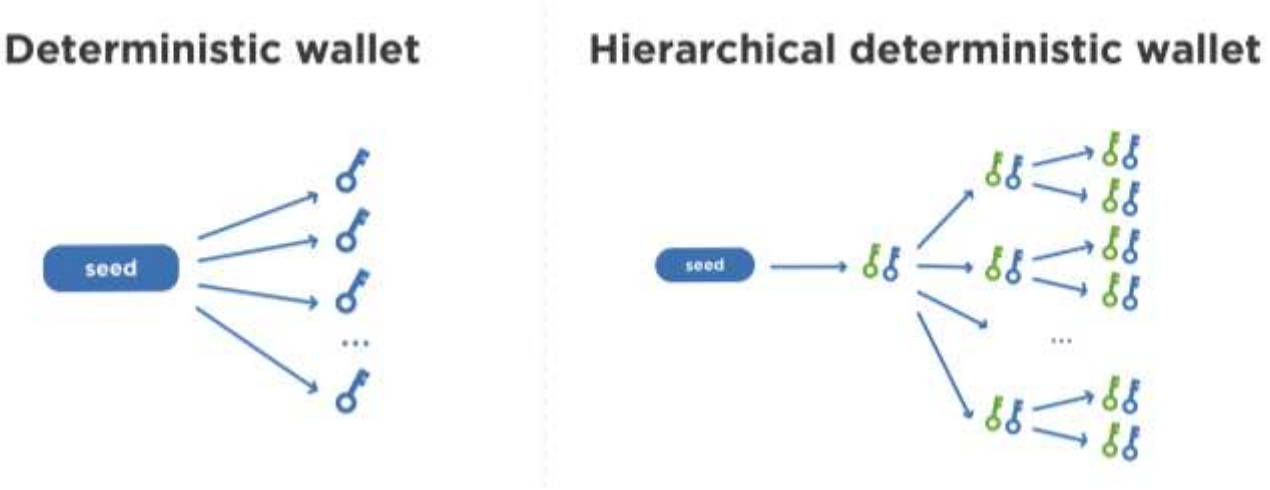


Рисунок 2.8 – Види детерміністичних гаманців

Звичайний детерміністичний гаманець має деяку мнемонічну фразу, з якої безпосередньо генерується безліч особистих ключів. Їх кількість може бути обмежена лише розмірністю індексу, який конкатенується до секрету перед хешуванням. Зазвичай це 4 байта, тобто простір можливих варіантів допускає близько 4 мільярдів унікальних ключів детерміністичного гаманця. На практиці цього повинно вистачити для будь-якої ситуації[13].

Перейдемо до ієрархічного детерміністичного гаманцю, схема розгортання ключів якого представлена поки в спрощеному вигляді. Є мнемонічна фраза, з якого безпосередньо виходить пара майстер-ключів. Якщо в звичайному детерміністичному гаманці ми отримуємо особистий ключ, то тут ми отримуємо пару ключів. Більш того, є рівні ієрархії, на кожному з яких ми розраховуємо індекс для породження дочірнього ключа. Ми також можемо будувати гілки відкритих ключів і гілки особистих ключів.

Відносно ІД гаманців варто відзначити, що згідно з правилами BIP32 на кожному рівні ієрархії вузол породження має три об'єкти: особистий ключ (privatekey), відкритий ключ (publickey) і код ланцюжка (chaincode), який використовується для породження наступного рівня ієрархії[13].

Розглянемо більш детально схему генерації ключів по BIP32 на рис.2.9:

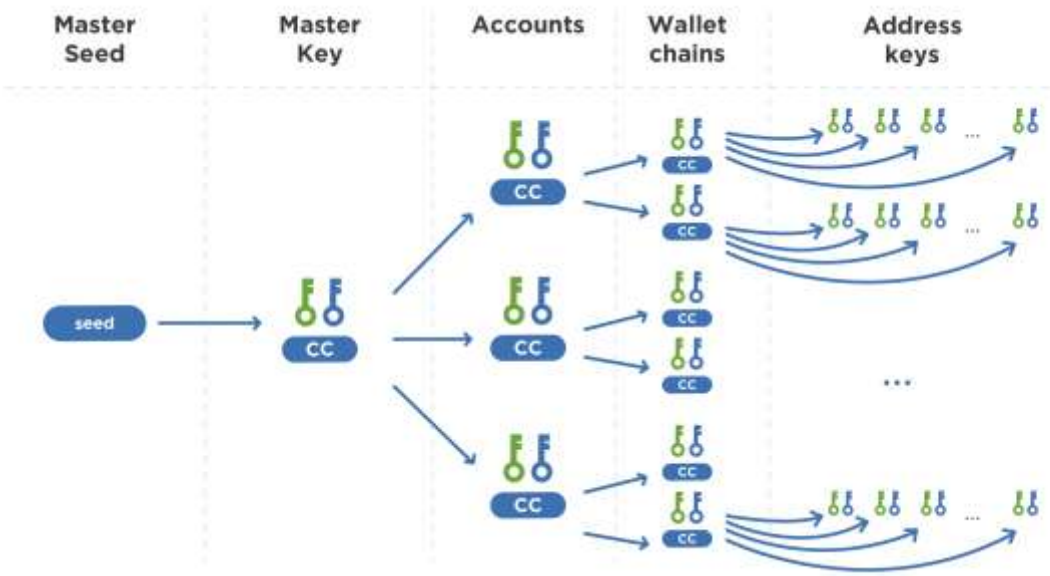


Рисунок 2.9 – Схема генерації ключів по BIP32

Все починається з мнемонічної фрази (seed), її ще називають masterseed, з якої розраховується нульовий рівень ієрархії - пара masterkeys і chaincode.

З пари майстер-ключів може генеруватися безліч пар ключів з певними індексами. Формується новий рівень ієрархії, який використовується для генерації аккаунтів. Припустимо, у одного користувача є мнемонічна фраза і він хоче створити декілька адрес, які будуть відрізнятися один від одного. Монети цих адрес не будуть змішуватися, не будуть публікуватися разом, а в готових транзакціях між ними не можна буде знайти зв'язок. Дані ключі будуть використовуватися повністю окремо один від одного. В одному з аккаунтів група ключів буде використовуватися для робочого бюджету, в іншому - для особистого бюджету, а ще один аккаунт - для чорної бухгалтерії. Монети не будуть змішуватися один з одним[13].

Наступний рівень ієрархії визначає різні ланцюжки генерування ключів. Найчастіше використовуються ланцюжки з індексами 0 і 1. Ланцюжок з індексом 0 буде генерувати кінцеві ключі для формування адреси для вхідних платежів, а ланцюжок з індексом 1 буде генерувати гаманці, на які будуть приходити монети, відправлені користувачем собі, тобто здача. Це потрібно, щоб гаманець на програмному рівні відрізняв відправлені ззовні платежі від здачі, розраховував зміни балансу кожної транзакції і становив наочний список з історією всіх платежів. Це спрощує розробку гаманця і його використання для повсякденних платежів.

Тепер перейдемо до математичної складової процесів ієрархічної генерації ключів. Почнемо з того, що розглянемо hash-basedmessageauthenticationcode. Це інший клас розрахунку хеш-функцій. Відрізняється вона тим, що приймає на вхід два значення, а не одне. Перше значення - це секретний ключ, а друге - саме повідомлення.

$$HMAC(K, m) = H((K^{opad})||H(K^{ipad})||m)),$$

де K – ключ,

m – повідомлення,

opad, ipad - деякі константні значення, необхідні для формування відрізняються один від одного ключів на різних етапах хешування.

В якості хеш-функції використовується SHA-512.

Особливість полягає в тому, що для використання HMAC, потрібно володіти секретним ключем, для того щоб отримати правильне хеш-значення повідомлення.

Отже, для розрахунку хеш-значення по HMAC значення ключа XOR-иться з сталим значенням ipad, після чого результат хешується. До цього значення конкатенуються повідомлення, після чого розраховується XOR ключа з сталим значенням, конкатенується з хеш-значенням, після чого знову хешується. У підсумку отримуємо 512 біт хеш-значення.

Розглянемо декілька функцій, які використовуються при розрахунку ієрархічних ключів[14].

Derivation functions:

- Master seed => master key;
- Private parent key => private child key;
- Private parent key =>public child key;
- Public parent key =>public child key;
- Public parent key X private child key;

Насамперед нас цікавить перетворення masterseed в пару masterkey. Після цього потрібно отримати з особистого батьківського ключа дочірній особистий ключ і

дочірній відкритий ключ. У другому випадку використовується така сама функція, як і в першому, але додається множення числа на базову точку, тому окремо не розглядатиметься. Далі слідує отримання з відкритого батьківського ключа дочірнього відкритого. Варто зазначити, що отримання з батьківського відкритого ключа дочірнього особистого неможливо. Це обмеження обумовлено певними властивостями, притаманними HD гаманцях, які ми розглянемо далі.

Отже, розглянемо кожну з функцій породження окремо.

- Master seed => master key

$$hash = HMAC(\text{Bitcoin seed}, Seed)$$
$$hash = (k||CC)$$

де k – головний приватний ключ,
CC – код ланцюга.

Для отримання майстер-ключа з майстер-сіда використовується функція HMAC, де в якості ключа передається константний рядок "Bitcoinseed", а в якості повідомлення самі дані основного секрету. Таким чином, виходить хеш-значення довжиною 512 біт, яке ми розглядаємо як дві частини: ліву і праву. Ліва частина є головним приватним ключем, а права частина буде кодом ланцюга. Далі ці значення будуть використовуватися для породження наступних рівнів дочірніх ключів.

Для того щоб отримати головний публічний ключ, досить помножити значення базової точки на значення головного приватного ключа. Як бачимо, це відбувається за аналогією зі звичайними ключами в групі точок на еліптичній кривій.

Тепер подивимося як отримується дочірній приватний ключ з батьківського приватного ключа.

- Private parent key => private child key;

$$I = HMAC(CC_{par}, ser_P(point(k_{par}))||ser_{32}(i))$$
$$k_i = I_L + k_{par}(mod\ n)$$
$$CC_i = I_R$$

$$K_i = point(k_i)$$

Знову скористаємося функцією HMAC. В якості ключа передаємо код ланцюга поточного рівня ієрархії, а в якості повідомлення - конкатенацію, де першою частиною буде приватний батьківський ключ, помножений на базову точку. Фактично це приведення до точки і серіалізація цієї точки. Конкатенація відбувається з індексом дочірнього ключа, який ми хочемо отримати, серіалізованим в 32 біта, тобто в 4 байта.

За результатом роботи функції HMAC ми отримуємо значення I, і знову його розглядаємо окремо: ліву і праву частини вихідних значень по 256 біт. Тоді дочірній приватний ключk_i ми розраховуємо шляхом додавання до лівого вихідного значення I_Lзначення батьківського приватного ключа. Додавання виконуємо по модулю n, де n - це порядок базової точки еліптичної кривої, для того щоб не перевищити максимальне значення приватного ключа. Таким чином, ми отримали дочірній приватний ключ[14].

Відповідно, дочірній код ланцюга буде дорівнює правому значенню функції HMAC, тобтоI_R. Якщо ми хочемо з приватного батьківського ключа знайти дочірній публічний ключ, то потрібно множити значенняk_i на значення базової точки на еліптичній кривій. Так ми отримаємо відкритий ключK_i.

- Public parent key => public child key;

$$I = HMAC(CC_{par}, ser_p(K_{par}) || ser_{32}(i))$$
$$K_i = point\left(parse_{256}(I_L)\right) + K_{par}$$
$$CC_i = I_R$$

Тут розрахунок буде трохи іншим. Ми передаємо в якості ключа код ланцюга поточного рівня ієрархії в функцію HMAC, після чого ми серіалізуємо батьківський публічний ключ і конкатенуємо його з потрібним індексом, серіалізованим в 32 біта. Вхідні дані отримані точно таким же чином, як і в попередньому випадку. Для розрахунку публічного ключа ми беремо ліву частину вихідного значення функції

HMAC, і розглядаємо його, як 256 біт, взятих по модулю порядку базової точки, наводимо до точки на еліптичній кривій, тобто множимо на базову точку, після чого складаємо результат з батьківським публічним ключем . Результатом складання буде теж точка, і це буде публічний дочірній ключ. Код ланцюга для даного ключа буде права частина вихідного значення функції HMAC[14].

2.5 Аналіз шляхів породження ключів

На кожному рівні ієрархії є певний індекс, який визначає аспекти породження ключів. Шлях від Майстер-ключа до кінцевого ключа може записуватися через «/» у вигляді індексів. Якщо мова йде про приватні ключі то запис починається з маленької "m", а якщо мова про породження публічного ключа, то з великою "M". Якщо індекс позначений апострофом, то слід розуміти, що мова йде про ієрархічне породження, без апострофа - звичайне.

Розглянемо один з популярних шляхів породження ключів, який використовується в BIP32, де і були визначені ієрархічні ключі.

$$m/accounts`/chains/addresses$$

$$m/0`/0/0$$

Він використовує такий шлях, де нульовим рівнем ієрархії є майстер-ключ. Далі йдуть індекси аккаунтів, які позначають одного і того ж користувача, після чого йдуть ланцюжки, де можуть бути ланцюжки адрес, які публікуються зовні для прийняття вхідних платежів, а з індексом 1 створюватимуться ті ланцюжки, на які сам користувач відправляє собі платежі (частіше за все це решта). Кінцевий індекс буде використовуватися для породження тих ключів, з яких будуть розраховуватися адреси[16].

Для того щоб за стандартом BIP32 розрахувати найперший ключ з індексом 0, ми будемо мати m, 0 з породженням ієрархії, ланцюжок - 0, індекс - 0 (m / 0 '/ 0/0). Так ми отримаємо шлях для першого ієрархічно породженого ключа.

Існує пропозиція щодо поліпшення Біткоїна, вона називається BIP43, яка передбачає запис в перший рівень ієрархії номера поліпшення, яке пропонує новий шлях породження (m / bip_number '/ *).

$$m \,/\, \text{bip\_number}\,'/\,*$$

Таким чином, з'явився BIP44, який використовував особливість попередньоїпропозиції, тобто для першого рівня ієрархії записується індекс 44, а запропонував наступні поліпшення: в індексі другого рівня ієрархії записувати певне значення, яке буде відповідати типу монети, яку ми використовуємо для даного гаманця. Тепер в одному гаманці можуть розгортатися і використовуватися ключі для різних валют.

$$m/44`/currency\_number/account/chain/address$$

Для Bitcoin шлях буде виглядати, як "m / 44 '/ 0' / 0 '/ 0/0", для тестової мережі Bitcoin - "m / 44' / 1 '/ 0' / 0/0", для Litecoin - "m / 44 '/ 2' / 0 '/ 0/0 ", для Dash -" m / 44' / 5 '/ 0' / 0/0 ". Цікаво, що Ethereum використовує точно таку ж еліптичну криву для розрахунку ключів і електронно-цифрового підпису і для його гаманця шлях буде виглядати так "m / 44 '/ 60' / 0 '/ 0/0".

Є ще одне поліпшення - BIP45. Поліпшення націлене на те, щоб визначити правила породження ключів в разі їх використання в multisignature гаманцях і формування адрес по BIP16, тобто P2SH. Він включає в себе пропозицію BIP43 і вказує на першому рівні ієрархії індекс 45, на другому ж рівні ієрархії він вимагає вказівки підписувача (cosigner).

$$m/45`/cosigner/chain/address$$

Наприклад, є правило мультипідпису 3-з-5. Отже є 5 підписантів, але щоб витратити монети, потрібно як мінімум 3 підписи. Таким чином, кожен з підписантів матиме HD гаманець зі своїм майстер-сідом, а в своєму шляху буде вказувати свій

порядковий номер. Він може бути розрахований, як індекс при сортуванні ключів, породжених на першому рівні ієрархії кожного користувача. Припустимо, на першому рівні відбулося породження ключів у кожного користувача, вони обмінюються один з одним, сортують і дізнаються, у кого який індекс для другого рівня ієрархії. Це потрібно, щоб в подальшому виключити необхідність взаємодіяти подібним чином, а відразу правильно генерувати адреси і знати свій порядковий номер.

Тобто можна один раз обмінятися розширеним відкритим ключем, щоб потім самостійно, незалежно від інших учасників групи, формувати multisignature адреси і приймати на них платежі[16].

2.6 Висновки

У другому розділі було розглянуто і проаналізовано існуючі алгоритми консенсусу, що використовуються у сучасних криптовалютах. Наведено їх переваги та недоліки. Проведено детальний аналіз криптостійкості, що надає технологія блокчейн. Порівняно найпоширеніші хеш-функції, що використовуються в різних видах блокчейнів, та наведено їх переваги та недоліки.

Розглянуто як відбувається ієрархічна генерація ключів, що таке детерміністичні гаманці і яких видів вони бувають. Проаналізовано функції, які використовуються для ієрархічної генерації ключів.

Проведено аналіз шляхів породження ключів. Розглянуто їх особливості, порівно їх між собою та описано їх переваги і недоліки.

3. ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ УПРАВЛІННЯ КРИПТОВАЛЮТНИМИ АКТИВАМИ

3.1 Обґрунтування вибору мови програмування та середовища розробки

Web, як гіпертекстову систему, можна розглядати з двох точок зору. По-перше, як сукупність відображуваних сторінок, пов'язаних гіпертекстовими переходами. По-друге, як безліч елементарних інформаційних об'єктів, що становлять відображаються сторінки (текст, графіка, мобільний код і т.п.). В останньому випадку безліч гіпертекстових переходів сторінки - це такий же інформаційний фрагмент, як і вбудована в текст картинка.

При другому підході гіпертекстова мережа визначається на безлічі елементарних інформаційних об'єктів самими HTML-сторінками, які і виступають в ролі гіпертекстових зв'язків. Цей підхід більш продуктивний з точки зору побудови відображуваних сторінок "на льоту" з готових компонентів.

При генерації сторінок в Web виникає дилема, пов'язана з архітектурою "клієнт-сервер". Сторінки можна генерувати як на стороні клієнта, так і на стороні сервера. Останнє реалізується через механізм підстановок на стороні сервера (Server SiteIncludes). Компанія Netscape розповсюдила в 1995 році механізм управління сторінками і на клієнта, розробивши мову програмування JavaScript[13].

Таким чином, JavaScript - це мова керування сценаріями перегляду гіпертекстових сторінок Web на стороні клієнта. Якщо бути більш точним, то JavaScript - це не тільки мова програмування на стороні клієнта. Liveware, прабатько JavaScript, є засобом підстановок на стороні сервера Netscape. Однак, найбільшу популярність JavaScript забезпечило програмування на стороні клієнта.

До можливостей JavaScript можна, наприклад, віднести наступне:

• відображати дані, що змінюються, такі як поточний час або дата;

• програмувати змінне значення в залежності від дати, броузера користувача або інших умов;

• змінювати зовнішній вигляд елементів сторінки, якщо користувач клацнув мишею або провів курсор миші над елементом[13].

Для мови високого рівня JavaScript володіє досить сильними можливостями. Він не дозволяє працювати на рівні машинних кодів, однак ви отримуєте доступ до багатьох можливостей браузерів, Web-сторінок, а іноді і системи, в якій працює броузер. На відміну від Java ™ або C, програми на JavaScript обходяться без

компіляції, а вашому браузеру не доведеться завантажувати віртуальну машину для виконання програмного коду.

JavaScript також працює в об'єктно-орієнтованої архітектури, що нагадує Java або C ++. Такі можливості мови, як конструктори або спадкування на базі прототипів, додають в схему розробки новий рівень абстракції, що сприяє багаторазовому використанню програмного коду.

Одна з головних причин, по якій Web-розробники обрали JavaScript, - можливість виконання на стороні клієнта багатьох функцій, які раніше виконувалися виключно на стороні сервера. Кращим прикладом є перевірка форм. Програмісти старої школи ще пам'ятають, що кілька років тому для перевірки користувальницького введення в формах HTML доводилося пересилати інформацію на Web-сервер і передавати її сценарієм CGI, де і проходила перевірка введених даних[13].

Якщо дані не містили помилок, сценарій CGI продовжував роботу. Однак при виявленні помилок сценарій повертав користувачеві повідомлення з описом проблеми. Хоча це рішення працює, уявіть, скільки зайвої роботи при цьому відбувається. Для передачі форми необхідний спеціальний запит HTTP від сервера. Після пересилки даних в Мережі доводиться заново виконувати сценарій CGI. Цей процес повторюється кожного разу, коли користувач припускається помилки при заповненні форми. Користувач дізнається про помилку лише після того, як повідомлення про помилку повернеться до нього.

Але ось на сцені з'являється JavaScript. Тепер елементи форми можна перевірити до того, як користувач передасть інформацію Web-сервера. Це призводить до зменшення кількості транзакцій HTTP, а також помітного зниження ймовірності помилки при повторному заповненні форми. Крім того, JavaScript дозволяє читати і записувати cookie - колись ця операція виконувалася виключно засобами Web-сервера для роботи з заголовками.

При виборі мови програмування найбільш значущим критерієм був досвід програмування на даній мові, і, оскільки, я має досвід та працюю використовуючи JS та його переваги дуже значущі, то це однозначний вибір[13].

Для роботи з JS розглядалось декілька найбільш популярних середовищ програмування:

1) SublimeText

На думку багатьох, SublimeText - це кращий з кращих, найшвидший редактор коду. Низький вбудований функціонал з лишком компенсується величезною кількістю плагінів. Але на відміну від того ж Vim, він має низький поріг входження і людський інтерфейс. SublimeText також можна розглядати в якості легковагій IDE.

Переваги SublimeText:

У SublimeText дуже приємна смуга прокрутки з превью коду, і, звичайно ж, можливість редагувати текст за допомогою кількох курсорів. Ще з фішок є підсвічування і автокомпліт практично чого завгодно, непоганий пошук по проекту, що дозволяє знайти потрібний рядок або файл всього по декількох буквах, автоматичний перенос слів по заданій ширині рядка, перевірка граматики, підтримка різних кодувань і переносів рядків, що настроюється ширина відступів - загалом, все, щоб вам було комфортно і зручно працювати. Ще до плюсів можна віднести гнучке налаштування шрифтів і колірних схем.

Недоліки SublimeText:

До мінусів редактора можна віднести проблеми з зворотну сумісність і відсутність вбудованої консолі.

SublimeText - легкий і неймовірно розширюваний редактор, який, при грамотному використанні, може виконувати будь-які завдання, пов'язані з редагуванням тексту, будь то код, файли конфігурацій або послання інопланетян на невідомій мові. Якщо ви цінуєте зручність роботи і вважаєте, що великовагові IDE не для вас, то SublimeText - однозначно ваш вибір[13].

2) VisualStudioCode

VisualStudioCode – багатоплатформений редактор коду, що підтримує базові можливості інтегрованого середовища розробки (IDE), створений в Microsoft.

Переваги VisualStudioCode:

Позиціонується як «легкий» редактор коду для кроссплатформенної розробки веб і хмарних додатків. VisualStudioCode поширюється безкоштовно і розробляється як програмне забезпечення з відкритим вихідним кодом.

VisualStudioCode дозволяє розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ.

У редакторі присутні вбудований відладчик, інструменти для роботи з Git і засоби рефакторинга, навігації по коду, автодоповнення типових конструкцій і контекстної підказки. Продукт підтримує розробку для платформ ASP.NET і Node.js, і вважається легковажним рішення, яке дозволяє обійтися без повної інтегрованого середовища розробки. Великим плюсом редактора є підтримка великої кількості мов, таких як C ++, C #, Python, PHP, JavaScript та інших.

VisualStudioCode вийшов відносно недавно і вже почав поступово набирати свою популярність. Якщо вам хочеться спробувати в цьому році щось новеньке, то варто сміливо зупинити ваш вибір на цьому редакторі[13].

3) WebStorm

WebStorm - середовище розробки для JavaScript, так само підходить як для frontend'a, так і для створення додатків на Node.js. Цей інструмент розроблено компанією JetBrains і є платним, проте можно використовувати 30-денну безкоштовну ліцензію чи ліцензійні сервери

Переваги WebStorm

Його головною перевагою є зручний і розумний редактор JavaScript, HTML і CSS, який підтримує також і інші мови, наприклад TypeScript, CoffeeScript, Dart, Less, Sass і Stylus і фреймворки, наприклад, Angular, React і Meteor.

WebStorm робить розробку проекту простий і зручною, забезпечуючи підсвічування і автодоповнення коду, його аналіз по ходу редагування, швидку навігацію і рефакторинг. Він має потужні інструменти налагодження та інтеграції з системами управління версіями (Git, GitHub, Subversion, Perforce, Mercurial, CVS), розуміє структуру проекту і код, відстежує помилки за допомогою систем ESLint,

JSHint, JSLint, TSLint, Stylelint і пропонує їх рішення. Вбудовані в IDE інструменти для тестування і роботи з проектом допомагають в розробці і роблять її зручніше і продуктивніше[13].

4) Atom

Atom - це безкоштовний текстовий редактор з відкритим вихідним кодом і підтримкою плагінів, написаних на Node.js, і вбудованих під керуванням GitControl. Його творцями є програмісти веб-сервісу Github, які вирішили допомогти своїй цільовій аудиторії і створити щось, що розробники можуть використовувати кожен день.

Переваги Atom

Головна особливість Atom - багато можливостей по налаштуванню. Редактор можна налаштувати на свій смак. Спочатку в нього вбудовані файл-менеджер, просунуті функції пошуку і заміни, різноманітні курсори, опції згортання коду, ясний інтерфейс, можливість імпорту правил і тим з TextMate.

Десктопних програм Atom має повний доступ до файлової системи, природні для операційної системи меню та панель команд. При цьому воно ідеально пристосоване для веб-програмування: можна додавати власні функції для редагування CSS, HTML і JavaScript. Потрібно відзначити також інтеграцію з Node.js, включаючи запуск веб-сервера прямо з редактора. Архітектура програми проста і зрозуміла кожному: можна замінити будь-який пакет своїм власним і закачати його в центральний репозиторій, щоб їм скористався кожен охочий.

Такий воістину прекрасний редактор все ж більш важкоатлет, ніж SublimeText. Але можна сказати точно, що Atom - гідний редактор для зручної роботи програміста з безліччю приємних доповнень[13].

Отже, врахувавши усі переваги і недоліки та власний досвід використання було обрано WebStorm як оптимальне середовище програмування.

3.2 Обґрунтування вибору технологій, фреймворків і бібліотек

Javascript, який спочатку був основною технологією динамічних веб-сайтів в Інтернеті, захоплює світ розробки додатків штурмом. Завдяки впровадженню Javascript фреймворків для створення мобільних додатків, бізнес та розробники знайшли оптимальне рішення для створення інтерактивного інтерфейсу з меншою кількістю рядків коду.

Враховуючи популярність Javascript, є сотні фреймворків Javascript, доступних для створення додатків в Інтернеті, для настільних ПК та для мобільних пристрїв. Розглянемо найпопулярніші фреймворки Javascript, які допомагають розробникам створювати крос-платформні рідні та гібридні мобільні додатки[15].

1) React Native

Представлений Facebook у 2015 році, React Native - це програма з відкритим кодом для створення рідних крос-платформних додатків. Використовуючи React і Javascript в якості мови програмування, розробники можуть створювати мобільні додатки, які не відрізняються від рідних додатків, створених за допомогою Objective-C, Swift або Java.

React Native - це розширення React, яке є бібліотекою Javascript для створення блоків інтерфейсу для веб-додатків. Декларативний стиль програмування, віртуальний DOM, багаторазові компоненти для побудови інтерфейсу - це одні з переваг, що надають цьому JS-фреймворку перевагу над десятками інших варіантів для створення мобільних додатків

Переваги React Native:

- Розробники можуть комбінувати компоненти рідного коду, написаного на Objective-C, Swift або Java з кодом React Native для створення масштабованих програм.
- Завдяки сильній підтримці мови п-рограмування Javascript, розробники легко засвоюють React і таким чином створюють додатки React Native.
- Цей фреймворк JavaScript дозволяє створювати крос-платформні мобільні додатки. Додатки React Native займають 80% ринку для платформ Android та iOS, написаних на JS, а це означає, що на розробників даного фреймворку є попит, коли мова йде про створення складних мобільних додатків[15].

2) NativeScript

NativeScript - це фреймворк з відкритим кодом для створення рідних додатків між платформами з Angular, Vue.js, TypeScript або Javascript. Це дає змогу веб-розробникам використовувати свій набір навичок для створення власного програмного забезпечення на мобільних пристроях.

NativeScript дозволяє розробникам використовувати Vue CLI, VueX та інші можливості фреймворку Vue для створення рідного мобільного додатка. Також NativeScript інтегрується із сучасними full-stack можливостями Angular, такими як підтримка маршрутизатора, генерація коду, інтеграція з Angular CLI тощо.

Переваги NativeScript:

- З NativeScript розробники можуть скористатися наявними плагінами з npm, CocoaPods (iOS), Gradle (Android), а також сотнями плагінів NativeScript.
- Єдину кодову базу можна записати та розгорнути на декількох платформах (Android та iOS). Для спільного використання коду можна використовувати Angular або Vue.js.
- Веб-розробники можуть використовувати наявні веб-навички для (Javascript, CSS, Native UI Markup) для розробки рідних мобільних додатків[15].

3) Ionic

Ionic - ще один популярний фреймворк javascript для створення гібридних додатків. Для розробників, які ознайомлюються з веб-технологіями та розробкою веб-додатків, зрозуміти структуру додатка Ionic досить просто.

Гібридні програми працюють у повноекранному браузері під назвою WebView, який невидимий для користувачів. За допомогою налаштованих рідних плагінів вони можуть отримати доступ до вбудованих функцій мобільних пристроїв, таких як камера, ідентифікатор сенсорного пристрою тощо, без основного коду, підключеного до пристрою.

Переваги Ionic:

- Кодова база даних гібриду може використовуватися для декількох платформ (Android та iOS), тим самим знижуючи витрати на розробку (порівняно з рідними додатками) та час на ринок.

- Створюючи гібридні програми за допомогою Ionic, розробники можуть отримати доступ до плагінів Cordova, що допомагає їм отримати доступ до апаратних та програмних можливостей мобільного пристрою.
- Наявна експертиза веб-розробки (HTML, CSS та Javascript) може бути використана для створення[15].

4) Apache Cordova

Apache Cordova (раніше PhoneGap) - це гібридний фреймворк розробки додатків, який загортає додаток HTML або Javascript у рідний контейнер. Існує довгий список інструментів, рамок та хмарних сервісів, які доступні для збільшення продуктивності Cordova. Деякі з популярних імен включають Visual Studio, Ionic, Framework7, Monaca, Mobiscroll тощо. Враховуючи потенціал, який приносить Cordova, учасниками цього фреймворку є деякі технічні гіганти, зокрема Adobe, Microsoft, Blackberry, IBM, Intel тощо .

Переваги Apache Cordova:

- Існує величезна спільнота, яка розробляє плагіни для Cordova. В результаті фреймворк полегшує доступ до широкого спектру плагінів, дозволяючи розробникам отримувати доступ до функцій пристрою, тим самим розширюючи сферу застосування та масштаб програми.
- Використовуючи Cordova, можна розгорнути код на декількох платформах (iOS, Android), що зробило розробку додатків вигідною справою[15].

5) OnSenUI

OnSen UI - це програма JS для створення гібридних мобільних додатків за допомогою HTML, CSS та Javascript. Інтерфейс Onsen сумісний як з AngularJS, так і з Angular 2+, React, Vue та jQuery, щоб розробники могли перемикатися між різними бібліотеками та рамками для побудови інтерактивних інтерфейсів.

Переваги OnSenUI:

- OnSenUI пропонує великий набір багатих компонентів інтерфейсу, розроблених спеціально для мобільних додатків.
- OnSenUI чудово працює з Monaca. Monaca пропонує потужний інструмент командного рядка та настільний додаток для спрощення складних завдань[15].

Проаналізувавши найбільш популярні фреймворки для мобільної розробки на Javascript будо обрано ReactNative як найбільш широко застосовуваний, а також як той, що має підтримку гіганта Facebook.

3.3 Тестування інформаційної технології управління криптовалютними активами

Відповідно до результатів дослідження, усі криптовалютні гаманці оцінювались за такими характеристиками, як:

- можливість імпортувати і експортувати мнемонічну фразу і якої довжини;
- мультивалютність;
- мультиплатформність;
- можливість обміну криптовалют всередині додатку:
    - швидкість;
    - комісії;
    - курс;
    - необхідність верифікації;
- можливість купівлі криптовалют всередині додатку:
    - швидкість;
    - комісії;
    - курс;
    - необхідність верифікації;
- можливість продажу криптовалют всередині додатку:
    - швидкість;
    - комісії;
    - курс;
    - необхідність верифікації;

Отже, після проведення ряду тестів (при купівлі і продажу за основну фіатну валюту було обрано гривню, а за криптовалюту – Ethereum, а обмінювався Bitcoin на

Ethereum) отримано результати дослідження, які наведені у таблицях 3.1(для купівлі), 3.2(для продажу), 3.3(для обміну).

Таблиця 3.1 – Результати тестування(купівля)

Назва гаманця	Кастодіальність	Мульти-валютність	Мульти-Платформність	Купівля			
				Швидкість	Комісія	Курс	Верифікація
Metamask	Ні (12 слів)	Ні	Розширення та мобільний додаток	Відсутня			
Ledger	Ні (24 слова)	Так	Десктопний і мобільний додаток	Відсутня			
Exodus	Так	Так	Десктопний додаток	35 сек	4%	3694:1	Так
Paytomat	Ні (24 слова)	Так	Мобільний додаток	110 сек	5%	3820:1	Так
Розроблена інформаційна технологія	Ні (24 слова)	Так	Мобільний додаток	45 сек	1.7%	3602:1	Ні

Таблиця 3.2 – Результати тестування(продаж)

Назва гаманця	Кастодіальність	Мульти-валютність	Мульти-Платформність	Продаж			
				Швидкість	Комісія	Курс	Верифікація
Metamask	Ні (12 слів)	Ні	Розширення та мобільний додаток	Відсутня			
Ledger	Ні (24 слова)	Так	Десктопний і мобільний додаток	Відсутня			
Exodus	Так	Так	Десктопний додаток	40 сек	4%	3360:1	Так
Paytomat	Ні (24 слова)	Так	Мобільний додаток	190-? сек	5%	3200:1	Так
Розроблена інформаційна технологія	Ні (24 слова)	Так	Мобільний додаток	50 сек	1.7%	3440:1	Ні

Таблиця 3.3 – Результати тестування(обмін)

Назва гаманця	Кастодіальність	Мульти-валютність	Мульти-Платформність	Обмін			
				Швидкість	Комісія	Курс	Верифікація
Metamask	Ні (12 слів)	Ні	Розширення та мобільний додаток	Відсутня			
Ledger	Ні (24 слова)	Так	Десктопний і мобільний додаток	Відсутня			
Exodus	Так	Так	Десктопний додаток	120 сек	0.75%	49.01:1	Так
Paytomat	Ні (24 слова)	Так	Мобільний додаток	? сек	5%	48.61:1	Так
Розроблена інформаційна технологія	Ні (24 слова)	Так	Мобільний додаток	600 сек	0.5%	49.39:1	Ні

Відповідно до одержаних результатів можна зробити висновки:

- У кастодільних рішень швидкість проведення операцій купівлі, продажу і обміну вище, якщо мова не йде про вивід коштів з їх адрес (як було проаналізовано раніше, це локальні адреси і вони не знаходяться в блокчейні, а є підконтрольними біржам і тд), що також призводить до великих ризиків з боку обмеження контролю ваших коштів, а при виводі їх з біржі ще потрібно заплатити комісію і чекати виконання транзакцій вже у блокчейні;
- Серед некастодіальних рішень не усі гаманці використовують мнемонічну фразу з 24 слів, що понижує криптостійкість;
- Серед некастодіальних рішень швидкість розробленої інформаційної технології є найвищою.
- Завдяки використання розробленого алгоритму купівлі і продажу через p2p зменшено комісію та підвищено швидкість порівняно з іншими рішеннями.
- Отримані найкращі курсу для купівлі, продажу і обміну криптовалюти завдяки алгоритму пошуку найкращого курсу.

3.4 Тестовий приклад роботи інформаційної технології управління криптовалютними активами

Для початку роботи потрібно встановити додаток собі на мобільних телефон за допомогою Play Market (Android) або App Store (iOS). Після чого потрібно увійти у нього і Вам буде запропоновано створити гаманець (рис 3.1). Тут можна обрати зі скількох слів буду складатись Ваша мнемонічна фраза. Краще обирати 24 слова, це гарантує більший захист.

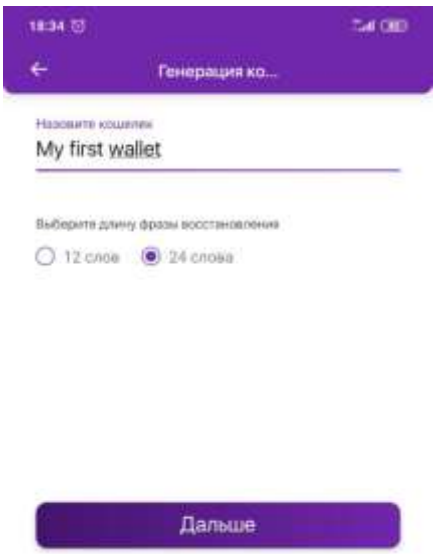


Рисунок 3.1 – Вікно створення гаманця

Далі Вам буде показана Ваша мнемонічна фраза і її потрібно буде зберегти, записати або запам’ятати (рис 3.2). Це обов’язково потрібно зробити, оскільки лише вона дає доступ до цього гаманця. Нікому її не повідомляйте!

Після того як мнемонічна фраза буде підтверджена відкриється головне вікно додатку (рис 3.3). У цьому вікні зверху можна побачити назву, що була обрана для цього гаманця та загальний баланс усіх наявних у Вас криптовалют в еквіваленті обраної основної валюти. Також тут зображено криптовалюти, що були обрані для показу і їх баланс. Ще тут можна перейти до функцій додання активних криптовалют, обміну, купівлі, продажу та вікна налаштувань.

Якщо Ви хочете додати більше криптовалют на головне вікно, то, після обрання потрібної опції, Вам відкриється вікно редагування списку криптовалют (рис 3.4), де Ви зможете додавати та забирати криптовалюти з головного вікна.

При натисканні будь-яку криптовалюту буде відкрито вікно роботи з обраною криптовалютою (рис. 3.5), де буде показано її баланс, еквівалент у обраній фіатній валюті, її адреса, транзакції, а також 2 функції: отримання (рис. 3.6) і відправка (рис. 3.7).

Рисунок 3.2 – Вікно запису мнемонічної фрази

При обрані функції купівлі буде відкрито вікно (рис. 3.8), де буде зображено процес купівлі. Тут можна обрати за яку валюту буде куплятись обрана криптовалюта, а також, через який платіжний метод. Після підтвердження зі сторони банку, чи іншої платіжної установи, котрою проводиться оплата, відбудеться купівля.

При обрані функції продажу буде відкрито вікно (рис. 3.9), де буде зображено процес купівлі. Тут можна обрати за яку валюту буде куплятись обрана криптовалюта, а також, через який платіжний метод. Криптовалюта спишеться з Вашої адреси, а на Вашу карту прийде фіатна валюта.

Рисунок 3.3 – Головне вікно додатку

Рисунок 3.4 – Вікно редагування списку криптовалют

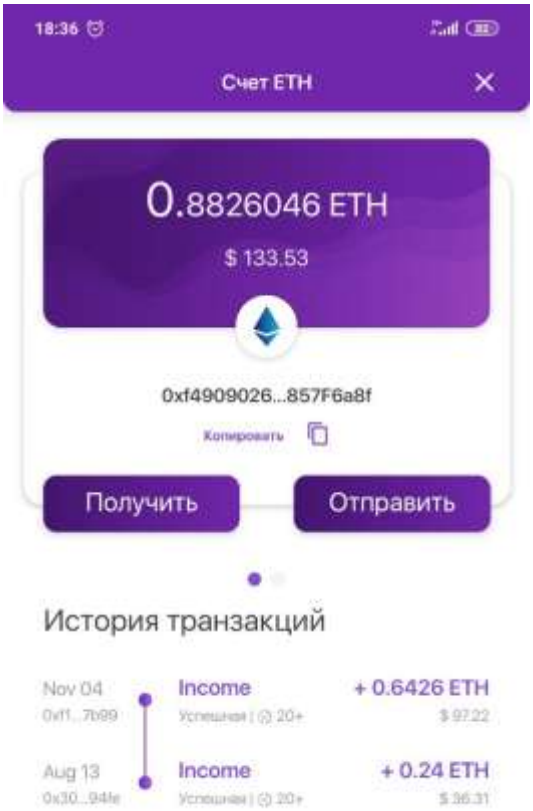


Рисунок 3.5 – Вікно обраної криптовалюти

Рисунок 3.6 – Вікно отримання криптовалюти

Рисунок 3.7 – Вікно відправки криптовалюти

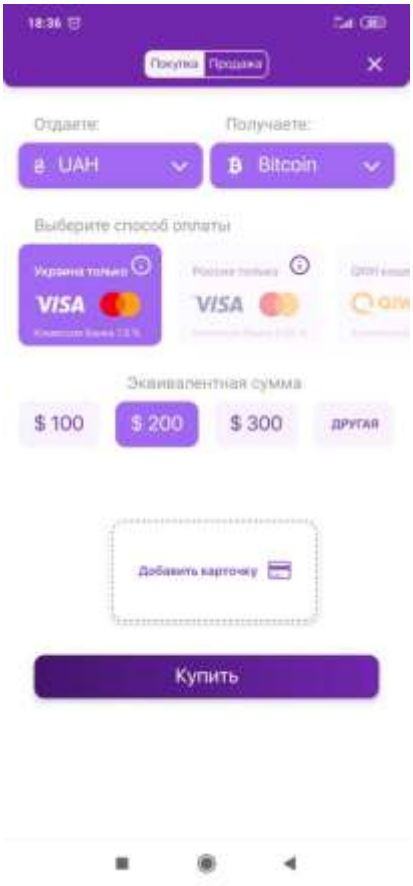


Рисунок 3.8 – Вікно купівлі

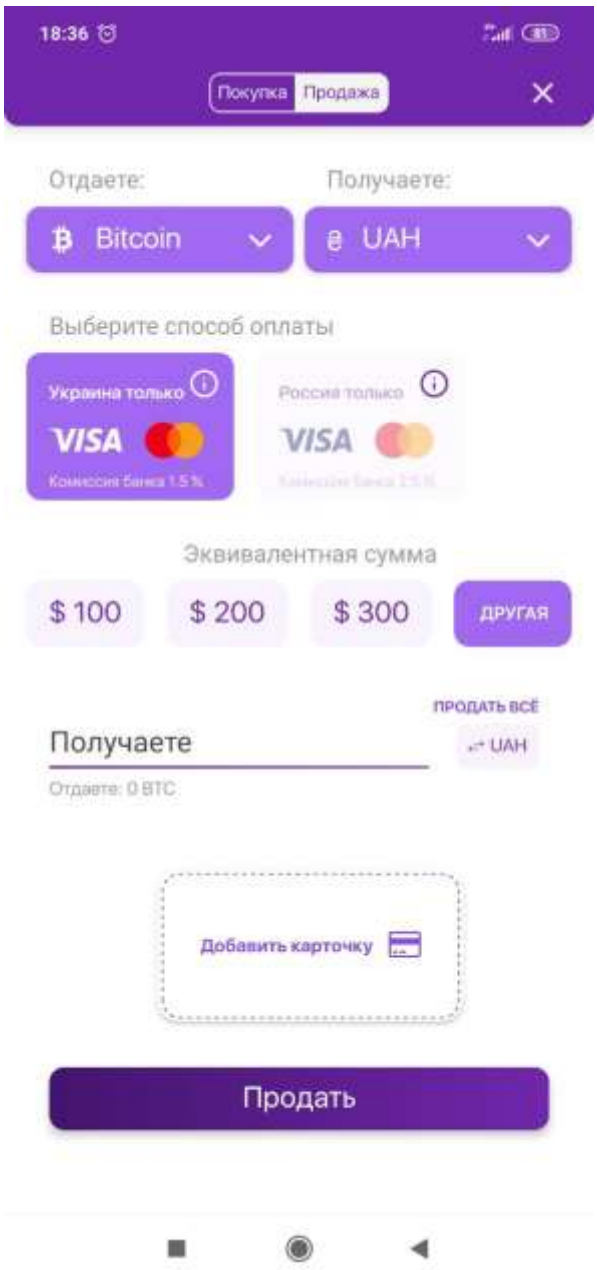


Рисунок 3.9 – Вікно продажу

При обрані функції продажу буде відкрито вікно (рис. 3.10), де буде зображено процес обміну. Тут обираються криптовалюти, що підлягають обміну, а також провайдер через якого обмін відбудеться. Швидкість обміну залежить лише від поточної завантаженості блокчейн мережі.

В вікні налаштувань (рис 3.11) користувача буде чекати велика кількість налаштувань різних типів. Від резервування, імпортування і створення гаманця та додавання активів до налаштувань захисту, локальної валюти, мови та інших.

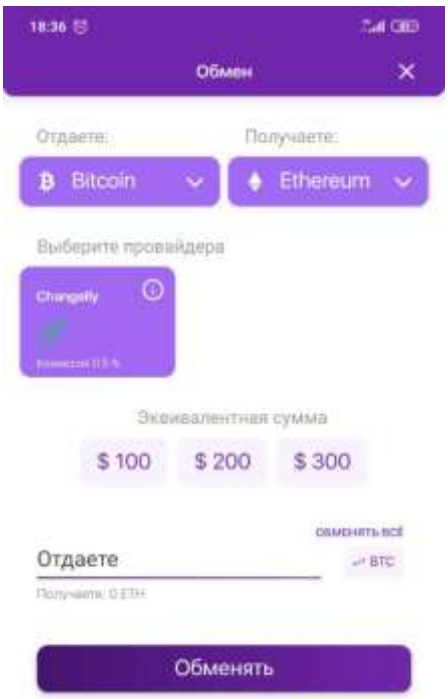


Рисунок 3.10 – Вікно обміну

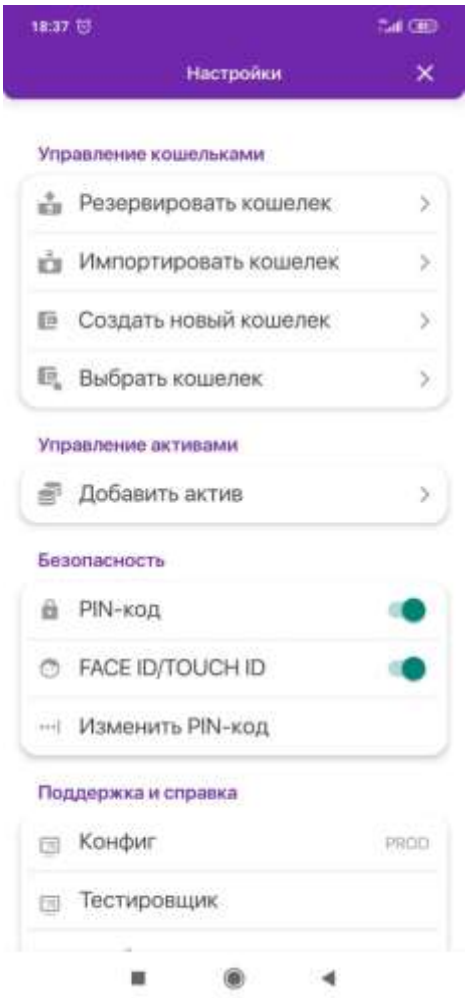


Рисунок 3.11 – Створення гаманця

3.5 Висновки

В даному розділі описано етап програмної реалізації інформаційної технології управляння криптовалютними активами.

Було проведено тестування створення, імпортування і експортування мнемонічної фрази гаманця, що показало найбільший захист коштів при використання мнемонічної фрази з 24 слів, котру потім можна дуже швидко розгорнути на інших некастодіальних рішеннях.

Були протестовані операції купівлі, продажу і обміну серед рішень, які мали таку можливість, і визначено, що серед некастодіальних рішень створена інформаційна технологія має переваги у швидкості, а між усіма по курсу, найнижчим комісіям, а також в ряді випадків не потребує верифікації.

Здійснено обґрунтування вибору мови програмування, а також інших програмних засобів. В результаті було обрано мову програмування JS і React Native, що дозволяються суттєво пришвидшити розробку.

Проведено тестування розробленого додатку. При тестуванні проводилось декілька запусків для кожної з операцій, змінюючи комісію мережі, та у різний час її завантаженості.

Додаток працює у режимі реального часу, завжди відпрацьовував вірно і у межах очікуваного часу. Результати, які були отримані під час тестування роботи додатку, відповідають результатам, які слідувало очікувати.

4. ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено 2-х незалежних експертів. Такими експертами будуть Сілагін О.В. та Бальзан М.В.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями за 5-ти бальною шкалою.

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 4.1.

Таблиця 4.1 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта	
	1. Експерт 1	2. Експерт 2
	Бали, виставлені експертами:	
1	4	4
2	3	3
3	4	4
4	4	4
5	3	3
6	3	4
7	4	3
8	3	4
9	4	4
10	4	4
11	3	4
12	3	4
Сума балів	$СБ_1 = 43$	$СБ_2 = 45$
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{1}^{3} СБ_i}{2} = 44$	

Отже, з отриманих даних таблиці 4.1 видно, що нова розробка має високий рівень комерційного потенціалу.

4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи

Для розробки нового програмного продукту необхідні такі витрати.

Основна заробітна плата для розробників визначається за формулою (4.1):

$$З_о = \frac{М}{Т_р} \cdot t, \quad (4.1)$$

де М- місячний посадовий оклад конкретного розробника;

$Т_р$ - кількість робочих днів у місяці, $Т_р = 22$ дні;

t - число днів роботи розробника, t = 45 днів.

Розрахунки заробітних плат для керівника і програміста наведені в таблиці 4.2.

Таблиця 4.2 – Розрахунки основної заробітної плати

Працівник	Оклад М, грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник	6000	272,72	5	1363,6
Інженер-програміст	3500	159,09	45	7159,05
Всього:				8522,65

Розрахуємо додаткову заробітну плату:

$$З_{дод} = 0,1 \cdot 8522,65 = 852,26 \text{ (грн.)}$$

Нарахування на заробітну плату операторів НЗП розраховується як 37,5…40% від суми їхньої основної та додаткової заробітної плати:

$$Н_{зп} = (З_о + З_р) \cdot \frac{\beta}{100}, \quad (5.2)$$

$$Н_{зп} = (8522,65+852,26) \cdot \frac{36,3}{100} = 3403,09 \text{ (грн.)}.$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за такою формулою:

$$А=\frac{Ц \cdot Н_а}{100} \cdot \frac{Т}{12}, \quad (4.3)$$

де Ц – балансова вартість обладнання, грн;

$Н_а$ – річна норма амортизаційних відрахувань % (для програмного забезпечення 25%);

Т – Термін використання (Т=3 міс.).

Таблиця 4.3 – Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань,грн
Персональний комп’ютер	10000	25	3	625
Всього:				625

Розрахуємо витрати на комплектуючі. Витрати на комплектуючі розрахуємо за формулою:

$$K = \sum_1^n H_i \cdot Ц_i \cdot K_i, \quad (4.4)$$

де n – кількість комплектуючих;

Ні - кількість комплектуючих і-го виду;

Ці – покупна ціна комплектуючих і-го виду, грн;

Кі – коефіцієнт транспортних витрат (приймемо Кі = 1,1).

Таблиця 4.4 - Витрати на комплектуючі, що були використані для розробки ПЗ.

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Пачка паперу	уп.	150	1	150
Ручка	шт.	10	1	10
Всього з урахуванням транспортних витрат				176

Витрати на силову електроенергію розраховуються за формулою:

$$В_е=В·П·Ф·К_п ; \quad (4.5)$$

де В – вартість 1кВт-години електроенергії (В=1,7 грн/кВт);

П – установлена потужність комп'ютера (П=0,6кВт);

Ф – фактична кількість годин роботи комп'ютера (Ф=150 год.);

$К_п$ – коефіцієнт використання потужності ($К_п< 1$, $К_п = 0,9$).

$$В_е= 1,7 ·0,6 ·150· 0,9 = 137,7 \text{ (грн.)}$$

Розрахуємо інші витрати $В_{ін}$.

Інші витрати $І_в$ можна прийняти як (100…300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$B_{ін} = (1..3) \cdot (З_о + З_р). \quad (4.6)$$

Отже, розрахуємо інші витрати:

$$B_{ін} = 1 * (8522,65 + 852,26) = 9374,91 \text{ (грн)}.$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$B = З_о + З_д + Н_{зп} + А + К + В_е + І_в$$

В = 8522,65+852,25+3403,09+625+176+137,7+9374,91= 23091,61 (грн.)

Розрахуємо загальну вартість наукової роботи $B_{заг}$ за формулою:

$$B_{заг} = \frac{B_{ін}}{\alpha} \quad (4.7)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$B_{заг} = \frac{23091,61}{1} = 23091,61$$

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{B_{заг}}{\beta} \quad (4.8)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{23091,61}{0,9} = 25657,34 \text{ (грн.)}$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Спрогнозуємо отримання прибутку від реалізації результатів нашої розробки. Зростання чистого прибутку можна оцінити у теперішній вартості грошей. Це забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності .

Оцінка зростання чистого прибутку підприємства від впровадження результатів наукової розробки. У цьому випадку збільшення чистого прибутку підприємства $\Delta П_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta П_i = \sum_1^n (\Delta П_я \cdot N + П_я \Delta N)_i \qquad (4.9)$$

де $\Delta П_я$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$П_я$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

В результаті впровадження результатів наукової розробки витрати на виготовлення інформаційної технологіїзменшатьсяна 25 грн (що автоматично спричинить збільшення чистого прибутку підприємства на 25 грн), а кількість користувачів, які будуть користуватись збільшиться: протягом першого року – на 200 користувачів, протягом другого року – на 175 користувачів, протягом третього року – 150 користувачів. Реалізація інформаційної технологіїдо впровадження результатів наукової розробки складала 1000 користувачів, а прибуток, що отримував розробник до впровадження результатів наукової розробки – 200 грн.

Спрогнозуємо збільшення чистого прибутку від впровадження результатів наукової розробки у кожному році відносно базового.

Отже, збільшення чистого продукту $\Delta П_1$ протягом першого року складатиме:

$$\Delta П_1 = 25 \cdot 1000 + (200 + 25) \cdot 200 = 70000 \; грн.$$

Протягом другого року:

$$\Delta П_2 = 25 \cdot 1000 + (200 + 25) \cdot (200 + 175) = 109375 \; грн.$$

Протягом третього року:

$$\Delta П_3 = 25 \cdot 1000 + (200 + 25) \cdot (200 + 175 + 150) = 143125 \; грн.$$

4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Абсолютна ефективність $E_{абс}$ вкладених інвестицій розраховується за формулою:

$$E_{абс} = (ПП - PV), \qquad (4.10)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

т – період часу, протягом якого виявляються результати впровадженої НДДКР, 3 роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки 2, 3,4.

Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, рисунок 4.1.

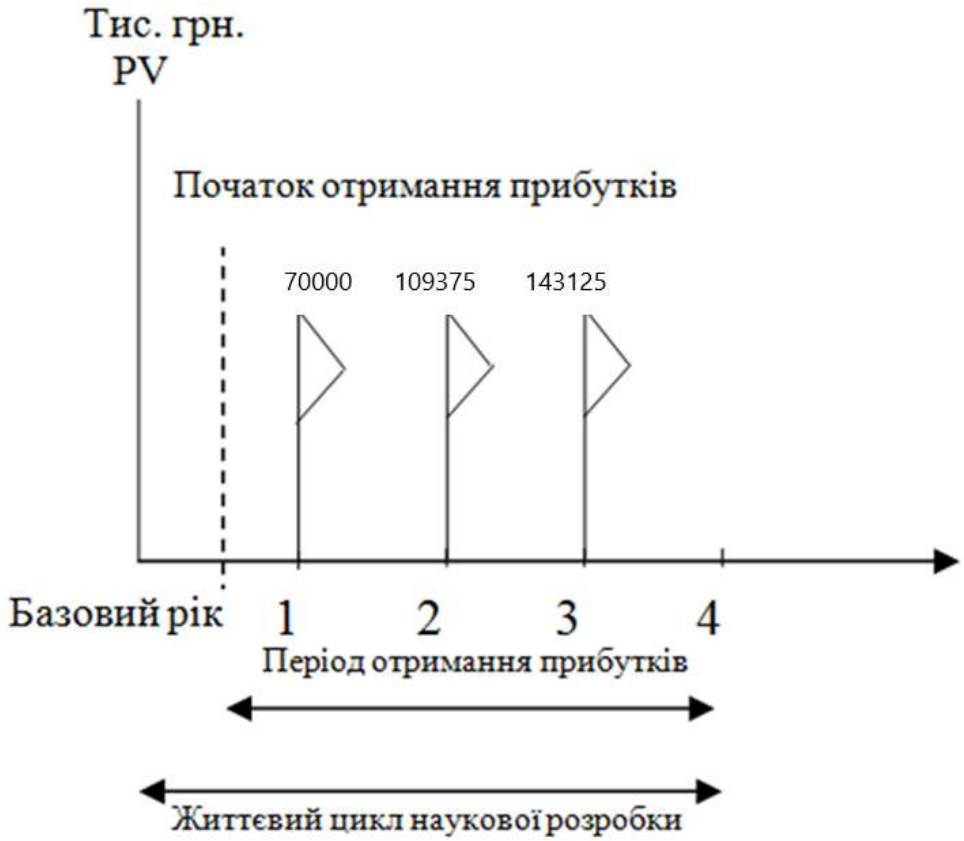


Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

Розрахуємо вартість чистих прибутків за формулою:

$$ПП = \sum_1^m \frac{\Delta П_i}{(1+\tau)^t} \quad (4.11)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

т – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки.

Отже, розрахуємо вартість чистого прибутку:

$$ПП = \frac{25657,34}{(1+0,1)^0} + \frac{70000}{(1+0,1)^2} + \frac{109375}{(1+0,1)^3} + \frac{143125}{(1+0,1)^4} = 263439,92 \text{ (грн.)}$$

Тоді розрахуємо $E_{абс}$:

$$\text{Е}_{\text{абс}} = 263439,92 - 25657,34 = 237782,58 \text{ грн.}$$

Оскільки $E_{абс} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $\text{Е}_\text{в}$ за формулою:

$$\text{Е}_\text{в} = \sqrt[\text{Т}]{1 + \frac{\text{Е}_{\text{абс}}}{\text{PV}}} - 1 \qquad (4.12)$$

де $\text{Е}_{\text{абс}}$ – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій PV = ЗВ, грн;

$\text{Т}_\text{ж}$ – життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$\text{Е}_\text{в} = \sqrt[3]{1 + \frac{237782,58}{25657,34}} - 1 = 1,17 \text{ або } 117\ \%$$

Далі, розраховану величина $\text{Е}_\text{в}$ порівнюємо з мінімальною (бар'єрною) ставкою дисконтування $\tau_{\text{мін}}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

$$\tau = d + f,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні d = 0,2;

f – показник, що характеризує ризикованість вкладень, величина f = 0,1.

$$\tau = 0{,}2 + 0{,}1 = 0{,}3$$

Оскільки $E_в = 117\% > \tau_{мін} = 0{,}3 = 30\%$, то у інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок}$розраховується за формулою:

$$T_{ок} = \frac{1}{E_в}$$

$$T_{ок} = \frac{1}{1{,}17} = 0{,}85 \text{ року}$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним.

4.5 Висновки

Під час виконання економічної частини магістерської кваліфікаційної роботи на основі обчислень було доведено, що розробка інформаційної технології

управління криптовалютними активами є доцільною та економічно обґрунтованою.

Проведено аудит із залученням експертів у даній сфері розробки, який показав, що рівень комерційного потенціалу розробки є високим.

У даному розділі обчислено витрати на заробітну плату, на амортизаційні відрахування, витрати на комплектуючі, на силову електроенергію. Загальна вартість яких склала 25 657.37 грн.

Також спрогнозовано деякі комерційні ефекти від реалізації результатів розробки. Прогнозується, що протягом 3-х років буде отримано прибуток.

Впровадивши розробку, приведена вартість всіх чистих прибутків підприємства за три роки становитиме 263 439.92 грн.

Також здійснено розрахунок ефективності інвестицій та періоду їх окупності. Оскільки життєвий цикл розробленого продукту становить 0.85 року.

Тому фінансування розробки є рентабельним для фінансування і може принести прибутки.

ВИСНОВКИ

В ході виконання магістерської кваліфікаційної роботи розроблено інформаційну технологію управління криптовалютними активами. Під час проведення аналізу предметної області, було визначено, що на сьогоднішній день

майже все якимось чином пов’язано з фінансовими операціями, а отже потрібно буде впевненими у їх надійності. Було визначено, що криптовалюти можуть надати як високий рівень захисту так і швидкість операцій. Проте багато сучасних рішень так само намагаються більше, або повністю, контролювати кошти користувачів. Було визначено, що обраний метод управління криптовалютними активами дозволяє користувачу не втрачати контроль своїх коштів, але з’являється складність роботи з некастодіальним підходом.

У другому розділі проаналізовано та отримано чітке розуміння про технологої, які будуть використані в плануванні, реалізації та тестуванні програмого продукту. Було ретельно проаналізовано алгоритми консенсусу, які застосовуються для підтвердження і захисту блокчейн мереж. Детально проаналізовано, як працюють різні алгоритми хешування у різних блокчейн мережах, визначено їх недоліки і переваги. Було спроектовано майбутню структуру проекту, створено діаграму класів, розроблено алгоритми для купівлі, продажу і обміну криптовалют. Проаналізовано ієрархічну генерацію ключів, що буде використовуватись у додатку. Проаналізовано і порівняно шляхи, по яким можна буде генерувати ключі і те, чим вони відрізняються, які переваги несе кожен із них.

У третьому розділі було проведено обґрунтування мови програмування і програмних засобів для розробки додатку. В результаті проведеного аналізу було обрано мову JS і React Native, як пару, що може сильно скоротити час розробки мобільного додатку, оскільки розробка проводиться одночасно для операційних систем Android і iOS, що являється величезним плюсом. Було проведено тестування розробленого додатку і порівняння його по різним характеристикам з іншими. Якщо порівнювати з кастодіальними рішеннями, то розроблена інформаційна технологія не потребує контролювати ключі, а отже, і фінанси користувача, що захищає його від їх втрати при зломі біржі, від заморозки його рахунків та інших проблем кастодіальних підходів, таких як неможливість перенести свої адреси на інший гаманець. В порівняні з іншими некастодіальними рішеннями можна виділити наступні переваги:

- Швидкість купівлі суттєво зросла, а саме в 2.4 рази;
- Швидкість продажу зросла в 3.8 рази.

- Обмін був успішно проведений, а його швидкість рівна швидкості знаходження блоку у одній з двох обраних криптовалют (Наприклад, у мережі Bitcoin цей час, в середньому, становить до 10 хвилин).
- Для операцій не потрібно проходити верифікацію.

Загалом, між усіма рішеннями, де були доступні операції купівлі, продажу та обміну:

- Комісія за купівлю і продаж була краща, від найближчого кращого варіанту, на 2.3%;
- Комісія за обмін на 0.25%;
- Курс купівлі одного Ethereum за гривню був кращий на 2.49%;
- Курс продажу одного Ethereum за гривню був кращий на 2.38%;
- Курс обміну Ethereum на 1 Bitcoin був кращий на 0.77%.

У четвертому розділі на основі обчислень було доведено, що розробка інформаційної технології управління криптовалютними активами є доцільною та економічно обґрунтованою. Проведено аудит із залученням експертів у даній сфері розробки, який показав, що рівень комерційного потенціалу розробки є високим. У даному розділі обчислено витрати на заробітну плату, на амортизаційні відрахування, витрати на комплектуючі, на силову електроенергію. Загальна вартість яких склала 25 657.37 грн. Прогнозується, що протягом 3-х років буде отримано прибуток. Впровадивши розробку, приведена вартість всіх чистих прибутків підприємства за три роки становитиме 263 439.92 грн. Також здійснено розрахунок ефективності інвестицій та періоду їх окупності. Оскільки життєвий цикл розробленого продукту становить 0.85 року. Тому фінансування розробки є рентабельним для фінансування і може принести прибутки.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Марков Д.Е., Сілагін О.В., Особливості розробки програмних продуктів на базі технології blockchain // Міжнародна наукова-практична конференція «Наукові відкриття та фундаментальні наукові дослідження: світовий досвід 2019» зб. наук. праць «ΛΌГОΣ» з матеріалами міжнар. наук.-практ. конф., м. Полтава, 20 травня, 2019 р. Полтава : ГО «Європейська наукова платформа», 2019. Т.5. с. 114.
2. Марков Д.Е., Сілагін О.В., Особлівості програмних додатків на базі технології blockchain // «ІНТЕРНЕТ-ОСВІТА-НАУКА-2019», Одинадцята міжнародна науково-практична конференція ІОН-2019, 22-25 травня, 2019 : Збірник праць. – Вінниця : ВНТУ, 2019 –343 с.
3. Марков Д.Е., Сілагін О.В., Аналіз інформаційної технології зберігання і управління криптовалютними активами // «Молодь в науці: дослідження, проблеми, перспективи», Вінниця, 2019. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/index.php/mn/mn2020/paper/view/8394. Дата звернення: Листопад 2019.
4. Ashton K. That Internet of Things / K. Ashton // Thing. RFID Journal, 22 July 2009. [Electronic resource]. – Mode of access http://www.rfidjournal.com/articles/view?4986.
5. Gartner Says 6.4 Billion Connected "Things" Will Be in Use in 2016, Up 30 Percent From 2015. [Electronic resource]. – Mode of access http://www.gartner.com/newsroom/id/3165317.
6. Shancang Li. The internet of things: a survey / Li Shancang, Li Da Xu, and Shanshan Zhao // Information Systems Frontiers 2015, 17.2. – Pp. 243-259.
7. Whitmore Andrew. The Internet of Things – A survey of topics and trends / Whitmore Andrew, Anurag Agarwal, and Li Da Xu // Information Systems Frontiers 17.2, 2015. – Pp. 261-274.
8. Dorri, Ali. Kanhere, and Raja Jurdak / Ali Dorri, S. Salil // Blockchain in internet of things: Challenges and Solutions" arXiv preprint arXiv:1608.05187, 2016.

9. Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. [Electronic resource]. – Mode of access https://bitcoin.org/bitcoin.pdf.

10. Christidis Konstantinos, Michael Devetsikiotis. Blockchains and Smart Contracts for the Internet of Things. [Electronic resource]. – Mode of access http://ieeexplore.ieee.org/iel7/6287639/6514899/07467408.pdf?arnumber=7467408.

11. Brody, Paul. Device democracy: Saving the future of the Internet of Things / Paul Brody, Pureswaran Veena // IBM, September, 2014.

12. Veena P. Empowering the Edge-Practical Insights on a Decentralized Internet of Things. Empowering the Edge-Practical Insights on a Decentralized Internet of Things / P. Veena, S. Panikkar, S. Nair, P. Brody // IBM Institute for Business Value, 17 Apr. 2015. [Electronic resource]. – Mode of access http://www-01.ibm.com/common/ssi/cgi-bin/ssialias? infotype=PM&subtype=XB&htmlfid= GBE03662USEN#loaded.

13. Boohyung Lee. Blockchain-based secure firmware update for embedded devices in an Internet of Things environment / Lee Boohyung, Lee Jong-Hyouk. The Journal of Supercomputing, 2016. – Pp.

1-16.

14. Ferrer E.C. The blockchain: a new framework for robotic swarm systems. arXiv preprint arXiv:1608.00695, 2016.

15. Девід Фленаган. JavaScript. Подробное руководство. 6 издание, 2017 – 1080 ст.

16. Пабло Ділєман. Изучаем Angular 2, 2017 – 354 ст.

17. Стоян Стефанов. React JS. Быстрый старт, 2017 – 224 ст.

18. Bahga Arshdeep. Blockchain Platform for Industrial Internet of Things / Bahga Arshdeep, Vijay K. Madisetti // Journal of Software Engineering and Applications. – 2016. – № 9. – Pp. 533-546.

19. Мельник А.О. Кіберфізичні системи: проблеми створення та напрями розвитку // Вісник Національного університету "Львівська політехніка". – Сер.: Комп'ютерні системи та мережі. – Львів : Вид-во НУ "Львівська політехніка". – 2014. – № 806. – С. 154-161.

20. Andreas M. Antonopoulos. Mastering Bitcoin: unlocking digital cryptocurrencies. "O'Reilly Media, Inc.", 2014. – 298 p.