

Вінницький національний технічний університет

Факультет комп'ютерних систем і автоматики

Кафедра автоматизації та інтелектуальних інформаційних технологій

Пояснювальна записка

домагістерської кваліфікаційної роботи

магістр

(освітньо-кваліфікаційний рівень)

на тему «Система обміну токенами блокчейну ETN»

Виконав: студент II курсу, групи 1АКІТ-18м (д/ф)
напряму підготовки

151 – Автоматизація та комп'ютерно-

інтегровані технології

(шифр і назва напряму підготовки, спеціальності)

Кушнір В.І.

(прізвище та ініціали)

Керівник: к.т.н., доцент Паламарчук Є.А.

(прізвище та ініціали)

Рецензент: к.т.н., доцент Биков М.М.

(прізвище та ініціали)

Вінниця – 2019 року

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет комп'ютерних систем і автоматики

Кафедра автоматизації та інтелектуальних інформаційних технологій
Освітньо-кваліфікаційний рівень магістр

Напрямок підготовки 151 – Автоматизація та комп'ютерно-інтегровані технології
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри АІВТ

д.т.н., проф. Р.Н. Кветний

ЗАВДАННЯ

НА ДИПЛОМНУ РОБОТУ (ПРОЕКТ) СТУДЕНТУ

Кушніру Владиславу Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Система обміну токенами блокчейнуETH
Керівник проекту (роботи) ПаламарчукЄ. А., к.т.н., доц. каф. АІТ

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від “ ___ ” _____ 2019 року № ___

2. Строк подання студентом проекту (роботи) _____

3. Вхідні дані до проекту (роботи) Мобільні пристрої на операційних системах андроїд таіos. Середовище розробки – PHPStorm. Мова програмування –JavaScript. Бібліотека – ReactNative.Середня швидкість виконання транзакції 3хв. Максимальна сума разового переказу 25ETH.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)вступ; аналіз методів та засобів розробки модулів роботи з технологією блокчейн та мобільних-сервісів; розробка структури та дизайну мобільного-додатку; програмна реалізація мобільного-додатку; тестування мобільного-додатку; економічна частина; висновки; перелік посилань; додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): Додаток А (обов'язковий) Технічне завдання, Додаток Б (обов'язковий) Графічна частина, Додаток В (обов'язковий) Лістинг головного екрану додатку, Додаток Г (обов'язковий) Лістинг екрану з інформацією для отримання транзакцій, Додаток Д (обов'язковий) Лістинг екрану з формою для відправки транзакцій, Додаток Е (обов'язковий) Лістинг екрану з даними про гаманець, Додаток Є (обов'язковий) Лістинг скрипту для відправлення транзакції.

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	Завдання прийняв
1	Паламарчук Є. А., к.т.н., доц. каф.АІТ		
2	Паламарчук Є. А., к.т.н., доц. каф.АІТ		
3	Паламарчук Є. А., к.т.н., доц. каф.АІТ		
4	Паламарчук Є. А., к.т.н., доц. каф.АІТ		
5	Козловський В.О., к.е.н., професор		

Дата видачі завдання _____ 2019р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітки
1	Аналіз стану проблеми і постановка задач		
2	Аналіз процесу розробки модулів роботи з технологією блокчейн та мобільних-сервісів		
3	Розробка структури мобільного-додатку		
4	Розробка алгоритму обміну токенами блокчейнуЕТН		
5	Програмна реалізація додатку		
6	Тестування роботи додатку		
7	Оформлення матеріалів до захисту		
8	Попередній захист		
9	Остаточний захист		

Студент _____ Кушнір В. І.
(підпис) (прізвище та ініціали)

Керівник проекту (роботи) _____ Паламарчук Є. А.
(підпис) (прізвище та ініціали)

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ РОЗРОБКИ МОДУЛІВ РОБОТИ З ТЕХНОЛОГІЄЮ БЛОКЧЕЙН ТА МОБІЛЬНИХ-СЕРВІСІВ	12
1.1 Аналіз стану проблеми	12
1.2 Аналіз процесу розробки мобільних-застосунків.....	14
1.3 Аналіз технології блокчейн як засобу зберігання даних	16
1.4 Аналіз інструменту для створення мобільних-застосунків	19
1.5 Порівняльний аналіз аналогів	21
1.6 Постановка задач розробки	25
1.7 Висновки	25
2 РОЗРОБКА СТРУКТУРИ ТА ДИЗАЙНУ МОБІЛЬНОГО-ДОДАТКУ	26
2.1 Вибір методів відображення інформації.....	26
2.2 Розробка структури мобільного-додатку.....	27
2.3 Розробка інтерфейсу екранів.....	28
2.4 Розробка дизайну мобільного-застосунку	31
2.5 Розробка структури програмної системи обміну токенами блокченку ЕТН....	35
2.6 Розробка алгоритму обміну токенами блокченку ЕТН.....	38
3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО-ДОДАТКУ	40
3.1 Обґрунтування інструментів для розробки	40
3.2 Обґрунтування вибору мови та середовища програмування	41
3.3 Програмна реалізація мобільного-додатку.....	43
3.4 Висновки	48
4 ТЕСТУВАННЯ МОБІЛЬНОГО-ДОДАТКУ	49
4.1 Аналіз методик тестування	49
4.2 Функціональне тестування мобільного-додатку.....	51
4.3 Тестування кросплатформності	57
4.4 Висновки	60

5. ЕКОНОМІЧНИЙ РОЗДІЛ	61
5.1 Технологічний аудит розробленої системи обміну токенами блокчейну ЕТН	61
5.2 Розрахунок витрат на виконання роботи.....	66
5.3 Розрахунок економічного ефекту від можливого впровадження нашої розробки	70
ВИСНОВКИ.....	76
ПЕРЕЛІК ПОСИЛАНЬ	78
ДОДАТКИ.....	82
Додаток А (обов'язковий). Технічне завдання.....	83
Додаток Б (обов'язковий). Графічна частина.....	86
Додаток В (обов'язковий). Лістинг головного екрану додатку	93
Додаток Г (обов'язковий). Лістинг екрану з інформацією для отримання транзакцій.....	94
Додаток Д (обов'язковий). Лістинг екрану з формою для відправки транзакцій .	96
Додаток Е (обов'язковий). Лістинг екрану з даними про гаманець	98
Додаток Є (обов'язковий). Лістинг скрипту для відправлення транзакції	100

АНОТАЦІЯ

В магістерській дипломній роботі розроблено систему для обміну токенами блокчейну ETH за допомогою технології блокчейн. Обмін працює на основі розробленого у роботі алгоритму, який включає в себе створення гешу транзакції та подальшого проведення транзакції в мережу блокчейн з передачею гешу в відповідному полі тіла транзакції. Реалізовано обмін цифровими активами та перевірка історії проведених операцій над гаманцем.

Мобільний-додаток «CryptoPay» був розроблений з використанням бібліотек jQuery та ReactNative, мови програмування JavaScript та середовища розробки PHPStorm. Мобільний-додаток може працювати на мобільних платформах IOS та Android.

АННОТАЦИЯ

В магистерской дипломной работе разработано систему для обмена токенами блокчейна ETC с помощью технологии блокчейн. Обмен работает на основе разработанного в работе алгоритма, который включает в себя создание хэша транзакции и последующего проведения транзакции в сеть блокчейн с передачей хэша в соответствующем поле. Реализовано обмен цифровими активами и проверка проведенних операций над кошельком.

Мобильное-приложение «Crypto Pay» было разработано с использованием библиотек jQuery и ReactNative, языка программирования JavaScript и среды разработки PHP Storm. Мобильное-приложение может работать на мобильных платформах IOS и Android.

ANNOTATION

The master's thesis developed a ETH blockchain token exchange system using blockchain technology. The exchange works on the basis of an algorithm developed during the work, which involves creating a transaction hash and a subsequent transaction into a blockchain network with a hash transmission in the corresponding field of the transaction body. The exchange of digital assets and checking of the wallet transaction history were implemented.

The «CryptoPay» mobile application was developed using jQuery and ReactNative libraries, JavaScript programming language and PHP Storm development environment. The mobile application can run on both iOS and Android mobile platforms.

ВСТУП

Актуальність. На даному етапі цивілізованого розвитку, постає низка питань щодо інструментів для обміну фінансовими активами. У зв'язку з розширенням меж ринкових відносин, проблема грошових переказів стає все актуальнішою. Все частіше фінансові відносини між людьми виходять за межі фізичного обміну і тоді виникає питання цифрових грошей.

Незважаючи на наявність цілого спектру допоміжних інструментів, щодня люди зіштовхуються з проблемою, коли їм необхідно перевести кошти на рахунки родичів, працівників, виконати оплату послуг. Одразу вони зіштовхуються з обмеженнями. На сьогоднішній день існує декілька варіантів, як отримати послуги онлайн переказу. Найпоширеніші це онлайн перекази через банк, використання сторонніх ресурсів з цифровими грошима та використання крипто-гаманців.

Якщо відправляються кошти з банківського рахунку певного банку, переважніше, щоб у отримувача був рахунок того ж банку, в інакшому випадку, ви будете вимушені оплатити велику комісію за виконання даного переказу, переказ коштів може займати декілька робочих днів. Міжнародні перекази не завжди можливі, також користувач зіштовхується з конвертацією переказу в валюту країни отримувача, що не завжди здійснюється по вигідному для клієнта курсу. Часто люди звертаються до послуг компаній які працюють з традиційними цифровими валютами, в них є низка тих самих проблем, що присутні в звичайних банків.

Криптовалюта позбавлена даних проблем, так як за своєю концепцією має однакову цінність влюбій точці планети. Використання розподіленої бази блокчейн дає змогу підвищити безпеку грошовим переказам. Більшість існуючих криптовалютних гаманців дають змогу виконувати транзакції доволі швидко [1].

Метою магістерської дипломної роботи є підвищення ефективності та пришвидшення процесу розробки мобільних-додатків обміну фінансовими активами.

Для досягнення поставленої мети в роботі вирішуються такі завдання:

- аналіз стану проблеми і постановка задач;

- аналіз технології блокчейн як засобу зберігання даних;
- розробка структури мобільного-сервісу;
- розробка візуальної частини мобільного-додатку;
- розробка алгоритму обміну токенами блокчейну ETC;
- програмна реалізація мобільного-сервісу;
- тестування роботи мобільного-сервісу.

Об'єктом є процес створення модулів роботи з технологією блокчейн та мобільних-сервісів.

Предмет дослідження являє собою методи і засоби розробки модулів роботи з технологією блокчейн та мобільних-сервісів.

Виконання даної магістерської роботи дозволяє отримати практичні навички розробки модулів роботи з технологією блокчейнта створення мобільних-сервісів з використанням сучасних JavaScript бібліотек.

Наукова новизна одержаних результатів:

Удосконалено модель захисту персональних даних користувача, що на відміну від існуючих відрізняється застосуванням мнемонічної фрази, що дає змогу забезпечити додаткову безпеку персональних даних, за рахунок використання seed-фрази.

Практичне значення результатів роботи. Створені алгоритми та програмне забезпечення можуть бути використані для подальшої розробки ефективніших систем обміну цифровими фінансовими активами та окремих модулів для них.

Апробація та публікації результатів роботи. Результати досліджень були представлені на VIII Міжнародній науково-практичній інтернет-конференції «Сучасний рух науки» (жовтень 2019).

Робота складається з 5 розділів. В першому розділі проаналізовано сучасний стан розвитку Інтернет-технологій, поставлено задачі, які потрібно виконати у магістерській роботі, проаналізовано методи вирішення задач, проведено порівняльний аналіз аналогів. В другому розділі розроблено структуру мобільного-додатку та алгоритми роботи його основних модулів. Розроблено шаблони графічного інтерфейсу сторінок додатку. В третьому розділі обґрунтовано вибір

середовища розробки PHPStorm і технологій ReactNative і JavaScript. Описано програмну реалізацію елементів керування і відображення інформаційного забезпечення. В четвертому розділі проведено функціональне тестування роботи мобільного-додатку на операційних системах Android та IOS. Також проведено тестування роботи обміну токенів блокчейні ETH. В п'ятому розділі розроблено економічну частину. У додатках міститься технічне завдання, лістинг коду основних частин програми та ілюстративний матеріал.

1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ РОЗРОБКИ МОДУЛІВ РОБОТИ З ТЕХНОЛОГІЄЮ БЛОКЧЕЙН ТА МОБІЛЬНИХ-СЕРВІСІВ

1.1 Аналіз стану проблеми

Сучасній людині важко уявити своє життя без високотехнологічних пристроїв, як-то смартфони, планшетні ПК, бо саме ці науково-технічні новинки дозволяють не тільки підтримувати зв'язок з людьми із різних куточків світу, а й допомагають у формуванні та розвитку сучасного інформаційного простору. Швидкий та невпинний розвиток інформаційних технологій, в особливості глобальної Інтернет-мережі, зумовив збільшення потреби людей у ресурсах, що допомогли б обмінюватись фінансовими активами дистанційно. Щодня зростає кількість користувачів, що використовують онлайн перекази для задоволення потреб таких як отримання чи оплати заробітної плати, міжнародних переказів рідним з-за кордону, оплати послуг інтернет ресурсів у вигляді інтернет-магазинів чи web-сайтів що надають послуги навчання. Існуючі ресурси, що надають дані послуги, не завжди дають змогу виконати такі транзакції зручно для користувача, або взагалі не мають можливості надати свої послуги і причин може бути безліч як відсутність представництв на території держави де знаходиться отримувач. Тоді користувач зіштовхується або з неможливістю отримати послуги, або з додатковими витратами на комісії міжбанківських переказів та конвертацію валюти, що може виконатись не по найкращому курсу для користувача, що викликає ряд незручностей для нього. В зв'язку з цим виникає потреба продовжувати наукові дослідження в даному напрямі.

Дана сфера є доволі специфічною, через те, що необхідно забезпечити безпеку фінансів користувача систему. Користувачі традиційних банківських систем піддаються низкі ризиків пов'язаних з їхніми збереженнями. Коли людина укладає угоду з банком на зберігання грошових ресурсів це фактично угода на довіру, тому що гроші які знаходяться на рахунку людини, фізично зберігаються у банку тобто особа не володіє грошима і у разі настання банкрутства банку, людина втрачає все. Що стосується криптовалют, вони зберігаються не у того хто надає послуги у

виглядіресурсу для обміну криптовалютою, а безпосередньо у користувача на його адресі в блокчейні, самі системи використовують ваш приватний ключ тільки для того щоб ви могли взаємодіяти з блокчейном, відправляти чи отримувати транзакції. Завантаживши ваш приватний ключ в дані застосунки, криптовалюта буде зберігатися в блокчейні а ви лише отримаєте можливість розпоряджатися ними через інтерфейс, що одразу вирішує проблему втрати фінансових активів у зв'язку з банкрутством. Також традиційні банківські системи, хоч і малоімовірно через те що існує велика кількість засобів безпеки, але можуть піддаватися хакерським атакам з метою викрадення активів, так як використовують централізовані бази даних в яких зберігаються дані користувачів. Що стосується криптовалют, через їхню специфіку унеможлиблює викрадення фінансів у такий спосіб, так як блокчейн являє собою розподілену базу даних, тобто одночасно зберігається у всіх користувачів мережі, що викликає труднощі у підробці даних.

За існуючими на даний момент способами обміну фінансами, особа може звернутися до банку. Зазвичай процедура займає до 3 робочих днів, користувач зіштовхується з обмеженням максимальної суми переказу, додатковою оплатою послуги переказу, обмеженим вибором країн з якими працює банк. Весь цей процес викликає незручності у користувача. Для усунення цих проблем було прийнято рішення розробити власний мобільний-додаток з простим та багато функціональним інтерфейсом, який дозволив би користувачеві швидко, вигідно та без зайвих обмежень виконувати обмін фінансовими активами. Таким чином сервіс повинен мати можливість повноцінного користування цифровими грошима у вигляді криптовалюти, а саме містити в собі функціонал отримання та відправлення токенів, відображати останні проведені операції.

В результаті було визначено, що необхідність покращення засобів обміну фінансовими активами є актуальною, та було прийнято рішення розробити власний додаток, який дозволяв би розв'язати цю проблему.

1.2 Аналіз процесу розробки мобільних-застосунків

Мобільні застосунки - це програмні застосунки, призначені для роботи на смартфонах, планшетах та інших мобільних пристроях[2]. Вони, як правило, доступні через магазини застосунків, якими керують власники мобільної операційної системи. Розробка мобільного-застосунку повинна бути розбита на певні етапи, щоб зменшити кількість похибок в подальшій реалізації поставлених задач.

Першим кроком є визначення цілі. В першу чергу потрібно визначитися, для чого потрібен мобільний-додаток. Від цього буде залежати структура і наповнення додатку, а також способи просування.

Необхідно вивчити аудиторію: хто майбутні клієнти і як вони шукають інформацію про продукт або послугу. Від цього і буде залежати, який мобільний-сервіспотрібно створювати.

Далі необхідно зайнятись просуванням бренду. Якщо є готовий продукт, який утвердився на ринку, але він невідомий в інтернеті: в різних рейтингах , на сторінках форумів або на інформаційних сайтах немає згадок про продукт, підійде створення сайту-візитки, який буду підвищувати лояльність клієнтів.

Скорочення навантаження на службу підтримки. Клієнтам не вистачає інформації про продукт, і створення інформаційного простору дозволило б вирішити цю проблему. Тобто тут потрібен лендинг з акцентом на інформаційний контент. Також такий сайт підійде для навчання нових працівників.

Якщо необхідно, наприклад підвищити попит, проінформувати потенційних користувачів про існуючий продукт, або немає можливості запуснути повноцінний сайт, варто почати з landingpage для одного товару або послуги.

Далі необхідно скласти технічне завдання. Технічне завдання, як можна здогадатися з назви, це документ, в якому будуть зібрані всі вимоги до майбутнього застосунку. Ступінь деталізації залежить від списку ваших бажань і складності ресурсу.

Виділимо основні розділи:

- опис мети і призначення застосунку, а також коротка інформація про цільову аудиторію;
- вимоги до оформлення застосунку. У загальних рисах необхідно описати, як буде виглядати дизайн майбутнього ресурсу, визначити бажану колірну гамму. Також в даному розділі потрібно вказати логотип або позначити, яким він повинен бути;
- список засобів перегляду. Визначитись з платформою на якій буде використовуватись застосунок;
- розробка макету мобільного додатку: на основі технічного завдання фахівці дизайну розроблять унікальний дизайн майбутнього додатку. В процесі створення макету додатку важливо врахувати всі можливі функції, які в майбутньому мають функціонувати;
- створення прототипу і проектування структури UI / UX - для повного та детального розуміння, як майбутній додаток буде виконувати свої функції дизайнер повинен промалювати всі можливі робочі екрани, карту взаємодії користувача з додатком.

Дизайн відіграє одну із ключових ролей при розробці мобільного-застосунку і він йде наступним кроком. Багато помилково починають з цього етапу, тому в підсумку виходять гарні, але незручні застосунки.

Перед зверненням до дизайнера необхідно зробити міні-добірку застосунків, які подобаються, але не потрібно забувати про особливості тематики продукту. Орієнтуватися необхідно на конкурентів і потрібно ознайомитися з трендами в дизайні, щоб продумати унікальний і сучасний дизайн. При виборі компанії або фрілансера обов'язково подивитися портфолію. Воно повинно вам подобатися і, в ідеалі, містити приклади з вашої тематики.

Останнім кроком є наповнення застосунку контентом, який буде залежати від конкретної вибраної тематики.

Виконавши кожен пункт, можна отримати робочий мобільний-застосунок, готовий до просування і повноцінного використання.

1.3 Аналіз технології блокчейн як засобу зберігання даних

Блокчейн – це розподілена база даних, у якій пристрої зберігання даних не підключені до загального сервера. Ця база даних зберігає постійно зростаючий список упорядкованих записів, які називаються блоками. Кожен блок містить мітку часу і посилання на попередній блок, а також транзакції, які були здійснені в мережі[3]. Для того, щоб зберегти дані в мережі блокчейн необхідно провести транзакцію, яка потім потрапить в блок і буде добута (рис. 1.1).

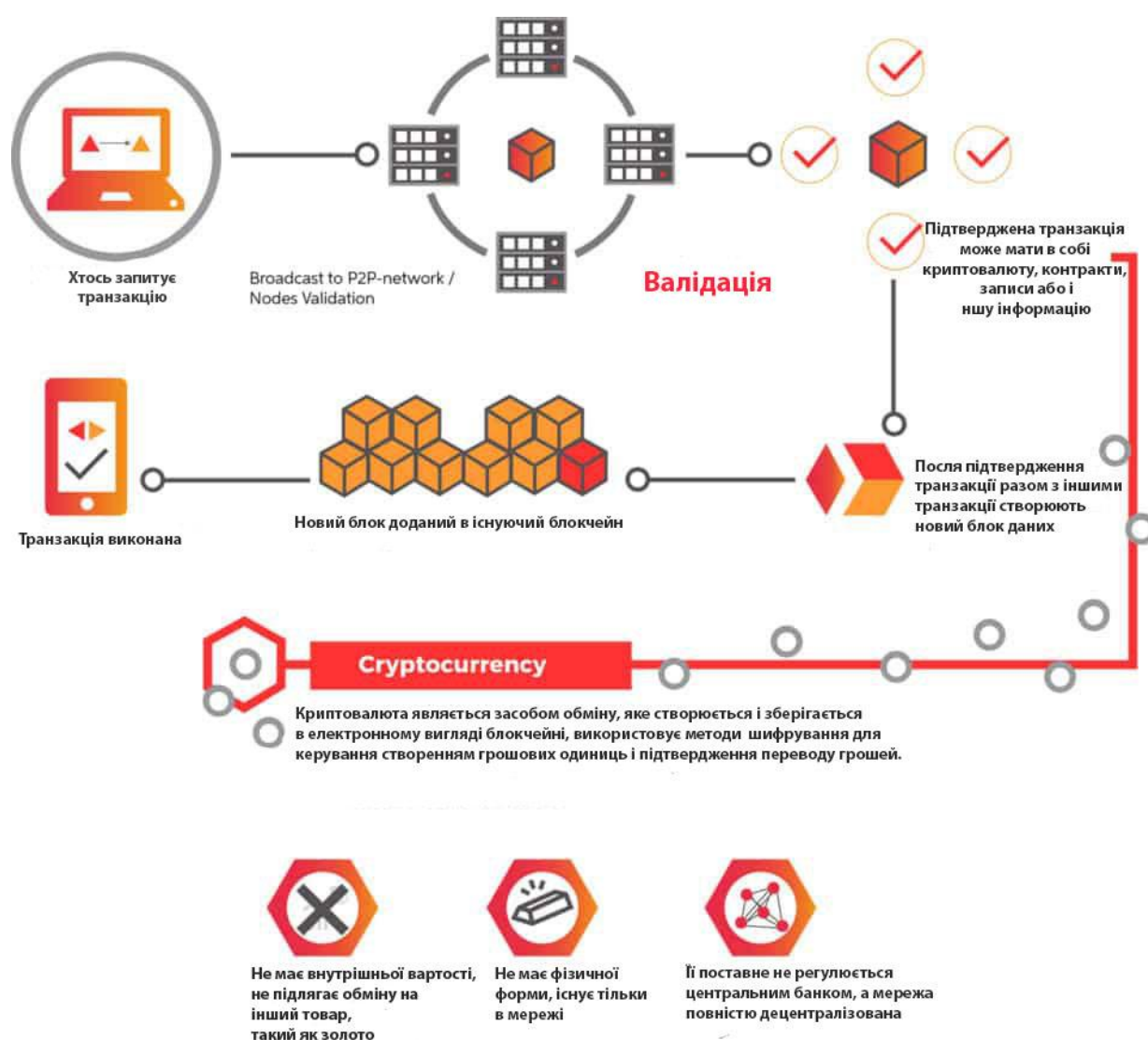


Рисунок 1.1 – Зображення роботи технології блокчейн

Застосування шифрування гарантує, що користувачі можуть змінити тільки ті частини ланцюжка блоків, якими вони володіють в тому сенсі, що у них є закриті ключі, без яких запис не можливий. Крім того, шифрування гарантує синхронізацію копій розподіленого ланцюжка блоків у всіх користувачів.

Кожен створений блок містить хеш попереднього. Таким чином створюється ланцюжок блоків, який бере початок від так званого genesis block до останнього знайденого системного блоку. Редагувати інформацію в блоці, яка знаходиться в ланцюгу все довготривалий час – не практично, так як в такому випадку так само прийшлося би редагувати інформацію в всіх попередніх блоках. Завдяки цим властивостям double-spending атака майже не можлива.

Блоки одночасно формуються безліччю «майнерів». Блоки, які задовольняють критерії, відправляються в мережу, включаючись у розподілену базу блоків. Регулярно виникають ситуації, коли кілька нових блоків в різних частинах розподіленої мережі називають попереднім один і той же блок, тобто ланцюжок блоків може розгалужуватися. Спеціально чи випадково можна обмежити ретрансляцію інформації про нові блоки (наприклад, один з ланцюжків може розвиватися в рамках локальної мережі). У цьому випадку можливе паралельне нарощування різних гілок.

У кожному з нових блоків можуть траплятися як однакові транзакції, так і різні, що входять тільки в один з них. Коли ретрансляція блоків поновлюється, майнери починають вважати головним ланцюжок з урахуванням рівня складності хешу і довжини ланцюжка. При рівності складності і довжини перевага віддається тому ланцюжку, кінцевий блок якого з'явився раніше. Транзакції, що увійшли тільки у відхилену гілку (в тому числі з виплати винагороди), втрачають статус підтверджених.

Якщо це операції з передачі біткоїнів, то вона буде поставлена в чергу і потім включена в черговий блок. Транзакції отримання винагороди за створення відсічених блоків не дублюються в іншій гілці, тобто «зайві» біткоїни, виплачені за формування відсічених блоків, не отримують подальших підтверджень і «втрачаються».

Розподілена база даних Blockchain формується як безперервно зростаючий ланцюжок блоків з записами про всі транзакції. Копія бази даних або її частини одночасно зберігаються на безлічі комп'ютерів та синхронізуються відповідно до формальних правил побудови ланцюжка блоків. Інформація в блоках не шифрована і доступна у відкритому вигляді, однак захищена від змін криптографічно через хеш-ланцюжок[4].

Принцип роботи технології блокчейн можна порівняти з роботою мережі Інтернет. Ви розміщуєте в мережі будь-яку інформацію, і користувачі Інтернету можуть отримати доступ до неї незалежно від місця свого знаходження. Так і тут: інформацію, яка міститься в блоці ланцюга, можуть отримати всі користувачі мережі, які мають до неї доступ, в будь-якій точці світу. Щоб отримати доступ до цієї інформації, ви повинні мати спеціальний закритий ключ, який створений відповідно криптографічному алгоритму. Це робить зберігання даних в ланцюгу блокчейн захищеним і повністю безпечним.

Для того, щоб зберегти дані, як було сказано вище необхідно провести транзакцію, тобто перевести кошти з однієї адреси на іншу, або заплатити мінімальну комісію і відправити транзакцію просто в мережу. Після цього вона буде проведена на визначену мережею адресу. Транзакція потрапить в блок і після того як її добудуть залишиться в мережі блокчейн, в одному із ланцюгів, які майже не можливо підмінити.

Через високий поріг захищеності технології її вигідно використовувати для збереження будь яких даних, які можуть бути скомпрометовані при типовому підході, наприклад використання документо-орієнтованої бази даних MongoDB. Для зміни даних в звичайні базі вам необхідно всього лиш отримати до неї доступ, в самому банальному і поширеному варіанті - це логін і пароль. Після цього ви можете змінювати будь-які дані. Для того, щоб це зробити з мережею блокчейн, вам необхідно підмінити 51% блоків, що при поточних потужностях просто фізично не можливо.

Як результат вибір технології блокчейн як засобу для збереження даних в сучасних реаліях є актуальним і швидко набирає обертів в безлічі країн по всьому світу.

1.4 Аналіз інструменту для створення мобільних-застосунків

Швидкий розвиток ринку мобільних-застосунків викликає в свою чергу великий асортимент інструментів для їх створення. Інструменти можуть існують як виключно для конкретної платформи так і кросплатформені.

Для розробки під операційну систему андроїд існують такі інструменти, як Java та Kotlin. Що стосується IOS використовуються Swift та ObjectiveC.

Кросплатформена розробка являється досить складною. Інструменти, що використовуються є досить новими, і це викликає певні труднощі під час розробки. З відомих являється Cordova. Це інструмент для розробки додатків написаний на JavaScript, який дозволяє розробляти застосунки на операційні платформи IOS та Android. Також для кросплатформеної розробки використовують ReactNative, який дозволяє розробляти застосунки на дві платформи з використанням одного коду.

Для розробки застосунків під платформу Android сьогодні є використовується мова програмування Kotlin. Kotlin (Котлін) — статично типізована мова програмування, що працює поверх JVM і розробляється компанією JetBrains. Також компілюється в JavaScript[6].

Автори ставили перед собою ціль створити лаконічнішу мову, ніж Java, і простішу, ніж Scala. Наслідками спрощення, порівняно з Scala стали також швидша компіляція та краща підтримка IDE.

Kotlin чудово підходить для розробки додатків Android, що приносить всі переваги сучасної мови для платформи Android без введення нових обмежень:

- сумісність: Kotlin повністю сумісна з JDK 6, тож додатки Kotlin можуть працювати на старих пристроях Android без проблем. Інструмент Kotlin повністю підтримується в AndroidStudio і сумісний з системою Android build;

- продуктивність: додаток на Kotlin працює так само швидко, як еквівалентний Java, завдяки дуже схожим структурам байт-кодів. За підтримки Kotlin для вбудованих функцій кодування за допомогою лямбда часто виконується навіть швидше, ніж той же код, написаний на Java;

- оперативна сумісність: котлін є 100% сумісним з Java, що дозволяє використовувати всі існуючі бібліотеки Android у програмі Kotlin. Це включає в себе обробку анотацій, таким чином, також доступні прив'язка даних та Dagger;

- час компіляції: котлін підтримує ефективну комбіновану поетапну компіляцію, так що при наявності додаткових накладних витрат на чисті збірки додаткові збірки зазвичай стають швидкими та швидшими, ніж з Java.

У підсумку: простота дозволяє використовувати мову майже будь-якому Java-розробнику, який готовий витратити півгодини на те, щоб переглянути інструкції або специфікацію мови, зворотна сумісність дозволяє використовувати мову у вже існуючому проекті.

У випадку з розробкою під операційну систему IOS зазвичай використовується мова програмування Swift. Swift — багатопарадигмова компільована мова програмування, розроблена компанією Apple для того, щоб співіснувати з Objective-C і бути стійкішою до помилкового коду. Основний філософією мови є його простота, швидкодія і захищеність. Поріг входження змінився в більшу сторону, в порівнянні з тим же Objective C - навіть не знаючи будь-яких інших мов програмування або основ, можна сміливо приступати до вивчення мови[8].

У цієї сучасної й потужної мови програмування є ряд переваг, які виділяють його на тлі інших. І це нові потужні можливості мови, гранично сувора типізація даних, більш лаконічний синтаксис, істотно більш висока продуктивність в операціях доступу до колекцій (сортування і пошук), підтримка переважної більшості сучасних синтаксичних конструкцій, що існують в інших високорівневих мовах програмування. І однією з головних особливостей Swift є те, що він має повну зворотну сумісність з Objective-C. Таким чином це дозволяє поступово

переписувати проекти з однієї мови на іншу, до того ж дає можливість використовувати функціонал Objective-C всередині самого Swift.

Наступним інструментом є ReactNative. React Native - це фреймворк для розробки кроссплатформених додатків. Він дає можливість створювати і використовувати компоненти так саме, як зазвичай ми це робимо в React, ось тільки рендеритись вони будуть не в HTML, а в нативні контроли операційної системи, під яку буде зібрано наш додаток [9].

Переваги ReactNative:

- перший і найголовніший плюс: використовуючи технологію React Native, ви платите за два додатки за ціною одного;
- загальний набір інструментів для web і mobile. Ви можете користуватись шаблоном для створення web-застосунків на React JS. Принципової різниці в розробці немає. Залишається тільки змінити шлях установки і роутинг додатку за допомогою навігатора React Native;
- один з головних плюсів React Native. Підтримка Flexbox «з коробки» робить написання UI менш болючим в порівнянні з імперативним стилем нативної платформи[10].

1.5 Порівняльний аналіз аналогів

Так як технологія блокчейн тільки набирає оборотів розвитку в різних спеціалізаціях, досить мало рішень які дозволяють виконати грошові перекази. Розглянемо деякі з них.

BitPay – постачальник платіжних послуг Bitcoin, з дуже мінімалістичним дизайном[25]. Основний екран застосунку зображений на рисунку 1.2.

Основні переваги BitPay:

- інтуїтивно зрозуміла робота з застосунком;
- можливість вибирати швидкість проведення транзакції, що безпосередньо дає змогу регулювати комісію;

- вмонтований перелік онлайн ресурсів де можна здійснювати покупки через даний застосунок.

Основні недоліки BitPay:

- часткова локалізація застосунку;
- погана оптимізація застосунку, при користуванні застосунком часто зависав або припиняв роботу;
- розрахований лише для обміну токенами Bitcoin.

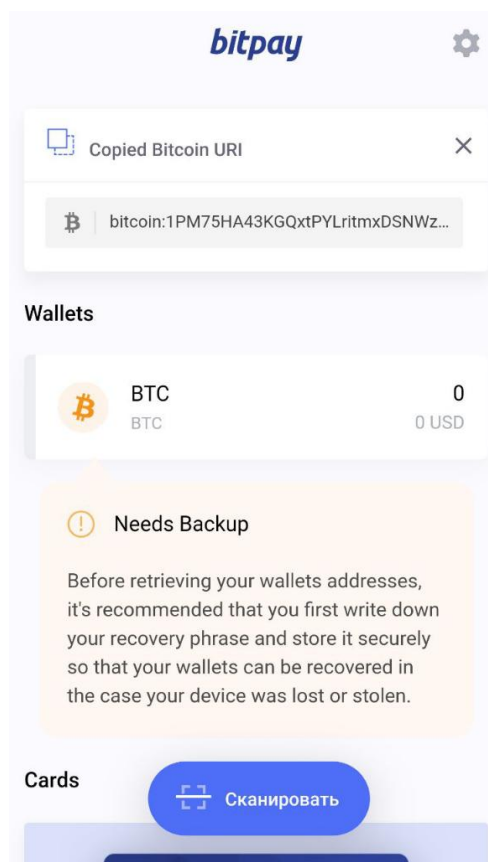


Рисунок 1.2 – Зображення роботи мобільного-застосунку “BitPay”

Electrum - застосунок для виконання обміну криптовалютою. Тут ви можете згенерувати собі гаманець для подальшого використання[26]. Екран з основним функціоналом Electrum зображено на рисунку 1.3.

Основні перевагиElectrum:

- при створенні гаманця використовується seedфраза що забезпечує високу безпеку.

Основні недоліки BitPay:

- незручний UI, дані розкидані на різні екрани що погіршує зручність та ефективність використання застосунку;
- без додаткової інструкції з додатком розібратись доволі важко;
- розрахований лише для обміну токенами Bitcoin;
- відсутня можливість регулювати швидкість виконання транзакцій, що обмежує користувача в можливості заощадити кошти у вигляді зменшення комісії;

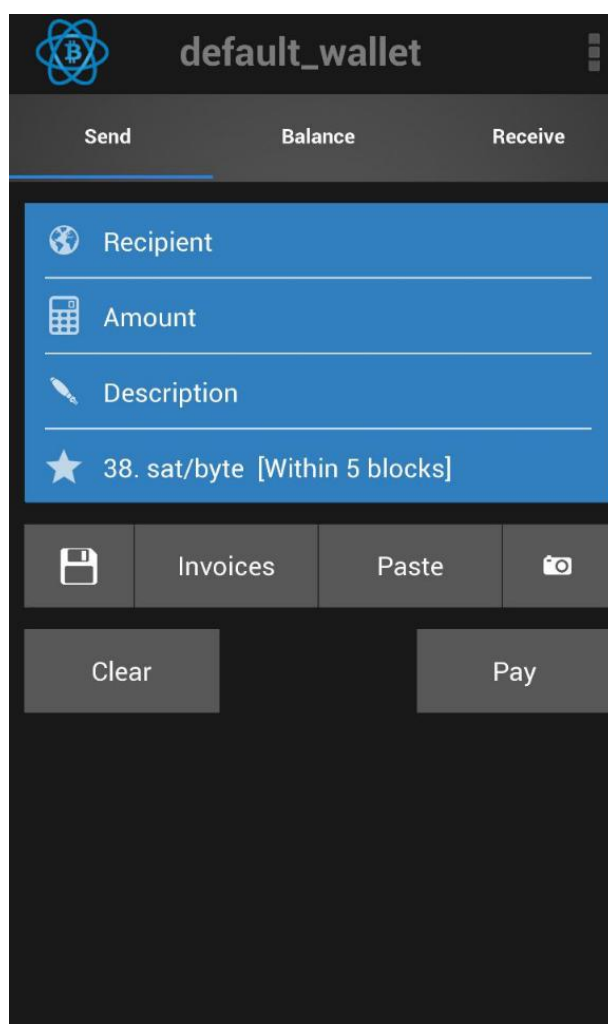


Рисунок 1.3 – Зображення роботи мобільного-застосунку “Electrum ”

Тому, проаналізувавши відомі аналоги програмного продукту, було вирішено розробити власний мобільний-додаток, в якому повинно підтримуватися основні функції аналогів та з максимально можливим усуненням їх недоліків.

1.6 Постановка задач розробки

Для розробки мобільного-додатку в рамках виконання магістерської дипломної роботи необхідно виконати наступні задачі:

- аналіз стану проблеми і постановка задач;
- аналіз інструментів для розробки мобільних-застосунків;
- аналіз технології блокчейн як засобу зберігання даних;
- розробка структури мобільного-сервісу;
- розробка візуальної частини мобільного-додатку;
- розробка алгоритму обміну токенами блокчейну ETH;
- тестування роботи мобільного-сервісу.

1.7 Висновки

У результаті проведеного аналізу було визначено, що технологія блокчейн та потреба людей обмінюватись цифровими активами має великий попит і аудиторію, що в свою чергу призводить до створення нових сервісів та удосконалення існуючих.

Також було розглянуто процес створення мобільних-додатків та проаналізовано технологію блокчейн і визначено, що використання кросплатформених інструментів для розробки мобільних-додатків є ефективним у процесі виконання такого роду завдань.

Так як обміну фінансовими активами у мережі існує, було визначено, що створення мобільного-додатку для обміну цифровими активами є актуальним. У зв'язку з цим було проведено порівняльний аналіз аналогів.

2 РОЗРОБКА СТРУКТУРИ ТА ДИЗАЙНУ МОБІЛЬНОГО-ДОДАТКУ

2.1 Вибір методів відображення інформації

Інформаційним забезпеченням додатку є п'ять екранів, які містять відображення інформації різної ступені складності, для того, щоб користувач міг максимально зрозуміти, для чого необхідний цей сервіс. Так сторінки містять наступні елементи UI представлення необхідної інформації:

- поле відображення балансу та назви рахунку;
- активні поля з назвою криптовалюти, курсом відносно фіатної валюти користувача та балансом;
- поле з адресою гаманця;
- блок історії транзакцій;
- блок отримання криптовалюти;
- блок відправлення криптовалюти;
- блок імпорту гаманця;
- блок контактів;
- блок створення нового гаманця.
- інформаційний блок щодо використання додатку.

В остаточному результаті контент буде представлятися у вигляді текстової та графічної інформації та анімації. Для кожного виду представлення потрібно розробити власну реалізацію, яка буде максимально чітко і просто давати користувачу можливість зрозуміти для чого йому потрібний конкретно цей сервіс.

Екран проведення транзакцій буде складною UI/UX частиною, тому що передбачає декілька етапів. Спочатку потрібно показати дані користувача, а саме адрес рахунку та баланс. При проведенні транзакції необхідно зробити анімацію, так як буде встановлюватися зв'язок з стороннім сервісом що може зайняти деякий час. Далі в залежності від того чи вдалось виконати транзакцію, чи ні необхідно показати відповідний блок успішного або невдалого виконання.

Список проведених транзакцій буде відображений у вигляді таблиці. Таблиця у цьому випадку підійде найкраще, так як даних в рядку буде небагато. Також необхідно передбачити відображення статусу транзакції в таблиці і поле, яке буде змінювати свій колір, в залежності того була транзакція успішна чи повернула помилку під час того як була добута.

Блок перегляду статусу транзакції буде знаходитись на сторінці з інформацією про конкретний гаманець і буде містити тільки текстову інформацію. В залежності від статусу транзакції буде відображатись будуть відображатись різні блоки.

Блок допомоги також буде містити тільки текстову інформацію. Надмірність анімацій у цьому випадку тільки зашкодить і заплутає користувача.

Щодо блоку контактів, він буде представляти собою невеликий опис можливих способів зв'язку з розробником.

Інформаційний блок буде містити найпоширеніші питання і відповіді у вигляді блоків, які можна буде згорнути при необхідності.

2.2 Розробка структури мобільного-додатку

Щоб створити мобільний-додаток потрібний план. Необхідно повністю описати, який зміст додатку і де буде знаходитись. План має назву розробка структури застосунку та малюється на папері. Структуру застосунку можна умовно розділити на зовнішню і внутрішню. Основними видами внутрішньої структури є лінійна, ієрархічна та мережна[27].

Лінійна структура – це найпростіша структура застосунків, в якій матеріал не поділяється на рівні. Всі екрани ієрархічно рівноправні. Такий застосунок є доволі простим у реалізації, але має обмежене застосування (рис. 2.1).

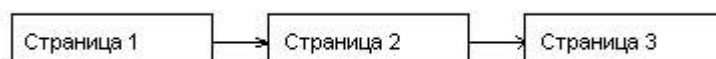


Рисунок 2.1 – Лінійна організація

Ієрархічна структура – це найбільш розповсюджений вид структури в сучасній розробці, цей варіант підходить для застосунків різних типів. Він передбачає застосування головної екрану, на якій розміщено меню з посиланнями на розділи застосунку, що розташовані на наступних екранах. Розділи можуть містити посилання на підрозділи або іншу детальну інформацію (рис. 2.2).

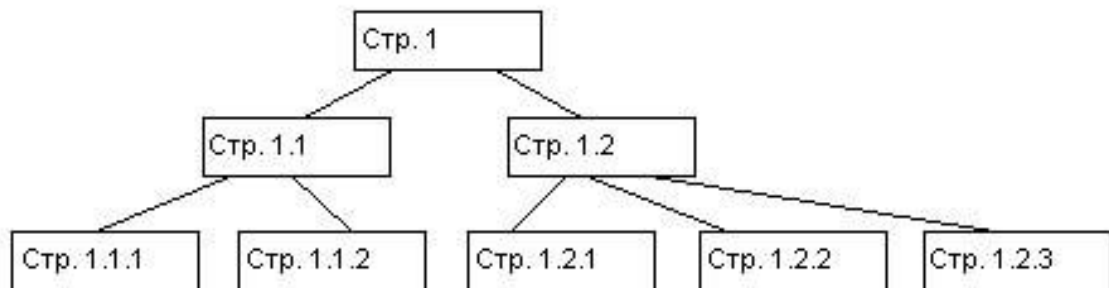


Рисунок 2.2 – Ієрархічна організація

На основі розділів додатку та взаємозв'язках між ними було обрано ієрархічну організацію, що забезпечить легке та звичне переміщення по додатку.

2.3 Розробка інтерфейсу екранів

Кожний екран додатку поділений на логічні блоки. Зверху розміщений хедер, з назвою гаманця та станом рахунку. Далі змінюється тільки контент в основній області, яка знаходиться посередині екрану. В залежності від того який екран вибрав користувач, основна область буде змінюватись. Нижня частина сайту, тобто футер, знаходиться в незвичному місці, а саме нижній частині основної області. Це дозволить зекономити місце, та не ламати UI структуру застосунку. Макет екрану зображено на рисунку 2.4.



Рисунок 2.4 – Макет екрану

На рисунку 2.5 зображено макет сторінки для вибору типу операцій що ми хочемо провести з рахунком та список транзакцій. У основній області знаходиться блок для вибору типу транзакцій та список вже проведених операцій.



Рисунок 2.5 – Макет для операцій з гаманцем

Після того як користувач вибрав тип транзакції, він переадресовується на безпосередньо екран з відповідною операцією (рис. 2.6).

The wireframe shows a rectangular frame containing several elements. At the top is a header box labeled 'Хедер'. Below it are two input fields: the first is labeled 'Адреса отримувача' and the second is labeled 'Сума переказу'. At the bottom center is a button labeled 'Кнопка відправити'.

Рисунок 2.6 – Макет сторінки відправлення фінансових активів

Екран допомоги користувачу (рис. 2.7) буде містити статистичну інформацію. Статистично відібрані питання будуть вибрані і на них дасть відповідь підтримка.

The wireframe shows a rectangular frame with a vertical stack of sections. It starts with a header box labeled 'Хедер'. This is followed by three identical pairs of boxes: the top box in each pair is labeled 'Питання' and the bottom box is labeled 'Відповідь'. At the very bottom is a footer box labeled 'Футер'.

Рисунок 2.7 – Макет сторінки затвердженого файлу

2.4 Розробка дизайну мобільного-застосунку

Основи дизайну мобільного-додатку засновуються на етапі розробки технічного завдання – документа, який містить в собі опис вимог стосовно структури додатку та візуального представлення. У технічному завданні також описуються побажання стосовно програмної реалізації застосунку.

Після розробки та затвердження техзавдання починається етап розробки дизайну застосунку. Першочергово при розробці необхідно створити модульну сітку додатку. Модульна сітка описує розташування елементів на екранах додатку.

Модульна сітка будь-якого екрану містить, як мінімум, два блоки: для меню і для основного тексту додатку. Якщо на сторінці необхідно розмістити додаткові елементи, наприклад, ще одне меню, «підвал» («footer») або «хедер» («header»), то модульну сітку розбивається на ще декілька блоків [28].

На практиці модульна сітка використовується при розробці дизайну візиток, книг, логотипів і схожих речей. Модульна сітка обов'язково повинна використовуватися і при розробці мобільних-застосунків.

За допомогою добре розробленої сітки можна зручно скомпонувати середу для всіх складових застосунку, починаючи від текстових закінчуючи графічними елементами. З використанням модульної сітки значно простіше розмістити в додатку різні елементи. Розташуємо певним чином різні елементи, можна зробити той чи інший з них більш помітним і при цьому пов'язати в одне ціле інформацію про додаток [29].

Модульна сітка, яка найчастіше використовується в дизайні сторінок в інтернеті – це три основні колонки осередки. Верхній і нижній колонтитули при цьому мають ширину, яка дорівнює їх розмірами. Такий варіант зазвичай при створенні сайтів використовують любителі. У деяких випадках ширина одного осередку може бути збільшена за рахунок іншої. Другий варіант модульних сіток - нестандартний. Він під силу тільки тим користувачам, які професійно займаються мобільним-дизайном. Точно в цьому випадку визначається тільки відстань між окремими елементами і ширина колонок. У дизайні інтернет-журналів і різного роду

новинних ресурсів модульна сітка зазвичай розробляється таким способом. Спершу створюється колонтитул.

У ньому може розташовуватися назва ресурсу, навігаційне меню і різні відеоматеріали. Як правило, сторінка розбивається на три колонки. Дві при цьому мають однаковий розмір. Третя колонка розділена на дві частини за допомогою вертикальної лінії. В результаті виходить чотири розділи: фотогалерея, новинний розділ, місце для реклами і останні інформаційні зміни. Для інтернет-магазинів різної спеціалізації модульна сітка створюється трохи інакше. Тут важливо забезпечити відвідувачам зручність навігації. Всі елементи при цьому групуються одноманітно і стандартно.

У блоці «хедер» прийнято вказувати назву сайту, яке може бути вибрано відповідно до назви компанії або ім'ям автора сайту, а також при необхідності логотип сайту. Основний блок меню в європейських сайтах традиційно розташований зліва, відповідно до напрямку писемності (зліва-направо). Також основне меню може бути розташоване зверху, під «хедером».

Коли інформаційна модель майбутнього додатку готова, можна братися за художню частину дизайну. Зазвичай дизайн виконується у вигляді шаблонів – набору елементів і зв'язків між ними. Саме шаблони дають змогу відокремити розробку візуального дизайну додатку від змісту.

Під час проектування мобільного-додатку потрібно правильно підібрати набір кольорів і врахувати можливість їх поєднання з іншими елементами дизайну при розробці застосунку. Перед вибором кольорів варто враховувати різні фактори і для цього необхідне розуміння кольорів і їх значення.

Чорний – колір, який найчастіше використовується для оформлення текстового контенту. Він підтримує витонченість, формальність і елегантність і може бути використаним на будь-якому сайті.

Синій з його королівським закликком і спокійним ефектом є хорошим кольором для мобільного-дизайну. Його поєднання з іншими кольорами, наприклад з червоним, жовтим, зеленим, надає бажаний ефект. Це найкращий колір для корпоративних сайтів.

Зелений – колір природи, росту, достатку і злагоди. Він є найкращим для рекламування природніх чи органічних товарів на сайті. Для сайтів про здоров'я доречним буде саме цей колір.

Коричневий – вважається земним, природнім, простим кольором. Він передає відчуття надійності, корисності і консерватизму. З іншої сторони він сприймається трохи скучно, але його можна комбінувати з більш яскравими кольорами. Цей колір також пов'язується з старовинними фотографіями, матеріалами і може застосовуватися для навіювання ностальгії.

Фіолетовий зазвичай асоціюється з таємницею і духовністю, а глибокий фіолетовий – з процвітаням, багатством, роялті і благородством. Досить сильний колір, який в обережному використанні може забезпечити хороший контакт на сайт.

Під час вибору кольорової гами часто використовують різні кольорові схеми. Вони допомагають швидко підібрати необхідний головний колір та відповідні йому відтінки. Однією з таких схем, яка і була використана при розробці, являється кольорова схема у вигляді кола. Використовувались монохромні і аналогічні кольори.

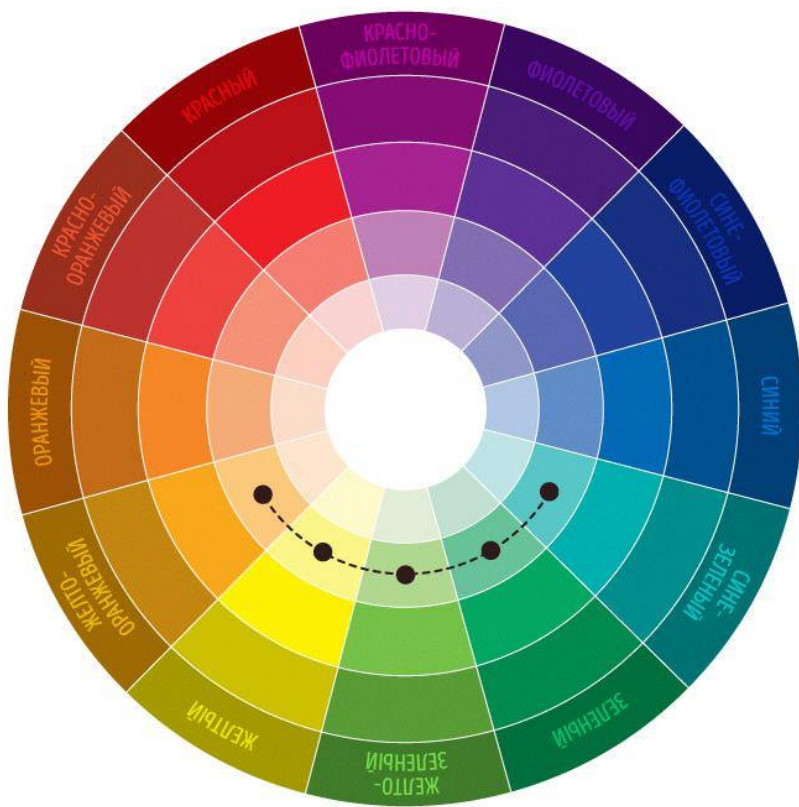


Рисунок 2.8 – Кольорова схема

Тематика розроблюваного застосунку пов'язана загалом з обміном фінансовими, тому можна обмежитись створенням строгого, простого дизайну та використанням простих анімацій, які не будуть відволікати від головної тематики додатку. Кольорова гама повинна була підібрана у світло зелених та синіх кольорах[30].

Система навігації – важлива частина дизайну будь-якого додатку.

За функціями, які виконує система навігації:

- вибір мов - мовна навігація додатку. За допомогою цього виду навігації по додатку відвідувач може вибрати, який з можливих мов відображення він бажає використовувати;
- основні важливі розділи додатку або його сторінок розміщуються в основній навігації додатку (наприклад, основне меню);
- глобальна навігація додатку містить у собі посилання на екрани, які повинні бути доступні з будь-якого додатку. Наприклад, посилання на екран налаштувань;
- для того, щоб привернути увагу відвідувачів до певних екранів застосунку, на яких, наприклад, розміщуються рекламні товари або послуги, існує рекламна навігація сайту;
- вказівна(довідкова) навігація сайту дозволяє відвідувачеві визначити, у якому місці додатку він у цей момент перебуває;

За видом візуального оформлення:

- текст – текстова навігація по додатку найбільш "давній" вид навігації, використовується в основному для навігації в тексті, рідше – для інших функцій;
- графіка – графічна навігація застосовується для всіх функціональних видів навігації додатку;

Наступним складним елементом дизайну є блок операцій з гаманцем, тому що він повинен крім змоги вибору типу транзакції відображати два стани операції, а саме успішне виконання, відображення інформації, дані транзакції, та блок

сповіщення, що транзакція була проведена. В залежності від цього необхідно розробити різний дизайн. В даному випадку було вирішено, змінити колір блоків та текст в залежності від результатів (рис. 2.11).



Рисунок 2.11 – Зображення станів блоку

Дані про транзакції будуть представлятись у вигляді стандартної таблиці, з однією відмінністю – буде можливість пересувати дані по горизонталі, бо даних досить багато (рис. 2.12). Також в таблицях буде відображатись статус.

File hash	Status	Registered on our servers	Registered in Ethereum blo
513658eb0f453ec6a80ffd2b3a51592a42...	success	2018-12-24 12:26:48	2018-12-24 12:
f8266d91d6af126d45263edce678f84929...	success	2018-12-19 16:50:40	2018-12-19 16:
e02e38e0a9ba54f47f3a1db43c47161f04...	success	2018-12-19 16:04:22	2018-12-19 16:
41d8f173daad30a73e3e1c89120620fa52...	success	2018-12-12 15:10:59	2018-12-12 15:
2f2c6ab832d7848bffebe7d423c41c543c8...	success	2018-12-06 17:18:53	2018-12-06 17:
eaf599fd44df486b8ff4f37c972f4c348032...	success	2018-11-24 19:57:17	2018-11-24 19:
64b24d9b77cb501be5d95b894dce2b247...	success	2018-11-21 13:16:17	2018-11-21 13:
b9c390fe2753cf3cac3771ea7a52a8cb4c6...	success	2018-11-21 12:18:53	2018-11-21 12:
435b39fc59cdb96df0571929e030eaffdbf...	success	2018-11-06 09:07:43	2018-11-06 09:

Рисунок 2.12 – Зображення списку транзакцій

2.5 Розробка структури програмної системи обміну токенами блокченку ЕТН

Основним складовими програмної системи є модулі. Всі види модулів у будь-якій мові програмування мають певні загальні властивості.

Модуль складається з великої кількості операторів мови програмування, записаних послідовно; модуль має назву, за якою до нього можна звертатися як до суцільного фрагмента; модуль може приймати й/або передавати дані як параметри в зухвалій послідовності або зв'язувати дані через фіксовані осередки або загальні області[31].

Структурні карти Константайна є моделлю відносин ієрархії між програмними модулями. Вузли структурних карт відповідають модулям і областям даних, потоки зображують міжмодульні виклики. При цьому циклічні й умовні виклики модулів моделюються спеціальними вузлами, тому потоки повинні бути зображені минаючими через ці спеціальні вузли. Міжмодульні зв'язки з даними й керування також моделюються спеціальними вузлами, прив'язаними до потоків (тобто до викликів модулів), стрілками вказуються напрямки потоків і зв'язків. На першому етапі побудови було визначено вузли структурної схеми: модуль взаємодії, модуль відображення, модуль розрахунку комісії та локальні дані. На другому етапі було визначено відношення між модулями і побудовано структурну схему, яка представлена на рисунку 2.14.

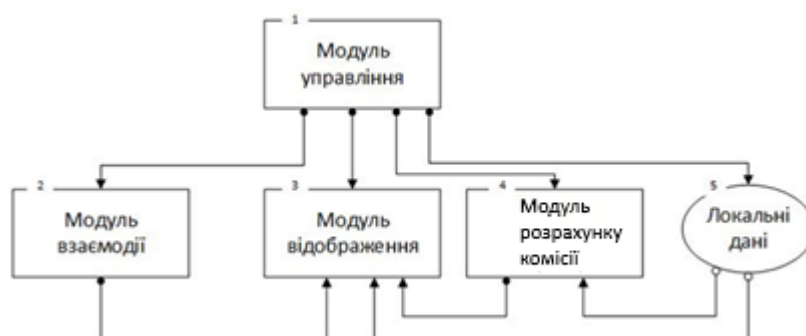


Рисунок 2.14 – Структурна карта Константайна

Розроблена структурна карта Константайна є відображенням моделі відношень ієрархічних зв'язків між програмними модулями (модуль взаємодії,

модуль відображення, модуль розрахунку комісії та локальні дані) мобільного-сервісу.

2.6 Розробка алгоритму обміну токенами блокченку ЕТН

В даний час створення алгоритмів – написання програм для електронних обчислювальних машин – стало видом людської діяльності. Найважливіший конструктивний компонент програмування, що не залежить від особливостей синтаксису програмної системи мов програмування та специфіки функціонування конкретних обчислювальних машин, – розробка алгоритму.

Підходи до створення алгоритмів і вимоги до них істотно змінювалися в ході еволюції комп'ютерів, але і зараз вони є актуальними. Спочатку, в епоху ЕОМ 1 - го і 2-го поколінь, коли вони були ще мало поширені, машинний час було дорогим, а можливості ЕОМ дуже скромні (з точки зору сьогоденних досягнень), основною вимогою до алгоритму була його вузько розуміється ефективність.

Модуль обрахунку комісії та обміну токенами є головним і найскладнішим в реалізації та потребує знань не тільки в області створення мобільних сервісів, а і застосуванні технології блокчейн при розробці мобільних додатків.

Алгоритм роботи додатку, який представлений на рисунку 2.13, складається з наступних кроків:

- початок;
- ініціалізація мобільного сервісу;
- зчитування вибраних параметрів на стороні клієнта;
- надсилання транзакції;
- перевірка статусу транзакції;
- оновлення даних по транзакції, після того як вона була добавлена в блок;
- кінець.



Рисунок 2.13 – Блок-схема алгоритму роботи додатку

2.7 Висновки

Для опису структури мобільного-додатку було визначено організацію екранів, та вибрано ієрархічну структуру. Також було визначено головні модулі гарантія та відношення між ними. Для наочного зображення даної структури було побудовано структурну схему Константайна.

Розроблено алгоритм обміну токенами блокчейну ETC та а побудовано блок-схему даного алгоритму.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО-ДОДАТКУ

3.1 Обґрунтування інструментів для розробки

У розділі 1.4 було систематизовано головну інформацію про поширені інструменти для мобільної розробки, такі як ReactNative, Kotlin таSwift. Також було зроблено висновок про їх застосування при розробці додатків для різноманітних платформ.

Серед наведених у розділі 1.4інструментів, ReactNativeнайбільш вигідний для розробки мобільних додатків, так як робить можливою кросплатформену розробку. Також ReactNativeмає дуже детальну документацію з усіх доступних функцій, які можна використовувати при розробці власного додатку.

Однією із важливих особливостей створення мобільних-додатків являється використання нативних мов та застосунків, тобто засобів розробки, які спеціально створюються для цієї платформи, в даному випадку мобільних телефонів. Використання ReactNativeробить роботу з мобільним застосунком схожою до роботи з браузером, дозволяє сильно не навантажувати процесор під час виконання безлічі завдань, що в свою чергу зменшує навантаження на центральний процесор.

Складною річчю при розробці мобільних-додатків являється можливість асинхронного програмування. Асинхронне програмування передбачає виконання одразу декількох операцій, тобто якщо відображається один об'єкт на екрані, паралельно може відображатись рух іншого. Іншими словами, мобільному не потрібно чекати доки закінчиться попередня операція, щоб почати наступну.Ця проблема виникла при необхідності відобразити безліч інформації на сторінці. Вони приходять від сервера асинхронно і ReactNative використовує асинхронністьна відміну від конкурентів.

Для поставлених цілей можна було б використовувати Kotlin, але використання написаної професіоналами бібліотеки цієї мови програмування значно полегшує процес розробки будь-якого мобільного-застосунку. Замість десяти

стрічок коду вам потрібно написати одну, яку ви зможете зрозуміти навіть після декількох місяців роботи над проектом.

Також використання ReactNative дає можливість розбити ваш продукт на основні модулі. Для кожного з них вже створені класи та методи роботи, будь-то підключення до серверу по сокетам, чи звичайне підключення DOM елементів.

Проаналізувавши проблеми, які виникли під час проектування мобільного-сервісу, поставлені цілі та переваги бібліотеки ReactNative було вирішено у ході виконання магістерської дипломної роботи використовувати саме цей інструмент для розробки мобільних-додатків.

3.2 Обґрунтування вибору мови та середовища програмування

Відповідно до поставленого завдання, потрібно розробити мобільний-додаток з використанням мови розмітки сторінок, каскадних таблиць стилів мови програмування java-script та його фреймворку. Розробка буде проводитись у ручному режимі. Перевагами використання цього режиму є хороша оптимізація, гнучкість та великий діапазон дій при створенні. Ці переваги лежать в основі якісного мобільного-продукту. З недоліків можна виділити довший процес розробки, але, знову ж таки, він компенсується якістю. Елементи дизайну можна буде змінити в будь-який момент з низькою витратою часу, так як кожен модуль пишеться вручну і не залежить один від одного. Код є відкритим, тому доступний для оптимізації, що добре відобразиться на швидкості роботи [32].

Використання мов програмування, стилів та мови розмітки створює всі можливості для реалізації власного дизайну. Написання коду з використанням методологій «Яндекс», надає можливість розробляти будь-який якісний мобільний-додаток у звичайному блокноті, але використання спеціалізованого програмного забезпечення є необхідним, так як воно має безліч різних розширень, які автоматизують рутинну роботу і прискорюють процес написання коду. Також

Існують інтернет портали, де можна знайти відповіді майже на всі питання, які виникають в ході розробки, що також полегшує роботу.

Середовищем для розробки було вибрано PHPStorm від компанії JetBrains. Доступна ліцензія на використання для студентів. Функціонал цього програмного забезпечення перевищує в декілька разів можливості звичайного блокноту і, звичайно, його досить для розробки мобільного-додатку [33]. До основних функцій PHPStorm входять такі:

- автодоповнення;
- контроль версій;
- система пошуку та виправлення помилок;
- підтримка найновіших стандартів і технологій розробки;
- навігація по коду;
- відлагодження.

JSX – це препроцесор, який виявляє синтаксис HTML до JavaScript тобто можна реалізувати будь-яку DOM структуру, використовуючи при цьому JavaScript. Всі дані, які відображаються знаходяться безпосередньо в документо-орієнтованій моделі.

CSS використовується авторами та відвідувачами веб-сторінок, щоб визначити кольори, шрифти, верстку та інші аспекти вигляду сторінки. Одна з головних переваг – можливість розділити зміст сторінки (або контент, наповнення, зазвичай HTML, XML або подібна мова розмітки) від вигляду документу (що описується в CSS) [40].

Таке розділення може покращити сприйняття та доступність контенту, забезпечити більшу гнучкість та контроль за відображенням контенту в різних умовах, зробити контент більш структурованим та простим, прибрати повтори тощо. CSS також дозволяє адаптувати контент до різних умов відображення (на екрані монітора, мобільного пристрою, у роздрукованому вигляді, на екрані телевізора, пристроях з підтримкою шрифту Брайля або голосових браузерів та ін.).

JavaScript (JS) – динамічна, об'єктно-орієнтована мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується як

частина браузера, що надає можливість коду на стороні клієнта (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати додатком, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд сторінки додатку.

Враховуючи переваги і недоліки використання конструкторів та засобів ручного створення сайтів, для розробки додатку було вирішено використати середовище розробки PHPStorm та засоби технологій JSX і JavaScript.

3.3 Програмна реалізація мобільного-додатку

Для анімації появи та приховування модального вікна з інформацією про успішну транзакцію було використано базовий компонент `ReactNative`, так як він зменшує кількість коду та його застосування є практичнішим. Керування показом та прихованням модального вікна відбувається за допомогою стану компоненту. При натисненні кнопки відправлення транзакції виконується код показаний на рисунку 3.1.

```

import React, {Component} from 'react';
import {Modal, Text, TouchableHighlight, View, Alert} from 'react-native';

export class ModalSend extends Component {
  state = {
    modalVisible: false,
  };

  setModalVisible(visible) {
    this.setState( state: {modalVisible: visible});
  }

  render() {
    return (
      <View style={{marginTop: 22}}>
        <Modal
          animationType="slide"
          transparent={false}
          visible={this.state.modalVisible}
          onRequestClose={() => {
          }}>
          <View style={{marginTop: 22}}>
            <View>
              <Text>Транзакцію відправлено</Text>

              <TouchableHighlight
                onPress={() => {
                  this.setModalVisible(!this.state.modalVisible);
                }}>
                <Text>Ок</Text>
              </TouchableHighlight>
            </View>
          </View>
        </Modal>
      </View>
    );
  }
}

```

Рисунок 3.1 – Код модального вікна

Навігація додатком відбувається за допомогою модуля ReactNavigation. Код для навігації приведений на рисунку 3.2.

```

render() {
  if (this.state.isLoading) {
    return <WelcomeScreen />;
  }
  return (
    <View style={{ flex: 1}}>
      <AppContainer balance={this.state.balance}/>
    </View>
  );
}

const StackNavi = createStackNavigator({
  routeConfigMap: {
    Home: {
      screen: HomeScreen,
      navigationOptions: { header: null}
    },
    Wallet: {
      screen: WalletScreen,
      navigationOptions: { header: null}
    },
    Send: {
      screen: SendScreen,
      navigationOptions: { header: null}
    },
    Get: {
      screen: GetScreen,
      navigationOptions: { header: null}
    },
  },
  stackConfig: {
    initialRouteName: 'Home',
  },
});

```

Рисунок 3.2 – Код для навігації

Для реалізації обміну цифровими активами у мережі блокчейн, необхідно реалізувати досить складний процес. Першим етапом є збирання транзакції. Вибір адреси відправника береться з глобального вмістилища даних в якому знаходяться данні про гаманець користувача. Далі коли користувач знаходиться на відповідній сторінці йому необхідно ввести дані про транзакцію, а саме адресу отримувача та суму переказу. Коли користувач ввів всі необхідні дані вони передаються в наш скрипт, який збирає тіло транзакції.

Після збирання тіла транзакції, відбувається відсилання транзакції в мережу (рис. 3.3).

```

web3.eth.getTransactionCount("0x831F09e124Ee21634E8D3D9d7B51231c46F7dfa5", "pending")
  .then((result) => {
    web3.eth.accounts.signTransaction({
      nonce: result,
      to: '0xD497e2C79564A444C9E2f9E53a821790A70EdF22',
      gasLimit: 1000000,
      data: `0x${inputData.fileHash}`
    }, '0x' + key)
    .then((signData)=>{
      console.log(signData);
      web3.eth.sendSignedTransaction(
        signData.rawTransaction
      )
      .on('transactionHash', (transHesh) => {
        tmpTransHesh = transHesh;

        let data = {
          txHash: transHesh,
          fileHash: inputData.fileHash,
          status: 'pending'
        };

        createTransaction(data, (txHash, id) => {
          let dataToDispatch = {
            id,
            txHash,
            status: 'success'
          };

          successCallback(dataToDispatch);
        });
      })
    .on('receipt', (res) => {
      console.log('*****');
      console.log('Transaction receipt');
      console.log(tmpTransHesh);
      console.log('*****');
      updateTransaction({ txHash: res.transactionHash });
    })
  })

```

Рисунок 3.3 – Код відправки транзакції

Код посилення транзакції досить складний і передбачає декілька івентів: транзакція була добавлена в блок, транзакція скоро вирахується, транзакція прорахована. Після того як транзакція виконана, буде додано дані про транзакцію на сторінці з історією транзакції (рис. 3.4).

```

render() {

  let accountLinkFirstPart = accountLink.substr( from: 0, length: 4)
  let accountLinkSecondPart = accountLink.substr( from: 38, length: 4)

  return (

    <View style={styles.history_container}>
      <View style={styles.history_data_container}>
        <View style={styles.history_from}>
          <Text style={styles.history_from_text}>{accountLinkFirstPart}...{accountLinkSecondPart}</Text>
        </View>
        <View style={styles.history_type}>
          <Text style={styles.history_type_text}> {this.state.typeofTransaction} </Text>
        </View>
        <View style={styles.history_value}>
          <Text style={styles.history_value_text}> {this.state.value}</Text>
        </View>
      </View>
    </View>
  );
}

```

Рисунок 3.4 – Код компоненту з історією транзакцій

Отримання історії транзакцій є також досить складним процесом, так як передбачає у собі спілкування з мережею блокчейн(рис. 3.5).

```

let myAddr = '0x6b800539925A6a98C27189EA97E4af59A273fdB6';
let currentBlock = eth.blockNumber;
let n = eth.getTransactionCount(myAddr, currentBlock);
let bal = eth.getBalance(myAddr, currentBlock);
for (let i=currentBlock; i >= 0 && (n > 0 || bal > 0); --i) {
  try {
    let block = eth.getBlock(i, true);
    if (block && block.transactions) {
      block.transactions.forEach(function(e) {
        if (myAddr == e.from) {
          if (e.from != e.to)
            bal = bal.plus(e.value);
          console.log(i, e.from, e.to, e.value.toString(10));
          --n;
        }
        if (myAddr == e.to) {
          if (e.from != e.to)
            bal = bal.minus(e.value);
          console.log(i, e.from, e.to, e.value.toString(10));
        }
      });
    }
  } catch (e) { console.error("Error in block " + i, e); }
}

```

Рисунок 3.5 – Код отримання історії транзакцій

При створенні облікового запису користувача йому пропонується використати пароль який генерується за допомогою seedфрази, що дозволить покращити захист персональних даних користувача. Для створення мнемонічної фрази було встановлено модуль bip39 та описано функцію створення паролю (рис. 3.6).

```
export function setPassword() {  
  
  let password = bip39.generateMnemonic()  
  
}
```

Рисунок 3.6 – Код створення паролю

Розбиття всіх сторінок додатку на логічні блоки здійснено за допомогою засобів JSX. Для забезпечення однакового відображення стилів і шрифтів у різних смартфонів, було використано набір FlexBox. Для реалізації основного функціоналу затвердження файлу використано ReactNative.

3.4 Висновки

Обґрунтовано використання інструменту ReactNative для розробки мобільного-додатку.

Проведено порівняльний аналіз середовищ розробки та мови програмування. Проаналізовано отримані дані та обрано мову програмування JavaScript. В якості інтегрованого середовища розробки було обрано RHPStorm.

Описано програмну реалізацію мобільного-додатку. Повний лістинг розроблених скриптів наведено в Додатку Д.

4 ТЕСТУВАННЯ МОБІЛЬНОГО-ДОДАТКУ

4.1 Аналіз методик тестування

Тестування додатку дає змогу розробнику знайти слабкі місця в проекті, і побачити, чи може робота додатку (а, отже, і враження користувача) бути покращена.

Існує два основних аспекти роботи мобільного-додатку. Перше, це функціональність (включаючи інтерфейс, за допомогою якого користувачі отримують доступ до наповнення додатку та сервісів). По-друге, це доступність (включаючи швидкість завантаження ресурсу і доступ до додатку) [41]. У більшості випадків, ці аспекти залежать один від одного: деякі операції, такі як, покупка через аукціон, сильно залежить від часу завантаження мобільного-додатку, в той час як час завантаження додатку на пряму залежить від відвідувачів, які користуються різними функціями додатку і створюють навантаження на сервер, також це залежить від ресурсів мобільного пристрою.

Багато розробників не мають достатнього терпіння, не можуть дочекатися того моменту, коли їх проект розпочне роботу, і тому занадто швидко починають запускати передані застосунки одразу після того, як створено основний каркас. Часто відбувається так, що квапливість і брак попереднього тестування застосунку в наслідках призводить до невідповідності, ресурс не витримує трафік, і це стає причиною важких наслідків для розробника. Також через наявність великого списку різновидів операційної системи можуть виникати проблеми з працездатністю додатку. Тут все залежить від першого враження, виробленого додатком на користувачів – у додатку буде поганий рейтинг, якщо він не буде працювати швидко та коректно на пристроях користувачів. Ще гірше, якщо застосунок не буде завантажуватися взагалі.

Хоча є змога проводити тест на функціональність вручну вже на ранніх стадіях розробки застосунку, тестування доступу вимагає від розробників виконання складніших дій. В основі тестування лежить проста ідея перевірки додатку і його

різних характеристик в різних умовах навантаження, перевірки працездатності на різних версіях платформ та інші. Для проведення тестування доступу, нам необхідно мати тисячі відвідувачів, які користуються різними телефонами та операційними системами для користування додатками, які використовують провайдерів з різною швидкістю мережі і знаходяться в різних куточках світу. Звісно, проведення цього тестування вручну потребує занадто великої кількості ресурсів, тому краще проводити даний тип тестування автоматично [42].

Існує велика кількість різних онлайнресурсів, які виконують тестування додатку для вас. Проте, як вже було сказано вище, клієнт та розробник при тестуванні додатку не можуть бути розділені. Для того щоб переконатися в тому, що результати не хибні та точні, вимає розробити власні сценарії тестування мобільного-додатку і виправляти їх по ходу дії. Сценарії тестування додатків мають містити в собі всі дії, які тільки можна виконати в застосунку, тому що використання різних функцій призводить до відкриття нових сторінок, що впливає на роботу додатку, в цілому.

Програми для тестування додатків звісно мають різний інтерфейс і особливості. Проте загальний принцип їх роботи залишиться незмінним: вони створюють лімітне навантаження на додатки і, таким чином, знаходять всі слабкі місця додатку. Завдяки звітності даних програм, розробник знає, що потрібно удосконалити [42]. Деякі утиліти, крім тестування поточних параметрів дають змогу спрогнозувати майбутню роботу, період, коли додаток стане знаменитим і зможе зосередити на собі увагу великої кількості користувачів.

Тестування додатків призначено для того, щоб зробити процес користування додатком приємнішим, тому тестування має велике значення в процесі розробки додатку. Воно являє собою поєднання людського управління і можливостей смартфона. Тестування дозволить вам створити якісний і оптимізований додаток, який обов'язково стане визнаним.

4.2 Функціональне тестування мобільного-додатку

Функціональне тестування полягає у перевірці на відсутність мертвих сторінок або недійсних переадресацій, коректність полів вводу, виводу, робочого процесу системи та цілісності даних.

Першою сторінкою для тестування була «Сторінка гаманців». Було перевірено коректне відображення балансу та валюти гаманця(рис. 4.1).

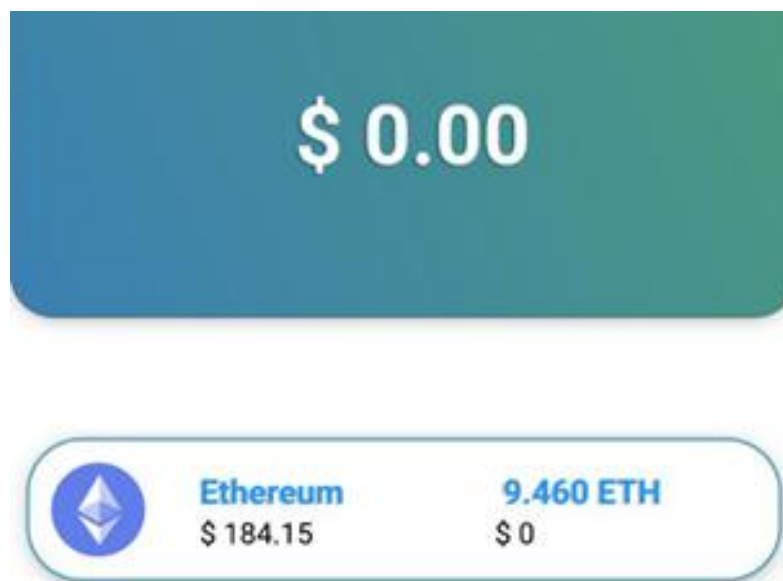


Рисунок 4.1 – Сторінкагаманців

На другій сторінці, а саме сторінці «Гаманець» проблем виявлено не було. Окремо було переглянуто компонент що відповідає за відображення даних про гаманець, а саме адресу та баланс у криптовалюти(рис. 4.2).

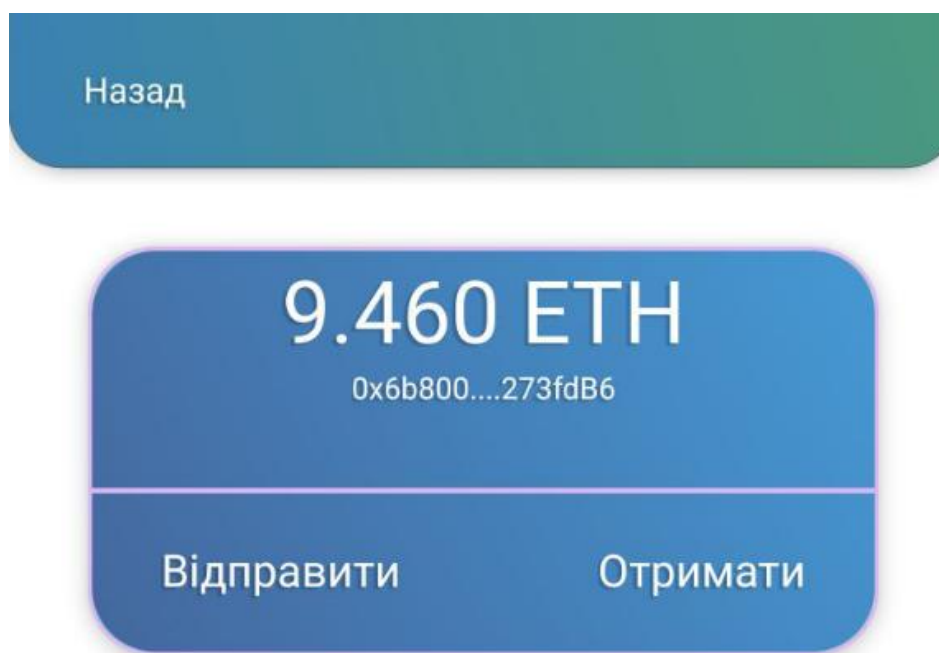


Рисунок 4.2 – Компонентз даними гаманця

Компонент відображення історії транзакцій, відображає останні операції проведені з гаманцем. Було перевірено відображені дані на коректність. Перевірка здійснювалась співставленням даних в додатку та на сайті з інформацією про транзакції за адресою гаманця. При активації компонента у виділеному місці відображається інформація про останні транзакції (рис. 4.3). Перевірка коректності даних здійснювалась на відповідному ресурсі з даними про гаманець (рис. 4.4)

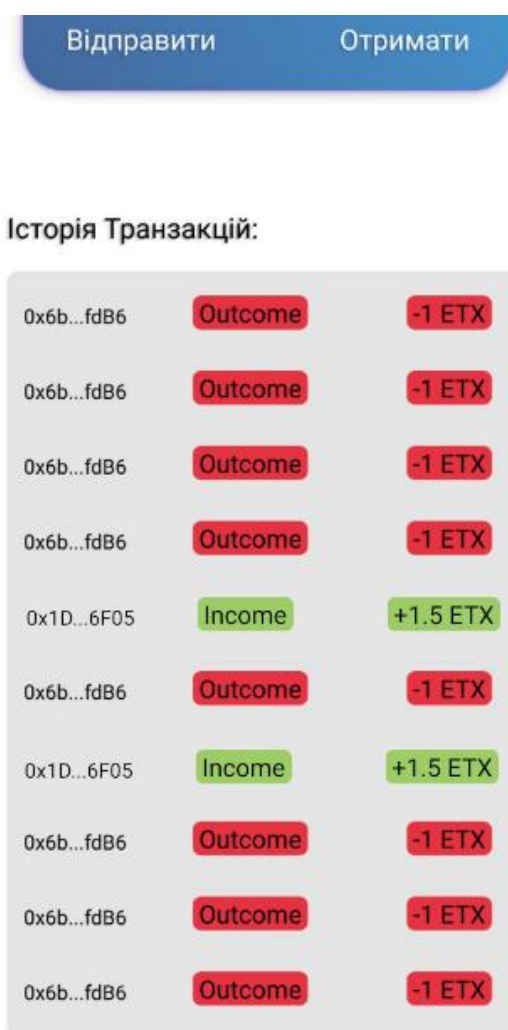
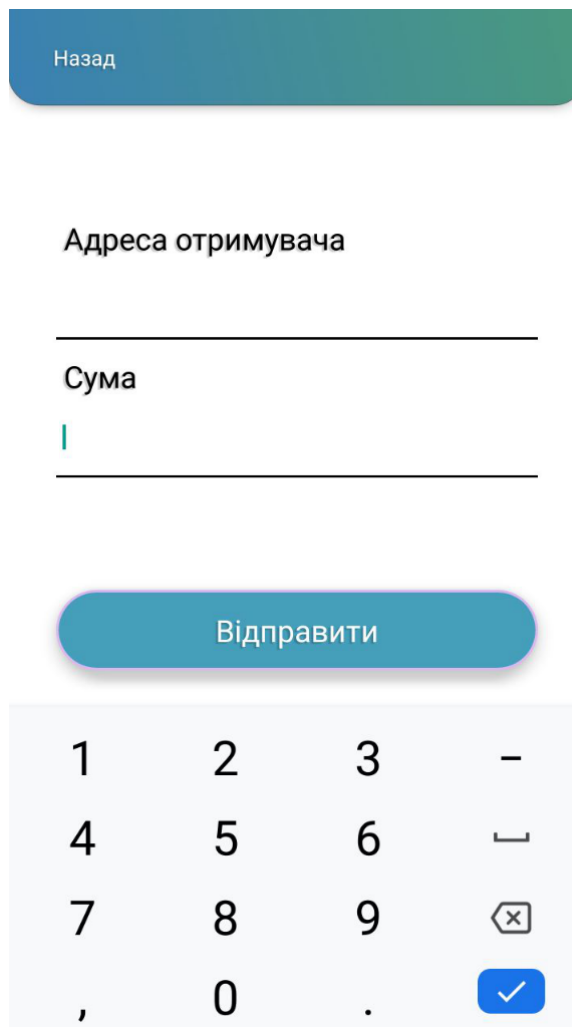


Рисунок 4.3 – Компонент з історією транзакцій

Block	Age	From	To	Value	[Txn Fee]
6799965	1 day 8 hrs ago	0x6b800539925a6a...	OUT Contract Creation	1 Ether	0.000053
6791103	2 days 17 hrs ago	0x6b800539925a6a...	OUT Contract Creation	1 Ether	0.000053
6790584	2 days 19 hrs ago	0x6b800539925a6a...	OUT 0xbaad249abaa27fc...	1 Ether	0.000021
6790579	2 days 19 hrs ago	0x6b800539925a6a...	OUT Contract Creation	1 Ether	0.000053
6790575	2 days 19 hrs ago	0xbaad249abaa27fc...	IN 0x6b800539925a6a...	1.5 Ether	0.00021
6790572	2 days 19 hrs ago	0x6b800539925a6a...	OUT 0xbaad249abaa27fc...	1 Ether	0.000021
6790530	2 days 19 hrs ago	0x1d219502aa0d2d...	IN 0x6b800539925a6a...	1.5 Ether	0.00021
6790157	2 days 20 hrs ago	0x6b800539925a6a...	OUT Contract Creation	1 Ether	0.000053
6790144	2 days 20 hrs ago	0x6b800539925a6a...	OUT Contract Creation	1 Ether	0.000053
6790137	2 days 20 hrs ago	0x6b800539925a6a...	OUT Contract Creation	1 Ether	0.000053

Рисунок 4.4 – Результат відкриття посилання на сторінку з транзакціями

Наступною сторінкою є сторінка відправлення транзакцій, перехід на сторінку відбувся коректно, поля вводу даних також працюють. В залежності від типу поля показується відповідна віртуальна клавіатура(рис. 4.5)



Назад

Адреса отримувача

Сума

Відправити

1	2	3	-
4	5	6	└
7	8	9	⊗
,	0	.	✓

Рисунок 4.5 – Сторінка відправлення транзакції

При введенні некоректних даних відбувається повідомлення та транзакція не відправляється. При заповненні форми коректними даними та нажаті кнопки «Відправити» показується модальне вікно про успішне відправлення транзакції(рис. 4.6)

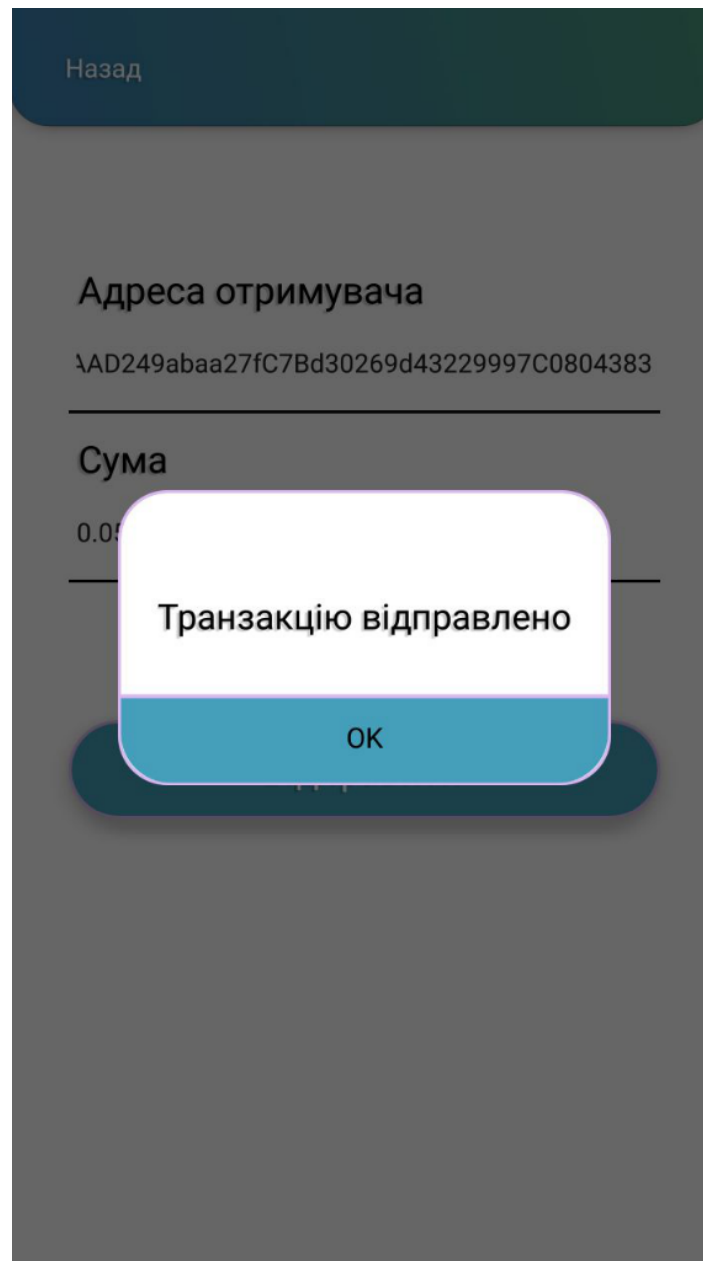


Рисунок 4.6 – Модальне вікно успішного відправлення транзакції

На сторінці з історією транзакцій появились дані про транзакцію, що підтвердилось при використанні додаткового ресурсу(рис. 4.7)

Txn Hash	Block	Age	From	To	Value	[Txn Fee]
0xf90403268c1b4b3...	6809126	39 secs ago	0x6b800539925a6a...	OUT 0xbaad249abaa27fc...	0.5 Ether	0.000021

Рисунок 4.7 – Результат відправлення транзакції

Для перевірки наступної сторінки необхідно повернутися на сторінку гаманця. Здійснюється перехід за допомогою верхньої навігації по додатку, а саме натисканням кнопки «назад». Перехід здійснюється успішно. Для переходу на сторінку з інформацією необхідною для отримання переказів необхідно натиснути кнопку «Отримати». Вся інформація на сторінці «Отримати» відображається коректно (рис. 4.8).



Рисунок 4.8 – Сторінка з даними для отримання активів

При натисканні на текст з адресою гаманця, адреса копіюється до буфера обміну. Для перевірки QR коду було використано другий смартфон. Результатом сканування було отримання адреси гаманця, що зійшлась з вказаною на сторінці (рис. 4.9).

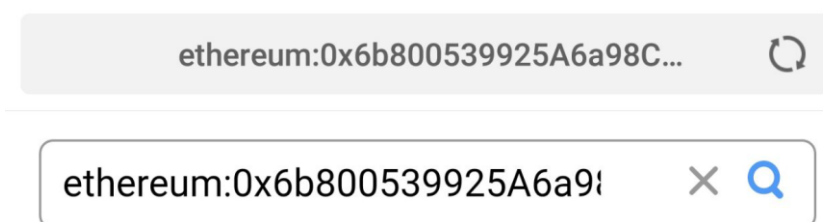


Рисунок 4.9 – Результат сканування QR коду камерою смартфона

Після проведення комплексного функціонального тестування проблем виявлено не було, а саме коректність внутрішніх посилань, посилання, що посилаються до головної сторінки, виявлення сторінок, на які не вказано посилання, відсутність непрацюючих сторінок, тестування форм та всього процесу надсилання транзакції.

4.3 Тестування кросплатформеності

Щоб перевірити мобільний-додаток потрібно провести тестування в на обох платформах, а саме Android та IOS.

Завантаження додатку на операційній системі Android7, а саме на смартфоні MeizuPro7+ відбулося помилок. Сторінка завантаження відобразилась коректно. Перенаправлення на сторінку з гаманцями відбулося без збоїв (рис. 4.10).

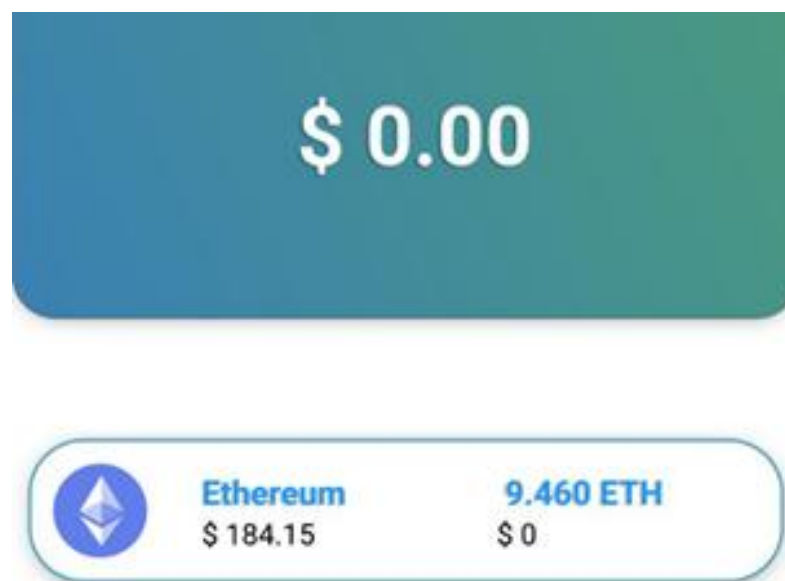


Рисунок 4.10 – Сторінка з гаманцями на операційній системі Android 7.0

Смартфон на операційній системі Android помилок не видав, елементи управління працюють коректно, тому тест на цій платформі успішний.

Наступною була віртуальна машина під керівництвом операційної системи IOS. (рис. 4.11).

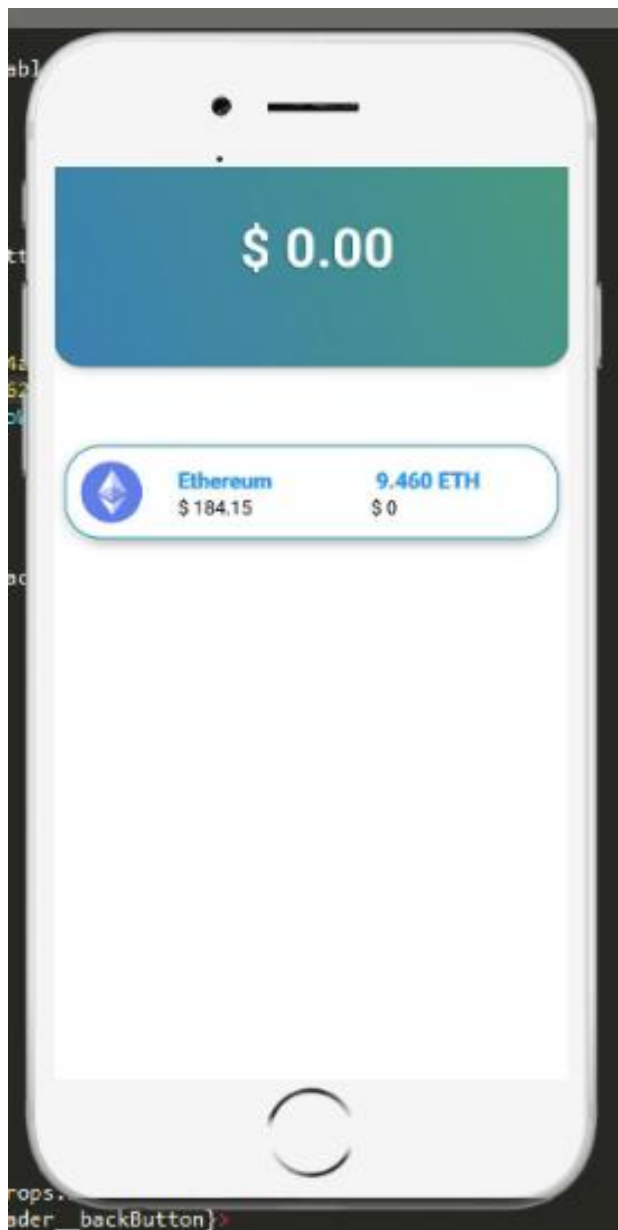


Рисунок 4.11 – Сторінка з гаманцями на операційній системі IOS

Текст та дані відобразились коректно. Перейшовши на сторінку з гаманцем, збоїв не виникло та дані з історією транзакцій відтворились вірно (рис. 4.12).

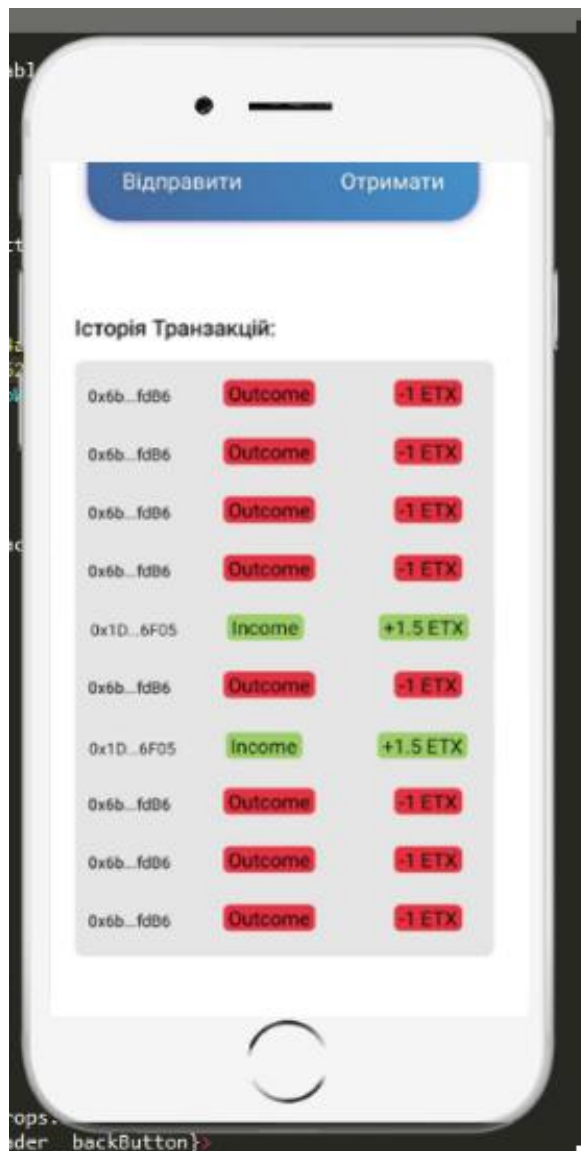


Рисунок 4.12 – Сторінка з історією транзакцій на операційній системі IOS

При спробі відправити транзакцію на операційній системі IOS ніяких збоїв не виявлено, всі дані відображаються вірно, модальне вікно про успішну транзакцію відобразилось коректно(рис. 4.13).

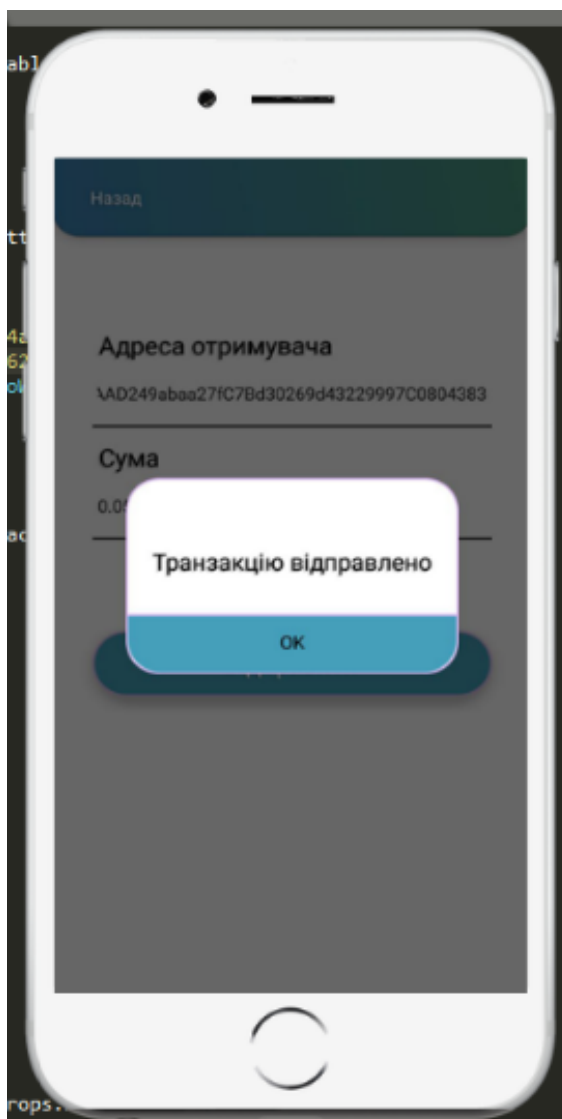


Рисунок 4.13 – Відправлення транзакції на операційній системі IOS

Після проведення функціонального тестування проблем виявлено не було. Всі сторінки працюють коректно. Також тест кросплатформеності показав, що мобільний додаток відображається нормально на обох платформах.

4.4 Висновки

Було наведено та описано методики тестування мобільних-додатків, а саме функціональне тестування, тестування кросплатформеності, зручності користувацького інтерфейсу та інші. Проблем виявлено не було, всі стандарти підтримуються без помилок.

5. ЕКОНОМІЧНИЙ РОЗДІЛ

5.1 Технологічний аудит розробленої системи обміну токенами блокчейнуETH

Як було підкреслено у попередніх розділах роботи, проблема підвищення надійності та терміновості грошових переказів між суб'єктами стає все актуальнішою, особливо коли фінансові відносини між людьми виходять за межі фізичного обміну і переходять у форму цифрових грошей.

Незважаючи на наявність різних допоміжних інструментів, люди щодня зіштовхуються з проблемою, коли їм необхідно перевести свої кошти на рахунки родичів, працівників, виконати оплату наданих послуг тощо. Найпопулярнішими способами переказу грошей є онлайн-перекази через банки, використання ресурсів з цифровими грошима та використання крипто-гаманців.

Якщо кошти відправляються з банківського рахунку певного банку, бажано, щоб отримувач цих коштів також мав рахунок у цьому ж банку, інакше переказовачу прийдеться сплатити велику комісію за виконання цього переказу. Якщо здійснюються міжнародні перекази, то переказовачі зіштовхуються з проблемою конвертації переказу в валюту країни отримувача, що не завжди здійснюється по вигідному для клієнтів курсу тощо.

Криптовалюта позбавлена цих проблем, оскільки за своєю суттю має однакову цінність (вартість) у будь-якій точці планети, а використання розподіленої бази блокчейн дає змогу підвищити безпеку грошових переказів.

Тому у виконаній нами магістерській кваліфікаційній роботі було заплановано розробити спеціальні методи і засоби, які б на базі технології блокчейну та створення мобільних-сервісів з використанням сучасних JavaScript бібліотек підвищували ефективність розробки мобільних-додатків обміну фінансовими активами.

Для досягнення поставленої мети нами було: проаналізовано технології блокчейну як засобу зберігання даних; розроблено структуру мобільного-сервісу; розроблено візуальну частину мобільного-додатку; розроблено алгоритм обміну

токенами блокчейну EТН; зроблено програмну реалізацію мобільного-сервісу; проведено тестування роботи мобільного-сервісу.

В результаті проведених нами досліджень отримано практичні навички розробки модулів роботи з технологією блокчейну та створення мобільних-сервісів з використанням сучасних JavaScript бібліотек.

З метою визначення технічного рівня та комерційного потенціалу розробленої нами системи обміну токенами блокчейну EТН проведемо її технологічний аудит. Для проведення технологічного аудиту запросимо 3-х експертів: к.т.н., доцента Папінова В.М.; к.т.н. доцента Кривогубченка С.Г. та к.т.н. Кулика Я.А.

Оцінювання технічного рівня та комерційного потенціалу нашої розробки здійснювалося за рекомендаціями [44], наведеними в таблиці 5.1 та таблиці 5.2.

Таблиця 5.1 – Критерії оцінювання технічного рівня та комерційного потенціалу розробки

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-тер.	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів

Продовження таблиці 5.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-тер.	0	1	2	3	4
Ринкові перспективи					
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Витрати на рівні експлуатаційних витрат	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

Продовження таблиці 5.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-тер.	0	1	2	3	4
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Таблиця 5.2 – Технічні рівні та комерційний потенціал будь-якої розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Технічний рівень та комерційний потенціал розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

В результаті проведеного технологічного аудиту запрошені експерти виставили нашій розробці такі оцінки (див. таблицю 5.3).

Таблиця 5.3 – Результати технологічного аудиту нашої розробки

Критерії	Прізвище, ініціали експертів		
	Папінов В.М.	Кривогубченко С.Г.	Кулик Я.А.
	Бали, виставлені експертами:		
1	4	4	3
2	3	3	4

Продовження таблиці 5.3

3	4	4	4
4	3	4	4
5	4	3	4
6	3	4	4
7	3	3	4
8	3	3	4
9	3	3	4
10	3	4	3
11	3	4	4
12	4	4	4
Сума балів	СБ ₃ = 40	СБ ₁ = 43	СБ ₂ = 46

Середньоарифметична сума балів, що їх виставили експерти, дорівнює:

$$\overline{\text{СБ}} = \frac{\sum_{i=1}^3 \text{СБ}_i}{3} = \frac{40 + 43 + 46}{3} = \frac{129}{3} = 43.$$

Оскільки середньоарифметична сума балів, що їх виставили експерти, складає 43 бали, то це свідчить, що розроблена нами система обміну токенами блокчейну ЕТН має технічний рівень та комерційний потенціал, який вважається «високим».

Це пояснюється тим, що нами удосконалено модель захисту персональних даних користувача, що дає змогу забезпечити додаткову безпеку його персональних даних за рахунок використання seed-фрази. Окрім того, створені алгоритми та програмне забезпечення можуть бути використані для подальшої розробки ще більш ефективних систем обміну цифровими фінансовими активами та окремих модулів для них.

5.2 Розрахунок витрат на виконання роботи

Витрати на розробку системи обміну токенами блокчейну ЕТН складаються із:

а). Основна заробітна плата Z_o розробників системи обміну токенами блокчейну ЕТН, яка визначається за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (5.1)$$

де M – місячний посадовий оклад розробника, грн;

$M = (4173...16320)$ грн/місяць;

T_p – число робочих днів в місяці; приймемо $T_p = 22$ дні;

t – число днів роботи розробників.

Зроблені розрахунки зведемо до таблиці 5.4:

б). Додаткова заробітна плата Z_d виконавців роботи розраховується як (10...12)% від величини їх основної заробітної плати, тобто:

$$Z_d = \alpha \cdot Z_o = (0,1...0,12) \cdot Z_o \quad (5.2)$$

Таблиця 5.4 – Основна заробітна плата розробників

Найменування посади виконавця	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на оплату праці, грн
1. Науковий керівник магістерської роботи	12550	570,45	25 год. / 6 год.	≈2377
2. Виконавець-магістрант	1700	77,27	56	≈ 4327
3. Консультант з ЕЧ	12800	581,82	2,5 години	≈242
4. Інші фахівці	7000	318,18	2,5	≈ 795
Загалом витрат на заробітну плату				$Z_o = 7741$ грн

Прийmemo, що $\alpha = 0,101$. Тоді для нашого випадку отримаємо:

$$З_d = 0,101 \times 7741 = 781,84 \approx 782 \text{ (грн.)}$$

в). Нарахування на заробітну плату виконавців роботи $H_{зп}$ розраховуються за формулою:

$$H_{зп} = (З_o + З_d) \cdot \frac{\beta}{100}, \quad (5.3)$$

де β – ставка обов'язкового єдиного внеску на державне соціальне страхування, %. $\beta = 22\%$. Тоді:

$$H_{зп} = (7741 + 782) \times 0,22 = 1875,06 \approx 1875 \text{ (грн.)}$$

г). Амортизація A , приміщень, обладнання, комп'ютерів, принтерів тощо, які використовувались під час виконання роботи:

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12} \text{ [грн.]}, \quad (5.4)$$

де $Ц$ – загальна балансова вартість основних засобів, обладнання, комп'ютерів, принтерів тощо грн;

H_a – річна норма амортизаційних відрахувань. Для нашого випадку можна прийняти, що $H_a = (2,5...25)\%$;

T – термін використання основних засобів, місяці.

Зроблені розрахунки зведено в таблицю 4.5.

д). Витрати на матеріали M розраховуються за формулою:

$$M = \sum_1^n H_i \cdot C_i \cdot K_i - \sum_1^n B_i \cdot C_b \text{ [грн.]}, (5.5)$$

де H_i – витрати матеріалу i -го найменування, кг; C_i – вартість матеріалу i -го найменування; K_i – коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$; B_i – маса відходів матеріалу i -го найменування; C_b – ціна відходів матеріалу i -го найменування; n – кількість видів матеріалів.

Таблиця 5.5 – Розрахунок амортизаційних відрахувань

Найменування обладнання, приміщень тощо	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
1. Комп'ютери, принтери, обладнання тощо	39500	21,5	2,2 (при 25% використанні)	389,24
2. Приміщення університету, факультету, кафедри	11800	4,5	2,2 (при 50% використанні)	48,68
Всього				$A = 437,92 \approx 438$

е). Витрати на комплектуючі K розраховуються за формулою:

$$K = \sum_1^n H_i \cdot C_i \cdot K_i \text{ [грн.]}, (5.6)$$

де H_i – кількість комплектуючих i -го виду, шт.; C_i – ціна комплектуючих i -го виду; K_i – коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$; n – кількість видів комплектуючих.

Під час виконання нашої роботи загальні витрати на матеріали та комплектуючі (за аналогією з іншими розробками) склали приблизно 1070 грн.

ж). Витрати на силову електроенергію V_e розраховуються за формулою:

$$V_e = \frac{V \cdot \Pi \cdot \Phi \cdot K_{\Pi}}{K_d}, \quad (5.7)$$

де V – вартість 1 кВт-год. електроенергії, в 2019 р. $V \approx 2,2$ грн/кВт;

Π – установлена потужність обладнання, кВт; $\Pi = 1,20$ кВт;

Φ – фактична кількість годин роботи обладнання, годин.

Прийmemo, що $\Phi = 180$ годин;

K_{Π} – коефіцієнт використання потужності; $K_{\Pi} < 1 = 0,8$.

K_d – коефіцієнт корисної дії, $K_d = 0,6$.

Тоді витрати на електроенергію будуть дорівнювати:

$$V_e = \frac{V \cdot \Pi \cdot \Phi \cdot K_{\Pi}}{K_d} = \frac{2,2 \cdot 1,20 \cdot 180 \cdot 0,8}{0,6} = 633,60 \approx 634 \text{ [грн.]}$$

і). Інші витрати $V_{\text{інш}}$ становлять (100...300)% від суми основної заробітної плати виконавців, тобто:

$$V_{\text{інш}} = (0,1 \dots 3) \times Z_o. \quad (5.8)$$

Для нашого випадку отримаємо:

$$V_{\text{інш}} = 1,9 \times 7741 = 14707,9 \approx 14708 \text{ (грн.)}$$

и). Сума всіх попередніх статей витрат складає витрати на виконання цього етапу роботи безпосередньо магістрантом – V .

$$V = 7741 + 782 + 1875 + 438 + 1070 + 634 + 14708 = 27248 \text{ (грн.)}$$

к). Загальні витрати ZB на можливе завершення нашої розробки $V_{\text{заг}}$ розраховуються за формулою:

$$B_{\text{заг}} = \frac{B}{\beta}, \quad (5.9)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання роботи. Аналізуючи стан виконання роботи (а вона потребує доопрацювання), можна стверджувати, що $\beta \approx 0,5$ [45]. Тоді:

$$B_{\text{заг}} = \frac{27248}{0,5} = 54596 \text{ (грн.)}$$

Тобто прогнозовані загальні витрати на розробку системи обміну токенами блокчейну EТН становлять приблизно 55 тис. грн.

5.3 Розрахунок економічного ефекту від можливого впровадження нашої розробки

Аналіз місткості ринку подібної продукції показує, що на сьогодні в Україні кількість реальних та зацікавлених користувачів подібних розробок складає приблизно 500 осіб, а середня вартість подібної розробки дорівнює приблизно 3 тис. грн. Оскільки розроблена нами система обміну токенами блокчейну EТН має значно кращі функціональні характеристики (тобто значно вищу захищеність), то нашу розробку можна буде реалізовувати на ринку дещо дорожче, приблизно за 3,5 тис. грн, або на 0,5 тис. грн дорожче.

Примітка. Є варіант, коли нашу розробку інвестор буде реалізовувати на ринку безкоштовно, а дохід буде отримувати у вигляді комісійних (наприклад, 0,2% від вартості фінансової операції), але через відсутність достовірної інформації з цього приводу цей варіант розглядати не будемо.

Окрім того, доцільно очікувати стрімке зростання попиту на нашу розробку принаймні протягом 3-х років після її впровадження, оскільки все більше і більше

людей будуть користуватися криптовалютою. Результати будуть виявлятися протягом 2021-го, 2022-го та 2023-го років. Прогноз зростання попиту на нашу розробку складає по роках:

а) 2021 р. – приблизно + 200 шт.;

б) 2022 р. – приблизно + 300 шт.;

в) 2023 р. – приблизно +400 шт.

Можливе збільшення чистого прибутку $\Delta\Pi_i$, що його може отримати потенційний інвестор від впровадження нашої розробки становитиме [45]:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_o \cdot N + \Pi_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\upsilon}{100}\right), \quad (5.10)$$

де $\Delta\Pi_o$ – зміна основного якісного показника від впровадження результатів розробки у цьому році. Зазвичай таким показником є зміна ціни нової розробки, грн. Для нашого випадку $\Delta\Pi_o = (3,5-3) = 0,5$ тис. грн;

N – основний кількісний показник, який визначає обсяг діяльності у цьому році до впровадження результатів розробки; прийmemo, що $N = 500$ шт.;

ΔN – покращення основного кількісного показника від впровадження результатів розробки. Таке покращення становитиме по роках, відповідно: +200 (2021), +300 (2022) та + 400 (2023) шт.;

Π_o – основний якісний показник, який визначає ціну реалізації розробки у цьому році (після впровадження результатів розробки); $\Pi_o = 3,5$ тис. грн;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки; для нашого випадку $n = 3$;

λ – коефіцієнт, який враховує сплату податку на додану вартість; $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати $\rho = (0,2...0,5)$; візьmemo $\rho = 0,5$;

υ – ставка податку на прибуток. У 2019 році $\upsilon = 18\%$.

Величина чистого прибутку $\Delta\Pi_1$ для потенційного інвестора протягом першого року від можливого впровадження нашої розробки (2021 р.) складе:

$$\Delta\Pi_1 = [0,5 \cdot 500 + 3,5 \cdot 200] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) = 324,57 \approx 325 \text{ тис. (грн.)}$$

Величина чистого прибутку $\Delta\Pi_2$ для потенційного інвестора від можливого впровадження нашої розробки протягом другого (2022 р.) року складе:

$$\Delta\Pi_2 = [0,5 \cdot 500 + 3,5 \cdot 300] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) = 444,14 \approx 44 \text{ тис. (грн.)}$$

Величина чистого прибутку $\Delta\Pi_3$ для потенційного інвестора від можливого впровадження нашої розробки протягом третього (2023 р.) року складе:

$$\Delta\Pi_3 = [0,5 \cdot 500 + 3,5 \cdot 400] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) = 563,72 \approx 564 \text{ тис. (грн.)}$$

Теперішня вартість інвестицій PV, що можуть бути вкладені у впровадження нашої розробки, становлять [44].

$$PV = K \times 3B = (1 \dots 5) \times 3B, \quad (5.11)$$

де 3B – витрати на виконання даної роботи, 3B = 55 тис. грн;

K = (1...5) – додатковий коефіцієнт, що враховує витрати потенційного інвестора на впровадження результатів нашої розробки.

Прийmemo, що K = 2,4. Тоді отримаємо:

$$PV = 2,4 \times 55 = 132 \text{ тис. (грн.)}$$

Абсолютний ефект від можливих вкладених інвестицій $E_{\text{абс}}$ складе:

$$E_{\text{абс}} = \text{ПП} - \text{PV}, \quad (5.12)$$

де ПП – приведена вартість всіх можливих чистих прибутків від можливого впровадження нашої розробки, грн;

PV – теперішня вартість інвестицій PV = 132 тис. грн.

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$\text{ПП} = \sum_1^t \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.13)$$

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

t – період часу, протягом якого виявляються результати впровадженої роботи, роки. Для нашого випадку t = 3 роки;

τ – ставка дисконтування. Прийmemo ставку дисконтування $\tau = 0,075$ (7,5%);

t – період часу від моменту отримання прибутків до початку впровадження розробки.

Тоді приведена вартість всіх можливих чистих прибутків ПП, що їх може отримати потенційний інвестор від впровадження нашої розробки, складе:

$$\text{ПП} = \frac{325}{(1 + 0,075)^2} + \frac{444}{(1 + 0,075)^3} + \frac{564}{(1 + 0,075)^4} \square 281 + 357 + 422 = 1060 \text{ тис. (грн.)}$$

Абсолютний ефект від можливого впровадження нашої розробки (при прогнозованому ринку збуту) за три роки складе:

$$E_{\text{абс}} = 1060 - 132 = 928 \text{ тис. (грн.)}$$

Оскільки $E_{abc} > 0$, то вкладання коштів на впровадження нашої розробки може бути доцільним.

Далі розрахуємо внутрішню норму дохідності E_B вкладених інвестицій:

$$E_B = T_{\text{ж}} \sqrt[4]{1 + \frac{E_{abc}}{PV}} - 1, (5.14)$$

де E_{abc} – абсолютний ефект вкладених інвестицій; $E_{abc} = 982$ тис. грн;

PV – теперішня вартість інвестицій $PV = 132$ тис. грн;

$T_{\text{ж}}$ – життєвий цикл розробки, роки. $T_{\text{ж}} = 4$ роки (2020-2023 рр.).

Для нашого випадку отримаємо:

$$E_B = \sqrt[4]{1 + \frac{982}{132}} - 1 = \sqrt[4]{1 + 7,439} - 1 = \sqrt[4]{8,439} - 1 = 1,704 - 1 = 0,704 = 70,4\%.$$

Далі визначимо ту мінімальну дохідність, нижче за яку потенційний інвестор не буде займатися впровадженням нашої розробки. Мінімальна дохідність або мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

$$\tau_{\text{мін}} = d + f, (5.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні $d = (0,11...0,18)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05...0,50)$, але може бути і значно більше.

Для нашого випадку отримаємо:

$$\tau_{\text{мін}} = 0,15 + 0,50 = 0,65 \text{ або } \tau_{\text{мін}} = 65\%.$$

Оскільки величина $E_B = 70,4\% > \tau_{\min} = 65\%$, то потенційний інвестор може бути зацікавлений у фінансуванні та комерціалізації нашої розробки.

Далі розраховуємо термін окупності коштів, вкладених у нашу розробку. Термін окупності $T_{ок}$ розраховується за формулою:

$$T_{ок} = \frac{1}{E_B} \cdot (5.16)$$

Для нашого випадку термін окупності $T_{ок}$ коштів складе:

$$T_{ок} = \frac{1}{0,704} \approx 1,42 \text{ років,}$$

що свідчить про потенційну доцільність комерціалізації нашої розробки.

Таблиця 5.6 – Економічна частина роботи. Зведена таблиця

Показники	Задані у ТЗ	Досягнуті у магістерській кваліфікаційній роботі	Висновок
1. Витрати на розробку системи обміну токенами блокчейну ЕТН	Не більше 60 тис. грн	55 тис. грн.	Досягнуто
2. Абсолютний ефект від впровадження розробки тис. грн	Щороку не менше 300 тис. грн	Щороку по 309 тис. грн протягом 3-х років	Виконано
3. Внутрішня норма дохідності інвестицій, %	не менше 60%	70,4%	Досягнуто
4. Термін окупності інвестицій, роки	до 3-х років	1,42 років	Виконано

Таким чином, основні техніко-економічні показники розробленої системи обміну токенами блокчейну ЕТН, визначені у технічному завданні, виконані.

ВИСНОВКИ

В ході виконання магістерської дипломної роботи був проведений аналіз сучасного стану проблеми обміну фінансовими активами. За результатами проведеного аналізу було визначено, що проблема є актуальною і не зважаючи на різкий темп зростання долі онлайн ресурсів, залишається невирішеною. Тому створення системи обміну токенами блокчейну ЕТН є доцільним. У зв'язку з цим було проведено порівняльний аналіз аналогів, що дозволило оцінити сучасний ринок додатків, які мають можливість надати послуги обміну цифровими фінансовими активами. У подальшому процесі розробки мобільного-додатку враховані результати аналізу для досягнення поставлених цілей магістерської дипломної роботи.

Для опису структури мобільного-додатку «CryptoPay» було визначено головні модулі та відношення між ними, що дозволило набагато швидше розробити остаточний мобільний-застосунок. Для наочного зображення даної структури було побудовано структурну схему Константайна, яка показує зв'язок модулів мобільного-додатку.

Для розробки мобільного-додатку «CryptoPay» було обрано мову програмування JavaScript та фреймворк ReactNative. Було проведено порівняльний аналіз інструментів, які можна використовувати для написання додатку. Проаналізувавши отримані результати, було вирішено, що ReactNative є найкращим рішенням для виконання поставлених задач. В якості інтегрованого середовища розробки було обрано PHPStorm.

Описано програмну реалізацію мобільного додатку «CryptoPay». Повний лістинг розроблених скриптів наведено в додатках.

Також описано створення візуальної частини мобільного додатку, що включає в себе розробку сторінок, інформаційних блоків та її модулів. В розробленому мобільному додатку «CryptoPay» було реалізовано наступні функціональні можливості:

- відображення в реальному часі інформації про баланс гаманця;

- відображення детальної інформації про минулі транзакції;
- можливість відправляти транзакції;
- відображення даних про гаманець за якими можна отримати переказ;

Було наведено та описано методики тестування програмного забезпечення, а саме методи функціонального тестування, тестування кросплатформності та інші. Для тестування розробленого мобільного додатку використано функціональне тестування та тестування кросплатформності. Проблем виявлено не було. Мобільний додаток підтримує всі сучасні стандарти мобільної розробки.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Кушнір В.І. Розробка системи обміну токенами блокчейну ETH [Електронний ресурс] / Є.Паламарчук В.Кушнір. – Режим доступу до ресурсу: <http://www.wayscience.com/konferentsiya-8-3-4-zhovtnya-2019/>- Назва з екрану.
2. Мобільний застосунок – [Електронний ресурс]. Матеріал з вікіпедії: https://uk.wikipedia.org/wiki/Мобільний_застосунок
3. Що таке блокчейн простими словами – [Електронний ресурс]. – Режим доступу: <https://prostocoin.com/blog/blockchain-guide>
4. Що таке блокчейн? – [Електронний ресурс]. – Режим доступу: <https://interesgroup.com.ua/article/what-is-the-blockchain.html>
5. Нараян Прасти. Блокчейн. Разработка приложений. – BHV, Русская редакция, 2016 – 256 с. - ISBN: 978-5-9775-3976-0.
6. Kotlin – [Електронний ресурс]. – Режим доступу: <https://kotlinlang.org>
7. Исакова С., Жемеров Д. Kotlin в действии. – Издательство «ДМК-Пресс», 2017. – 402 с. – ISBN 978-5-97060-497-7.
8. Swift – [Електронний ресурс]. – Режим доступу: <https://developer.apple.com/swift>
9. ReactNative – [Електронний ресурс]. – Режим доступу: <https://facebook.github.io/react-native/>
10. Плюсы и минусы платформы ReactNative – [Електронний ресурс]. – Режим доступу: <https://dou.ua/lenta/articles/react-vs-real/>
11. ReactNativevsRealNativeApps– [Електронний ресурс]. – Режим доступу: <https://codeburst.io/react-native-vs-real-native-apps-ad890986f1f>
12. ReactJS – [Електронний ресурс]. – Режим доступу: <https://habr.com/post/338068/>
13. ReactNative – одного Jsмало – [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/post/323214/>
14. React Navigation, complete Redux state management, tab-bar and multiple navigators– [Електронний ресурс]. – Режим доступу: <https://codeburst.io/react->

navigation-with-complete-redux-state-management-tab-bar-and-multiple-navigators-ed30a69d9a4d

15. Бодух А. React и React Native / А. Бодух. – 2017. – 500с. – ISBN 978-1-78646-565-8

16. Стефанов С. React.js Быстрый старт/ С. Стефанов. – изд.СПБ: Питер , 2016. – 365с. – ISBN 978-5-4461-0889-3

17. Айзенман Б. Изучение ReactNative: создание собственных мобильных приложений с помощью JavaScript / Б. Айзенман. – 2015. – 272с. – ISBN 978-1-49198-914-2

18. Вард Д. React Native Cookbook / Д. Вард. – 2019. – 592с. – ISBN 978-1-78899-192-6

19. Крокфорд Д. JavaScript. Сильные стороны. Пер. с англ. – СПб.: Питер, 2012.

20. Томас М. React в действии / М. Томас. – изд. Эксмо , 2016. – 720с. – ISBN 978-5-4461-0999-9

21. Замметти Ф. Практический ReactNative: создайте два проекта и одну игру с использованием ReactNative/ Ф. Замметти. – 2018. – 352с. – ISBN 978-1-4842-3938-5

22. Васильев О. JavaScript в примерах и задачах/ О. Васильев. – изд.СПБ: Питер , 2016. – 368с. – ISBN 978-5-699-95459-9

23. Филипс Б. и др.. Андроид.Програмирование для профессионалов: Пер. С англ./Б.Филлипс, К. Стюарт, К. Марсикано. – СПб.: Издательство «Питер», 2017. – 688 с. – ISBN 978-5-4461-0413-0.

24. Бэнкс Алекс, Порселло Ева Реакти Redux.Функциональная разработка. – Издательство «Питер», 2018. – 336 с. – ISBN 978-1-49195-462-1.

25. BitPay– [Электронный ресурс]: Матеріал з BitPay– сайт з мобіольними-додатками із використанням технології блокчейн. Режим доступу до матеріалу: <https://bitcoin.org/ru/wallets/mobile/android/bitpay/>

26. Electrum – [Электронный ресурс]. Матеріал з Electrum– сайт з мобіольними-додатками із використанням технології блокчейн. Режим доступу до матеріалу: <https://electrum.org/#home>

27. Процесс разработки дизайна мобильного приложения с нуля – [Электронный ресурс]. – Режим доступа: <https://medium.com/product-design/процесс-разработки-дизайна-мобильного-приложения-с-нуля-909c0443ef92>
28. Иттен И. Искусство цвета / И. Иттен. – изд. Д. Аронов, 2011. – 96с. – ISBN 978-5-94056-021-0
29. Сучасний дизайн – [Электронный ресурс]. – Режим доступа: <http://kitchenriots.com/lvldesign-inprocedural-generation/>
30. Структурные карты Константайна - [Электронный ресурс]. - Режим доступа до матеріалу: <http://www.interface.ru/fset.asp?Url=/case/defs71>
31. Вільна енциклопедія. JavaScript – [Электронный ресурс]. – Режим доступа: <https://uk.wikipedia.org/wiki/JavaScript>
32. Введение в JavaScript – [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru/intro>
33. PHPStorm [Электронныйресурс]: The smartest JavaScript IDE. Режим доступа до матеріалу: <https://www.jetbrains.com/webstorm>
34. 6 Simple ways to speed up your react native app– [Электронныйресурс]. – Режимдоступу: <https://codeburst.io/6-simple-ways-to-speed-up-your-react-native-app-d5b775ab3f16>
35. How do I sign transactions with Web3 – [Электронныйресурс]. – Режимдоступу: <https://medium.com/finnovate-io/how-do-i-sign-transactions-with-web3-f90a853904a2>
36. Interacting With Smartcontracts Web3.js – [Электронныйресурс]. – Режимдоступу: <https://medium.com/swlh/interacting-with-smartcontracts-via-web3-js-367ea980652c>
37. Signing and making transactions on Ethereum using web3.js – [Электронныйресурс]. – Режимдоступу: <https://medium.com/coinmonks/signing-and-making-transactions-on-ethereum-using-web3-js-1b5663207d63>
38. Аквино Крис, Ганди Тодд Front-end. Клиентская разработка для профессионалов. Node.js – Издательство «Питер», 2016. – 512 с. – ISBN 978-5-496-02930-8.

39. Айзенберг Б. Тестирование и оптимизация веб-сайтов: руководство по Google Website Optimizer / Б. Айзенберг, Дж. Кварто вон Тивадар, Л.Т. Дэвис. – изд. Диалектика, 2009. – 336 с. – ISBN 978-5-8459-1542-9

40. Основные этапы тестирования мобильных приложений – [Электронный ресурс]. – Режим доступа: <https://itvdn.com/ru/blog/article/mob-test-blog>

41. Автоматизація тестування мобільних додатків – [Електронний ресурс]. – Режим доступу: <https://live.ithillel.ua/ua/avtomatyzatsiia-testuvannia-mobilnykh-dodatki>

42. Тестирование мобильных приложений для начинающих – [Электронный ресурс]. – Режим доступа: <https://qa-academy.by/qaacademy/news/testirovanie-mobilnykh-prilozhenij-dlya-nachinayushhix/>

43. Канер Сэм и др.. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений: Пер. с англ./Сэм Канер, Джек Фолк, Енг Кек Нгуен. – К.: Издательство «ДиаСофт», 2001. – 544 с. – ISBN 966-7393-87-9.

44. Методичні рекомендації з комерціалізації розробок, створених в результаті науково-технічної діяльності – К.: Наказ Державного комітету України з питань науки, інновацій та інформатики (Лист № 1/06-4-97 від 13.09.2010 р.).

45. Козловський В. О. Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт. – Вінниця: ВНТУ, 2012.

ДОДАТКИ

Додаток А (обов'язковий).**Технічне завдання**

ЗАТВЕРДЖЕНО
Зав. кафедри АІТ
_____ Кветний Р.Н.

" ____ " _____ 2019 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

«Система обміну токенами блокчейну ЕТН»

08-02.МКР.000.00.006ТЗ

Керівник роботи:

_____ к.т.н., доц. кафедри АІТ Паламарчук Є. А.

" ____ " _____ 2019 р.

Виконав:

_____ студент гр.1АКІТ-18м Кушнір В.І.

" ____ " _____ 2019 р.

Вінниця 2019

1. Назва та галузь застосування.

Засіб для проведення обмінів цифровими активами у зручний спосіб з використанням блокчейну ETH. Розроблений додаток дозволяє значно спростити процес обміну цифровими активами.

2. Підстава для розробки.

Підставою для розробки магістерської кваліфікаційної роботи є наказ ВНТУ №254 від 02.10.2019

3. Мета та призначення розробки.

Мета виконання магістерської кваліфікаційної роботи – є підвищення ефективності та пришвидшення процесу розробки мобільних-додатків обміну фінансовими активами.

Призначення роботи – розробка мобільного-додатку «CryptoPay».

4. Джерела розробки:

1. Кушнір В.І. Розробка системи обміну токенами блокчейну ETH [Електронний ресурс] / Є.Паламарчук В.Кушнір. – Режим доступу до ресурсу: <http://www.wayscience.com/konferentsiya-8-3-4-zhovtnya-2019/>- Назва з екрану.

2. How do I sign transactions with Web3 – [Електроннийресурс]. – Режимдоступу: <https://medium.com/finnovate-io/how-do-i-sign-transactions-with-web3-f90a853904a2>

3. Interacting With Smartcontracts Web3.js – [Електроннийресурс]. – Режимдоступу: <https://medium.com/swlh/interacting-with-smartcontracts-via-web3-js-367ea980652c>

4. Signing and making transactions on Ethereum using web3.js – [Електроннийресурс]. – Режимдоступу:<https://medium.com/coinmonks/signing-and-making-transactions-on-ethereum-using-web3-js-1b5663207d63>

5. Технічні вимоги:

- мобільний прилад під керівництвом операційної системи –IOS, Android;
- середовище розробки – PHPStorm;
- середня швидкість виконання транзакції 3хвилини;
- мова програмування – JavaScript;
- максимальна сума разового переказу 25 ETH;
- бібліотеки – web3, ReactNative.

6. Економічні показники:

- прогнозовані витрати на розробку – 55тис. грн;
- абсолютна ефективність розробки – не менше 300тис. грн;
- термін окупності витрат для виробника – 1,42 років.

7. Стадії та етапи розробки:

Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)
Аналіз стану проблеми і постановка задач	
Аналіз процесу розробки мобільних-додатків	
Розробка структури мобільного-додатку	
Розробка алгоритму роботи обміну токенами в мережі блокчейну ETH	
Обґрунтування вибору інструменту для розробки додатку	
Програмна реалізація мобільного-додатку	
Тестування роботи мобільного-додатку	
Оформлення матеріалів до захисту	

8. Порядок контролю та прийняття.

1. Рубіжний контроль. Провести до 10.11.2019.
2. Попередній захист магістерської кваліфікаційної роботи. Провести до 17.12.2019.
3. Захист магістерської кваліфікаційної роботи. Провести в період з 18.12.2019 до 19.01.2019.

Додаток Б (обов'язковий).**Графічна частина**

Завію Каф АІТ

_____ канд. техн. наук, доцентПаламарчук Є.А

(підпис)

(науковий ступінь, вчене звання, ініціали та прізвище)

Науковий керівник

_____ канд. техн. наук, доцентПаламарчук Є.А

(підпис)

(науковий ступінь, вчене звання, ініціали та прізвище)

Тех. Контроль

_____ канд. техн. наук, доцентПаламарчук Є.А

(підпис)

(науковий ступінь, вчене звання, ініціали та прізвище)

Нормконтроль

_____ канд. техн. наук, доцентПаламарчук Є.А

(підпис)

(науковий ступінь, вчене звання, ініціали та прізвище)

Рецензент

_____ канд. техн. наук, доцентБиков М.М

(підпис)

(науковий ступінь, вчене звання, ініціали та прізвище)

Студент гр. 1АКІТ-18м

(підпис)

_____ Кушнір В. І.

(ініціали та прізвище)

Продовження додатка Б



Рисунок Б.1 – Схема роботи програми

Продовження додатка Б

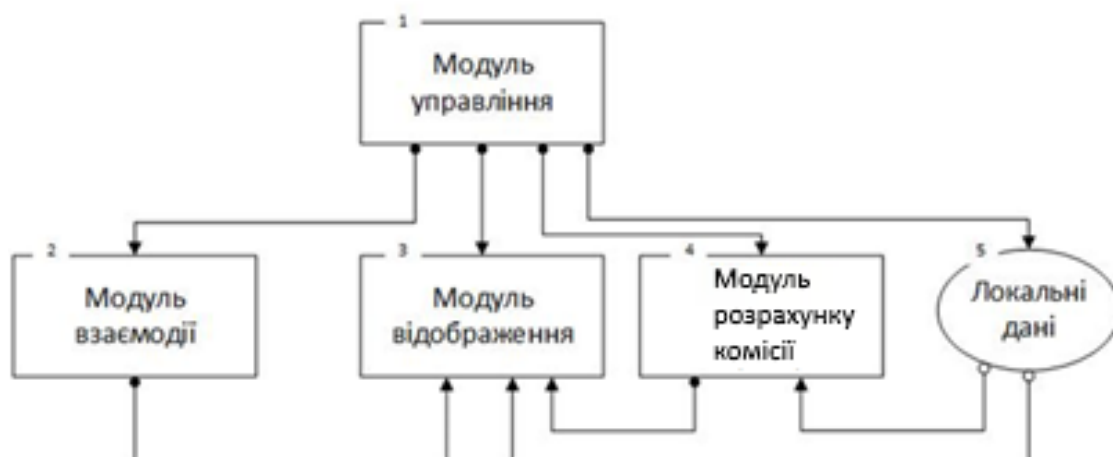


Рисунок Б.2 – Структурна карта Константайна

Продовження додатка Б

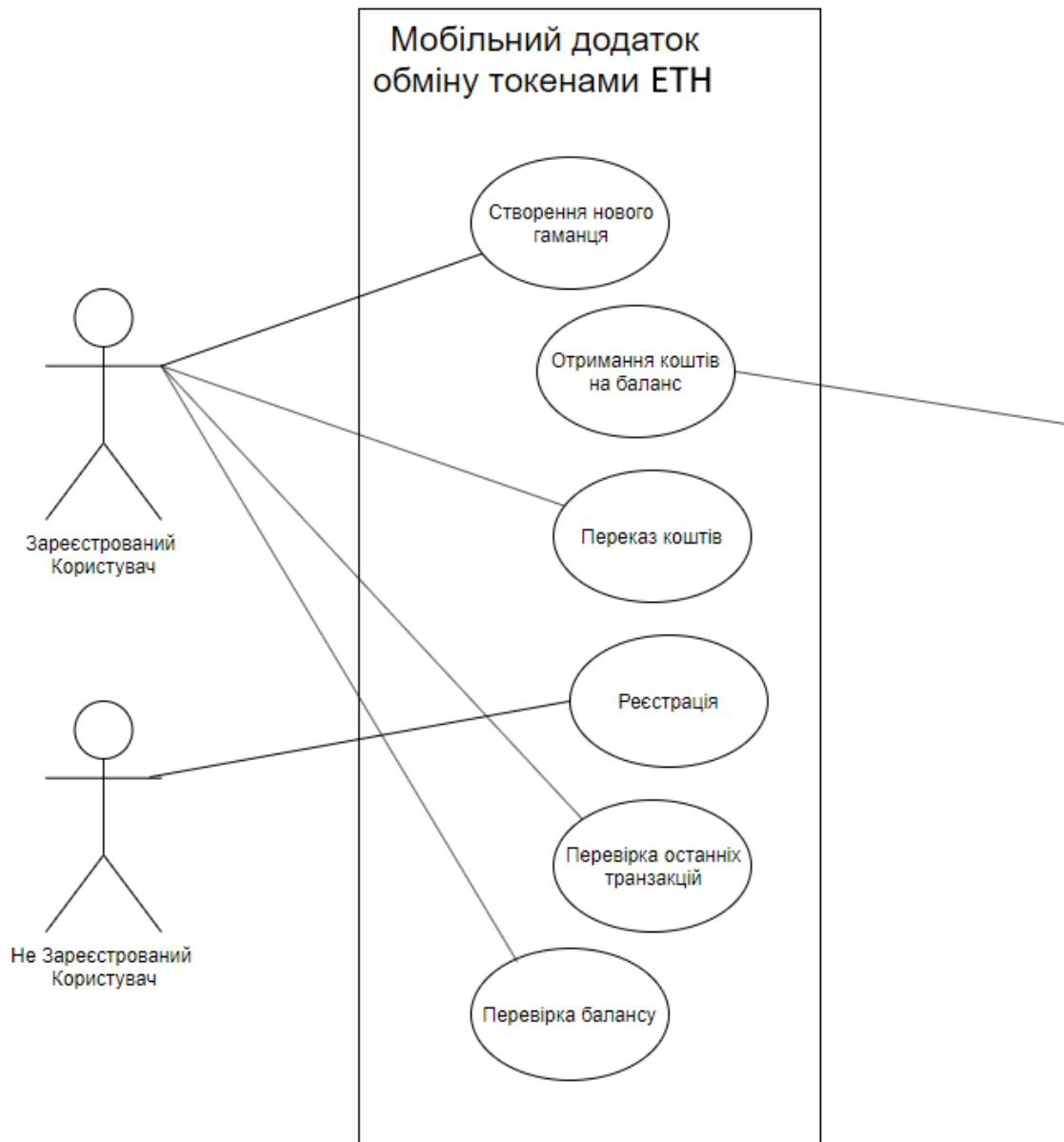


Рисунок Б.3 – UML Діаграма прецедентів

Продовження додатка Б

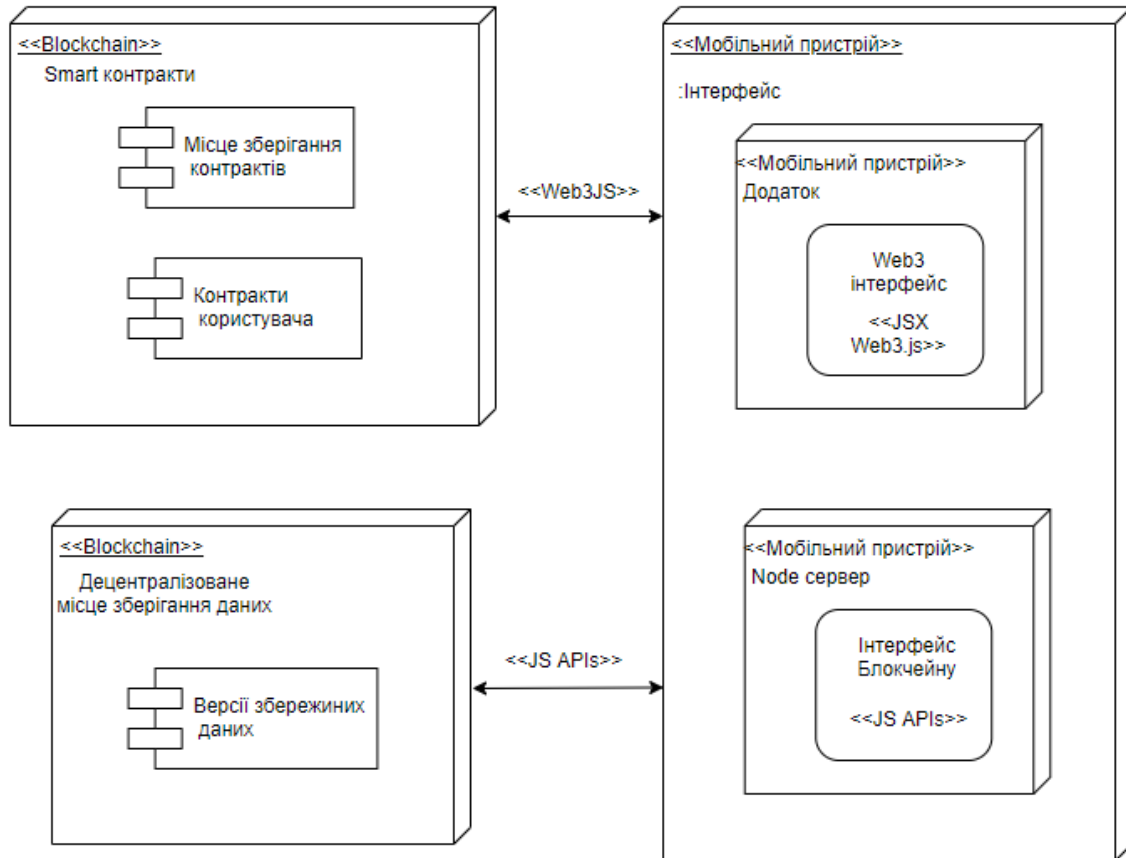


Рисунок Б.4 – UML Діаграма компонентів

Продовження додатка Б

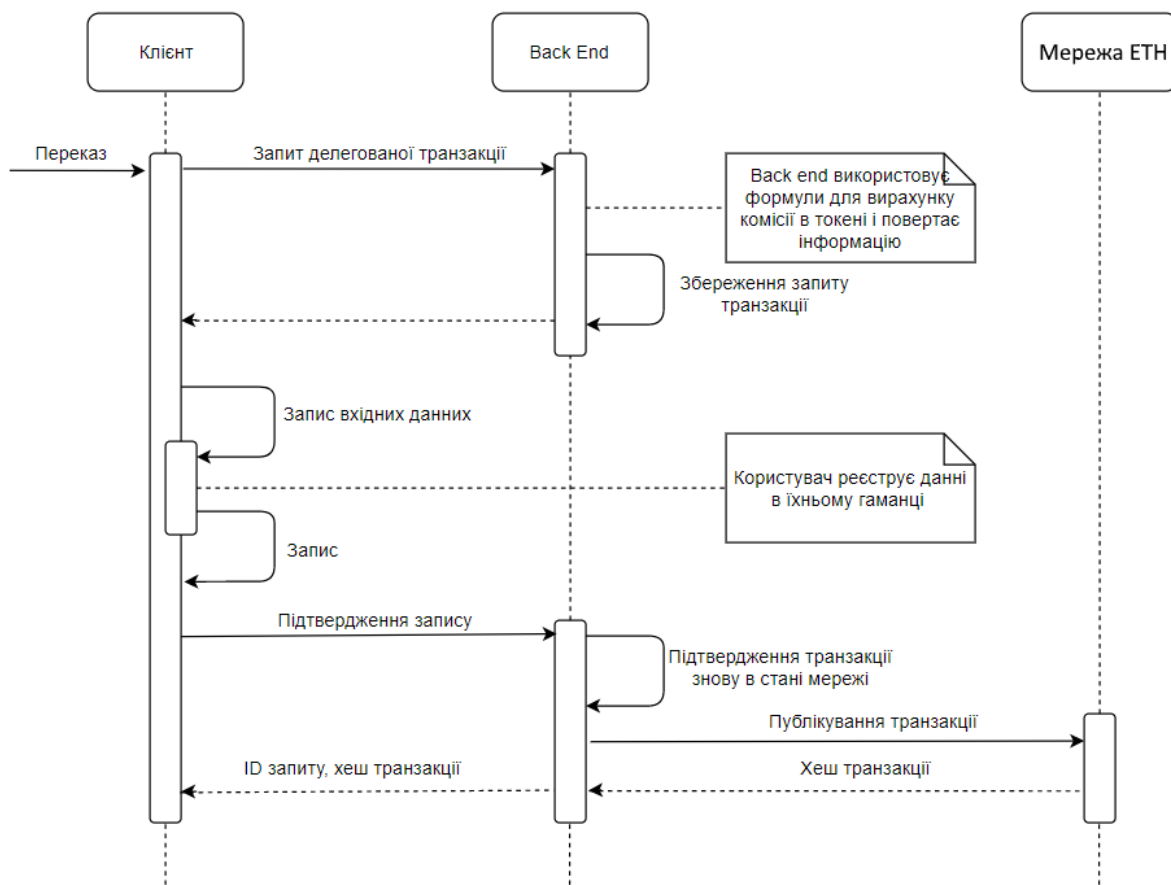


Рисунок Б.5 – UML Діаграма послідовностей

Продовження додатка Б

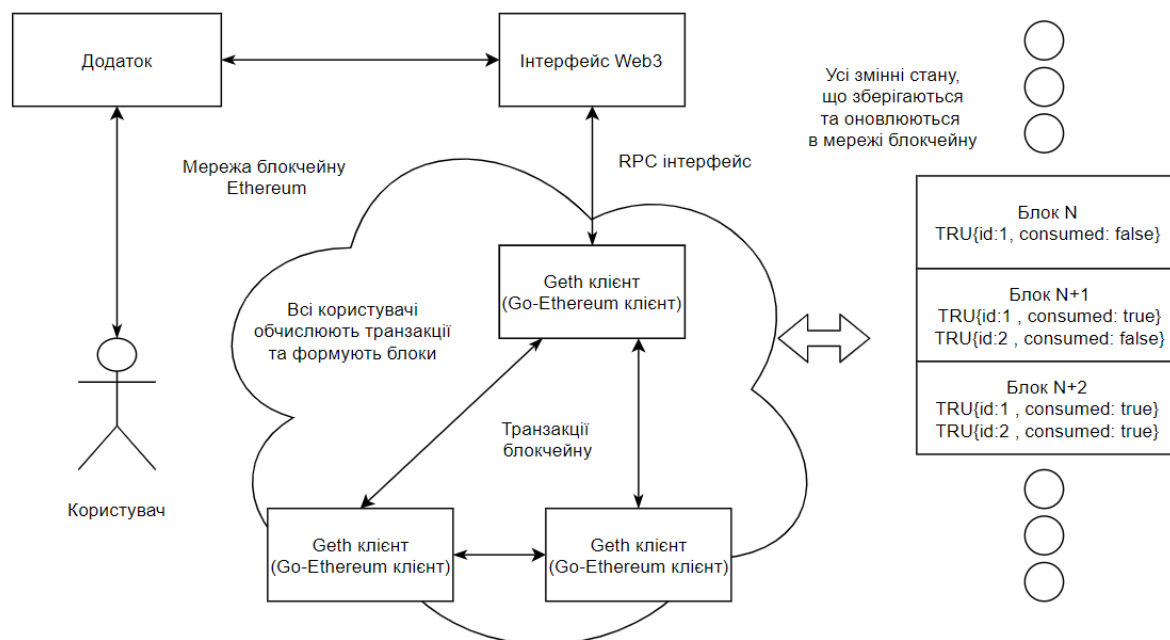


Рисунок Б.6 – UML Діаграма взаємозв'язку користувача з блокчейном ETH

Додаток В (обов'язковий).

Лістинг головного екрану додатку

```

import * as React from 'react';
import { View,Text, StyleSheet,TouchableOpacity, } from 'react-native';
import {Header,CryptoItem} from "../components";
class HomeScreen extends React.Component{
  constructor(props) {
    super(props);
  }
  render(){
    return (
<View >
<View>
<Header>
<View style={ styles.balance__container}>
<Text style={ styles.balance}>$ 0.00</Text>
</View>
</Header>
</View>
<View>
<TouchableOpacity style={styles.cryptoItem} onPress={() => this.props.navigation.navigate("Wallet")}>
<CryptoItem />
</TouchableOpacity>
</View>
</View>
);
  };
};
const styles = StyleSheet.create({
  cryptoItem: {
    marginTop: 25,
  },
});
export {HomeScreen};

```

Додаток Г (обов'язковий).

Лістинг екрану з інформацією для отримання транзакцій

```

import * as React from 'react';
import {View, Text, StyleSheet, TextInput, TouchableOpacity, Image} from 'react-native';
import {Header} from "../components";
class GetScreen extends React.Component{
  constructor(props){
    super(props)
  }
  render() {
    return (
<View>
<View>
<Header navigation={this.props.navigation} style={styles.header}>
<View style={styles.header__backButton}>
<TouchableOpacity onPress={() => this.props.navigation.goBack()} >
<Text style={ styles.header__backButton__text}>Назад</Text>
</TouchableOpacity>
</View>
</Header>
</View>
<View style={styles.qr}>
<Image source={require('../sources/qr.png')} style={styles.imageView} />
</View>
<View style={styles.get__text__container} >
<View>
<Text style={styles.get__text} > Ваш Ethereum адрес:</Text>
</View>
<View style={styles.get__address__text__container}>
<Text style={styles.get__address__text}> {window.accountLink}</Text>
</View>
</View>
</View>
</View>

```

```
    );  
  }  
};  
const styles = StyleSheet.create({  
  header: {  
    height: 90,  
  },  
  header__backButton__text: {  
    padding: 25,  
    color: "white",  
    textShadowColor: 'rgba(0, 0, 0, 0.2)',  
    textShadowOffset: {width: -1.5, height: 1},  
    textShadowRadius: 2  
  },  
  header__backButton: {  
    top: 15,  
    left: 3,  
  },  
  qr: {  
    alignItems: "center",  
  },  
  get__text: {  
    fontSize: 18,  
    color: "grey",  
    textShadowColor: 'rgba(0, 0, 0, 0.2)',  
    textShadowOffset: {width: -1.2, height: 0.5},  
    textShadowRadius: 1  
  },  
});  
export {GetScreen};
```

Додаток Д (обов'язковий).

Лістинг екрану з формою для відправки транзакцій

```

import * as React from 'react';
import { View,Text, StyleSheet,TextInput,TouchableOpacity,Button } from 'react-native';
import Modal from "react-native-modal";
import {sendTransaction} from "../scripts/sendTransaction";
class SendScreen extends React.Component{
  constructor(props){
    super(props)
    this.state ={
      isModalVisible: false
    }
  }
  toggleModal = () => {
    this.setState({ isModalVisible: !this.state.isModalVisible });
  };

  render() {
    return (
<View>
<View>
<Header navigation={this.props.navigation} style={styles.header}>
<View style={styles.header__backButton}>
<TouchableOpacity onPress={() => this.props.navigation.goBack()}>
<Text style={styles.header__backButton__text}>Назад</Text>
</TouchableOpacity>
</View>
</Header>
</View>
<View style={styles.send__form}>
<View>
<Text style={styles.send__form__text}> Адреса отримувача</Text>
<TextInput style={styles.send__form__input}
      onChangeText = {(text)=> {

```



```

                this.setState({toAccount: text})
            }}
        />
</View>
<View>
<Text style={styles.send__form__text}> Сума</Text>
<TextInput style={styles.send__form__input}
            keyboardType = 'numeric'
            onChangeText = {(text)=> {
                this.setState({value: text*1000000000000000000})
            }}
        />
</View>
<View>
<TouchableOpacity style={styles.send__form__button} onPress={() =>
    {sendTransaction(this.state.toAccount,this.state.value); this.toggleModal() }} >
<Text style={styles.send__form__button__text}>
    Відправити
</Text>
</TouchableOpacity>
<Modal isVisible={this.state.isModalVisible} backdropOpacity={0.6} styles={styles.modal}>
<View style={styles.modalContainer}>
<Text style={styles.modalText}>Транзакцію відправлено</Text>
<View>
<TouchableOpacity onPress={this.toggleModal} style={styles.modalButton}>
<Text style={styles.modalButton__text}>
    ОК
</Text>
</TouchableOpacity>
</View>
</View>
</View>
</Modal>
</View>
</View>
</View> );
};

```

Додаток Е (обов'язковий).

Лістинг екрану з даними про гаманець

```

import * as React from 'react';
import { ScrollView, View, Text, StyleSheet, TouchableOpacity, } from 'react-native';
import { Header, History } from "../components";
import LinearGradient from "react-native-linear-gradient";

window.accountLink = "0x6b800539925A6a98C27189EA97E4af59A273fdB6"

let accountLinkFirstPart = accountLink.substr(0,7)
let accountLinkSecondPart = accountLink.substr(35,7)

class WalletScreen extends React.Component{
  constructor(props) {
    super(props);
    this.state = {
      balance: window.AccountBalance,
    }
  }
  render() {
    return (
<ScrollView showsVerticalScrollIndicator={false}>
<View>
<View>
<Header navigation={this.props.navigation} style={styles.header}>
<View style={styles.header__backButton}>
<TouchableOpacity onPress={() => this.props.navigation.goBack()} >
<Text style={ styles.header__backButton__text}>Назад</Text>
</TouchableOpacity>
</View>
</Header>
</View>
<LinearGradient colors={['#4598D3',"#44679D"]} start={{ x: 1, y: 0 }} end={{ x: 0, y: 1 }}
style={styles.wallet__info}>
<View>
<Text style={styles.wallet__info_balance}>{this.state.balance} ETH</Text>
<Text style={styles.wallet__info_adress}>{accountLinkFirstPart}....{accountLinkSecondPart}</Text>

```

```

</View>
<View style={styles.wallet__info__line}/>
<View style={styles.wallet__info__buttons__container}>
  <TouchableOpacity style={styles.wallet__info__button} onPress={() => this.props.navigation.navigate("Send")}>
    <Text style={styles.button__text}>Відправити</Text>
  </TouchableOpacity>
  <TouchableOpacity style={[styles.wallet__info__button, styles.wallet__info__button__right]} onPress={() =>
    this.props.navigation.navigate("Get")}>
    <Text style={styles.button__text}>Отримати</Text>
  </TouchableOpacity>
</View>
</LinearGradient>
<View style={styles.wallet__info__history}>
  <Text style={styles.wallet__info__history__text}>
    Історія Транзакцій:
  </Text>
  <View style={styles.wallet__info__history__container}>
    <History/>
  </View>
</View>
</View>
</View>
</ScrollView>
  );
}
};

```

Додаток Є (обов'язковий).

Лістинг скрипту для відправлення транзакції

```

import Web3 from 'web3';
import {View} from "react-native";
let web3 = new Web3(
  new Web3.providers.HttpProvider('https://ropsten.infura.io/mXinyGPyBIRXE9dEXwee')
);
let key ='CCAF57B9DFB20806E49E1A44751478F6BB8742D9752EFE6A5234A3E90A4B2483';
let tmpTransHesh = null;
export function sendTransaction(toAccount,value) {
  web3.eth.getTransactionCount("0x6b800539925A6a98C27189EA97E4af59A273fdB6", "pending")
    .then((result) => {
      web3.eth.accounts.signTransaction({
        nonce: result,
        to: toAccount,
        gasLimit: 1000000,
        value: value,
      }, '0x' + key)
        .then((signData) => {
          console.log(signData);
          web3.eth.sendSignedTransaction(
            signData.rawTransaction
          )
            .on('transactionHash', (transHesh) => {
              tmpTransHesh = transHesh;
            })
        })
    })
}
export function balanceCheck() {
  return( web3.eth.getBalance("0x6b800539925A6a98C27189EA97E4af59A273fdB6"))
}

```