МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра програмних систем і технологій

**ЗВІТ**

про проходження технологічної практики

в умовах кафедри програмних систем і технологій

Індивідуальне завдання: Software Architecture Document (SAD)

Виконав: ст. гр. ІПЗ-33

Гоша Давід

Керівник практики:

доцент Зубик Людмила Володимирівна

Київ – 2023

Зміст

[1. Вступ 3](#_Toc135993818)

[1.1 Мета 3](#_Toc135993819)

[1.2. Сфера застосування 3](#_Toc135993820)

[1.3. Визначення, акроніми та скорочення 4](#_Toc135993821)

[1.4 Список використаних джерел 6](#_Toc135993822)

[1.5 Огляд 8](#_Toc135993823)

[2. Загальний опис 9](#_Toc135993824)

[2.1 Перспектива продукту 9](#_Toc135993825)

[2.2 Особливості продукту 9](#_Toc135993826)

[2.3 Класи користувачів та характеристики 10](#_Toc135993827)

[2.4 Операційне середовище 12](#_Toc135993828)

[2.5 Обмеження при розробці та впровадженні 13](#_Toc135993829)

[2.6 Документація користувача 14](#_Toc135993830)

[2.7 Припущення та залежності 15](#_Toc135993831)

[3. Архітектура системи 17](#_Toc135993832)

[3.1.1 Структура блоків 17](#_Toc135993833)

[3.1.2 Структура транзакцій 17](#_Toc135993834)

[3.1.3 Перевірка транзакцій і блоків 18](#_Toc135993835)

[3.1.4 Зберігання та розповсюдження блокчейну 18](#_Toc135993836)

[3.1.5 Механізм консенсусу 18](#_Toc135993837)

[3.2 Інтерфейс користувача та досвід 18](#_Toc135993838)

[3.3 Мережа та комунікація 19](#_Toc135993839)

[3.4 Архітектурна діаграма 20](#_Toc135993840)

[4 Детальний дизайн системи 22](#_Toc135993841)

[4.1 Дизайн інтерфейсу користувача 22](#_Toc135993842)

[4.2 Структури даних 22](#_Toc135993843)

[4.3 Дизайн класів 23](#_Toc135993844)

[4.4 Деталі реалізації 23](#_Toc135993845)

[5. Висновки та подальша робота 24](#_Toc135993846)

[5.1 Підсумок досягнень 24](#_Toc135993847)

[5.2 Вивчені уроки 24](#_Toc135993848)

[5.3 Майбутня робота 24](#_Toc135993849)

# 1. Вступ

У цьому документі надається вичерпний архітектурний огляд блокчейн-додатку, розробленого на Go. Блокчейн-додаток забезпечує безпечну і децентралізовану систему для зберігання і передачі токенів з акцентом на однорангові транзакції. Він використовує гібридний механізм консенсусу, який поєднує в собі підтвердження пройденого часу (PoET) і підтвердження роботи (PoW) для забезпечення безпеки і масштабованості.

## 1.1 Мета

Метою Документа про архітектуру програмного забезпечення (SAD) є надання детального архітектурного плану блокчейн-додатку. Цей документ призначений для широкого кола зацікавлених сторін, включаючи розробників, тестувальників, менеджерів проектів і бізнес-аналітиків, щоб допомогти їм зрозуміти архітектурний дизайн і функціональність системи.

Блокчейн-додаток відповідає ключовим системним вимогам, таким як безпека, продуктивність і масштабованість. Пріоритетом є безпека зберігання і передачі токенів, а також висока продуктивність і масштабованість за допомогою гібридного механізму консенсусу PoET і PoW.

Цей документ служить довідником для підтримки і подальшого розвитку системи. Його можна використовувати для розуміння структури системи, оцінки архітектурних компромісів, перевірки відповідності архітектури вимогам системи та планування майбутніх удосконалень.

SAD також має на меті гарантувати, що архітектура програми розроблена і задокументована в чіткий, стислий і послідовний спосіб, що сприяє кращій комунікації між зацікавленими сторонами, полегшує процеси прийняття рішень і підвищує загальну якість і зручність використання програми.

У наступних розділах ми розглянемо особливості архітектури системи, включаючи основні компоненти системи (ядро блокчейну, веб-гаманець і консольний інтерфейс) і те, як вони взаємодіють, використання гібридного механізму консенсусу PoET і PoW, а також підхід системи до безпеки, продуктивності і масштабованості

## 1.2. Сфера застосування

Цей Документ архітектури програмного забезпечення (SAD) охоплює три основні компоненти додатку на основі блокчейну, розробленого з використанням Go. До них відносяться

1. **Ядро блокчейну:** Цей компонент є серцем системи. Ядро являє собою базову інфраструктуру блокчейну, яка забезпечує безпечні, децентралізовані транзакції. Воно використовує гібридну модель консенсусу, що поєднує в собі підтвердження часу, що минув (PoET), і підтвердження роботи (PoW) для перевірки транзакцій і додавання блоків, пропонуючи баланс між безпекою, продуктивністю і масштабованістю. Проектування та реалізація ядра блокчейну були здійснені відповідно до стандартів ГОСТ для забезпечення надійності та стійкості.
2. **Веб-гаманець:** Цей компонент надає користувачам інтерфейс для взаємодії з блокчейном. Він дозволяє користувачам здійснювати транзакції та перевіряти баланс своїх токенів. Ключовою особливістю веб-гаманця є робота з приватними ключами. Після реєстрації користувача система завантажує приватний ключ, який прив'язаний до його облікового запису. Цей ключ має вирішальне значення для підписання транзакцій і перевірки особи користувача. Важливо зазначити, що механізм управління приватними ключами суворо дотримується найкращих практик кібербезпеки та відповідних стандартів ГОСТ для підтримки найвищого рівня безпеки.
3. **Консольний інтерфейс:** Консольний інтерфейс - це система взаємодії нижчого рівня, призначена в першу чергу для розробників, тестувальників та інших технічних зацікавлених сторін. Він надає набір інструментів командного рядка для прямої взаємодії з ядром блокчейну, пропонуючи більш детальний контроль і зворотній зв'язок, ніж веб-гаманець. Це робить його безцінним інструментом для налагодження, тестування, моніторингу та оптимізації системи.

Ці компоненти працюють разом, щоб створити безпечну, зручну для користувача систему для однорангового децентралізованого зберігання і передачі токенів. Цей SAD надає вичерпний опис програмної архітектури цих компонентів, пояснюючи, як вони задовольняють вимоги проекту, дотримуючись відповідних стандартів ГОСТ і кращих практик розробки програмного забезпечення.

Цей документ призначений для широкої аудиторії, включаючи розробників програмного забезпечення, тестувальників, менеджерів проектів та бізнес-аналітиків, але не обмежуючись ними. Він структурований таким чином, щоб бути доступним як для технічних зацікавлених сторін, які можуть заглибитися в тонкі деталі архітектури системи, так і для нетехнічних зацікавлених сторін, які можуть отримати уявлення про загальний дизайн і функціональність системи.

У наступних розділах цього ЗЗД буде розглянуто архітектурне представлення системи, важливі архітектурні рішення та обґрунтування цих рішень. Також буде показано, як система відповідає вимогам до продуктивності, безпеки та масштабованості. В рамках прагнення до прозорості та надійності, в цьому документі будуть використані авторитетні наукові та промислові джерела.

## 1.3. Визначення, акроніми та скорочення

Для забезпечення ясності та взаєморозуміння, у цьому розділі наведено перелік та пояснення визначень, акронімів та абревіатур, що використовуються у цьому документі з архітектури програмного забезпечення (SAD).

1. Блокчейн (Blockchain): Цифровий реєстр транзакцій, який дублюється і розподіляється по всій мережі комп'ютерних систем на блокчейні.
2. Ядро: Центральний компонент обчислювальної системи, який керує роботою комп'ютера та апаратного забезпечення.
3. Веб-гаманець: Тип гаманця, який дозволяє користувачам управляти своїми криптовалютами через веб-інтерфейс, що робить його доступним з будь-якого комп'ютерного пристрою з підключенням до Інтернету.
4. Приватний ключ: Складна форма криптографії, яка дозволяє користувачеві отримати доступ до своєї криптовалюти.
5. Механізм консенсусу: Відмовостійкий механізм, який використовується в комп'ютерних і блокчейн-системах для досягнення необхідної згоди щодо єдиного значення даних або єдиного стану мережі між розподіленими процесами або багатоагентними системами.
6. PoET: Абревіатура від англ. Proof of Elapsed Time (доказ часу, що минув). Це алгоритм консенсусу, який використовує систему справедливої лотереї, де кожен вузол мережі має рівні шанси на перемогу.
7. PoW: Абревіатура від Proof of Work (доказ роботи). Це механізм консенсусу в мережі блокчейн, який використовується для підтвердження транзакцій і створення нових блоків в ланцюжку.
8. SAD: Абревіатура від Software Architecture Document (Документ архітектури програмного забезпечення). Комплексний документ, який охоплює архітектуру програмної системи, включаючи поведінку, структуру та інші погляди на систему.
9. GO: Мова програмування, яка використовується для розробки ядра блокчейну, веб-гаманця і консольного інтерфейсу.
10. Хешування: Хеш-функція - це процес, який отримує вхідні дані і повертає рядок байтів фіксованого розміру, як правило, "дайджест". Хешування є наріжним каменем технології блокчейн. У контексті криптовалют, криптографічна хеш-функція - це особливий клас хеш-функцій, який особливо добре підходить для безпечної обробки великих обсягів даних.
11. Nonce: Nonce ("число, що використовується тільки один раз") - це число, додане до хешованого або зашифрованого блоку в блокчейні, яке при повторному хешуванні відповідає обмеженням рівня складності. Він використовується в системах підтвердження роботи блокчейну, щоб зробити процес майнінгу нового блоку (або запису в реєстрі) для блокчейну обчислювально дорогим.
12. Майнер: У контексті криптовалют майнер - це фізична або юридична особа, яка підтверджує і перевіряє нові транзакції та додає їх до блокчейну. Цей процес передбачає розв'язання складних математичних задач, що вимагає значних обчислювальних потужностей.
13. Криптовалюта: Це загальна абревіатура для "криптовалюти", типу цифрової або віртуальної валюти, яка використовує криптографію для забезпечення безпеки. Прикладами криптовалют є Bitcoin, Ethereum та Ripple.
14. Біткойн: Біткойн - перша децентралізована криптовалюта, яка залишається найвідомішою і найціннішою криптовалютою. Вона була створена у 2008 році невідомою особою під псевдонімом Сатоші Накамото. Біткоїни створюються як винагорода за процес, відомий як майнінг.
15. Gossip Protocol: Також відомий як "епідемічний протокол", це процедура або процес комп'ютерної однорангової комунікації, яка заснована на способі поширення епідемій. Деякі розподілені системи використовують протокол пліток для зв'язку через його високу надійність.
16. Оверлейна мережа: Оверлейна мережа - це комп'ютерна мережа, яка накладається на іншу мережу. Вузли в оверлейній мережі можна розглядати як з'єднані віртуальними або логічними посиланнями, кожне з яких відповідає шляху, можливо, через багато фізичних посилань, в базовій мережі.
17. Однорангова мережа (P2P): Мережа P2P - це мережа, в якій кожен комп'ютер (або "одноранговий") виступає в ролі сервера для інших, забезпечуючи спільний доступ до файлів і периферійних пристроїв без потреби в центральному сервері. P2P-мережі можуть бути створені вдома, на підприємстві або в Інтернеті.
18. Вузол: Пристрій або точка передачі даних у великій мережі. У контексті блокчейну нода - це будь-який комп'ютер, який підключений до мережі блокчейн.
19. Гаманець: У контексті криптовалюти гаманець - це цифровий інструмент, який дозволяє користувачам взаємодіяти з мережею блокчейн. Він дозволяє користувачам зберігати свої криптовалюти і керувати ними.
20. CLI (Command Line Interface - інтерфейс командного рядка): Текстовий інтерфейс користувача, який використовується для введення команд безпосередньо в комп'ютерну систему. В контексті цього проекту CLI призначений для досвідчених користувачів і майнерів.
21. Графічний інтерфейс користувача (GUI): Тип користувальницького інтерфейсу, який дозволяє користувачам взаємодіяти з електронними пристроями за допомогою графічних іконок і візуальних індикаторів, таких як вторинні позначення, замість текстових інтерфейсів, набраних командних міток або текстової навігації.
22. dAPP (децентралізований додаток): Це додаток, який працює в децентралізованій мережі, уникаючи єдиної точки відмови. dApps мають внутрішній код, що працює в децентралізованій одноранговій мережі, на відміну від традиційних додатків, де внутрішній код працює на централізованих серверах.
23. IP (Internet Protocol): Основний протокол зв'язку в наборі інтернет-протоколів для передачі дейтаграм через мережеві кордони.
24. TCP (Transmission Control Protocol): Один з основних протоколів у наборі інтернет-протоколів, TCP використовується програмами, які вимагають гарантованої доставки.
25. P2P (Peer-to-Peer): Децентралізована форма мережевої архітектури, в якій окремі вузли мережі ("піри") діють як незалежна мережа, здатна функціонувати безпосередньо з іншими вузлами за допомогою протоколів зв'язку P2P.
26. API (Application Programming Interface - інтерфейс прикладного програмування): Обчислювальний інтерфейс, який визначає взаємодію між декількома програмними посередниками. Він визначає типи викликів або запитів, які можуть бути зроблені, способи їх здійснення, формати даних, які повинні використовуватися, тощо.
27. JSON (JavaScript Object Notation): Легкий формат обміну даними, який легко читати і писати людям, а також легко аналізувати і генерувати машинам. JSON часто використовується, коли дані надсилаються з сервера на веб-сторінку.

Ці терміни та абревіатури складають основу мови, яка використовується в описі та обговоренні архітектури програмного забезпечення в наступних розділах цього документу.

## 1.4 Список використаних джерел

Цей розділ містить список основних джерел, які були використані при розробці цього проекту, а також джерела, які можуть надати додаткові поглиблені знання про використані технології та методології:

1. Nakamoto, S. (2008). "Біткоїн: пірингова система електронних грошей". Доступно за посиланням: https://bitcoin.org/bitcoin.pdf.
2. Wood, G. (2014). "Ефіріум: Безпечний децентралізований узагальнений реєстр транзакцій". Доступно за посиланням: https://ethereum.github.io/yellowpaper/paper.pdf.
3. Antonopoulos, A. M. (2014) (2014). "Освоєння біткоїна: розблокування цифрових криптовалют". O'Reilly Media, Inc.
4. "Go Programming Language". (2023). Доступно за посиланням: https://golang.org/doc/.
5. Бутерін, В. (2013). "Смарт-контракт наступного покоління та децентралізована платформа додатків". Доступно за посиланням: https://ethereum.org/en/whitepaper/.
6. "Стандарт верифікації та валідації систем та програмного забезпечення - IEEE Std 1012-2016". (2016). Стандарти IEEE.
7. "ГОСТ Р 34.10-2018: Інформаційні технології - Криптографічний захист інформації - Процеси підпису та перевірки електронного цифрового підпису". (2018). Національний стандарт України.
8. "ГОСТ Р 34.11-2012: Інформаційні технології - Криптографічний захист інформації - Хеш-функції". (2012). Національний стандарт України.
9. "ГОСТ Р ISO/IEC 27001-2016: Інформаційні технології - Методи забезпечення безпеки - Системи управління інформаційною безпекою - Вимоги". (2016). Національний стандарт України
10. Montresor, A., & Jelasity, M. (2013). "PeerSim: Масштабований симулятор P2P". У матеріалах Дев'ятої міжнародної конференції IEEE з пірингових обчислень (P2P'09). https://www.gsd.inesc-id.pt/~ler/docencia/rcs1314/papers/P2P2013\_041.pdf
11. SAFE Network. (2020). "Еволюція термінології з розвитком технологій: Децентралізована проти розподіленої". https://medium.com/safenetwork/evolving-terminology-with-evolved-technology-decentralized-versus-distributed-7f8b4c9eacb
12. Мова програмування Go. (2023). "Список відтворення мови програмування Go". https://www.youtube.com/playlist?list=PL4\_hYwCyhAvZmzpIjwewZOdBMFJooHIHx
13. Перейти до стандартного макету проекту. (2023). "Стандартний макет проекту Go." https://github.com/golang-standards/project-layout
14. Уолтер, К. (2023). "Огляд алгоритмів консенсусу в блокчейні". https://github.com/cedricwalter/blockchain-consensus
15. Antonopoulos, A. M., & Dhillon, V. (2017)."Освоєння біткоїна: програмування відкритого блокчейну". https://github.com/bitcoinbook/bitcoinbook
16. Superstas. (2023). "Реалізація Gcoin Mempool". https://github.com/superstas/gcoin/blob/master/gcoin/mempool/mempool.go
17. Kiayias, A., Russell, A., David, B., & Oliynykov, R. (2017)."Уроборос: Достовірно безпечний блокчейн-протокол з доказом частки". https://pdfs.semanticscholar.org/7dce/801b2b13001d0d3b0319c550ee1977e456df.pdf
18. Liao, K., Katz, J., & Zikas, V. (2018)."Протоколи BFT під вогнем". https://arxiv.org/pdf/1801.07447.pdf
19. Berini, M. (2015). "Розробка та впровадження додатку для біткоїн-гаманця". https://openaccess.uoc.edu/bitstream/10609/45861/6/mberiniTFM1215memoria.pdf

Ці посилання охоплюють як практичні аспекти технології блокчейн і мови програмування Go, так і теоретичні основи та стандарти криптографічних систем і систем управління інформаційною безпекою.

## 1.5 Огляд

У наступних розділах цього документу систематично викладено технічні та функціональні специфікації, принципи проектування, архітектурний план та інші важливі аспекти запропонованої системи на основі блокчейну.

Розділ "Загальний опис" містить загальний підсумок системи, окреслюючи її основні функції, ключові класи користувачів та їх характеристики, операційне середовище, обмеження при проектуванні та реалізації, документацію для користувачів та ключові залежності.

"Архітектура системи" заглиблюється в технологічну основу системи, розглядаючи структурні компоненти блокчейну, дизайн інтерфейсу користувача, мережеві протоколи та різні взаємодії блокчейну.

"Детальний дизайн системи" описує специфіку компонентів системи, детально розглядаючи дизайн інтерфейсу користувача, структури даних та їх взаємодію, дизайн класів та конкретні деталі реалізації, які є критично важливими для фактичного створення системи.

Розділ "Реалізація та тестування системи" описує процес втілення проекту в життя, надаючи уявлення про стратегію реалізації, процедури тестування, проблеми, що виникали, та їх вирішення, а також про заходи, вжиті для забезпечення якості та перегляду коду.

Нарешті, у розділі "Висновки та подальша робота" розглядаються досягнення проекту, отримані уроки та можливості для подальшого вдосконалення. Цей перспективний розділ підкреслює ітеративну природу процесу розробки програмного забезпечення і підкреслює потенціал для подальших удосконалень і оновлень у майбутньому.

Цей документ має на меті слугувати всеосяжним керівництвом до проекту, пропонуючи інформацію як для технічних (розробників, тестувальників), так і для нетехнічних (менеджерів проектів, бізнес-аналітиків) зацікавлених сторін

# 2. Загальний опис

## 2.1 Перспектива продукту

Блокчейн-додаток, що розробляється, є автономною системою, а не частиною більшої системи. Він призначений для забезпечення безпечної та децентралізованої платформи для однорангового зберігання та передачі токенів. Система буде використовувати принципи технології блокчейн, щоб гарантувати, що всі транзакції будуть постійно реєструватися і не можуть бути змінені або видалені.

Система складається з трьох основних компонентів: ядра блокчейну, веб-гаманця та консольного інтерфейсу.

* Ядро блокчейну є основою програми. Воно забезпечує базову інфраструктуру блокчейну, керує транзакціями, блоками та алгоритмами консенсусу.
* Веб-гаманець - це компонент, орієнтований на користувача, який дозволяє йому безпечно зберігати свої токени і керувати ними. Гаманець також включає в себе функцію завантаження приватних ключів, що підвищує безпеку токенів користувачів.
* Консольний інтерфейс - це інтерфейс командного рядка (CLI), який полегшує взаємодію між користувачами і блокчейном. Користувачі можуть використовувати цей інтерфейс для надсилання та отримання токенів, перевірки балансу та інших операцій у блокчейні.

Система використовує гібридний механізм консенсусу на основі доказу часу, що минув (PoET) і доказу роботи (Proof-of-Work, PoW). Цей підхід покликаний гарантувати, що всі транзакції підтверджуються чесно і ефективно, підтримуючи безпеку, продуктивність і масштабованість системи.

Продукт не має аналогів або попередніх версій, оскільки є унікальним поєднанням блокчейн-ядра, веб-гаманця та консольного інтерфейсу.

Ця система призначена для широкого кола користувачів, як технічних, так і нетехнічних. Розробникам, тестувальникам, менеджерам проектів та бізнес-аналітикам може бути корисно зрозуміти дизайн та роботу безпечної, ефективної та масштабованої блокчейн-системи. У той же час, нетехнічні користувачі можуть скористатися надійними функціями безпеки системи для безпечного зберігання і передачі токенів.

## 2.2 Особливості продукту

Блокчейн-додаток складається з декількох ключових функцій, призначених для підтримки потреб користувачів. Ось основні особливості системи:

**Ядро блокчейну**: Ядро - це серце програми, що забезпечує основну функціональність і операції, пов'язані з технологією блокчейн. Воно відповідає за управління транзакціями, блоками, алгоритмами консенсусу та загальною структурою блокчейну. Ядро призначене для забезпечення цілісності та безпеки блокчейну, перевірки всіх транзакцій та ведення децентралізованого реєстру.

**Веб-гаманець**: Цей компонент дозволяє користувачам зберігати та керувати своїми токенами в безпечному середовищі. Веб-гаманець розроблений таким чином, щоб бути зручним для користувача, забезпечуючи простий інтерфейс для взаємодії користувачів зі своїми токенами. Користувачі можуть переглядати свій баланс, надсилати та отримувати токени, а також здійснювати інші пов'язані з ними транзакції. Ключовою особливістю гаманця є можливість завантаження приватних ключів. Замість того, щоб генерувати нову пару ключів при реєстрації користувача, гаманець дозволяє користувачам завантажувати свої приватні ключі, забезпечуючи підвищену безпеку токенів користувачів.

**Консольний інтерфейс**: Консольний інтерфейс - це інтерфейс командного рядка (CLI), призначений для полегшення взаємодії між користувачами і блокчейном. Через цей інтерфейс користувачі можуть виконувати різні операції, такі як створення транзакцій, перевірка балансу і взаємодія з блокчейном. Інтерфейс консолі розроблений таким чином, щоб бути простим та інтуїтивно зрозумілим, що дозволяє як технічним, так і нетехнічним користувачам легко взаємодіяти з блокчейн-системою.

**Гібридний консенсус POET і PoW**: Додаток використовує гібридний механізм консенсусу, що поєднує Proof-of-Elapsed-Time (PoET) і Proof-of-Work (PoW). Цей унікальний підхід гарантує, що всі транзакції підтверджуються чесно і ефективно, а також підтримує безпеку, продуктивність і масштабованість системи.

**Функції безпеки**: Система приділяє велику увагу безпеці для захисту токенів і транзакцій користувачів. Вона використовує принципи технології блокчейн, такі як децентралізація і незмінність, щоб забезпечити безпечне середовище для зберігання і передачі токенів. Система також включає в себе стандарти, визначені в ГОСТ Р 34.10-2018 для процесів цифрового підпису та верифікації, ГОСТ Р 34.11-2012 для криптографічних хеш-функцій та ГОСТ Р ISO/IEC 27001-2016 для систем управління інформаційною безпекою.

**Масштабованість та продуктивність**: Додаток розроблено для забезпечення високої продуктивності та масштабованості. Завдяки використанню переваг мови програмування Go, система може обробляти велику кількість транзакцій і користувачів, зберігаючи при цьому високу швидкість обробки та ефективне використання ресурсів.

**Налаштовуваність**: Дизайн додатку дозволяє його кастомізацію. Користувачі можуть створювати власні токени, що відкриває широкий спектр потенційних варіантів використання. Це робить додаток придатним для різноманітних завдань, що виходять за рамки простого зберігання і передачі токенів, таких як відстеження активів, управління ланцюжками поставок тощо.

Разом ці функції забезпечують комплексну, безпечну і зручну платформу для однорангового зберігання і передачі токенів, що робить додаток цінним інструментом для всіх, хто цікавиться технологією блокчейн.

## 2.3 Класи користувачів та характеристики

Додаток Go пропонує широкий спектр функціональних можливостей і призначений для використання двома основними категоріями користувачів:

**Кінцеві користувачі**: Ці користувачі взаємодіють безпосередньо з блокчейн-додатком, в першу чергу через веб-гаманець і консольний інтерфейс. Вони можуть бути зацікавлені у використанні криптовалют або хочуть зрозуміти, як працює технологія блокчейн. Ключові характеристики цих користувачів такі:

* Початківці: Ці користувачі можуть не мати попереднього досвіду роботи з технологією блокчейн і зацікавлені в її вивченні та дослідженні. Простий та інтуїтивно зрозумілий інтерфейс веб-додатку допомагає їм орієнтуватися та виконувати основні операції, такі як створення нового користувача, завантаження користувача та перевірка балансу.
* Користувачі середнього рівня: Ці користувачі мають достатнє розуміння технології блокчейн і того, як вона працює. Їм зручно виконувати транзакції, перевіряти статус транзакцій і досліджувати блокчейн за допомогою команди "/chain" використовуючи CLI інтерфейс.
* Просунуті користувачі: Ці користувачі мають глибоке розуміння технології блокчейн. Їм зручно користуватися всіма функціями програми, включаючи здійснення транзакцій, перевірку балансів і розуміння основних процесів, що лежать в основі блокчейну.

**Розробники та адміністратори**: Ці користувачі взаємодіють з системою на більш технічному рівні, в першу чергу з ядром блокчейну і вихідним кодом. Їх ключові характеристики такі:

* Розробники блокчейну: Ці користувачі володіють технічними знаннями та навичками, пов'язаними з технологією блокчейн. Вони розуміють, як працює ядро блокчейну, механізм консенсусу і можуть взаємодіяти з системою за допомогою консольного інтерфейсу. Вони також можуть використовувати додаток як навчальний інструмент для кращого розуміння внутрішньої роботи блокчейну.

**Майнери**: Майнери в контексті блокчейну - це вузли мережі, які перевіряють і підтверджують нові транзакції та додають їх до блокчейну. Цей процес передбачає вирішення складних математичних задач, які вимагають значних обчислювальних потужностей, що часто називають "підтвердженням роботи" (Proof of Work, PoW). За свій внесок у мережу майнери отримують винагороду у вигляді токенів криптовалюти. Ключовими характеристиками майнерів є

* Розподіл ресурсів: Майнери виділяють свої обчислювальні ресурси (обчислювальну потужність, електроенергію тощо) для підтримки мережі блокчейн. На їхніх машинах працює програмне забезпечення, яке включає алгоритм консенсусу, який у випадку з біткоїном і Zcash є варіантом PoW.
* Забезпечення безпеки: Перевіряючи і додаючи транзакції в блокчейн, майнери сприяють загальній безпеці мережі. Обчислювальна робота, яку вони виконують, унеможливлює зловмисникам змінювати минулі транзакції з обчислювальної та фінансової точки зору.
* Перевірка транзакцій: Майнери відіграють вирішальну роль у перевірці транзакцій, гарантуючи, що вони є дійсними відповідно до правил протоколу, перш ніж додавати їх до блокчейну.

Майнери сприяють підтримці децентралізованої природи блокчейну, забезпечуючи надійність, безпеку та прозорість системи. Вони є невід'ємною частиною системи і допомагають забезпечити її постійну функціональність і надійність.

**Користувачі-початківці:** Ці користувачі представляють більшість кінцевих користувачів, які не обов'язково є технічно грамотними, але хочуть взаємодіяти з блокчейн-додатком за призначенням. У вашому випадку це користувачі, які хочуть керувати своїми криптовалютними активами. Для таких користувачів додаток надає веб-гаманець зі зручним графічним інтерфейсом користувача (GUI). Ключовими характеристиками для користувачів-початківців є

* Взаємодія з користувацьким інтерфейсом: Користувачі-початківці в основному взаємодіють з блокчейн-додатком через наданий веб-гаманець. Цей веб-інтерфейс спрощує процес проведення транзакцій, перевірки балансу рахунку та вивчення даних блокчейну, усуваючи необхідність взаємодії з командним рядком або прямих маніпуляцій з даними блокчейну.
* Низький технічний рівень: Від користувачів-початківців не очікується високого рівня технічної підготовки або глибокого розуміння технології блокчейн. Вони можуть не знати про такі процеси, як хеш-функції, цифрові підписи або механізми консенсусу, які забезпечують безпеку та функціональність блокчейну.
* Прості операції: Основні операції, які здійснюють користувачі-початківці, швидше за все, включають створення облікового запису, вхід в нього, перевірку балансу, надсилання та отримання токенів і, можливо, вивчення історії транзакцій. Всі ці операції доступні через веб-гаманець.
* Використання веб-гаманця: Гаманець служить важливим інструментом для користувачів-початківців. Він надає доступний та інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам керувати своїми цифровими активами. Веб-гаманець обробляє складні процеси, такі як управління приватними ключами та підписання транзакцій, за лаштунками, пропонуючи безперешкодний користувацький досвід.

**Взаємодія користувачів CLI:**

Основним способом взаємодії для користувачів є консольний інтерфейс, який надає такі команди, як "/exit" для виходу з програми, "/user" для доступу до специфічних для користувача команд, таких як перевірка адреси, гаманця та балансу. Команда "/chain" надає функціональні можливості, пов'язані з самим блокчейном, такі як друк ланцюжка і здійснення транзакцій.

Додаток надає чіткі повідомлення про успішні та невдалі операції, що сприяє легкій та інтуїтивно зрозумілій взаємодії. Крім того, дизайн дозволяє легко входити в систему новим користувачам і надає широкі функціональні можливості для вивчення досвідченими користувачами.

Такий підхід до дизайну, орієнтований на користувача, робить блокчейн-додаток придатним для широкого кола користувачів, від початківців до досвідчених ентузіастів блокчейну, а також розробників і системних адміністраторів.

## 2.4 Операційне середовище

**Ядро блокчейну**: Ядро, або серцевина блокчейну - це сама мережа, подібно до Біткоїну. Ядро розроблено для роботи на обладнанні з мінімальними вимогами - 1 ядро, 2 ГБ оперативної пам'яті та SSD на 20 ГБ. Ці вимоги роблять додаток доступним для широкого кола пристроїв і гарантують, що він може працювати без надмірного споживання ресурсів.

**Веб-гаманець**: Веб-гаманець надає користувачам інтерфейс для взаємодії з блокчейном. Він функціонує в середовищі браузера і, таким чином, не має додаткових апаратних вимог, окрім тих, що необхідні для запуску веб-браузера. Веб-гаманець не створює приватні ключі, а надає функціонал для їх імпорту, забезпечуючи безпечне управління активами користувача.

**Консольний інтерфейс**: Цей компонент надає майнерам повноцінний гаманець і провідник. Майнери запускають цей інтерфейс на своєму обладнанні і підключаються до одного з вузлів мережі. Оскільки цей інтерфейс також призначений для роботи на апаратному забезпеченні майнерів, вимоги до нього будуть аналогічні вимогам до ядра.

**Вузли майнінгу**: Вузли майнінгу підтримують мережевий консенсус, подібно до Біткоїна, використовуючи систему підтвердження часу, що минув (PoET). Як і ядро, ці вузли також розраховані на роботу з 1 ядром, 2 ГБ оперативної пам'яті і SSD на 20 ГБ, що забезпечує широку доступність для потенційних майнерів і сприяє децентралізованому характеру мережі.

**Вимоги до мережі**: Мережа розрахована на обробку приблизно двох транзакцій кожні дві хвилини. Така конструкція балансує між необхідністю обслуговувати транзакції користувачів і бажанням уникнути перевантаження мережі або окремих вузлів занадто великим трафіком.

Загалом, наш додаток розроблений таким чином, щоб бути легким і доступним, зі скромними вимогами до апаратного забезпечення та ефективною мережевою структурою. Він надає кілька точок доступу (через веб-гаманець і консольний інтерфейс), щоб задовольнити різні типи користувачів, забезпечуючи широку базу користувачів і сприяючи загальній надійності мережі блокчейн.

## 2.5 Обмеження при розробці та впровадженні

На розробку та реалізацію блокчейн-додатків накладається ряд обмежень, пов'язаних з дизайном та реалізацією. Ці обмеження стосуються мови програмування, відповідності законодавчим та нормативним вимогам, обмежень платформи, стандартів шифрування та впровадження найкращих галузевих практик.

**Обмеження щодо мови програмування**

Внутрішня частина додатку розроблена на мові програмування Go, а зовнішня частина реалізована з використанням HTML, SaaS та JavaScript. Це означає, що команда розробників повинна володіти навичками та досвідом роботи з цими мовами програмування. Крім того, будь-які допоміжні бібліотеки або інструменти розробки, що використовуються, також повинні бути сумісними з цими мовами.

**Правові та регуляторні обмеження**

Додаток, як і такі відомі платформи, як Ethereum або Solana, повинен відповідати всім застосовним місцевим і міжнародним нормам, пов'язаним з технологією блокчейн. Ці правила можуть охоплювати такі сфери, як конфіденційність користувачів, безпека даних і фінансові транзакції, серед іншого. Точний характер і обсяг цих норм може змінюватися залежно від географічного розташування користувачів і типу транзакцій, що виконуються на блокчейні.

**Обмеження платформи**

Система блокчейн розроблена з використанням алгоритму консенсусу Proof of Elapsed Time (PoET), який забезпечується через Trusted Execution Environments (TEE). Це накладає обмеження на типи систем, які можуть брати участь в мережі блокчейн в якості майнерів, оскільки можуть використовуватися тільки системи з підтримкою TEE.

**Обмеження щодо шифрування**

У додатку використовуються стандарти шифрування, подібні до тих, що використовуються в Ethereum. На додаток до них, додаток також використовує українські стандарти шифрування (ГОСТ). Це впливає на розробку і реалізацію всіх компонентів системи, де відбувається шифрування або дешифрування даних, включаючи, але не обмежуючись цим, обробку транзакцій і управління гаманцями.

**Обмеження найкращих практик**

Очікується, що розробка та реалізація системи будуть відповідати найкращим практикам, що переважають у розробці додатків блокчейну. Це включає в себе ефективне управління блокчейн-реєстром, безпечну реалізацію функціональності гаманця та зручний дизайн інтерфейсу консолі.

Врахування цих обмежень ставить перед командою розробників унікальні завдання. Під час проектування та впровадження системи вони повинні враховувати ці обмеження і прагнути оптимізувати продуктивність, безпеку та зручність використання системи.

## 2.6 Документація користувача

Користувацька документація є невід'ємною частиною будь-якого програмного проекту, пропонуючи керівництво і підтримку користувачам програми. В контексті нашого блокчейн-додатку, користувацька документація поділяється на документацію для кінцевих користувачів і розробників/адміністраторів.

Для кінцевих користувачів блокчейн-додатку, які в основному взаємодіють з веб-гаманцем і інтерфейсом консолі, документація розроблена з урахуванням різних рівнів кваліфікації та технічної обізнаності. В якості основного ресурсу надається вичерпний Посібник користувача. Цей посібник містить детальні інструкції для кожної операції користувача, починаючи від початкового налаштування облікового запису і закінчуючи виконанням транзакцій та запитами щодо балансу.

Крім того, система онлайн-довідки, вбудована в інтерфейс програми, пропонує користувачам негайну та контекстно-залежну підтримку. Ця функція покращує користувацький досвід, надаючи пряму допомогу, усуваючи потребу в зовнішніх ресурсах або підтримці. На додаток до цього, на веб-сайті програми є розділ поширених запитань (FAQ). Цей розділ відповідає на питання, що часто зустрічаються, і надає більш глибоке розуміння ключових концепцій блокчейну.

Щоб задовольнити різноманітну базу користувачів, також доступні відеоуроки, що демонструють ключові дії в додатку. Ці уроки слугують наочними посібниками для таких процесів, як створення облікового запису або виконання транзакцій, допомагаючи, зокрема, користувачам-початківцям, які можуть знайти візуальне представлення процесів більш зрозумілим.

З іншого боку, для розробників і адміністраторів, які займаються більш технічною роботою з системою, взаємодіючи безпосередньо з ядром і вихідним кодом блокчейна, документація набуває більш технічної форми. Детальний посібник для розробників пропонує поглиблене вивчення архітектури програми. Він ілюструє ядро блокчейну, механізм консенсусу та інші важливі компоненти за допомогою схем і пояснень. Цей посібник також допоможе розробникам у налаштуванні середовища розробки, налаштуванні системи та усуненні поширених проблем.

Крім того, вичерпна документація API слугує довідником для API програми. Вона описує кінцеві точки, їх призначення, необхідні вхідні дані та вихідний формат. Цей ресурс полегшує розробникам інтеграцію власних систем з блокчейном.

Нарешті, Посібник адміністратора призначений для осіб, які займаються управлінням і підтримкою мережі блокчейн. Він містить інструкції з налаштування та управління майнінг-вузлами, забезпечення безпеки системи та вирішення поширених проблем.

Уся документація курується з акцентом на ясність і лаконічність, мінімізуючи технічний жаргон, за винятком випадків, коли це необхідно. Документація регулярно оновлюється паралельно з оновленнями системи, щоб забезпечити постійну актуальність і точність. Основна мета цих зусиль - дати можливість всім користувачам оптимально використовувати блокчейн-додаток, незалежно від рівня їх технічної підготовки.

## 2.7 Припущення та залежності

У цьому розділі описані деякі припущення і залежності, пов'язані з розробкою і функціонуванням блокчейн-додатку.

**Припущення:**

1. Знання користувачів: Передбачається, що кінцеві користувачі мають базове розуміння того, як працювати з веб-додатками, і що майнери мають певний рівень технічних знань, особливо в роботі з технологією блокчейн.
2. Доступ до інтернету: Додаток передбачає, що користувачі мають стабільний і високошвидкісний доступ до інтернету. Успіх програми значною мірою залежить від безперебійного онлайн-з'єднання для оновлення транзакцій і блоків у режимі реального часу.
3. Регуляторне середовище: З огляду на те, що цей додаток працює в Україні, передбачається, що він відповідає всім чинним українським законам і нормативно-правовим актам, що стосуються технології блокчейн і криптовалют. Також передбачається, що ці нормативні акти залишаться відносно стабільними і підтримуватимуть технологію блокчейн в майбутньому.
4. Обслуговування та підтримка: Передбачається, що буде здійснюватися постійне технічне обслуговування і підтримка програми. Це необхідно для вирішення будь-яких потенційних проблем або помилок, впровадження оновлень і забезпечення безпеки та актуальності додатку.

**Залежності:**

1. Мова програмування Go: Додаток створено з використанням Go, а це означає, що його робота та майбутній розвиток залежать від постійної підтримки та розвитку цієї мови програмування.
2. Веб-технології: Веб-гаманець побудований з використанням HTML, SASS і JavaScript. Тому будь-які оновлення або зміни в цих технологіях можуть вплинути на роботу і розвиток веб-гаманця.
3. Механізм консенсусу PoET: Робота блокчейну залежить від механізму консенсусу PoET (Proof of Elapsed Time), який є менш енергоємним у порівнянні з традиційним консенсусом Proof of Work. Ця залежність означає, що будь-які зміни в алгоритмі PoET можуть вплинути на роботу блокчейну.
4. TEE (Trusted Execution Environment): Блокчейн-додаток також залежить від TEE для безпечної обробки транзакцій. Будь-які зміни або вдосконалення технології TEE можуть вплинути на продуктивність і безпеку додатку.
5. Мережева інфраструктура: Робота блокчейн-додатку залежить від мережевої інфраструктури, яка з'єднує майнерів і вузли для перевірки транзакцій і створення блоків. Будь-які збої або зміни в мережевій інфраструктурі можуть вплинути на загальну роботу блокчейну.

Розуміння цих припущень і залежностей має важливе значення для управління ризиками і планування майбутнього розвитку і роботи блокчейн-додатку. Рекомендується регулярно переглядати і контролювати ці фактори, щоб забезпечити постійний успіх додатк

# 3. Архітектура системи

Досліджувана система - це загальнодоступний блокчейн-додаток, розроблений на мові Golang, який реалізує структуру блокчейну, що зберігає незмінний запис усіх транзакцій, які відбуваються в мережі. У цьому розділі описано структуру та ключові компоненти блокчейн-додатку, приділяючи особливу увагу структурі блоків, структурі транзакцій, механізмам перевірки блоків і транзакцій, системі зберігання та механізмам консенсусу.

### 3.1.1 Структура блоків

Основним компонентом додатку блокчейн є блок. Блок слугує фундаментальною одиницею блокчейну, що містить запис про декілька транзакцій, і пов'язаний з іншими блоками, утворюючи структуру, подібну до ланцюжка.

Блок в додатку складається з декількох полів. Структура кожного блоку визначається в блоці Block struct, який включає в себе наступне:

* CurrHash: Зберігає хеш поточного блоку.
* PrevHash: Містить хеш попереднього блоку, який пов'язує блоки, утворюючи блокчейн.
* Nonce: Унікальний номер, який використовується в процесі майнінгу.
* Difficulty: Позначає складність проблеми майнінгу.
* Miner: Містить публічний ключ або ідентифікатор майнера, який додав блок до ланцюжка.
* Signature: Містить цифровий підпис для забезпечення цілісності блоку.
* TimeStamp: Зберігає час, коли блок було додано до ланцюжка.
* Transactions: Масив транзакцій, що містяться в блоці.
* Mapping: Карта, яка відстежує всі транзакції, наприклад, скільки криптовалюти було переведено з однієї адреси на іншу.

### 3.1.2 Структура транзакцій

Транзакції є рушійною силою блокчейну, оскільки вони представляють дії, що відбуваються в мережі. Структура кожної транзакції визначається в Структурі транзакції, яка включає в себе:

* RandBytes: Випадкові байти для ентропії.
* PrevBlock: Хеш попереднього блоку.
* Sender: Публічний ключ відправника транзакції.
* Reciver: Публічний ключ одержувача транзакції.
* Sum: Сума переказу криптовалюти.
* ToStorage: Кількість криптовалюти, переказаної на сховище.
* CurrHash: Хеш поточної транзакції.
* Sign: Цифровий підпис для підтвердження цілісності транзакції.

### 3.1.3 Перевірка транзакцій і блоків

Механізми верифікації є невід'ємною частиною підтримки безпеки і цілісності блокчейну. Система використовує спеціальні функції для перевірки транзакцій і блоків:

* IsValid(): Ця функція перевіряє транзакції шляхом перевірки хешу транзакції та цифрового підпису відправника.
* IsBlockValid(): Ця функція перевіряє блоки, перевіряючи різні елементи, такі як хеш, підпис, доказ, мітку часу та дійсність транзакцій.

### 3.1.4 Зберігання та розповсюдження блокчейну

Дані блокчейну зберігаються за допомогою бази даних SQLite, де кожен блок зберігається у вигляді запису. Цей метод забезпечує ефективне зберігання та пошук блоків, дозволяючи легко реплікувати базу даних між різними вузлами, забезпечуючи таким чином децентралізацію та стійкість до втрати даних.

### 3.1.5 Механізм консенсусу

Механізм консенсусу, що використовується в додатку, є гібридною моделлю, що поєднує в собі доказ часу, що минув (PoET) і доказ роботи (PoW). Цей механізм забезпечує справедливість, підтримуючи децентралізоване середовище, де кожен вузол-учасник має справедливу можливість видобувати блок, зберігаючи при цьому безпеку системи.

Структура і компоненти цього блокчейн-додатку забезпечують надійну, децентралізовану систему, гарантуючи безпеку і цілісність транзакцій. Його дизайн робить його придатним для різних застосувань, включаючи криптовалюти та децентралізовані додатки (dApps), пропонуючи багатообіцяючу перспективу для майбутніх досліджень і розробок у сфері блокчейн-технологій.

## 3.2 Інтерфейс користувача та досвід

Додаток блокчейн має як графічний інтерфейс користувача (GUI), так і інтерфейс командного рядка (CLI), щоб задовольнити різноманітне коло користувачів. Графічний інтерфейс в основному базується на веб-технологіях, що робить його доступним для користувачів з різним рівнем технічної підготовки. Він розроблений таким чином, щоб бути інтуїтивно зрозумілим і зручним для користувачів, особливо для тих, хто є новачком у світі криптовалют.

Графічний інтерфейс надає веб-гаманець, який дозволяє користувачам переглядати блокчейн, ініціювати транзакції та переглядати історію своїх транзакцій. Існує кілька функцій, які покращують користувацький досвід, в тому числі:

1. Головна сторінка: Користувачів зустрічає вступна головна сторінка, яка пропонує базову функціональність і спрямовує їх до подальших функцій.
2. Вхід/Реєстрація: Для посилення безпеки та персоналізації користувачі повинні увійти в систему, щоб отримати доступ до своїх гаманців. Існує також можливість реєстрації для нових користувачів.
3. Сторінка гаманця: Це місце, де користувачі керують своїми коштами. Вони можуть переглядати свій баланс, ініціювати нові транзакції та переглядати історію транзакцій.
4. Сторінка провідника: Ця сторінка дозволяє користувачам досліджувати блокчейн, включаючи перегляд всіх блоків і транзакцій.
5. Вихід: Користувачі можуть безпечно вийти зі свого гаманця, гарантуючи, що їхня інформація залишається в безпеці.

На додаток до веб-гаманця, існує CLI, призначений для більш просунутих користувачів і майнерів. Це дозволяє більш досвідченим користувачам взаємодіяти з системою на нижчому рівні, пропонуючи їм додатковий контроль і опції.

Що стосується зворотного зв'язку, то додаток розроблений таким чином, щоб тримати користувачів в курсі їхніх дій. Сповіщення надходять, коли транзакції ініціюються, підтверджуються або завершуються. Крім того, система пропонує чіткі та інформативні повідомлення про помилки, коли виникають проблеми, які допомагають користувачам вирішити проблему.

Додаток надає високий пріоритет безпеці. Приватні ключі користувачів зашифровані та надійно зберігаються в базі даних, що захищає їх від несанкціонованого доступу. Крім того, додаток включає в себе добре структурований механізм виходу з системи, щоб забезпечити безпечне завершення сеансів користувачів.

Підсумовуючи, можна сказати, що цей блокчейн-додаток забезпечує баланс між зручністю використання та функціональністю. Він надає комплексний, але простий в навігації інтерфейс для новачків, в той же час пропонуючи поглиблений контроль, який може знадобитися більш досвідченим користувачам. Додаток присвячений забезпеченню позитивного користувацького досвіду, що досягається завдяки продуманому дизайну, чіткій комунікації та надійним заходам безпеки.

## 3.3 Мережа та комунікація

Цей блокчейн-додаток використовує гібридну мережеву архітектуру, що поєднує в собі як клієнт-серверні, так і однорангові характеристики.

Потік мережевого зв'язку виглядає наступним чином:

* Клієнт -> Адресний сервер (через багатоадресну розсилку)
* Вузол -> Адресний сервер (через багатоадресну розсилку)
* Клієнт -> Вузол (одноранговий)
* Вузол -> Вузол (через однорангову мережу)
* Вузол -> Сервер пулу (через однорангову мережу)
* Вузол -> Сервер часу (через однорангову мережу)

Взаємодія клієнт-сервер в основному обертається навколо клієнтів, які взаємодіють з вузлами для отримання балансу, інформації про блок або для запису транзакції в блоці. Вузли надсилають запити до інших вузлів однорангової мережі, щоб додати новий блок до блокчейну, а також можуть запитувати певний діапазон майнінгу у пул-сервера або запитувати поточний стан часу у сервера часу.

Зв'язок між вузлами здійснюється за допомогою функції handleServer, яка прослуховує різні вхідні запити (ADD\_BLOCK, ADD\_TRNSX, GET\_BLOCK, GET\_LHASH, GET\_BLNCE, WAKEUP\_MSG) і реагує на них відповідним чином. Вузли спілкуються між собою за допомогою протоколу TCP, при цьому кожен вузол виступає як клієнт (ініціює з'єднання) і як сервер (приймає з'єднання). Така однорангова комунікація забезпечує децентралізацію, високу відмовостійкість і стійкість до поділу мережі.

Транзакції поширюються по мережі за допомогою функції makeTransaction в клієнті, де кожна транзакція надсилається всім підключеним вузлам. Потім вузли додають транзакцію до свого пулу пам'яті (пулу даних транзакцій, що очікують підтвердження) за допомогою функції handleTransaction.

Коли новий блок видобуто, він поширюється по мережі за допомогою функції pushBlockToNet. Для вирішення конфліктів і підтримки консенсусу в мережі вузли дотримуються правила "перемагає найдовший ланцюжок".

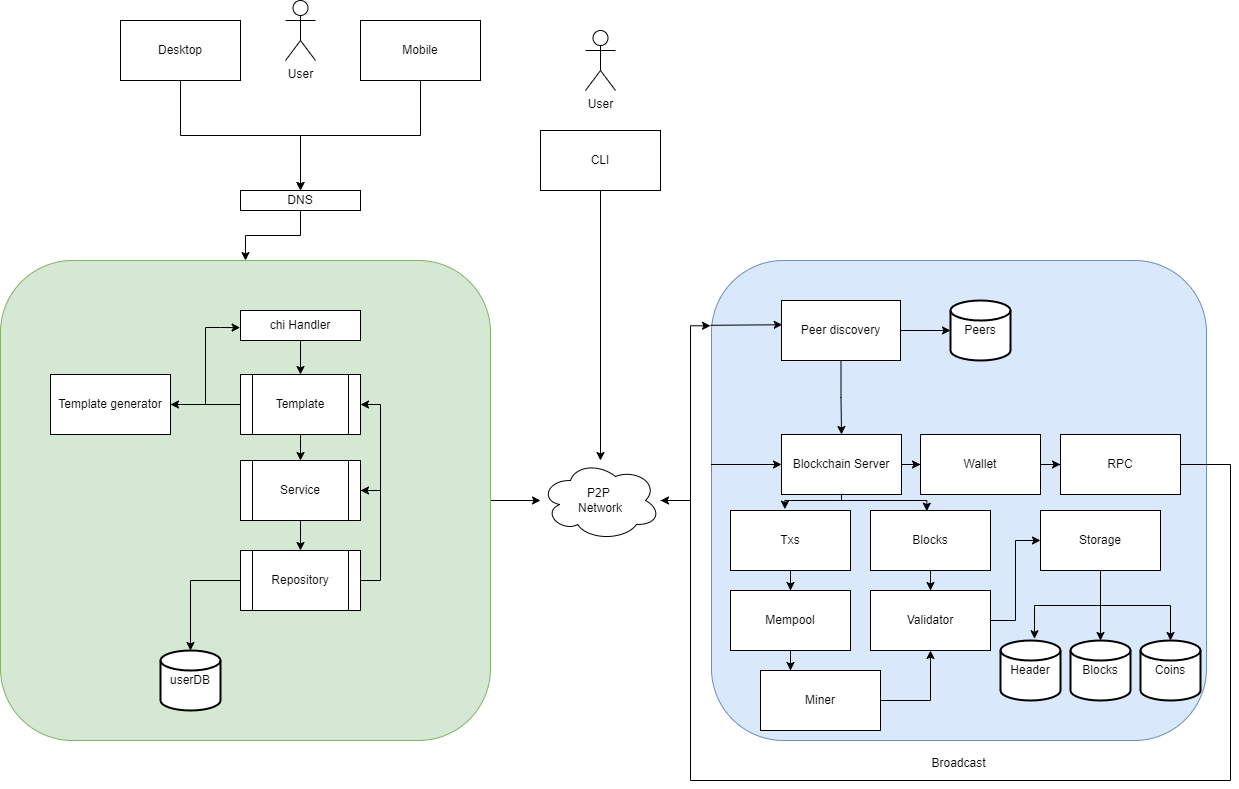
Для встановлення з'єднань вузли повинні мати файл, що містить IP-адреси довірених вузлів. Протокол не пропонує автоматичного виявлення вузлів.

Додаток блокчейн дотримується тих же стандартів шифрування і заходів безпеки, які використовуються в біткоїні для зв'язку між вузлами. Закриті ключі зашифровані для безпеки і зберігаються в базі даних.

Варто зазначити, що наявність вузлів, які звертаються до сервера пулу і сервера часу, вказує на те, що цей блокчейн-додаток підтримує стратегію спільного майнінгу і залежить від синхронізованого часу на всіх вузлах, що підвищує його надійність і точність.

В цілому, така архітектура призводить до створення високодецентралізованої, стійкої та безпечної мережі блокчейн, що дозволяє здійснювати безперебійні, прозорі та надійні транзакції.

## 3.4 Архітектурна діаграма



Архітектурна діаграма

# 4 Детальний дизайн системи

## 4.1 Дизайн інтерфейсу користувача

Інтерфейс користувача для додатку на основі блокчейну розроблений інтуїтивно зрозумілим і простим, щоб пристосуватись як для початківців, так і для досвідчених користувачів. Макет включає основні компоненти, подібні до популярних криптовалютних гаманців. Ключові елементи користувацького інтерфейсу включають

**Цільова сторінка**: Цільова сторінка додатку призначена для надання огляду функціональних можливостей додатку та заклику до дії для реєстрації або входу в систему.

**Інформаційна панель користувача**: Після входу в систему користувачі потрапляють на інформаційну панель, де вони можуть переглянути свій поточний баланс, історію транзакцій, ініціювати транзакції та отримати доступ до дослідника блокчейну.

**Процес транзакцій**: Процес транзакції розроблений таким чином, щоб бути простим. Користувачам потрібно ввести адресу одержувача і суму, яку вони хочуть переказати. Потім система підтверджує ці дані перед завершенням транзакції.

**Дані блокчейну**: Дані блокчейну представлені в прозорому і зрозумілому форматі, що дозволяє користувачам переглядати блокчейн і бачити деталі транзакцій для кожного блоку.

Реєстрація та вхід: Сторінки реєстрації та входу прості та безпечні. Користувачі повинні надати свої дані для реєстрації, після чого вони можуть використовувати ці дані для входу.

## 4.2 Структури даних

Структури даних системи в основному засновані на Go і розроблені таким чином, щоб імітувати структури в типовому блокчейні. Основні структури даних включають

**Блокчейн**: Це лінійний ланцюжок блоків, кожен з яких містить список транзакцій. Кожен блок пов'язаний з попереднім блоком, зберігаючи його хеш.

**Блок**: Кожен блок містить список транзакцій, мітку часу, хеш попереднього блоку і свій власний хеш.

**Транзакція**: Являє собою передачу криптовалюти від одного користувача до іншого. Містить адреси відправника і одержувача, суму переказу і мітку часу.

**Користувач**: представляє учасника мережі блокчейн з унікальними обліковими даними і гаманцем.

**MemPool**: Це колекція транзакцій, які були передані в мережу, але ще не включені в блок.

Блокчейн не зберігається в гаманці; скоріше, гаманець взаємодіє з блокчейном, що зберігається на вузлах мережі.

## 4.3 Дизайн класів

Оскільки цей додаток розроблено на Go, він в основному використовує структури, а не класи. Ключові структури включають в себе:

**BlockChain, Block, Transaction, User, MemPool**: Ці структури представляють фундаментальні компоненти блокчейну. Вони мають відповідні методи для таких операцій, як додавання транзакцій, перевірка блоків і так далі.

**Package**: Ця структура представляє собою пакет, що надсилається мережею, який містить опцію (що представляє тип повідомлення) і дані (корисне навантаження повідомлення).

Кожна з цих структур та їхні методи функціонують разом для підтримки блокчейну та обробки транзакцій.

## 4.4 Деталі реалізації

Додаток в основному реалізовано на мові Go, статично типізованій компільованій мові, відомій своєю простотою та ефективністю. Інтенсивно використовується стандартна бібліотека Go, а також додаткові бібліотеки сторонніх розробників, такі як Chi для маршрутизації, Logrus для ведення журналів та SQLite для управління базами даних.

Архітектура програми має багаторівневу структуру, зокрема, шаблон сервіс-репозиторій. Структура програми розділяє проблеми на окремі шари, що сприяє підтримці та масштабованості коду.

Обробка помилок в системі є невід'ємною частиною. Будь-які проблеми під час транзакцій, створення блоків або мережевих комунікацій належним чином відловлюються і обробляються, забезпечуючи надійність системи.

Безпека є першочерговим завданням, особливо з огляду на природу додатків, заснованих на блокчейні. Система використовує надійні методи шифрування для захисту транзакцій і даних користувачів. Наприклад, ключі шифруються в базі даних, що підвищує безпеку облікових даних користувачів

# 5. Висновки та подальша робота

## 5.1 Підсумок досягнень

Розробка і завершення цієї криптовалютної системи на основі блокчейну є значним досягненням. Ми успішно впровадили систему, яка включає в себе основні функції криптовалютної мережі, в тому числі можливість проводити транзакції, видобувати нові блоки та вести розподілену книгу. Гібридна архітектура системи, що поєднує однорангові та клієнт-серверні мережеві моделі, забезпечує надійний та ефективний зв'язок між різними вузлами мережі.

Крім того, помітним досягненням стало створення двох типів гаманців - веб-гаманця для нових користувачів і інтерфейсу командного рядка (CLI) для досвідчених користувачів і майнерів. Такий подвійний підхід робить систему доступною для широкого кола користувачів, пропонуючи розширені інструменти для більш досвідчених користувачів.

Система успішно дотримується правил консенсусу, запозичених з Біткоїна, таких як правило найдовшого ланцюжка для вирішення конфліктів і концепція доказу роботи для майнінгу блоків. Крім того, система забезпечує безпечне зберігання ключів користувачів у зашифрованому вигляді в базі даних.

## 5.2 Вивчені уроки

В ході реалізації проекту ми зіткнулися з численними викликами і отримали безцінний досвід. Реалізація однорангової мережі для розповсюдження блоків і перевірки транзакцій стала значним досвідом. Крім того, захист ключів в базі даних і забезпечення безпеки транзакцій було складним завданням, яке вимагало суворої уваги до деталей.

Процес розробки також підкреслив важливість ретельного тестування і налагодження. Враховуючи складність технології блокчейн, ретельне тестування було життєво важливим для виявлення та виправлення потенційних проблем. Якби проект було розпочато знову, приділення більшої уваги архітектурі та дизайну системи допомогло б передбачити та уникнути потенційних проблем.

## 5.3 Майбутня робота

Хоча поточний стан проекту є робочим і відповідає його початковим цілям, все ще є місце для розширення і вдосконалення. Майбутня робота може включати інтеграцію більш досконалих криптографічних методів для підвищення безпеки системи та впровадження більш зручних функцій в інтерфейсі веб-гаманця для покращення користувацького досвіду.

Крім того, масштабування мережі для підтримки більшої кількості транзакцій і користувачів є потенційним напрямком майбутньої роботи. Також можуть бути проведені дослідження більш ефективних алгоритмів консенсусу, щоб зменшити обчислювальну потужність, необхідну для майнінгу, і зробити систему більш енергоефективною.

Крім того, створення API для взаємодії сторонніх додатків з системою може бути корисною функцією, яку варто розглянути в майбутньому. Це дозволило б розробляти додаткові сервіси і додатки навколо криптовалютної системи, тим самим розширюючи її потенційні варіанти використання.