**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики

Кафедра інтелектуальних програмних систем

# Кваліфікаційна робота

**на здобуття ступеня бакалавра**

за спеціальністю 121 Програмна інженерія

на тему:

# ЗАСОБИ РОЗРОБКИ КРИПТОВАЛЮТНИХ ГАМАНЦІВ З ПІДТРИМКОЮ БАГАТЬОХ БЛОКЧЕЙНІВ

Виконав студент 4-го курсу

Юрій МОМОТЕНКО

(підпис)

Науковий керівник:

асистент, кандидат фіз.-мат. наук

Костянтин ЖЕРЕБ _____

(підпис)-

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент 

(підпис)

Роботу розглянуто й допущено до захисту на засіданні кафедри інтелектуальних програмних систем

«25» травня 2022 р., протокол № 10

Завідувач кафедри

Олександр ПРОВОТАР _____

(підпис)

Київ - 2022

# РЕФЕРАТ

Обсяг роботи 29 сторінок, 2 ілюстрації, 12 використаних джерел, 1 додаток

Ключові слова: БЛОКЧЕЙН, ЗАСІБ РОЗРОБКИ, КРИПТОВАЛЮТА, КРИПТОГАМАНЕЦЬ, МНЕМОНІЧНА ФРАЗА, ПРИВАТНИЙ КЛЮЧ, ПУБЛІЧНИЙ КЛЮЧ, ШИФРУВАННЯ, BITCOIN, ETHEREUM, WEB3

Об’єктом роботи є побудова криптовалютних гаманців з підтримкою багатьох блокчейнів за допомогою різних засобів розробки, в тому числі, «UniBDK». Предметом роботи є бібліотека для створення криптовалютних гаманців на основі блокчейну та прототип програмного засобу криптовалютного гаманця з модулем індексування блокчейну для розширеного функціоналу.

Метою кваліфікаційної роботи є створення власного засобу розробки для підтримки багатьох блокчейнів, створення прототипу криптовалютного гаманця з модулем індексування блокчейну.

Середовище розробки: Manjaro 21.2, інструмент для створення Jetbrains GoLand 2022.1, інструменти віртуалізації на рівні операційної системи Docker, Docker Compose, мова програмування Go.

Результати роботи: проведено аналіз існуючих засобів розробки, виділено основну модель криптовалютних гаманців, розроблено засіб розробки з підтримкою багатьох блокчейнів «UniBDK», створено прототип криптовалютного гаманця, з модулем індексування блокчейну.

Бібліотека «UniBDK» може застосовуватися розробниками для створення модуля управління ключами криптогаманця з підтримкою багатьох блокчейнів. Прототип криптовалютного гаманця може використовуватися як

приклад застосування бібліотеки «UniBDK», або ж бути розширеним до самостійного продукту.

**ЗМІСТ**

**ВСТУП** ..... **6**

**РОЗДІЛ 1. Блокчейн та смарт-контракти** ..... **8**

**1.1 Блокчейн Bitcoin** ..... **8**

**1.2 Блокчейн Ethereum** ..... **9**

**РОЗДІЛ 2. Проблема створення криптогаманців** ..... **11**

**2.1 Етапи розвитку блокчейну** ..... **11**

**2.2 Розвиток криптогаманців** ..... **13**

**2.3 Інструменти розробки криптогаманців** ..... **15**

**РОЗДІЛ 3. Структура криптовалютних гаманців та їх функціонування** **16**

**3.1 Поширені складові криптовалютних гаманців** ..... **16**

**3.2 Модуль управління ключами** ..... **16**

**3.3 Модуль індексування токенів** ..... **19**

**РОЗДІЛ 4. Створення засобу розробки та прототип бекенду для криптовалютних гаманців з підтримкою блокчейну** ..... **21**

**4.1 Створення засобу розробки** ..... **21**

**4.2 Створення бекенду для прототипу гаманця** ..... **23**

**4.3 Створення інтерфейсу командного рядка для взаємодії з прототипом гаманця** ..... **25**

**4.4 Вдосконалення прототипу криптовалютного гаманця з підтримкою багатьох блокчейнів** ..... **26**

**ВИСНОВКИ** ..... **27**

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ** ..... **28**

**ДОДАТОК А** ..... **30**

# ВСТУП

**Оцінка сучасного об’єкта розробки.** Станом на зараз, світ криптовалюти та блокчейну притягує до себе все більше та більше людей. З кожним днем з’являється все більше і більше криптовалют та додатків на основі блокчейну. Для розробки різноманітних програмних продуктів з використанням нових технології, гостро виникає потреба в бібліотеках та різних засобах розробки. Зокрема, ріст кількості криптовалют, змушує розробників криптовалютних гаманців вручну додавати підтримку того чи іншого блокчейну.

Криптовалютні гаманці є окремим сегментом в світі блокчейну та є ключовим зв’язком між новітньою технологією та кінцевим користувачем.

**Актуальність роботи та підстави її виконання.** При розробці критовалютних гаманців використання різних засобів розробки для виконання однієї і тієї ж фукнції є незручним. Розширення такого продукту для підтримки більшої кількості блокчейнів вимагатиме ручного підлаштування під засіб розробки бажаної мережі. Отже необхідно побудувати уніфікований засіб розробки для модуля управління ключами криптовалютного гаманця та наочно показати приклад його застосування.

**Мета і завдання роботи.** Метою кваліфікаційної роботи є створення засобу розробки та прототипу криптовалютного гаманця, з модулем індексування блокчейну. Для досягнення цієї мети поставлено такі завдання.

- Дослідити криптовалютний гаманець як програмний продукт.
- Дослідити існуючі засоби розробки для управління ключами на блокчейні.

- Розробити засіб розробки для управління ключами з підтримкою блокчейну Bitcoin та Ethereum.
- Розробити модуль індексування блокчейну Bitcoin та Ethereum.
- Розробити прототип криптовалютного гаманця за допомогою перелічених вище програмних засобів.

**Об’єкти, методи й засоби дослідження або розроблення.** Об’єктом створення засобу розробки «UniBDK» є побудова криптовалютних гаманців з підтримкою багатьох блокчейнів.

**Можливі сфери застосування.** Програмний продукт можна застосовувати при створенні криптовалютних гаманців з підтримкою багатьох блокчейнів.

# РОЗДІЛ 1. Блокчейн та смарт-контракти

## 1.1 Блокчейн Bitcoin

Ідею блокчейну розроблювали ще у 1991 році, що підтверджується працею [1]. Тоді вона поставала як криптографічно захищений ланцюг блоків для збереження даних. Пізніше Сатоші Накамото розвинув ідею та вперше реалізував її, створивши першу криптовалюту – Bitcoin. У статті [2] наведено перший документ опублікований творцем даної технології.

Блокчейн – це база даних, яка має особливу структуру. Записи до неї здійснюються за допомогою блоків, кожен з яких залежить від попереднього. Це відбувається шляхом хешування. Інформація попереднього блоку хешується та подається на вхід при створенні хешу наступного блоку.

В мережі існують валідатори та майнери. Майнери, в залежності від алгоритму консенсусу працюють над знаходженням необхідних чисел, необхідних для створення нового блоку. Валідатори, в свою чергу, перевіряють правильність інформації та зберігають всю історію блокчейну. Отримати певну роль у публічному блокчейні може кожен бажаючий.

В мережі Bitcoin працює алгоритм згоди Proof of Work – підтвердження роботи. За ним, майнер необхідний перебрати велику кількість чисел задля знаходження nonce – числа, яке є обов’язковим параметром хеш функції, використаної при створенні хешу блоку. Це відбувається за рахунок того, що виставлено обмеження на те, якого вигляду повинен мати хеш блоку, наприклад, починатися з чотирьох нулів. З 2009 року (старту Bitcoin) пройшло досить багато часу. Сатоші передбачив, що обчислювальні можливості будуть зростати з роками, тому додав коректування складності до свого проекту. Таким чином, якщо генерування нового блоку відбувається занадто швидко

(оптимальним часом вважаються десять хвилин), то відбувається корекція складності, зокрема шляхом редагування патерну, якому має слідувати хеш блоку.

Оскільки майнери працюють в різнобій, більш ніж вірогідно, що вони будуть включати в блоки різні транзакції, проте все одно знайдуть необхідний патерн. Виникне ситуація, коли в мережі буде два, три, або більше варіантів одного й того самого блоку. Ця проблема також була передбачена шляхом побудови найдовшого ланцюга. З часом до майнера через валідатора дойде найдовший ланцюг, він виявить що блок який він хотів побудувати уже змайнений, тому за протоколом, він буде змушений переключитися на нього, покинувши свій ланцюг.

Валідатори при отриманні блокчейну, займаються перевіркою всіх блоків на відповідність один одному. Таким чином, якщо хтось один захоче підмінити якийсь блок, то мережа просто не прийме його зміни, в першу чергу через можливу не відповідність хешам. Проте, існує така атака на блокчейн, яка називається «Атака 51%». Загроза даної атаки виникає, коли обчислювальні потужності зосереджуються у руках однієї особи у кількості більше половини хоча б на один відсоток. В такому випадку дана особа, або ж організація, матиме змогу контролювати блокчейн, блокувати транзакції. Тому важливим є запобігання можливості даної атаки. З ростом блокчейну на весь світ, шанси такої атаки зменшуються, проте все одно залишаються, оскільки майнери збираються в пули, задля колективного збільшення своїх обчислювальних можливостей. [3]

### 1.2 Блокчейн Ethereum

В 2013 році Віталій Бутерін – російський програміст, представив нову розробку – блокчейн з можливістю створення децентралізованих додатків на його основі. Це стало можливим завдяки введенню нового поняття – смарт-контракту.

Смарт-контракт – це своєрідна угода на блокчейні, яка задається набором інструкцій. Будь-хто бажаючий має змогу взаємодіяти з контрактом та його методами, з кодом, який виконається на блокчейні.

Задля написання коду для блокчейну, було розроблено різні бібліотеки, мови програмування та компілятори, для переведення інструкцій на мові програмування високого рівня в байт-код, зрозумілий блокчейну. Так було створено мову програмування Solidity. Вона повна за Тюрінгом та дає змогу створити абсолютно будь-який продукт. Для компіляції є низка варіантів, найпопулярніший з них, компілятор solc, який поширюється під різні платформи.

Альтернативою блокчейну Ethereum виступає його форк (проект побудований на вихідному коді) Binance Smart Chain. Це блокчейн, створений китайською криптовалютною біржею-гігантом Binance. Він завоював популярність серед користувачів дешевизною транзакцій. В той час, коли до оновлення Berlin, яке знизило вартість транзакцій, комісія за переказ коштів чи взаємодію з блокчейном Ethereum могла сягати сімдесяти доларів. В той же час вартість транзакції на альтернативному блокчейні Binance складає лише пятдесят центів в середньому. [4]

# РОЗДІЛ 2. Проблема створення криптогаманців

## 2.1 Етапи розвитку блокчейну

Технологія блокчейн швидко розвивалася з часом. Можна виділити декілька етапів розвитку, кожен з яких вирізнявся ростом популярності. Перший етап є найтривалішим, під час нього відбувалося тестування застосування криптографії для створення криптовалют. Найвдалішим експериментом того часу є вищезгадана криптовалюта Bitcoin. Саме цей проект показав досить вдале застосування блокчейну для отримання децентралізованої системи економіки.

Другим етапам, хотілося б виділити вдалий експеримент Віталія Бутеріна, який створив віртуальну машину для виконання смарт-контрактів. Саме це дало поштовх новому розвитку блокчейну, як технології для безпечних систем. Дана технологія надала змогу запускати верифіковані мікропрограми для менеджменту коштів інвесторів. Зокрема, було створено інтерфейс ERC-20 [5]. Саме дана технологія задала шаблон для створення взаємнозамінних токенів, певний стандарт, на основі якого можна було створювати власні криптомонети, надаючи їм певних властивостей в залежності від потреб. У цей час набуло поширення таке поняття як dapp (decentralized application – децентралізований застосунок). Це породило створення ряду проектів, які програмували логіку за допомогою смарт-контрактів. В свою чергу з’явилася проблема верифікації контрактів: як користувачі того чи іншого смарт-контракту могли бути впевнені в тому що наданий проектом код дійсно є відображенням того коду, який існує на блокчейні. Тут з’явилася необхідність у оглядачах блокчейну – вебсайтах, які могли відображати транзакції користувачів. Необхідною функцією для користувачів які хотіли впевнитися в безпечності проекту стала верифікація

контрактів. Дана особливість блокчейну надавала можливість переглянути точну копію коду який було додано до мережі. Водночас з'явилася нова професія – аудитор смарт-контрактів. Яка набула більшої популярності пізніше, під час наступних етапів.

В даний час розвиток пішов декількома напрямками. Виділимо епоху розвитку обмінників як третій етап. Почався розвиток централізованих бірж (CEX), а згодом і децентралізованих (DEX).

CEX виступають своєрідними банками у світі криптовалют. Зокрема більшість з таких організації використовують технологію KYC (know your customer – знай свого користувача), за допомогою якої можна однозначно ідентифікувати користувача. Це своєрідна спроба легалізації фінансів. Хоча більшість бірж повідомляють, що не співпрацюють з державними установами (по типу податкових служб, служб безпеки тієї чи іншої країни), проте яскравим прикладом у допомозі з легалізуванням криптодоходу є біржа Binance, яка створила функціонал для зручного підрахунку податку (зокрема для користувачів з США) [6]. Також вони офіційно оголосили боротьбу з "відмиванням грошей" зобов'язавши всіх користувачів, які бажають використовувати їх сервіси пройти обов'язкову верифікацію акаунту [7].

Альтернативою централізованих обмінників виступила децентралізація. Так були створені смарт-контракти, які виступають менеджерами коштів. Своєю відкритістю до аудитів та кодом у вільному доступі вони надають своєрідні гарантії користувачам. В своїй більшості вони не вимагають KYC, користувачем обмінника є будь-який публічний ключ на блокчейні – адрес. За ним закріпляються кошти. "Not your keys, not your wallet" декламують користувачі обмінники один між одним, для запобігання випадків викрадення коштів через підробні сайти, що, на жаль, не є дуже рідкісним явищем.

Наступним четвертим етапом є DAO – decentralized autonomous organization. Альтернативою сучасної економіки можна привести як приклад

акціонерне товариство. В даній організації немає центрального керівництва, важливі рішення в існуванні спільности відіграють інвестори. Зазвичай, такі рішення приймають за допомогою голосування, де користувачі-валідатори пропорційно до вкладених коштів мають певну вагу голосу. Прикладами таких організацій є Compound, BitDAO, тощо.

Останнім, поточним штабелем розвитку технології блокчейн є зростання попиту на невзаємозамінні токени - NFT. Дана технологія була вперше представлена ще у 2015 році, разом з розвитком блокчейну Ethereum, проте найактивніший ріст попиту на проекти з NFT став саме 2021 рік. Саме з того часу технологія невзаємозамінних токенів розвинулася, ставши не лише можливістю продажі електронних картин, проте і створенням різноманітних ігор, площадок для продажу NFT. Даний вид токенів повпливав також і на різноманітні блокчейни, та породило створенням нових інтерфейсів. Так, зокрема, хотілося б виділити блокчейн Solana, який створювався як проста альтернатива Ethereum зі своїми модифікаціями: збільшенням пропускної можливості блокчейну (кількості транзакцій за секунду), дешевшою ціною на газ та переходом повністю на proof of stake алгоритмом консенсусу. Проте ріст попиту на NFT, призвів до побудови проектів на Solana в тому числі. Дешевизна транзакцій та простота створення (в Solana NFT це звичайний токен з максимальною кількістю 1) заохотило багато компаній та стартапів на створення та розширення свого продукту саме там.  Хоча блокчейн був створений у 2017 році, та першим NFT проектом вважають Monkey Business, створених всередині 2021 році [8].

### 2.2 Розвиток криптогаманців

У попередньому розділі було згадано досить багато різних застосувань технології блокчейн, єдине що не було згадано, це те що існувало завжди –

криптогаманці. На них попит існував ще з часу створення Bitcoin. Найпростішим гаманцем був Bitcoin 0.1.0 створений Сатоші Накамото [9]. Дана програма підтримувала лише операційну систему Windows 2000, XP, Vista. Вона являла собою валідатор блоків, та допомагала блокчейну існувати, під'єднуючись до інших, таких само клієнтів, та поширюючи інформації про поточний стан блокчейну по мережі.

З часом зростали вимоги до криптогаманців, зокрема новостворені блокчейни, відходили від подібних до Bitcoin клієнтів, вони потребували власного менеджменту ключів, функціонал для створення декількох адресів (дана можливість з'явилася в Bitcoin пізніше), перегляду транзацій, з часом – перегляду NFT. Гостро зростала проблема у підтримці багатьох блокчейнів. Оскільки незручно було тримати по декілька програм для різних блокчейнів. Якщо у випадку з похідними від Ethereum блокчейнами (EVM-compatible) інтерфейс отримання інформації через API або веб-сокети не відрізнявся, у випадку з Bitcoin, Solana, Dogecoin, Litecoin необхідно мати чотири різних гаманці (у випадку відмови користуватися CEX). На ринку з'являлися рішення цієї проблеми. Так, гаманець Metamask – найпопулярніший серед EVM-compatible блокчейнів, Phantom надає чудову підтримку NFT з блокчейну Solana. Особливої уваги заслуговує мобільний додаток TrustWallet, який зробив прорив одними з перших створивши гаманець для криптовалют з непоєднаних блокчейнів починаючи Bitcoin, закінчуючи Near. Зокрема, нещодавно було додано навіть підтримку NFT, проте з EVM-compatible мереж.

Централізовані біржі також не залишили своїх користувачів без зручності управління фінансами зі свого телефону. Можна знайти різноманітні застосунки кожної великої фінансової організації, як от Binane, FTX, OKX, MEXC, тощо. Особливу увагу вони приділяють саме криптовалютам, гнатися за трендами NFT не кожна біржа може, тому на даний момент не досить часто можливо зустріти невзаємнозамінні токени на централізованих обмінниках.

## 2.3 Інструменти розробки криптогаманців

Кожен блокчейн намагається надати якомога більше інструментів для розробки на їх основі, щоб заохотити розробників з досвідом на різних мовах програмування. Найпоширенішою мовою Web3 Інтернету прийнято вважати JavaScript. Зокрема її застосовують при розробці та тестуванні смарт-контрактів, побудові скриптів для запуску смарт-контрактів на блокчейні, побудові бекенду різного призначення. Не меншу популярність для створення бекенду має мова Go. Її широко застосовують в криптографії, для створення сервісів азартних ігор.

Найпопулярніші блокчейни зазвичай підтримують 4-5 основних мов програмування. Навіть, якщо власники мережі не бажають імпортувати засоби розробки (SDK) на інші мови програмування, то найчастіше є потужні ком’юніті, які власноруч виконують такий імпорт. Зокрема, нерідко можна зустріти дуже популярні криптографічні, або блокчейн бібліотеки, розташовані на аккаунтах звичайних розробників.

# РОЗДІЛ 3. Структура криптовалютних гаманців та їх функціонування

## 3.1 Поширені складові криптовалютних гаманців

Криптовалютний гаманець є досить складним програмним рішенням, яке складається з різних модулів. Проаналізувавши декілька популярних додатків, таких як Metamask та TrustWallet, можна виділити декілька основних функцій які вони надають. Зокрема до таких функцій можна віднести наступне:

- Модуль управління ключами (основний)
- Індексація токенів (основний)
- Модуль управління транзакціями
- Зв’язок з обмінниками
- Модуль відображення вартості активів

Основними модулями можна назвати модулі управління ключами та індексації токенів. Сюди також можна було б віднести управління транзакціями. Два модуля, які залишаються, необхідні для того, щоб в користувача було уявлення про те, яку реальну вартість має його криптовалюта, оскільки ринок дуже волатильний.

## 3.2 Модуль управління ключами

Оскільки основна суть криптовалютного гаманця полягає в управлінні ключами, тому їх основною функцією є саме менеджмент приватних ключів. Найпопулярнішим дизайном створення приватного ключа є BIP39. Суть даної специфікації в наданні зручного для користувача набору наперед заданих із

2048 можливих слів англійської мови (є розширення для різних мов, зокрема китайської). Для генерації фрази, відбувається створення ентропії – числа в бітовому представлені, яке створюється не псевдорандомним чином за допомогою комп'ютера (шляхом збору різної інформації про стан системи, наприклад). Дане число, має мати довжину від 128 до 256 бітів, число 32 має бути дільником довжини бітового представлення даного числа. Наступним кроком до нього додають по одному біту хешу ентропії на кожен тридцять другий біт. Для цього хешування виконується за допомогою криптографічної функції SHA-256. Потім відбувається поділ отриманих бітів на 11. Кожна частинка є індексом одного із 2048 слів в списку прийнятих інтерфейсом BIP39 слів (максимальний індекс списку 2047 є одинадцяти бітовим числом). [10]

Саме таким чином генерується мнемонічна фраза. Даний метод представлення сіду був обраний, для можливості користувачів зберегти дану фразу для можливості відновлення доступу до свого гаманця. Даний метод доведено є криптографічно стійким. Не важко переконатися в кількості різних комбінацій даних слів в залежності від довжини фрази. Використовуючи методи комбінаторики, можна побудувати наступне розміщення та прикинути їх кількість.

$$\overline{A_n^k} = n^k$$

$$\overline{A_{2048}^{12}} = 2048^{12} = \left(2^{11}\right)^{12} = 2^{132} \approx 5.4 * 10^{39}$$

$$\overline{A_{2048}^{24}} = 2048^{24} = \left(2^{11}\right)^{24} = 2^{264} \approx 2.9 * 10^{79}$$

Звернемось до рейтингу TOP500 – проекту, який оцінює обчислювальну потужність 500 відомих найпотужніших суперкомп'ютерів. Станом на 2022 рік, даний список очолює проект Frontier з піковою швидкодією 1.5 екзафлопсів, або ж $1.5 * 10^{18}$ операцій в секунду. Задамо ідеальні умови для грубого перебору всіх можливих комбінацій мнемонічних фраз. Зокрема, генерація ключа буде виконуватися за один флопс (одну операцію в секунду)та

на перевірку даного ключа на коректність в 0 секунд, необхідно буде наступну кількість часу для перевірки всі мнемонічних фраз:

$$t = \frac{\textit{кількість можливих фраз}}{\text{кількість операцій в секунду}} = \frac{5.4 * 10^{39}}{1.5 * 10^{18}} \approx 3.6 * 10^{21}\text{с} = \frac{3.6 * 10^{21}}{60 * 60 * 24 * 365} = 1.14 * 10^{14}\text{років}$$

Отримана кількість років необхідна на перебір всіх фраз в $\frac{1.14*10^{14}}{13.7*10^{9}} \approx 8321$ разів перевищує час, який сплив до сьогодення від Великого вибуху.

Повертаючись до необхідних методів для ефективного управління ключами, він повинен мати наступні функції: генерація або імпорт існуючої мнемонічної фрази, підпис даних та валідація отриманого підпису. Будуючи власне рішення, стає можливим вибір довжини мнемонічної фрази. Зокрема, можливо делегувати даний вибір користувачеві, або ж обрати власноруч. Популярний “гарячий” гаманець Metamask генерує користувачам фразу з дванадцяти слів, на противагу “холодний” гаманець Ledger пропонує користувачам для генерації лише фразу з двадцяти чотирьох слів, проте також підтримує імпорт мнемоніки в дванадцять та вісімнадцять слів. Наступним кроком після отримання мнемонічної фрази тим чи іншим шляхом, є отримання приватного ключа. Це відбувається шлях конкатенації бітового представлення мнемонічної фрази (64 байти) та певної строки (наприклад для Bitcoin задають фразу “Bitcoin seed”). Для підвищення безпеки, також можливо задати власну строку, або доповнити загально прийняту з певним власним паролем. Наступ є хешування отриманої фрази, методом HMAC-SHA512 – запатентованою криптографічною функцією, яка повертає значення хешу довжиною 512 біт. Даний хеш і є приватним ключем. [11]

З приватного ключа відбувається генерація публічного ключа шляхом незворотного отримання шляхом передачі приватного ключа в функцію шифрування основану на еліптичних кривих. Еліптична крива над полем дійсних чисел це множина точок на площині, координати якої задовольняють рівнянню виду:

$$y^2 = x^3 + ax + b$$

Національний інститут стандартів та технологій США (NIST) рекомендує 15 еліптичних кривих для використання їх в цифрових підписах. Досить часто використовують псевдорандомні генератори для чисел $a$ та $b$. У випадку з Bitcoin, використовується не одна з рекомендованих кривих, а спеціально підібрані коефіцієнти для збільшення ефективності операцій. Так, було обрано $a = 0$, $b = 7$. Тому еліптична крива в Bitcoin задається рівнянням $y^2 = x^3 + 7$. Для цифрового підпису в даному блокчейнів використовують ECDSA, який є поширеним стандартом, та використовується в різних мережах. Публічний ключ є точками x та y на еліптичній кривій. Дані точки отримують наступним чином: public key = private key * G, де G – базова точка, яка є загальновідомою зі специфікації secp256k1. [12]

Також необхідно згадати інший стандарт BIP32. Дане запропоноване та прийняте покращення до гаманців, називається ієрархічно детерміновані гаманці (Hierarhical Deterministic Wallets), або HD-гаманці. Суть методу полягає в генерації з приватного ключа, отриманого із сід-фрази інших приватних ключів шляхом хешування. Таким чином можливо із одної мнемоніки отримати велику кількість приватних ключів.

### 3.3 Модуль індексування токенів

Створивши пару ключів на блокчейні, виникає проблема відображення всіх токенів певного адреса. Дана проблема виникає як наслідок двох причин: відсутність функціоналу на блокчейну (жодного API для отримання балансу за адресою), а бо ж наявністю різноманітних токенів на блокчені. Зокрема, якщо розглядати блокчейни з однією монетою, по типу Bitcoin, то в ньому відсутня саме можливість перегляду балансу довільного публічного ключа

(необхідний приватний ключ для даної операції). На противагу йому, в Ethereum, є необхідний функціонал для отримання балансу будь-якого адресу, проте присутня інша проблема: не має можливості без спеціальних засобів відстежити, якщо за якимось контрактом було отримано певну кількість токенів. Вирішенням даних проблем є якраз індексування. Ідея полягає в програмному опрацюванні всіх блоків певного блокчейну з отримання та зберіганням до бази даних певної інформації. У випадку з Bitcoin необхідно по всім транзакціям відстежувати як змінюється баланс всіх адресів. У випадку ж з Ethereum, можливо лише відстежувати з якими контрактами були пов’язані адреса, та запитувати в блокчейна напряму, скільки монет того чи іншого типу є на адресі. Зокрема такий функціонал реалізований в інтерфейсі токену мережі Ethereum – ERC20. Для зручності та швидкості такого програмного рішення зазвичай використовується нереляційна база даних типу “ключ-значення”.

# РОЗДІЛ 4. Створення засобу розробки та прототип бекенду для криптовалютних гаманців з підтримкою блокчейну

## 4.1 Створення засобу розробки

Написання багатьох застосунків досить часто є невід'ємно пов'язаним з використанням різноманітних бібліотек, які значно спрощують написання коду. При створенні криптовалютного гаманця розробники можуть зіткнутися з проблемою, що додаючи підтримку для різних блокчейнів, код додатку збільшується завдяки зайвому коду та реімплементації потрібних для гаманця методів.

Мова програмування Go отримала широку підтримку серед крипторозробників. Зокрема через швидкодію та мікросервісну архітектуру. Оберемо її для створення примітивного засобу розробки для підтримки багатьох блокчейнів. Дослідивши готові застосунки в мережі Інтернет, можна прийти до висновку що дана ніша є доволі вільною. Присутні рішення найчастіше є простою обгорткою над різним API, з неповним функціоналом. Найближчим аналогом, можна виділити бібліотеку "go-hdwlallet" користувача foxnut. Зокрема там реалізована підтримка інтерфейсу HD гаманців, згаданих у розділі 3.2. Аналізуючи різні рішення криптогаманців, було виділено основні аспекти, на які варто звернути увагу, та побудовано діаграму діяльності, зображену в додатку А. З неї бачимо, що засіб розробки має реалізовувати наступні функції: створення нового гаманця (генерація сід-фрази), імпорт сід-фрази. Метою користування криптогаманців зазвичай є переказ токенів. Тому додамо ще дві функції: підпис транзакції та валідація підпису.

В ході розробки, була використана рекомендована розробниками BIP39 бібліотека. Засіб розробки буде підтримувати сід фразу з 12 слів. Визначимо головний інтерфейс функцій нашого гаманця (Рисунок 1).

```
type Wallet interface {
    GetBlockchainName() Blockchain
    GetAddress() (string, error)
    GetKey() *Key
    SignData([]byte) ([]byte, error)
}
```

Рисунок 1. Інтерфейс класу гаманець

Він матиме наступні функції: отримати назву блокчейну – змінну-перелічення зі списком підтримуваних блокчейнів, отримати публічний адрес у вигляді строки, отримати посилання на клас мастер-ключа та функція підпису даних (яку використовуватиме менеджер транзакцій). В майбутньому, для того щоб додати підтримку інших блокчейнів, буде достатньо імплементувати інтерфейс Wallet та реалізувати можливість мастер ключа повернути уніфікований об’єкт класу типу Wallet, з реалізованими методами під інший блокчейн. Спроектовану систему класів можна зобразити наступним чином (Рисунок 2).

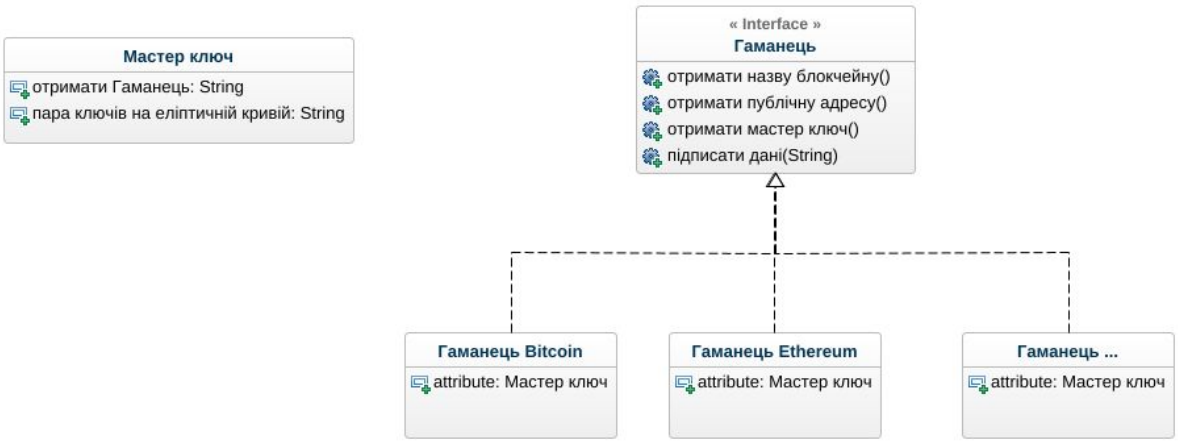


Рисунок 2. Система класів модуля управління ключами

### 4.2 Створення бекенду для прототипу гаманця

Як було зазначено в розділі 3.3, для повноцінного користування криптогаманцем необхідний модуль, який буде відстежувати події на блокчейні та записувати інформацію про нові токени, або ж баланс до бази даних. Для цього необхідно створити декілька мікросервісів, об'єднаних одним інтерфейсом для спілкування. Ціль даних сервісів – проіндексувати всі адреси на блокчейні та зберегти про них певну інформацію. Зокрема для Bitcoin, необхідним є підрахування балансу аккаунтів, для Ethereum індексування наявності токенів на певному адресі.

Опишемо логіку сервісу для підрахування балансу монет в мережі Bitcoin. Нам необхідно опрацювати всі блоки, починаючи з першого. Для кожного блоку, необхідно записати інформацію про кожну транзакцію: зокрема її суми її вихідних значень. Через особливості мережі, вихід однієї транзакції є входом іншої, тому при опрацюванні необхідно буде звертатися до попереднього значення транзакції. Наступним кроком буде власне додавання та віднімання від балансу різної кількість сатоші (найменшої одиниця криптовалюти Bitcoin) на певному адресі. Даний процес є досить довгим, що зумовлено віком блокчейну, та великою кількістю блоків в ньому. По завершенню синхронізації, сервіс матиме змогу опрацьовувати нові блоки (кожні 10 хвилин) та мати оновлену інформацію про блокчейн та баланси на адреси.

Сервіс для індексування токенів в мережі Ethereum працюватиме схожим чином. Потрібно аналогічно опрацювати всі транзакції в мережі. Зокрема, нас цікавлять всі події на контакті, які викликають контракти, які імплементують інтерфейс ERC-20. Для розробки системи відслідковування подій та їх збереження в базу даних скористаємось готовою бібліотекою наданою розробниками блокчейну Ethereum, під назвою go-ethereum. Вона написана мовою Go та надає можливість "підписуватися" на події деякого

контракту за його адресою та відслідковувати їх. Додатково, для кращої оптимізації під кінцевого користувача та зручності тестування, використаємо набір інструментів харківської блокчейн-компанії Distributed Lab, зокрема бібліотеку kit та figure. З їх допомогою, ми створимо єдиний файл з конфігураціями, яким можна буде модифікувати при подальшому налаштуванні.

Результатом роботи вийде сервіс, який буде працювати на сервері та не потребуватиме додаткової уваги, тобто, буде автономним. Основними частинами програми є:

- інтерфейс командної строки
- імпорт файлу конфігурації
- зв’язок з базою даних
- власне сервіс

Середовище виконання – дистрибутив Manjaro 2021.2. Стосовно бази даних, то вибір падає на нереляційну базу даних FoundationDB – розробку компанії Apple. Вона зберігає дані у вигляді ключ-значення. Її перевагою є можливість кластеризації, розширення бази на різних серверах та дисках.

Для поширення сервісу на сервери, можливо налаштувати Dockerfile, для побудови відповідного image. Для комфортного запуску, також можливо додати docker-compose файл. Тоді підняття та налаштування сервісу локально значно спроститься до виконання лише наступної команди.

*docker-compose up -d*

Окрім індексування, наш сервер криптогаманця повинен виконувати наступні функції: мати зв’язок з блокчейнами які він підтримує, підтримувати побудову транзакцій, валідацію підпису, отриманого від клієнта гаманця, надсилання транзакціїї до пулу транзакцій блокчейну.

Бекенд виконано як GRPC сервер. Задля цього, було створено окрему папку proto в коді серверу, де визначена специфікація серверу та інтерфейси з полями основних класів, вигляд запиту та відповіді на нього. Наступним кроком була генерація коду серверу, імплементування методів якого необхідно виконати. Генерація коду відбувається за допомогою спеціального сервісу розробленого компанією Google – *protoc-gen-go-grpc*.

### 4.3 Створення інтерфейсу командного рядка для взаємодії з прототипом гаманця

Для демонстрації виконаної роботи, було створено консольний застосунок з базовим функціоналом криптовалютного гаманця. Він має наступні команди:

- *create*
- *import <mnemonic>*
- *balance <blockchain>*
- *send <blockchain> <to address> <amount> [wallet number]*
- *sign <message>*

Розглядаючи даний функціонал, варто ще раз поглянути на додаток А. Введення команди *create* генерує користувачу нову мнемонічну фразу. Обов’язковим є запитання, чи дійсно користувач хоче перестворити дані свого гаманця. Наступним кроком є прохання задати пароль від криптогаманця. Користувач задає довільний пароль, за допомогою якого сід шифрується та зберігається локально, не передаючись через інтернет. У випадку, якщо користувач бажає імпортувати власну сід-фразу, то йому необхідно викликати команду *import*.

Команда *balance* відображає баланс користувача за всіма можливими токенами (у випадку з Ethereum). При цьому відбувається звернення до серверної частини застосунку. Аналогічно виконується команда відправки коштів. Транзакція створюється та підписується на локальному пристрої. Потім відбувається передавання інформації на сервер додатку. Там проходить процес валідації транзакції, зокрема перевірки правильності підпису. І вже сервер надсилає транзакцію на ноду блокчейну.

Зрештою команда *sign* є частиною модуля управління ключами та використовується для підпису будь-яких переданих даних.

### 4.4 Вдосконалення прототипу криптовалютного гаманця з підтримкою багатьох блокчейнів

Застосунок можливо значно покращити. Зокрема можна виділити основні напрямки розвитку. Один з них є підтримках багатьох блокчейнів. Оскільки даний продукт є прототипом, то тут реалізовано лише підтримку двох блокчейнів (Bitcoin та Ethereum). Наступним кроком в розробці є додавання різних мереж. Зокрема більшість Ethereum подібних блокчейнів не вимагають написання окремого коду, або індексера, та можуть мати взаємозамінний, конфігурований код.

Іншим кроком є запуск серверної частини на спеціальних серверах. Обрана база даних, вимагає вартісного обладнання, оскільки швидкодія відплачується ціною SSD-дисків. У випадку серйозного запуску даного продукту необхідно прорахувати вартість обладнання та, в більшості випадків, напряму спілкуватися з датацентрами, з приводу надання спеціалізованих потужностей.

# ВИСНОВКИ

У результаті виконання роботі було досягнуто поставленої мети. Зокрема, було досліджено блокчейни Bitcoin і Ethereum, криптогаманці Metamask та TrustWallet. Під час виконання кваліфікаційної роботи було розглянуто етапи розвитку блокчейнів та проблему створення криптогаманців. В результаті було виділено основні модулі, з яких складаються криптовалютні гаманці. Такими є:

- Модуль управління ключами
- Модуль індексації токенів
- Модуль управління транзакціями
- Модуль зв’язку з обмінниками
- Модуль відображення вартості активів

Використавши отриману з дослідження інформацію, було створено засіб розробки криптовалютних гаманців з підтримкою багатьох блокчейнів. Він має вигляд бібліотеки з уніфікованим інтерфейсом для створення та використання приватних ключів для блокчейнів Bitcoin та Ethereum.

Було побудовано два модулі для індексування блокчейну. З використанням відповідного комп’ютерного обладнання (зокрема серверів з потужними процесорами та накопичувачами розміром більше одного терабайта даних) з їх допомогою можливо проіндексувати весь блокчейн та мати змогу отримувати інформацію про баланс користувачів для відображення в критовалютному гаманці.

Використовуючи вищеописані створені продукти, було побудовано прототип криптовалютного гаманця у вигляді застосунку з інтерфейсом командної строки для управління приватними ключами та відображенням балансу даних ключів в реальному часі.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ


[1] Narayanan, Arvind. Bitcoin and cryptocurrency technologies: a comprehensive introduction / Arvind Narayanan, Joseph Bonneau, Edward Felten … [и др.]. — Princeton : Princeton University Press, 2016. — ISBN 978-0-691-17169-2

[2] Satoshi Nakamoto Bitcoin: A peer-to-peer Electronic Cash System [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: https://bitcoin.org/bitcoin.pdf

[3] Andreas M. Antopolous. Mastering Bitcoin [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: https://github.com/bitcoinbook/bitcoinbook

[4] Andreas M. Antopolous. Mastering Ethereum. – O`Reilly Media, Inc., 2018. – ISBN: 978-1-491-97194-9

[5] Ethereum Improvement Proposals. EIP-20: Token Standart [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: https://eips.ethereum.org/EIPS/eip-20

[6] Binance. How to Obtain Tax Reporting on Binance & Frequently Asked Questions [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: https://www.binance.com/en/support/faq/538e05e2fd394c489b4cf89e92c55f70

[7] Binance. Important Changes About Binance Identity Verification [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: https://www.binance.com/en/support/announcement/51bf294e26324211a4731ca998e110ca

[8] Decrypt. Monkey Business: Why Solana NFT Project SMB and Its DAO Are Battling Over Its Future [Електронний ресурс]:[Веб-сайт] – Режим доступу до

ресурсу: https://decrypt.co/93193/solana-monkey-business-monkedao-battling-over-future

[9] Bitcoin Wiki. Original Bitcoin client [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: https://en.bitcoin.it/wiki/Original_Bitcoin_client

[10] Learning me a bitcoin. Mnemonic seed [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: https://learnmeabitcoin.com/technical/mnemonic

[11] Learning me a bitcoin. Digital signatures (signing & verifying) [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: https://learnmeabitcoin.com/beginners/digital_signatures_signing_verifying

[12] Блокчейн и децентрализованные системы : учеб. пособие для студ. заведений высш. образования : в 3 частях. Ч. 1 / П. Кравченко, Б. Скрябин, О. Дубинина. – Харьков, 2019. – 488 с. : ил. 191; табл. 13; библиогр.: 124 назв. ISBN 978-617-7634-25-5 ISBN 978-617-7634-26-2 (ч. 1)

# ДОДАТОК А

## Модель автентифікації криптовалютних гаманців.