

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

«На правах рукопису»
УДК _____

До захисту допущено:
В. о. завідувача кафедри
_____ Олександр РОЛІК
«__» _____ 2021 р.

**Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Інформаційні управляючі
системи та технології»
зі спеціальності 126 «Інформаційні системи та технології»
на тему: «Інформаційна система підтримки управління власними
криптовалютними активами»**

Виконав (-ла):
студент (-ка) VI курсу, групи ІС-02мп
Халін Олег Ігорович _____

Керівник:
доцент, к.т.н., доцент,
Жданова Олена Григорівна _____

Рецензент:
доцент, к.т.н.
Новінський Валерій Петрович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.
Студент (-ка) _____

Київ – 2021 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В. о. завідувач кафедри

_____ Олександр РОЛІК

« ____ » _____ 2021р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Халіну Олегу Ігоровичу

1. Тема дисертації «Інформаційна система підтримки управління власними криптовалютними активами», науковий керівник дисертації Жданова Олена Григорівна, к.т.н, доцент кафедри АСОІУ, затверджені наказом по університету від «27» жовтня 2021 р. № 3587-с
2. Термін подання студентом дисертації «6» грудня 2021 р.
3. Об'єкт дослідження – процес підтримки управління криптовалютними активами.
4. Предмет дослідження – інформаційна система підтримки управління власними криптовалютними активами користувача.
5. Перелік завдань, які потрібно розробити:
 - 1) Провести аналіз проблеми та актуальності теми. Зробити огляд існуючих рішень;
 - 2) Розробити інформаційну систему підтримки управління власними криптовалютними активами, яка б дозволила збирати інформацію щодо транзакцій користувача в мережі блокчейн, модифікувати та на вихід давати звіти;
 - 3) Дослідити ефективності розробленого програмного забезпечення.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу

Діаграма сценаріїв використання
 Схема бази даних
 Діаграма класів
 Діаграма послідовностей
 Діаграма компонентів
 Діаграма розгортання
 Екранні форми

7. Орієнтовний перелік публікацій – одна публікація

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання «30» вересня 2020 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання	Примітка
1	Аналіз існуючих рішень	01.09.2021	
2	Формування вимог до системи	07.09.2021	
3	Сценарії використання системи та її структурна схема	14.09.2021	
4	Вибір та обґрунтування елементів та технологій	21.09.2021	
5	Розробка ER-діаграми, та архітектура бази даних	01.10.2021	
6	Реалізація бізнес-логіки системи	20.10.2021	
7	Розроблення інтерфейсу користувача	01.11.2021	
8	Виконання експериментальних досліджень	06.11.2021	
9	Оформлення документації	18.11.2021	
10	Подання дисертації на попередній захист	22.11.2021	
11	Подання дисертації на основний захист	6.12.2021	

Студент

Олег ХАЛІН

Науковий керівник

Олена ЖДАНОВА

РЕФЕРАТ

Розмір пояснювальної записки – 105 аркушів, містить 17 ілюстрацій, 37 таблиць, 48 джерел, 1 додаток.

Актуальність теми. У роботі розглянуто проблему підтримки управління власними криптовалютними активами, показано основні особливості існуючих рішень як фіатних засобів фінансового обліку, так і програмного забезпечення на базі мережі блокчейн, їх переваги та недоліки. На сьогоднішній день криптовалютні активи розвиваються досить швидко, нові криптовалюти випускаються кожний день. Досить багато мереж і бірж, які необхідно контролювати досвідченим користувачам мережі блокчейн. Виявлено потребу користувачів у систематизації та агрегації даних з різних ресурсів в одному місці, менеджменті та описі транзакцій, а також формування звітностей за заданими фільтрами.

Мета дослідження. Основною метою є спрощення управління власними криптовалютними активами за рахунок моніторингу, аналізу та агрегування даних з різних джерел.

Об'єкт дослідження: процес підтримки управління криптовалютними активами.

Предмет дослідження: інформаційна система підтримки управління власними криптовалютними активами користувача.

Для реалізації поставленої мети **сформульовані наступні завдання:**

- провести аналіз проблеми та актуальності теми. Зробити огляд існуючих рішень;
- розробити інформаційну систему підтримки управління власними криптовалютними активами, яка б дозволила збирати інформацію щодо транзакцій користувача в мережі блокчейн, модифікувати та на вихід давати звіти;
- дослідити ефективності розробленого програмного забезпечення.

Наукова новизна результатів магістерської дисертації полягає в тому, що запропоновано архітектурне рішення для побудови програмного забезпечення для

підтримки управління власними криптовалютними активами, яке на відміну від інших надає користувачеві агреговану інформацію з більш детальним описом транзакцій, економить час та ресурси користувача. Результат досягнутий шляхом розробки унікального програмного засобу.

Практичне значення отриманих результатів полягає в тому, що реалізовані методи поєднані в межах однієї Системи і максимально прості у використанні для користувача. Розроблено телеграм бот, який працює кросплатформенно, тому користувач має доступ до Системою в необхідний для нього момент часу. Також реалізовано API-інтерфейс, за допомогою якого результати роботи програми можуть з легкістю отримувати і застосовувати сторонні сервіси. Дана система може бути використана як власниками криптовалютних активів для менеджменту власного портфелю, так і розробниками для інтеграції з системою і розширення її можливостей.

Зв'язок з науковими програмами, планами, темами. Робота виконувалась на кафедрі інформаційних систем та технологій Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського".

Ключові слова: БЛОКЧЕЙН, КРИПТОВАЛЮТНІ АКТИВИ, УПРАВЛІННЯ ФІНАНСАМИ, ФІНАНСОВИЙ ОБЛІК, ТЕЛЕГРАМ БОТ, ІНФОРМАЦІЙНА СИСТЕМА.

ABSTRACT

Explanatory note size –105 pages, contains 17 illustrations, 38 tables, 48 sources, 1 application.

Actuality of theme. The paper considers the problem of supporting the management of own cryptocurrency assets, shows the main features of existing solutions for both fiat financial accounting and blockchain-based software, their advantages and disadvantages. Today, cryptocurrency assets are developing quite rapidly, new cryptocurrencies are issued every day. There are many networks and exchanges that need to be controlled by experienced blockchain users. The need of users to systematize and aggregate data from different resources in one place, management and description of transactions, as well as the formation of reports on the specified filters.

The aim of the study. The main goal is to simplify the management of own cryptocurrency assets through monitoring, analysis and aggregation of data from various sources.

Object of research: information system to support the management of the user's own cryptocurrency assets.

To achieve this goal, the **following tasks** were formulated:

- to analyze the problem and relevance of the topic. Review existing solutions;
- to develop an information system to support the management of own cryptocurrency assets, which would allow to collect information on user transactions in the blockchain network, modify and report on the output;
- to investigate the effectiveness of the developed software.

The scientific novelty of the results of the master's dissertation is that an architectural solution is proposed to build software to support the management of own cryptocurrency assets, which, unlike others, provides the user with aggregated information with a more detailed description of transactions, saves time and resources. The result was achieved by developing a unique software.

The practical value of the obtained results is that the implemented methods are combined within one system and as easy as possible for the user. Developed a telegram bot

that works cross-platform, so the user has access to the system at the right time for him. An API interface has also been implemented, through which the results of the program can be easily received and used by third-party services. This system can be used both by owners of cryptocurrency assets to manage their own portfolio, and developers to integrate with the system and expand its capabilities.

Relationship with working with scientific programs, plans, topics. The work was performed at the Department of Information Systems and Technologies of the National Technical University of Ukraine "Kyiv Polytechnic Institute named after Igor Sikorsky".

Keywords: BLOCKCHAIN, CRYPTOCURRENCY ASSETS, FINANCIAL MANAGEMENT, FINANCIAL ACCOUNTING, TELEGRAM BOTS, INFORMATION APPLICATION.

ЗМІСТ

ВСТУП.....	11
1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	13
1.1 ВЕБ-ЗАСТОСУНОК ZAPPER	13
1.2 ПОДАТКОВЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ZENLEDGER	13
1.3 ВЕБ- ТА МОБ-ЗАСТОСУНОК COINTRACKER	14
1.4 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ТА ПОДАТКОВИЙ ІНСТРУМЕНТ COINTRACKING.....	14
1.5 ПОДАТКОВЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ KOINLY	14
Висновок до розділу	15
2 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ.....	16
2.1 ПРИЗНАЧЕННЯ РОЗРОБКИ	16
2.2 МЕТА ТА ЗАДАЧІ РОЗРОБКИ.....	17
2.3 ОПИС ФУНКЦІОНАЛЬНОЇ МОДЕЛІ	18
Висновок до розділу	21
3 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ.....	22
3.1 ДІАГРАМА СЦЕНАРІЇВ ВИКОРИСТАННЯ	22
3.2 ОПИС ПРЕЦЕДЕНТІВ	22
Висновок до розділу	27
4 СТРУКТУРНА СХЕМА СИСТЕМИ.....	28
4.1 ВХІДНІ ДАНІ	28
4.2 ВИХІДНІ ДАНІ	29
4.3 ДІАГРАМИ ПОСЛІДОВНОСТІ	30
Висновок до розділу	30
5 ВИБІР ТА ОБГРУНТУВАННЯ ЕЛЕМЕНТІВ ТА ТЕХНОЛОГІЙ.....	31
5.1 MIRO	31
5.2 NODE.JS	32

5.3	Typescript	32
5.4	PostgreSQL	33
5.5	Apollo Server i GraphQL	33
5.6	Prisma	34
5.7	Docker	34
5.8	Telegraph	34
5.9	WebStorm	34
5.10	Trello	35
5.11	GitHub та GitHub CI	35
	Висновок до розділу	35
6	ER-діаграма	36
6.1	ER-діаграма	36
6.2	Діаграма компонентів	36
6.3	Опис схеми бази даних	36
	Висновок до розділу	45
7	Реалізація бізнес-логіки системи	46
7.1	Специфікація функцій серверної частини системи	46
7.2	Специфікація функцій клієнтської частини системи	51
7.3	Діаграма розгортання	52
	Висновок до розділу	52
8	Розроблення інтерфейса користувача	53
8.1	Стартовий екран телеграм боту	53
8.2	Екран редагування існуючих API ключів або додавання нових	56
8.3	Екран експорту існуючих транзакцій, за заданими фільтрами	60
8.4	Екран додавання нових готівкових транзакцій	66
8.5	Екран додавання коментарю до існуючих транзакцій	68
	Висновок до розділу	70
9	Тестування системи	71

9.1 МЕТА ВИПРОБУВАНЬ.....	71
9.2 ЗАГАЛЬНІ ПОЛОЖЕННЯ	71
9.3 РЕЗУЛЬТАТИ ВИПРОБУВАНЬ	72
Висновок до розділу	78
10 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ	79
10.1 Опис ідеї проєкту	79
10.2 Технічний аудит ідеї проєкту	83
10.3 Аналіз ринкових можливостей запуску стартап-проєкту	84
10.4 Розроблення ринкової стратегії проєкту	90
10.5 Розроблення маркетингової програми стартап-проєкту	95
Висновок до розділу	99
ВИСНОВКИ.....	100
ПЕРЕЛІК ПОСИЛАНЬ	102

ВСТУП

У сучасному світі фінансовий ринок розвивається дуже швидкими оборотами і одним із новітніх його представників являється ринок криптовалютних активів. На даний момент швидкість росту об'ємів грошей в децентралізованих фінансових продуктах досягає неймовірних показників.

Для прикладу розглянемо період літа 2020 року. Загальна заблокована вартість (TVL), показник обсягу капіталу, заблокованого в протоколах децентралізованих фінансових продуктах становив 1 мільярд доларів в травні 2020 року а під кінець року вже досяг 15,7 мільярдів доларів [1].

Ми розглянули лише DeFi частину систем мережі блокчейн. Давайте подивимось на загально ринкову капіталізацію криптовалют у світі. На даний момент вона становить 2.518.766.119.351 – два з половиною трильйонів доларів. 7 січня 2021 року вона становила один трильйон, це значить що лише за цей рік кількість грошей в криптовалютах збільшилось на 150% [2].

Люди активно купують, продають та обмінюють свої активи. Розробники криптовалютних фінансових продуктів розробляють все більше і більше веб- та моб-додатків для взаємодії з блокчейн мережею, такі як лендінг протоколи, які дозволяють проводити кредитування в децентралізованому світі, електронні гаманці для керування своїми активами, лотереї побудовані на технологіях блокчейну, фармінги, які дозволяють користувачам примножувати свої фінанси, а також біржи, які здебільшого являються кастадіальними і допомагають людям торгувати, прогнозувати і заробляти на своїх активах.

Також збільшується і кількість різноманітних блокчейнів – найпопулярніші з них Ethereum Blockchain [3] та Binance Smart Chain [4], також існує купа менш популярних чейнів, таких як Tezos [5], Solana [6], Cosmos Network [7], Cardano [8], Polygon (MATIC) [9], Tron [10], Polkadot [11] та інші.

На даний момент існує 5 основних і найбільших бірж – Binance [12], KuCoin [13], FTX [14], OKEx [15], Huobi [16]. Є навіть компанії, які створили конструктор криптовалютних бірж і продають їх як сервіс за підпискою.

Підсумував вищесказане можемо зробити висновок – світ криптовалютних активів досяг великих масштабів і це неможливо не замітити. Людям необхідно слідкувати за великою кількістю своїх акаунтів в різних DeFi інструментах і мати можливість управляти своїми активами. Налаштування, підтримка таких системи потребує багато зусиль, часу, а також спеціальних знань. Тому створення інформаційної системи з підтримки управління власними криптовалютними активами є актуальною задачею.

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Веб-застосунок Zapper

Zapper.fi [17] дозволяє вказавши ваш Ethereum адреса, показати всі ваші транзакції і активи з розрахунком прибутковості тієї чи іншої операції. При підключенні вашого Ethereum гаманця, наприклад за допомогою електронного гаманця Metamask [18], до цього сервісу користувач може не тільки відслідковувати звітність, а й виконувати всі можливі дії з обміну активами, вкладати свої активи в пули ліквідності і у різноманітні децентралізовані системи. Zapper.fi підтримує більше 50 відомих DeFi сервісів: Uniswap [19], Yearn Finance [20], 1Inch [21], Curve [22], Balancer [23], Sushiswap [24] і інші.

Але попри наведений функціонал дане програмне рішення надає декілька надлишковий функціонал та у результаті чого користувач отримує досить незручний інтерфейс користувача. Також дана система розрахована лише на блокчейн мережу Ethereum і не дозволяє побачити усі активи користувача в інших мережах або біржах.

1.2 Податкове програмне забезпечення ZenLedger

ZenLedger [25] є програмним забезпеченням, зосередженим на податковому управлінні та обліку DeFi, NFT та криптовалютах. Користувач має змогу створювати звіти про податки на криптовалюту та відстежувати свій криптопортфель. Платформа ZenLedger допомагає криптоінвесторам та податковим спеціалістам у поданні криптовалютних податків та фінансовому аналізі, забезпечуючи цифровий робочий процес для спрощення, оптимізації та автоматизації податкового та бухгалтерського процесу.

Як можна зрозуміти, функціонал даної системи досить зрозумілий, але вузько направлений. Дана система не має інтеграцій з провідними криптовалютними біржами, також немає адаптації для використання на мобільному девайсі. Даний продукт являється платним.

1.3 Веб- та моб-застосунок Cointracker

Cointracker [26] дозволяє користувачам відстежувати ефективність інвестицій та податків криптовалютних активів. Користувачу необхідно підключити свої біржі та гаманці. Дана системи збирає усі баланси та транзакції. Технологія підтримує відстеження комісій, маржинальну торгівлю та більшість інтеграцій, одночасно обробляючи трейдерів з мільйонами транзакцій. Також є функціонал формування звітів і інтеграція з програмами бухгалтерського обліку.

Даний продукт не має підтримки на деяких платформах, а саме: Linux та Windows і варто зауважити, що сумарно майже 90% (за даними на лютий 2020р.) користувачів ПК користуються саме цими операційними системами. Дане програмне забезпечення має систему платної підписки. Також як зазначено на офіційному сайті, дана система має повну підтримку функціоналу лише для 4 країн – США, Великобританії, Канади та Австралії, для всіх інших тільки часткова підтримка.

1.4 Програмне забезпечення для аналізу та податковий інструмент Cointracking

CoinTracking [27] аналізує операції користувачів та створює звіти в режимі реального часу про прибутки та збитки, вартість криптовалютних активів, реалізовані та нереалізовані прибутки, звіти про податки. Є інтеграція з частиною бірж. Експорт даних в Excel, PDF, CSV, XML і JSON форматі.

Дане програмне рішення має надлишковий функціонал. Система також є платно і обмежений функціонал у деяких країнах. Основна мета – формування податкових звітів, а не ведення фінансів.

1.5 Податкове програмне забезпечення Koinly

Koinly [28] - програмне забезпечення для оподаткування криптовалютних активів. Продукт має інтеграції з основними блокчейнами та біржами. Має функціонал створення податкових звітів, а також їх експорт.

Дане програмне рішення потребує спеціальний формат файлу для внесення облікових даних, а саме CSV і це є певним недоліком для користувача, тому що не завжди є можливість згенерувати CSV файли. Також основна мета даного продукту – формування звітностей для податків, а не управління фінансами. Дана система є також платною.

Висновок до розділу

Ми переглянули декілька рішень для бухгалтерського обліку криптовалют, і ось результати, як ми бачимо, ринок пропонує нам велику різноманітність програмних рішень щодо обліку криптовалютних активів та податків. Деякі з цих рішень не сумісні з більшістю чейнів, деякі з них застаріли або мають не досить зрозумілий інтерфейс для користувача. Деякі недоступні на мобільних пристроях, а також більшість із них – платні і не покривають більшість потреб користувачів. Основне направлення даних систем – формування податкової звітності, а не управління власними активами.

2 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ

2.1 Призначення розробки

Розроблювана система призначена для підтримки управління власними криптовалютними активами, її реалізацію у вигляді телеграм боту. Ця система може використовуватись будь-яким користувачем, який має серед своїх активів криптовалюту. Вона є не просто корисною, а необхідною для значної кількості користувачів, котрим необхідно агрегувати декілька джерел: різні електронні гаманці на різних чейнах, різні криптовалютні біржі.

На ринку існує ряд різноманітних систем, якими користуються більшість розвинутих користувачів, однак ми націлились на необхідні користувачу задачі, які не були реалізовані в більшості варіантів. А саме: автоматизація збирання активностей користувачів з основних чейнів і основних бірж на ринку. Автоматичне визначення еквіваленту до долара, та статус транзакції, а також можливість додавати коментар до кожної транзакції. Додаємо до цього всього зручний інтерфейс і кросплатформеність у вигляді телеграм бота, де користувач в найшвидшому вигляді може прослідкувати за новими транзакції. В результаті матимемо зручну систему, якою зможуть користуватися більшість людей для ведення обліку своїх активів.

Також, для більш розвинутих користувачів буде представлено готовий для використання API, завдяки якому розробники зможуть створювати власні інтерфейси та інтегрувати функціонал під будь-яким потреби.

Ми створюємо продукт персонально за заданими умовами і можемо в будь-який час його кастомізувати під потреби користувача. Для нас це не просто теорія або тестовий проект, для нас це реальна розробка, яка буде неодмінно в майбутньому приносити користь. Її функціонал буде економити час та гроші користувачу, а детально розроблений дизайн і юзер флоу клієнтської частини Системи дозволить користувачам легко і інтуїтивно зрозуміло виконати всі необхідні задачі та мати всю необхідну інформацію.

Інформаційна система повинна бути:

- надійною — для того, щоб усі процеси відбувались без перешкод;
- адаптивною — що значить система має бути зручною для майбутнього розширення функціоналу;
- кросплатформною — для того щоб не обмежувати цільову аудиторію користувачів у рамках певної платформи, так як криптовалютний світ представляє користувачам продукти під різні девайси;
- інтуїтивною у користуванні;
- не перенасиченою зайвими функціями, одна при цьому функціонально повною.

Даний проект може бути використаний у якості прикладу клієнт-серверного застосунку у будь-якій предметній області. За мету було взято створити як можна більш гнучкішу архітектуру для подальшого використання у інших кросплатформних систем.

2.2 Мета та задачі розробки

Мета розробки – спрощення управління власними криптовалютними активами за рахунок моніторингу, аналізу, агрегування та оптимізації даних з різних джерел.

Для досягнення поставленої цілі необхідно виконати наступні задачі:

- розробити прототипи та юзер флоу поведінки користувача при взаємодії з Telegram ботом;
- спроектувати архітектуру серверної частини Системи;
- розробити усі необхідні сцени Системи згідно з прототипами;
- інтегрувати серверну частину у клієнтські частини Системи;
- провести тестування Системи на предмет наявності логічних та програмних помилок.

2.3 Опис функціональної моделі

Програмний продукт буде реалізовано у вигляді телеграм боту та API-інтерфейсу, які буде розміщено на сервері. Користувачі матимуть змогу взаємодіяти з інтерфейсом Системи за допомогою телеграм боту.

У системі присутні 3 типи акторів:

- адміністратор;
- анонімний (неавторизований) користувач;
- авторизований користувач.

Для кожного користувача в базі даних визначений його тип. Тому після авторизації анонімний користувач змінить свій тип на той, який визначений у базі даних.

У таблиці 2.1 описані актори, варіанти використання та їх описи дій.

Таблиця 2.1– Актори, варіанти використання та їх описи дій.

Актор	Варіант використання	Опис дії варіанта використання
Анонімний користувач	Запуск бота	Користувач може, знайшовши адресу бота в пошуковій секції месенджера, почати з ним роботу натиснувши команду “/start”.
Анонімний користувач	Додавання API ключів	Користувач може додати свої API ключі для бірж: - Binance; - FTX; - OKEX; - Huobi; - KuCoin.

		Для цього йому необхідно ввести: - API key; - API secret; - API passphrase (для OKEX). Або адресу електронного гаманця, для цього його необхідно ввести адресу гаманця. При цьому його дані разом з ідентифікатором телеграму запишуться в базу даних.
Авторизований користувач	Зміна API ключів	Користувач може змінити API ключі для бірж або електронного гаманця за своїм ідентифікатором телеграму.
Авторизований користувач	Пошук по транзакціям	Користувач може зробити пошук по своїм транзакціям за наступними ресурсами: - Біржа; - Електронний гаманець на базі мережі Ethereum. Та за періодом часу: - Сьогоднішні; - Вчорашні; - Цього місяця; - Цього року; - Нестандартна дата. Як результат він отримає свої транзакції повідомленнями або повідомлення що транзакцій не було знайдено.

Авторизований користувач	Додавання/зміна коментару транзакції	Користувач може додати або змінити коментар транзакції, обравши необхідну для модифікації транзакцію.
Авторизований користувач	Додавання готівкової або фіатної транзакції	Користувач може додати нову транзакцію, яка була проведена готівкою або з платіжної карти. Увівши необхідні поля: - Час проведення; - Валюту: USD, UAH чи EUR; - Суму; - Чи є ця транзакція витратою; - Еквівалент до долара на момент проведення транзакції; - Коментар; - З якого акаунта була транзакція – особистого чи ФОПа.
Авторизований користувач	Пошук по транзакціям	Користувач може зробити експорт своїх транзакціям відфільтрувавши за наступними ресурсами: - Біржа; - Електронний гаманець на базі мережі Ethereum. Та за періодом часу: - Сьогоднішні; - Вчорашні; - Цього місяця; - Цього року; - Нестандартна дата.

		Як результат він отримає файл формату .csv який зможе редагувати та переглядати.
--	--	--

Висновок до розділу

У даному розділі описано призначення розробки, обґрунтовано цілі та задачі розробки. Розділ містить відомості про типи користувачів в межах розробленого програмного продукту та функції, які необхідно реалізувати. В процесі роботи було виявлено основні вимоги для подальшої розробки системи.

3 СЦЕНАРІЇ ВИКОРИСТАННЯ СИСТЕМИ

3.1 Діаграма сценаріїв використання

Згідно з заданими функціональними умовами було розроблено діаграму сценаріїв використання системи.

Користувач має змогу додавати та редагувати API ключі та адресу електронного гаманця на базі мережі Ethereum, отримати оповіщення про нові транзакції, додавати до них коментар, додавати готівкові транзакції, а також шукати, фільтрувати та експортувати транзакції.

Діаграму сценаріїв використання наведено у графічних матеріалах у додатку А.

3.2 Опис прецедентів

Згідно з заданими функціональними умовами та опираючись на діаграму сценаріїв використання було проведено детальний опис кожного прецеденту. Дана інформація описана в таблиці 3.1-3.3

Таблиця 3.1– Опис прецеденту додавання/редагування API ключа в систему

Назва	Додавання/редагування API ключа в систему
ID	UC_001
Опис	Користувач хоче додати новий API ключ або редагувати існуючий, щоб система відображала його транзакції з даного ресурсу
Актор	Користувач
Переваги для системи	Користувач більше використовує систему тому що може отримувати інформацію з більшої кількості ресурсів
Частота використання	90% користувачів системи будуть додавати API ключі, так як це основа роботи системи

	10% користувачів будуть редагувати API ключі, так як зазвичай їх рідко змінюють
Тригер	Користувач натискає кнопку “Add api keys” або “Edit api keys”
Передумови	API ключі обраного ресурсу збережені у системі
Постумови	1. Користувач обирає додати новий API ключ або оновити існуючий 2. Користувач обирає ресурс, який він хоче додати 3. Система пропонує ввести дані запису (AC1) 4. Система зберігає API ключі в базу даних (EX1) 5. Система виводить повідомлення “API ключі були успішно збережені” Користувачу відображаються нові транзакції у випадку, якщо вони існують
Основна поведінка	AC1 – Система обробляє введені дані 1. Користувач вводить API key 2. Користувач вводить API Secret 3. Система валідує API ключі (AC2) Система повертається на Постумову пункт 4
Альтернативні поведінки	AC2 – Система визначає, що API ключі не валідні 1. Система пропонує користувачу ввести валідні API ключі Система повертається на Постумову пункт 4
Виключення	EX1 – Сервер не відповідає або API ключі не були додані

	1. Система виводить повідомлення “Виникла проблема сервера” 2. Система повертається на Постумову пункт 1
--	---

Таблиця 3.2– Опис прецеденту додавання/редагування коментару до транзакції

Назва	Додавання/редагування коментару до транзакції
ID	UC_002
Опис	Користувач хоче додати новий коментар до транзакції або редагувати існуючий
Актор	Користувач
Переваги для системи	Користувач може більш якісно керувати своїми активами, що дає більший рейтинг системі.
Частота використання	80% користувачів системи будуть додавати API ключі, так як це важлива складова фінансового обліку
Тригер	Користувач отримує повідомлення про нову транзакцію та натискає кнопку “Add comment” або “Edit comment”
Передумови	Транзакція оновилась з полем коментару
Постумови	1. Користувач обирає додати новий коментар або оновити існуючий до транзакції 2. Система пропонує ввести текст коментару (AC1) 3. Система зберігає коментар в базу даних в базу даних (EX1) 4. Система виводить повідомлення “Коментар до транзакції був успішно збережений”

	Повідомлення транзакції оновлюється відповідно з коментарем користувача
Основна поведінка	АС1 – Система обробляє введені дані 1. Користувач вводить текст коментарю Система повертається на Постумову пункт 3
Виключення	ЕХ1 – Сервер не відповідає або коментар не було додано 1. Система виводить повідомлення “Виникла проблема сервера” 2. Система повертається на Постумову пункт 1

Таблиця 3.3– Опис прецеденту пошуку/експорту транзакцій

Назва	Пошук/експорт транзакцій
ID	UC_003
Опис	Користувач хоче знайти або експортувати транзакції за обраними фільтрами
Актор	Користувач
Переваги для системи	Користувач буде мати більш гнучкий функціонал, що дозволить підвищити рейтинг системи
Частота використання	65% користувачів системи будуть шукати/експортувати дані по транзакціям, так як це частина фінансового обліку
Тригер	Користувач натискає кнопку “Search” або “Export”

Передумови	Користувач отримує транзакції за заданими фільтрами у вигляді повідомлень або файлу формату .csv
Постумови	1. Користувач обирає пункт меню, що відповідає за пошук по транзакціям або їх експорт 2. Користувач обирає фільтрацію за ресурсом 3. Користувач обирає фільтрацію за діапазоном дат (AC1) 4. Система повертає список транзакцій за введеними фільтрами користувачу (EX1) Користувачу відображаються транзакції за введеними фільтрами або відправляє .csv файл
Основна поведінка	AC1 – Система обробляє введені дані 1. Користувач обирає нестандартний діапазон дат 2. Користувач вводить діапазон дат – початкову і кінцеву 3. Система валідує введену дату (AC2) Система повертається на Постумову пункт 4
Альтернативні поведінки	AC2 – Система визначає, що введена дата не валідні 1. Система пропонує користувачу ввести валідну дату Система повертається на Постумову пункт 4
Виключення	EX1 – Сервер не відповідає або стався збій при виведенні транзакцій

	1. Система виводить повідомлення “Виникла проблема сервера” 2. Система повертається на Постумову пункт 1
--	--

Висновок до розділу

Даний розділ присвячений сценаріям використання системи. Було аналізовано та побудовано use-case діаграму. Опираючись на вимоги до систему було проведено детальний аналіз поведінки користувача та описано кожний прецедент з усіма варіантами поведінки користувача та системи.

4 СТРУКТУРНА СХЕМА СИСТЕМИ

4.1 Вхідні дані

Вхідні дані отримуються з наступних джерел:

- Користувачі;
- API Binance біржі;
- API FTX біржі;
- API OKEX біржі;
- API Huobi біржі;
- API KuCoin біржі;
- API Covalent – для отримання даних за адресою електронного гаманця.

Дані від користувача надходять з телеграм боту, в якому користувач відправляє повідомлення з необхідними даними.

Для роботи системи користувач повинен у відповідному меню надати наступні дані:

- Binance API key;
- Binance API secret;
- FTX API key;
- FTX API secret;
- OKEX API key;
- OKEX API secret;
- OKEX API passphrase;
- Huobi API key;
- Huobi API secret;
- KuCoin API key;
- Адресу електронного гаманця на базі мережі Ethereum.

Для додавання коментару до транзакції – користувач повинен ввести сам текст коментару.

Інформація про транзакції отримується зі сторонніх API сервісів. Таким чином економиться час на розробку, розширюється функціонал системи, завдяки саме внутрішніх полів, яких немає в блокчейні, а також знімається відповідальність за правдивість інформації.

Інформація стосовно готівкових або фіатних транзакцій надходить також від користувача через телеграм бота.

4.2 Вихідні дані

Вихідними даними є інформація яку може бачити користувач у повідомленнях або в експортованих файлах. Вихідними даними є транзакції, які надходять з різних бірж або мережі Ethereum. Інформація щодо транзакції містить наступні поля:

- дата проведення;
- номер блоку;
- хеш транзакції;
- адреса відправника;
- адреса отримувача;
- валюта;
- значення;
- комісія;
- сума в доларах;
- баланс на момент проведення операції;
- чи є транзакція відправленням чи отриманням;
- коментар.

4.3 Діаграми послідовності

Діаграма послідовності описує процес додавання користувачем API ключа і витягування транзакції за заданими даними.

Діаграму послідовності наведено у графічних матеріалах у додатку Г.

Висновок до розділу

У даному розділі був проведений аналіз системи та створена її структурна схема. Було розглянуто усі вхідні дані та описано джерела з яких ці дані отримуються. В нашому випадку це користувачі та стороні API. По суті в системі один головний актор, який отримує транзакції з деталями як вихідні дані з системи. Також була розроблена діаграма послідовності, що описує взаємодію користувача із системою.

5 ВИБІР ТА ОБГРУНТУВАННЯ ЕЛЕМЕНТІВ ТА ТЕХНОЛОГІЙ

Для розробки основи програмного забезпечення, брейн штормів та формування задач і структури Системи використовувався сервіс Miro [29]. Для розробки використовувалося оточення розробки JetBrains WebStorm [30] та система контролю версій Github [31]. Мовою програмування як для серверного, так і для клієнтського частини Системи було обрано JavaScript [32] та його розширення TypeScript [33]. Для розробки серверну частини використовувався Node.js [34], для бази даних – PostgreSQL [35]. Вид API – GraphQL [36]. Для взаємодії з базою даних – бібліотека Prisma. Як допоміжний інструмент для створення клієнтської частини Системи а саме Telegram [37] бота – Telegraph [38]. Для постановки задач і контролю ходу розробку використовувалась система управління проектами Trello [39]. Для створення проектної документації використовувалися текстовий процесор Microsoft Word [40] та веб-застосування Draw.io [41].

5.1 Miro

Miro – це онлайн інструмент для взаємодії команди в процесі брейн штормів, створення прототипів, поставлення цілей та розробки аналізу Системи. Використовувалась дана система саме для швидкої комунікації і розроблення стратегії створення доданку. В даному інтерфейсі було розроблено усі ідеї і концепції Системи. Створено флоу продукту та початкову структуру базу даних.

Обрали саме дане програмне забезпечення ми з наведених нижче причин:

- безкоштовний інструмент, з обширною кількістю функцій;
- зберігається історія змінень, не раз даний функціонал допомагав повернути втрачені напрацювання;
- присутня вже готова колекція шаблонів для різних видів задач, що заощадило досить багато часу;
- синхронізація з google drive [42];
- швидкий доступ до системи з браузера.

5.2 Node.js

Node.js – крос-браузерна, open-source програмна платформа. Так як я довгий час розробляв клієнтські застосунки, то node.js для мене був інструментом з мінімальним порогом входження. На доданок, на JS можна розробляти як серверну частину, так і телеграм бот і клієнтську частину. Це зберігає досить багато часу і є досить доречним для такої невеликої команди. Однак також було проведено аналіз даної платформи, який допоміг мені виділити наступні переваги.

Ця система керована подіями і працює асинхронно. Вибудовує чергу за пріоритетністю. Коли до сервера одночасно підключаються тисячі людей, їй легко впоратися з навантаженням, тож при грамотному підході до розробку застосунків вона являє собою досить потужний інструмент.

При правильному підході до архітектури кодової бази, масштабування системи не викличе жодних проблем.

Велика кількість плагінів та пакетів, що допомагає скоротити час розробки.

Платформа досить швидко розвивається і оновлюється. Пропонує новітні підходи до вирішення проблем.

5.3 TypeScript

TypeScript – сучасний інструмент, який був розроблений компанією Microsoft. Він позиціонується, як засіб програмної розробки, який розширює можливості JavaScript. Здається на даний момент, будь-який поважаючий свій час і чистоту коду JavaScript розробник використовує та надає перевагу цій бібліотеці. Переваг досить багато, однак я виділив найбільш значущі нижче.

TypeScript працює поверх JavaScript, компілюється у звичайний JavaScript та обробляє будь-які можливі прогалини у функціоналі. Завдяки даному функціоналу є можливість налаштовувати конфігурації і компілювати код в необхідну версію ES, при цьому використовуючи останні переваги новітнього синтаксису.

Статична типізація. Допомагає зробити код більш передбачуваним, уникати значну кількість помилок ще на етапі розробки програмного коду.

Завдяки підтримки ООП є можливість створювати і наслідувати класи, а синтаксис нагадує синтаксис C# [43] або Java [44].

Можливість створювати власні типи і, при необхідності, наслідувати їх.

5.4 Postgresql

Postgresql – база даних для постійного зберігання даних застосунку. По-перше була обрана реляційна база даних, так як ми маємо досить велику кількість таблиць і зв'язків між ними. А дана база даних є найменш обмеженою і найбільш потужною на даний момент. Із найзначніших переваг, виділив наступні:

- підтримує необмежену кількість таблиць і їх розмірів;
- потужні та надійні механізми транзакцій та реплікації;
- опен-сорс;
- підтримка багатовимірні масивів;
- підтримка json;
- велика кількість бібліотек для інтеграції і взаємодії.

5.5 Apollo Server і GraphQL

Apollo Server [45] – завдяки GraphQL можна надіслати запит до свого API та отримати саме той набір даних, який необхідний, не більше і не менше. Це мабуть головна відмінність від REST API. Основні переваги, чому було обрано саме ці технології описано нижче.

Зводить до мінімуму обсяг даних, що передаються по API. Не додає не необхідних запитів і полів.

Кеширує запити і тим самим скорочує час відповіді серверу на запити клієнта.

Можливість створити декілька успадкованих запитів в одному. Тобто якщо існує модель користувача і модель коментарю – усувається необхідність двох запитів для витягування Користувачів і їх коментарів, це можна зробити в одному запиті.

Сумісність із сучасними системами.

5.6 Prisma

Prisma [46] – це бібліотека на стороні сервера, яка допомагає системі читати та записувати дані в базу даних інтуїтивно зрозумілим і безпечним способом. Він дозволяє взаємодіяти з базою даних і розуміти все об'єктами. Як для JavaScript розробника – це дуже звичний спосіб роботи з інформацією. Нижче описано переваги завдяки яким було обрано дану бібліотеку.

Усі результати запитів Prisma Client повністю типізовані. Так як в проєкті було використано Typescript – це дуже корисний функціонал.

Має вбудований адміністратор бази даних. Який працює як новітня версія PHPMyAdmin [47].

Дана бібліотека є open-source.

Також це абсолютно новий підхід для роботи з базами даних.

5.7 Docker

Docker [48] – програмне середовище для автоматизації розгортання та керування програмних середовищ. Спрощує процес розробки, розділяє все по контейнерам, ізолює запуск застосунки.

5.8 Telegraph

Telegraph – бібліотека, яка допомагає розробляти телеграм боти з наявними класами і функціями взаємодії з Telegram API. Має типізацію і сумісна з TypeScript. Корисний і зрозумілий інструмент для швидкої розробки.

5.9 WebStorm

WebStorm – новітній розумний редактор коду. Націлений в першу чергу на мови програмування JavaScript та TypeScript. Має в своєму арсеналі досить поширену бібліотеку плагінів. Головна перевага при виборі редактора – звичність для розробника. Я працюю з даним середовищем більше 5 років – це і було основною причиною вибору даної програми.

5.10 Trello

Trello - система управління проектами у вигляді канбана. Безкоштовна зручна у використанні система. Мінімалістичний інтерфейс і неперенасиченість функціоналом.

5.11 GitHub та GitHub CI

GitHub – систем контролю версій. Мабуть найпопулярніша платформа для розробки програмних продуктів.

GitHub CI - інструмент для автоматизації тестування та доставки кодової бази у production середовище.

Висновок до розділу

Даний розділ присвячений вибору та обґрунтуванню технологій, фреймворків, бази даних та інструментів, на яких будується система. Було проведено детальний аналіз кожної системи і їх документацій. Опираючись на важливі для розробки характеристики виписані основні причини вибору саме таких інструментів розробки.

6 ER-ДІАГРАМА

6.1 ER-діаграма

Діаграма описує схему графового інтерфейсу Системи. Кожна сутність є графовим об'єктом. В об'єктах є атрибути, які описують його характеристики. Також у деяких об'єктів присутні методи, вони дозволяють визначити деякі поля за певною логікою, яка описується у методі.

Так наприклад є мутації для додавання чи оновлення даних, а також резолвери для отримання даних з бази даних за допомогою серверної частини Системи.

ER-діаграму наведено у графічних матеріалах у додатку В.

6.2 Діаграма компонентів

На діаграмі компонентів відображені серверна та клієнтські частини Системи з внутрішніми складовими.

Ми маємо серверну частину, яка складається з бази даних та функціональної реалізації Системи, а також клієнтську частину, яка реалізована у вигляді телеграм боту.

В серверній реалізації у нас є графові об'єкти, моделі а також роутер та графова схема.

В клієнтській реалізації ми маємо функції, запити та компоненти. Також у нас телеграм бот має реалізацію серверної частини для прийому нових повідомлень, тому також має роутер та мутації.

Діаграму компонентів наведено у графічних матеріалах у додатку Г.

6.3 Опис схеми бази даних

В системі всього розроблено 8 сутностей, а саме:

- 3 сутностей транзакцій: по біржам, по гаманцям та готівка;
- 4 сутності API ключів, для наступних бірж: Binance, FTX, OKEX, Huobi;
- Сутність користувача.

Детальний опис сутностей наведено в таблиці 6.1. Схему бази даних зображено на рисунку 6.1.

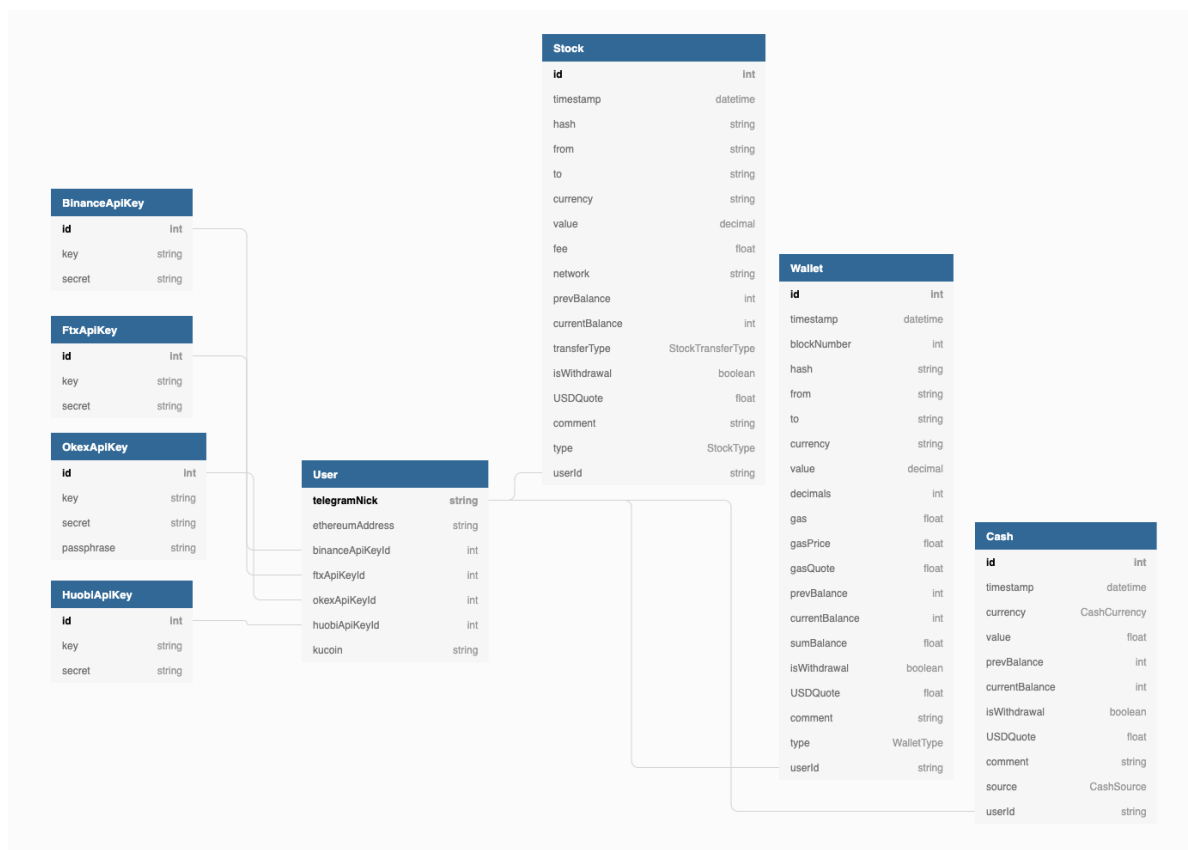


Рисунок 6.1 – Схема бази даних

Таблиця 6.1– Перелік таблиць, які описують бізнес логіку Системи.

Таблиця	Опис таблиці	Поля	Тип	Опис поля
BinanceApiKey	Таблиця з ключами доступу до біржи Binance	id	int	Ідентифікатор
		key	string	Ключ біржі, що дає змогу отримувати інформацію по транзакціям користувача
		secret	string	Секрет - шифрування біржі, для підключення до API

FtxApiKey	Таблиця з ключами доступу до біржи FTX	id	int	Ідентифікатор
		key	string	Ключ біржі, що дає змогу отримувати інформацію по транзакціям користувача
		secret	string	Секрет - шифрування біржі, для підключення до API
OkeхApi Key	Таблиця з ключами доступу до біржи OKEX	id	int	Ідентифікатор
		key	string	Ключ біржі, що дає змогу отримувати інформацію по транзакціям користувача
		secret	string	Секрет - шифрування біржі, для підключення до API
		passphrase	string	Перевірна фраза для доступу до API. Виступає в ролі пароля
HuobiApi Key	Таблиця з ключами доступу до біржи Huobi	id	int	Ідентифікатор
		key	string	Ключ біржі, що дає змогу отримувати інформацію по транзакціям користувача
		secret	string	Секрет - шифрування біржі, для підключення до API

User	Користувач	telegramNick	string	Ідентифікатор, що також є ніком користувача у телеграмі
		ethereumAddress	string	Адреса акаунта користувача в мережі Ethereum
		binanceApiKey	int	Вказівник на запис з ключами доступу користувача до біржі Binance
		ftxApiKey	int	Вказівник на запис з ключами доступу користувача до біржі FTX
		okexApiKey	int	Вказівник на запис з ключами доступу користувача до біржі OKEX
		huobiApiKey	int	Вказівник на запис з ключами доступу користувача до біржі Huobi
		kukoin	string	Ключ доступу користувача до біржі Kucoin
Stock	Транзакції, що були виконані через біржі	id	int	Ідентифікатор
		timestamp	datetime	Дата проведення транзакції
		hash	string	Хеш транзакції
		from	string	Адреса акаунта з якого

				відправили кошти. Для вихідних транзакцій – адреса користувача
		to	string	Адреса акаунта на який відправили кошти. Для вихідних транзакцій – адреса користувача
		currency	string	Валюта, в якій були відправленні кошти
		value	int	Сума в валюті, на яку була виконана транзакція
		fee	float	Значення комісії
		network	string	Назва чейну
		prevBalance	int	Баланс користувача в даній валюті перед проведенням транзакції
		currentBalance	int	Баланс користувача в даній валюті після проведенням транзакції
		transferType	StockTransferType	Тип транзакції. Його типом в базі даних є enum. Який може приймати наступні значення: INTERNAL – для транзакцій всередині біржи, EXTERNAL – для

				транзакцій з біржі в інші ресурси (наприклад гаманець)
		isWithdrawal	boolean	Чи транзакція є вихідною
		USDQuote	float	Ціна однієї одиниці валюти в доларах
		comment	string	Коментар користувача до транзакції
		type	stockType	Тип біржі. Його типом в базі даних є enum. Який може приймати наступні значення: BINANCE, OKEX, KUCOIN, FTX, HUOBI – відповідають назвам бірж
		userId	string	Вказівник на запис користувача, якому належить дана транзакція
Wallet	Транзакції, що були виконані через електроні гаманці	id	int	Ідентифікатор
		timestamp	datetime	Дата проведення транзакції
		blockNumber	int	Номера блока в блокчейні, в котрому була виконана транзакція
		hash	string	Хеш транзакції

		from	string	Адреса акаунта з якого відправили кошти. Для вихідних транзакцій – адреса користувача
		to	string	Адреса акаунта на який відправили кошти. Для вихідних транзакцій – адреса користувача
		currency	string	Валюта, в якій були відправленні кошти
		value	decimal	Сума, на яку була виконана транзакція. Сума в базі даних записується без decimals. Тобто помножена на 10 в ступені значення decimals
		decimals	int	Кількість десятків валюти
		gas	float	Значення комісії
		gasPrice	float	Ціна комісії
		gasQuote	float	Ціна одиниці комісії в доларах
		prevBalance	int	Баланс користувача в даній валюті перед проведенням транзакції
		currentBalance	int	Баланс користувача в даній валюті після проведення транзакції

		nse		валюті після проведенням транзакції
		sumBalance	float	Сумарний баланс у доларах з усіх валют
		isWithdrawal	boolean	Чи транзакція є вихідною
		USDQuote	float	Ціна однієї одиниці валюти в доларах
		comment	string	Коментар користувача до транзакції
		type	WalletType	Тип електронного гаманця. Його типом в базі даних є enum. Який може приймати наступні значення: ETHEREUM, TEZOS – відповідають мережам блокчейну гаманців
		userId	string	Вказівник на запис користувача, якому належить дана транзакція
Cash	Транзакції, що були виконані за готівку або фіат (гроші не в мережі	id	int	Ідентифікатор
		timestamp	datetime	Дата проведення транзакції
		currency	CashCurrency	Валюта, в якій була виконана транзакція. Його типом в базі даних є enum.

	блокчейн)			Який може приймати наступні значення: USD, UAH, EUR
		value	float	Сума в валюті, на яку була виконана транзакція
		prevBalance	int	Баланс користувача в даній валюті перед проведенням транзакції
		currentBalance	int	Баланс користувача в даній валюті після проведенням транзакції
		isWithdrawal	boolean	Чи транзакція є вихідною
		USDQuote	float	Ціна однієї одиниці валюти в доларах
		comment	string	Коментар користувача до транзакції
		source	CashSource	Вид, акаунт, з якого була виконана транзакція. Його типом в базі даних є enum. Який може приймати наступні значення: PERSONAL, FOP
		userId	string	Вказівник на запис користувача, якому

				належить дана транзакція
--	--	--	--	--------------------------

Висновок до розділу

В даному розділі на основі вимог до системи і попереднього аналізу було розроблено схему графового Системи інтерфейсу у вигляді ER-діаграми. Описано об'єкти, їх атрибути та методи. Розроблено зв'язки між ними та описані основні мутації та запити. Також була розроблена діаграма класів, що складається з бази даних, серверної частини та клієнтської частини.

Було розроблено схему бази даних та продемонстровано у вигляді ілюстрації з усіма зв'язками. Описано усі таблиці бази даних їх описи, поля, типи полів та основні характеристики.

7 РЕАЛІЗАЦІЯ БІЗНЕС-ЛОГІКИ СИСТЕМИ

7.1 Специфікація функцій серверної частини Системи

Специфікація основних функцій коду серверної частини Системи, задіяних у роботі програми наведено у таблицях 7.1-7.4.

У таблиці 7.1 наведено основні функції взаємодії з базою даних. В яку входять функції створення та оновлення інформації, а також функції для витягування необхідної інформації для користувача.

У таблиці 7.2 наведено функції для організації процесу збирання та обробки даних щодо криптовалютного електронного гаманця за заданою адресою на базі мережі Ethereum.

У таблиці 7.3 наведено функції для організації процесу збирання обробки даних з біржі Binance за заданими API ключами. Реалізація функцій роботи з іншими біржами досить подібна, тому вони не були додані то роботи.

У таблиці 7.4 наведено основні функції роботи серверної частини Системи а також допоміжні функції що використовуються в роботі системи.

Таблиця 7.1– Перелік основних функцій роботи з базою даних

Назва	Параметри	Примітка
resolveUserByTelegramNick()	telegramNick	Створює запит в базу даних для отримання API ключів користувача по його нікнейму в мережі Telegram. Віддає дані у вигляді відповіді на GraphQL запит.
resolveWalletTransactionsByUserTelegramNick()	telegramNick startDate endDate	Створює запит в базу даних для отримання транзакцій користувача по його нікнейму в мережі Telegram. У випадку необхідності фільтрує дані по

		заданим в запиті датам стартовою та кінцевою. Віддає дані у вигляді відповіді на GraphQL запит.
CreateOrUpdateUserMutation()	telegramNick ethereumAddress binance okex kucoin ftx huobi	Отримує нікнейм користувача та API ключі. Перевіряє чи дійсні API ключі. У випадку коли в базі не знайдено користувача із заданим нікнеймом – створює новий запис і додає API ключі. У випадку коли користувач вже існують – оновлює його API ключі.
CreateOrUpdateWalletTransactionCommentMutation()	id comment	Створює або оновлює коментар до транзакції, що була виконана через електронний гаманець за її ідентифікатором. Записує його до бази даних.
createOrUpdateStockTransactionCommentMutation()	id comment	Створює або оновлює коментар до транзакції, що була виконана через одну із бірж за її ідентифікатором. Записує його до бази даних.
CreateCashTransactionMutation()	timestamp currency value isWithdrawal USDQuote comment source userId	Створює нову транзакцію по готівковим або фіатним операціям. Записує її до бази даних.

CreateOrUpdateCashTransactionCommentMutation()	id comment	Створює або оновлює коментар до готівкової або фіатної транзакції. Записує його до бази даних.
--	---------------	--

Таблиця 7.2– Перелік основних функцій отримання та обробки інформації щодо транзакцій електронних гаманців мережі Ethereum

Назва	Параметри	Примітка
getCovalentData()	address startBlock	Створює запит на сторонній API для отримання інформації щодо транзакцій в мережі Ethereum. Повертає масив транзакцій.
getEthereumBalance()	address symbol	Створює запит на сторонній API для отримання інформації щодо балансу користувача за його адресом. У випадку якщо вказано деякий символ – повертає інформацію щодо балансу саме цього активу.
getEthereumTransactions()	telegramNick ethereumAddress	Перевіряє існуючу адресу електронного гаманця користувача. У випадку помилки – видаляє адресу з бази даних і відправляє користувачу відповідне повідомлення. Збирає всі транзакції користувача починаючи з дати останньої доданої в базу і додає їх до бази даних.

		Відправляє користувачам повідомлення з транзакціями і можливістю додати коментар до них.
getEthereumUse rsTransactions()		Отримує всі транзакції в мережі Ethereum усіх користувачів системи. Виконується раз на 10 хвилин.

Таблиця 7.3– Перелік основних функцій отримання та обробки інформації щодо транзакцій біржі Binance

Назва	Параметри	Примітка
getBinanceSign()	query secret	Створює і повертає Sign для доступу до API Binance і формування запитів.
getBinanceTime stamp()	date	Створює і повертає необхідний для API Binance формат теперішньої дати і часу.
getBinanceBala nces()	key secret	Перевіряє існуючі ключі доступу до API Binance користувача. У випадку помилки – видаляє ключі з бази даних і відправляє користувачу відповідне повідомлення. Отримує баланси користувача існуючих у нього монет.
getBinanceDepo sits()	startTime key secret	Отримує усі вхідні транзакції користувача починаючи з заданої дати.

getBinanceWithdrawals()	startTime key secret	Отримує усі вихідні транзакції користувача починаючи з заданої дати.
getBinanceTransactions()	telegramNick binance	Збирає всі транзакції користувача починаючи з дати останньої доданої в базу. Сортує транзакції від найстаріших до найновіших і додає їх до бази. Відправляє користувачам повідомлення з транзакціями і можливістю додати коментар до них.
getBinanceUsersTransactions()		Отримує всі транзакції з біржи Binance усіх користувачів системи. Виконується раз на 10 хвилин.

Таблиця 7.4– Перелік основних загальних функцій back-end частини системи

Назва	Параметри	Примітка
getUSDQuote()	currency	Створює запит на сторонній API для отримання інформації щодо ціни криптовалюти в доларах.
areAddressesEqual()	address1 address2	Допоміжна функція. Перевіряє чи рівні адреси електронних гаманців у мережі Ethereum.
schedule()		Функція, що виконує основні необхідні функції для отримання і обробки даних з заданою періодичністю.

getContext()		Отримує контекст Системи для роботи з базою даних.
--------------	--	--

7.2 Специфікація функцій клієнтської частини Системи

Специфікація основних функцій коду клієнтської частини Системи, задіяних у роботі програми наведено у таблиці 7.5.

Таблиця 7.5– Перелік основних функцій телеграм боту

Назва	Параметри	Примітка
setApi()	ctx	Створює ентріпоінт на клієнтській стороні для отримання інформації щодо нових транзакцій користувача.
getBackKeyboard()	ctx	Допоміжна функція. Виводить меню з функціоналом повернення системи до минулого положення.
getRetryKeyboard()	ctx	Допоміжна функція. Виводить меню з функціоналом повторного запиту на сервер у випадку отримання помилки від серверу.
getSelectTokenKeyboard()	ctx	Допоміжна функція. Виводить меню з функціоналом вибору необхідного джерела для додання ключів доступу та відображення додатих ключів.
getAddTokensKeyboard()	ctx	Допоміжна функція. Виводить меню з функціоналом додавання нового API ключа або модифікування існуючого.

getFilterKeyboard()	ctx	Допоміжна функція. Виводить меню з функціоналом фільтрації транзакцій за проміжком дат.
printTransaction()	ctx transaction	Функція, яка виводить транзакції користувача повідомленнями.
generateStr()	transaction	Функція, яка генерує зручне для читання користувачів повідомлення.
getWalletTransactions()	nick from to	Функція, яка створює запит на сервер та повертає транзакції користувача через гаманець.
createOrUpdateWalletTransactionComment()	transactionId comment type	Функція, яка створює або модифікує коментар щодо транзакції користувача через гаманець.

7.3 Діаграма розгортання

На діаграмі розгортання вказані обчислювальні вузли системи і програмні компоненти, що виконуються на цих вузлах.

Діаграму розгортання програмного забезпечення наведено у графічних матеріалах у додатку Д.

Висновок до розділу

Даний розділ присвячений реалізації бізнес-логіки системи. Було розділено все на функції за їх метою використання. Описано усі специфікації функції: назви, опис, аргументи, результати виконання. Всі функції діляться на функції роботи з базами даних, серверні та клієнтські.

Також була побудована діаграма розгортання продукту.

У разі розширення команди або передачі продукту іншій команді даний розділ надає усю необхідно інформацію щодо архітектури програмного коду Системи.

8 РОЗРОБЛЕННЯ ІНТЕРФЕЙСА КОРИСТУВАЧА

8.1 Стартовий екран телеграм боту

Після додавання телеграм боту до своїх поточних чатів, користувач побачить логотип програми, її назву та короткий опис функціоналу бота. Також йому буде запропоновано натиснути команду /start для початку роботи з ботом.

Коли користувач відправив функцію для початку роботи, система перевірить чи були у цього користувача додані API ключі та/або адреса електронного гаманця у мережі Ethereum.

Якщо у користувача не було додано ніяких даних – йому буде запропоновано зробити це. В іншому ж випадку йому буде продемонстровано головне меню бота, де він зможе:

- “Edit api keys” – відредагувати уже існуючі API ключі, або додати нові;
- “Export” – експортувати свої транзакції за заданими параметрами;
- “Search” – знайти необхідну транзакцію за заданими параметрами;
- “New cash transaction” – додати нову готівкову або фіатну транзакцію.

На рисунку 8.1 показане стартове вікно, на рисунку 8.2 – вікно з можливістю додавання нових API ключів, на рисунку 8.3 – вікно з можливістю редагування існуючих та додавання нових.

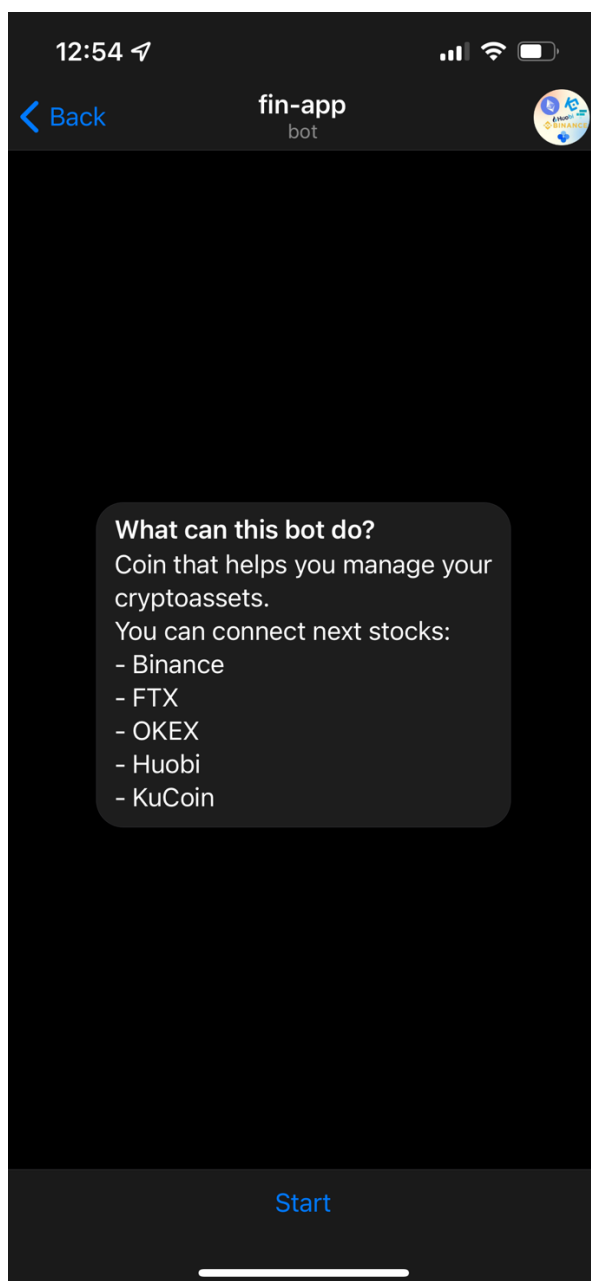


Рисунок 8.1 – Стартове вікно бота

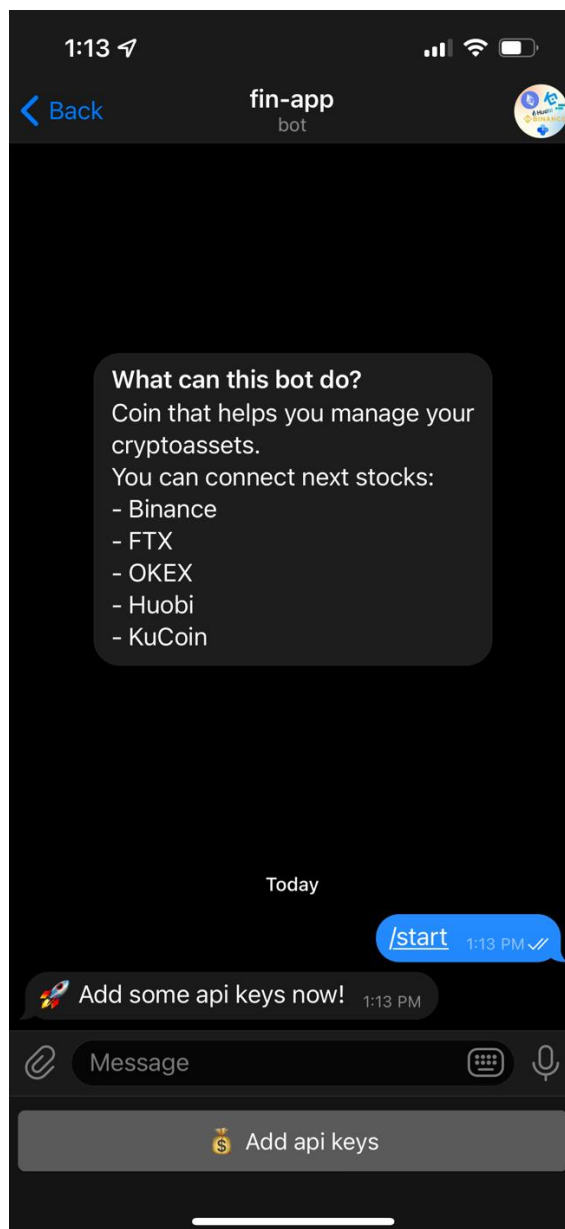


Рисунок 8.2 – Вікно з можливістю додавання нових API ключів

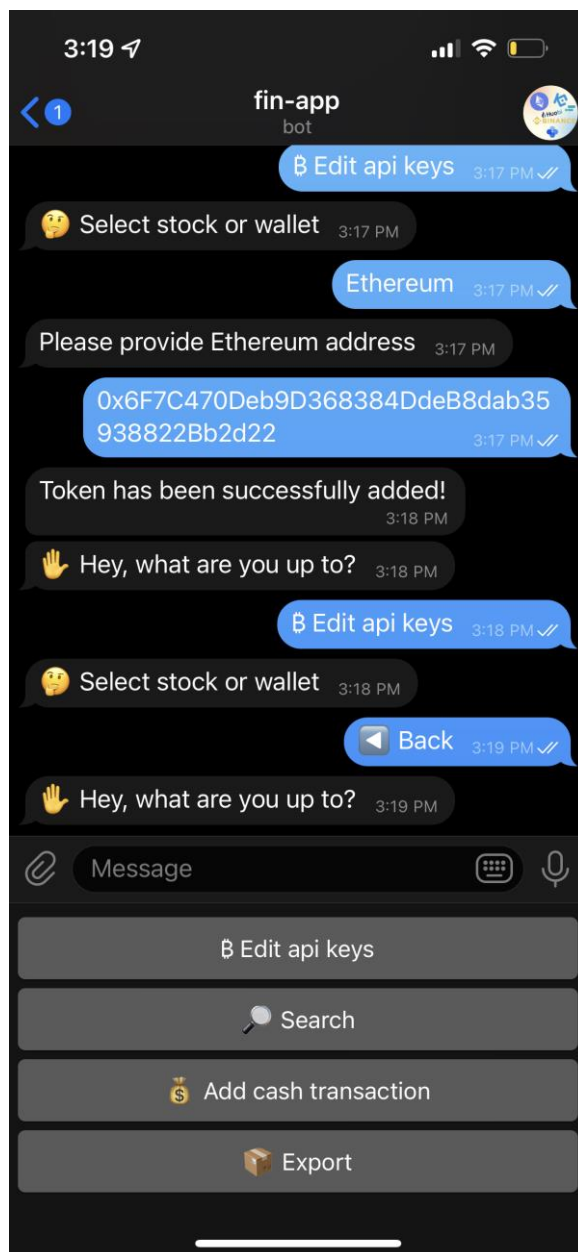


Рисунок 8.3 – Вікно з можливістю редагування існуючих API ключів

8.2 Екран редагування існуючих API ключів або додавання нових

Коли користувач обрав пункт меню “Add API keys” або “Edit API keys”, йому буде запропоновано обрати біржу, яку він хоче інтегрувати до системи із наступного списку: Binance, FTX, OKEX, Huobi, KuCoin, або адресу електронного гаманця, вже додані ресурси будуть відмічені галочкою. Також буде відображена кнопка повернення до попереднього меню.

Після вибору ресурсу – користувачу прийде повідомлення з запитом на ввід ключа, секрету та паролю (для OKEX) послідовно. У разі валідних даних, користувач отримає повідомлення щодо успішної інтеграції, а також останні транзакції, якщо такі існують. У разі помилки – побачить текст помилки.

На рисунку 8.4 продемонстровано пункти меню з вибором доступних ресурсів, на рисунку 8.5 – екран послідовного заповнення даних, на рисунку 8.6 – екран успішного додавання ключів, на рисунку 8.7 – помилка при валдації даних.

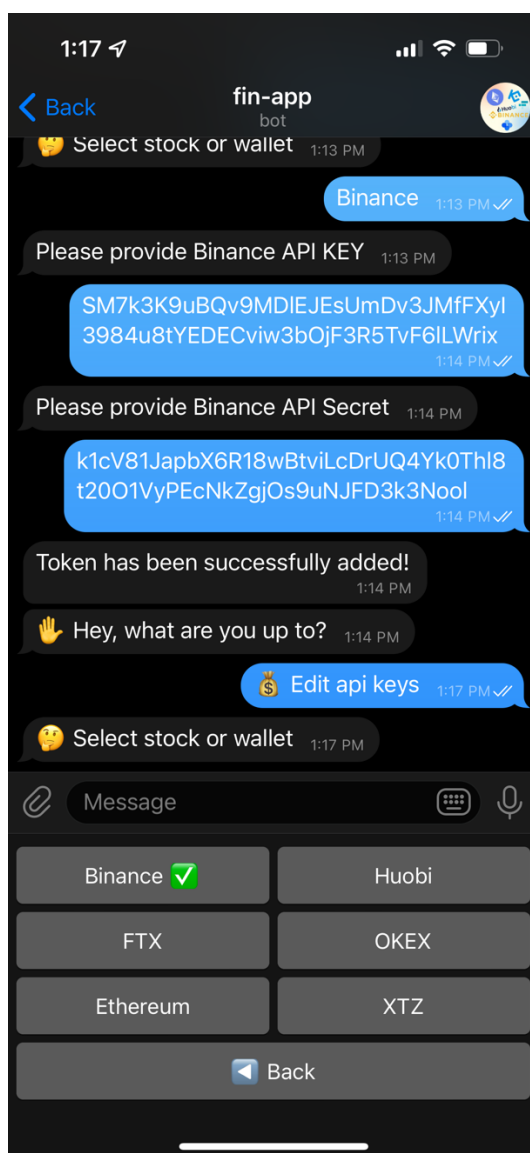


Рисунок 8.4 – Вікно з пунктами меню вибору доступних ресурсів

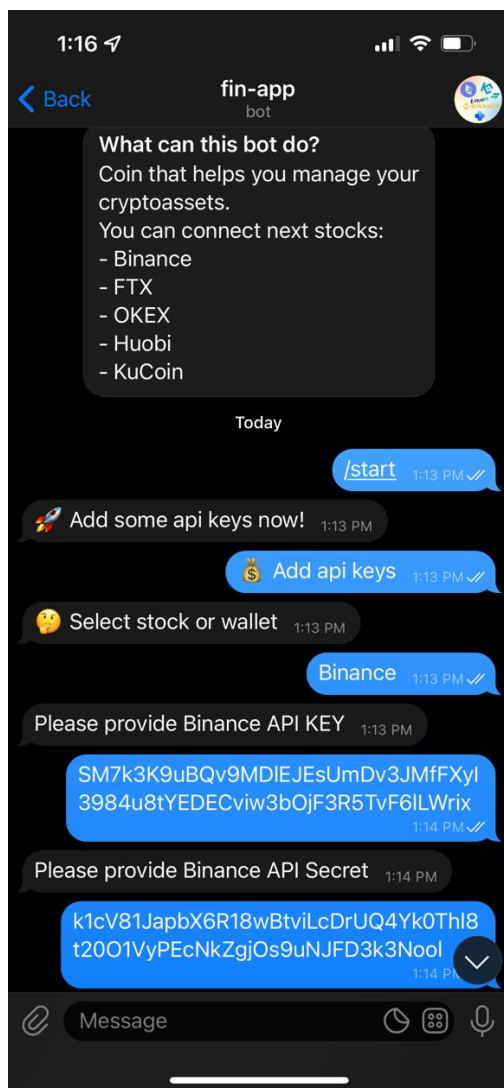


Рисунок 8.5 – Вікно послідовного заповнення даних

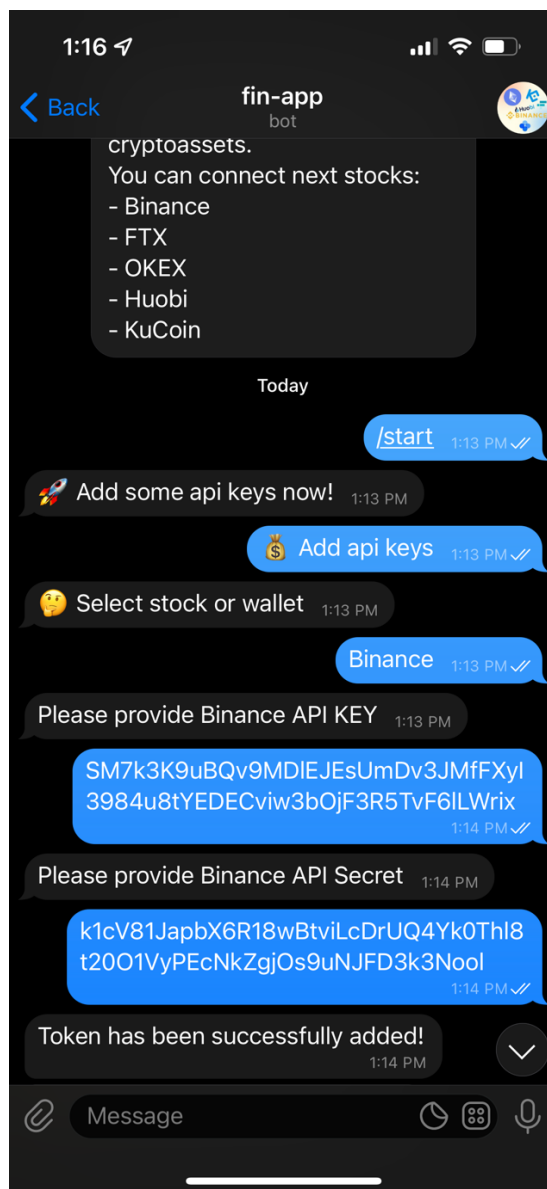


Рисунок 8.6 – Вікно успішного додавання API ключів

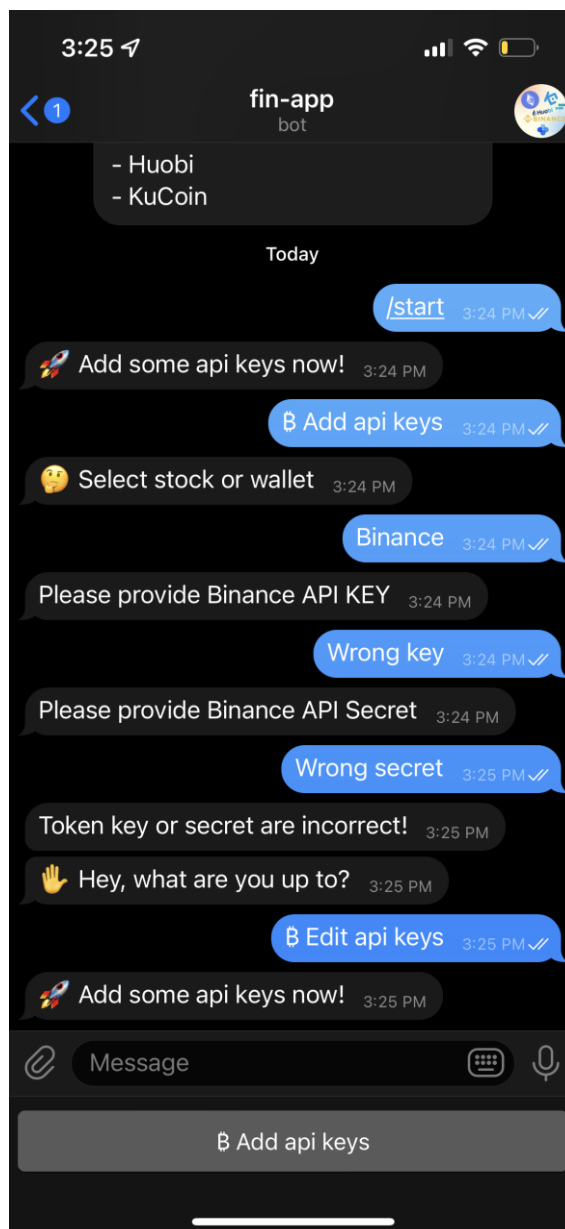


Рисунок 8.7 – Вікно помилки при валдації даних

8.3 Екран експорту існуючих транзакцій, за заданими фільтрами

Коли користувач обрав пункт меню “Export” або “Search”, йому буде запропоновано обрати фільтрацію з доступних в системі: ресурс та проміжок дати проведення транзакції. Задля зручності – серед дат запропоновано наступні варіанти:

- “Today” – відображає усі транзакції, які відбулися сьогодні
- “Yesterday” – відображає усі транзакції, які відбулися вчора
- “This month” – відображає усі транзакції, які відбулися поточного місяця

- “This year” – відображає усі транзакції, які відбулися поточного року
- “Custom data” – пропонує користувачу ввести діапазон дат від і до.

У випадку персоналізованого діапазону дат – буде запропонована ввести дату початку фільтрації і кінця послідовно. Після цього ці дані буде провалідовано і відображено помилки при необхідності.

Коли всі фільтри були установлені – користувачу будуть відображенні транзакції з можливістю додавання коментару, при пошуку, або буде відіслано .csv файл у випадку експорту.

На рисунку 8.8 показано пункти меню з вибором доступних діапазонів дат, на рисунку 8.9 – екран послідовного заповнення даних початкової і кінцевої дати, на рисунку 8.10 – помилка валідації дати, на рисунку 8.11 – результат відфільтрованих транзакцій, а на рисунку 8.12 – файл формату .csv з відфільтрованими транзакціями.

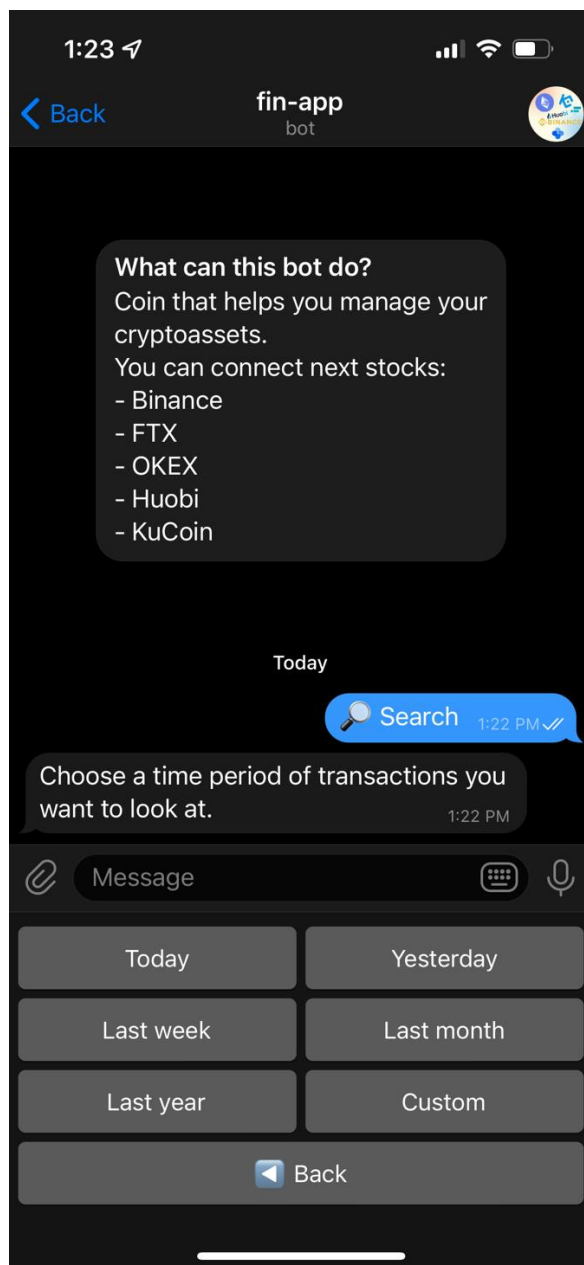


Рисунок 8.8 – Вікно з пунктами меню з вибором доступних діапазонів дат

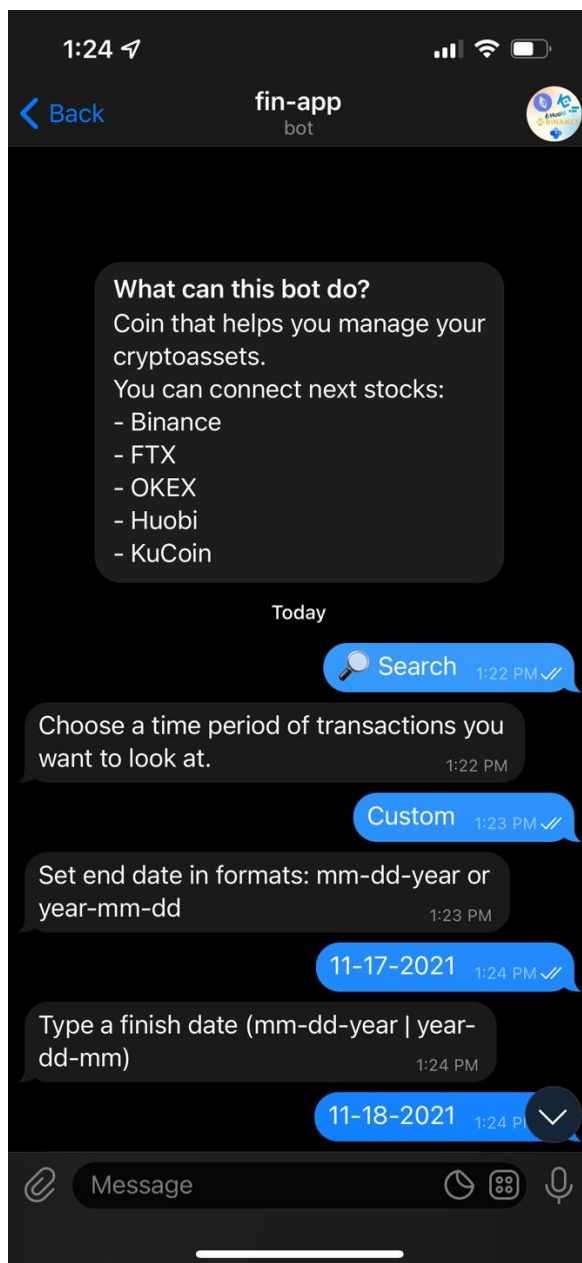


Рисунок 8.9 – Вікно послідовного заповнення даних початкової і кінцевої дати

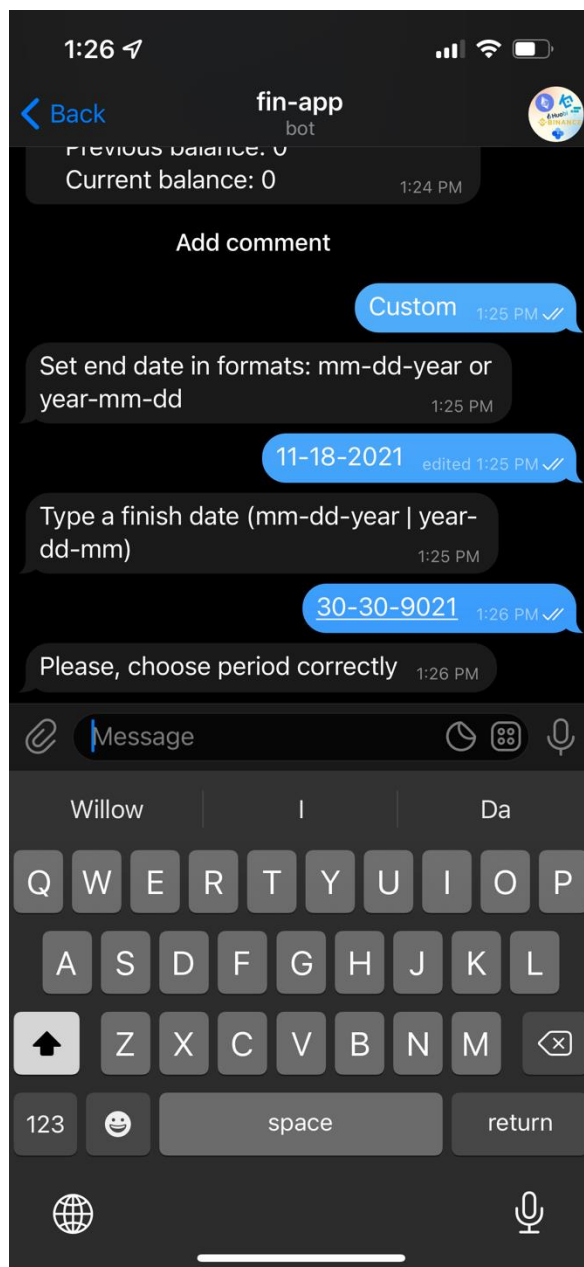


Рисунок 8.10 – Вікно помилки валідації дати

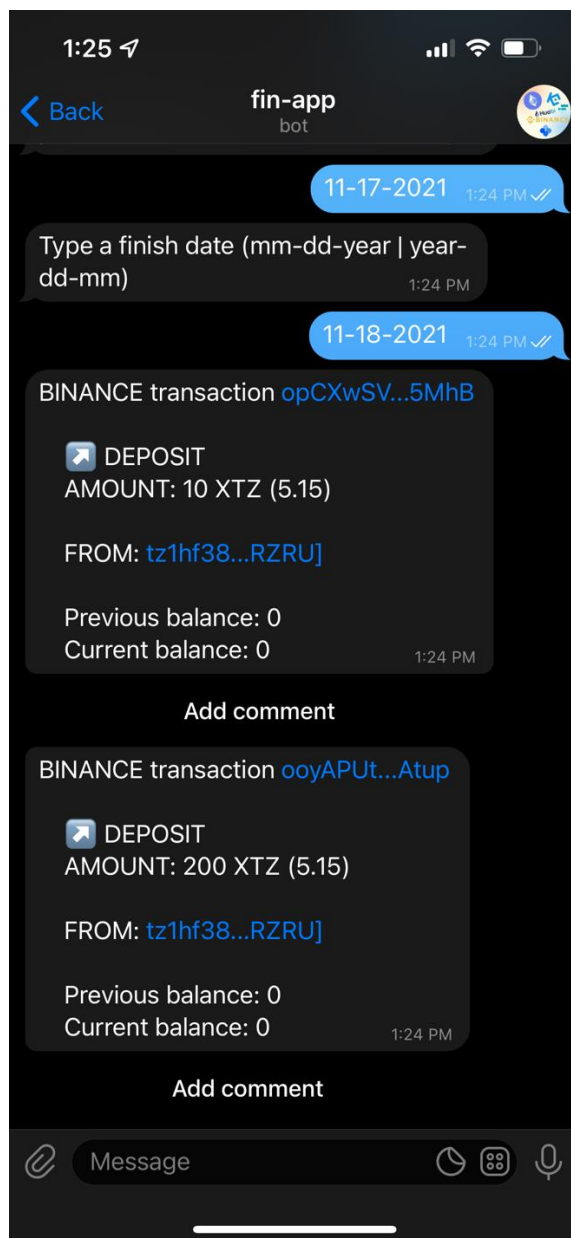


Рисунок 8.11 – Вікно результату відфільтрованих транзакцій

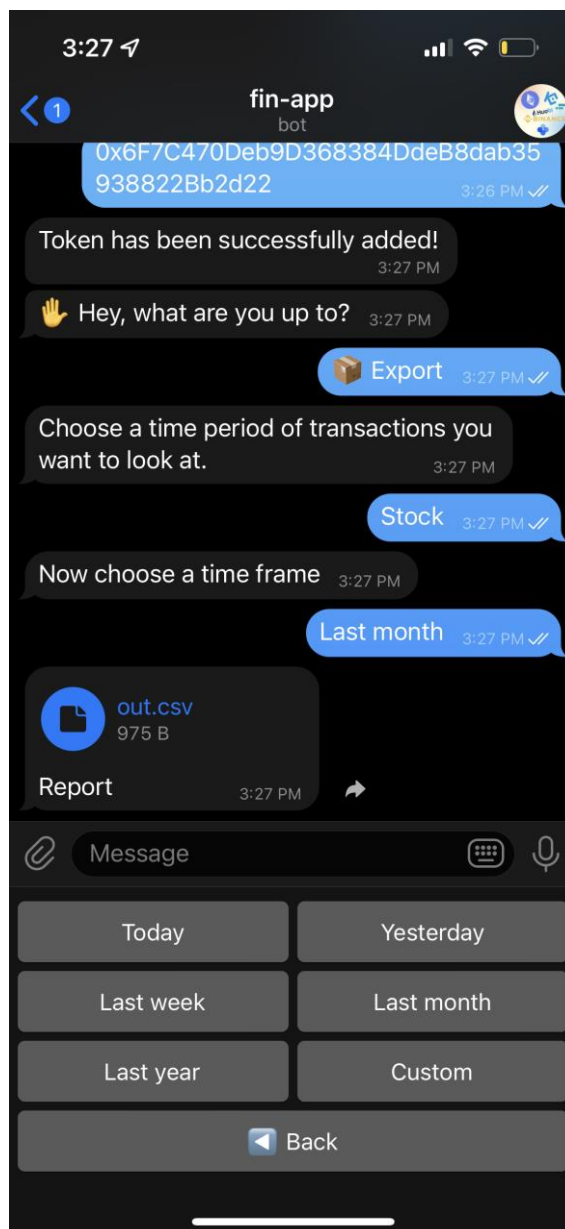


Рисунок 8.12 – Вікно з повідомленням з файлом формату .csv з відфільтрованими транзакціями

8.4 Екран додавання нових готівкових транзакції

Коли користувач обрав пункт меню “Add new cash transaction”, йому буде запропоновано ввести інформацію щодо транзакцій послідовно.

У випадку успішного додавання транзакції або помилки – буде виведено відповідне повідомлення

На рисунку 8.13 продемонстровано порядок заповнення даних, на рисунку 8.14 – повідомлення про успішне додавання транзакції.

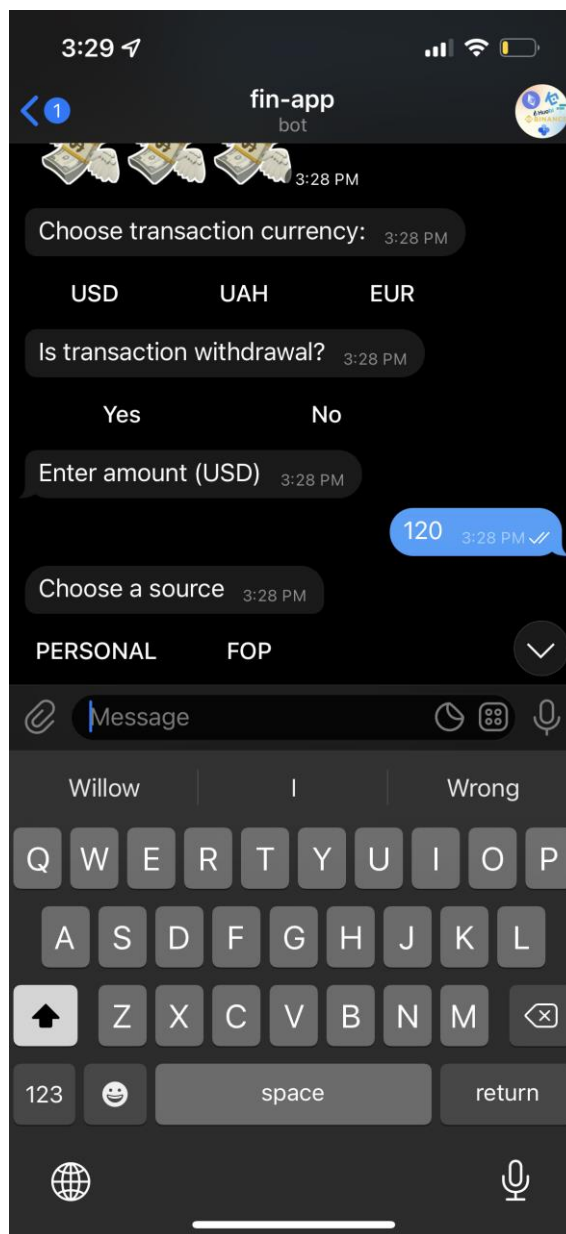


Рисунок 8.13 – Вікно з порядок заповнення даних готівкових транзакцій



Рисунок 8.14 – Вікно з повідомленням про успішне додавання транзакції

8.5 Екран додавання коментарю до існуючих транзакцій

Раз на 10 хвилин серверна частина Системи перевіряє біржі на наявність нових транзакцій по користувачам. У випадку коли у користувача знайдені нові транзакції – йому приходить відповідне повідомлення з можливістю додати коментар до транзакції.

При натисканні кнопки “Add comment” – користувачу буде запропоновано ввести коментар з клавіатури. Кнопка змінить свій статус на “Edit comment”.

На рисунку 8.15 показано нові транзакції користувача, на рисунку 8.16 – реалізація функціоналу редагування коментарю щодо транзакції.

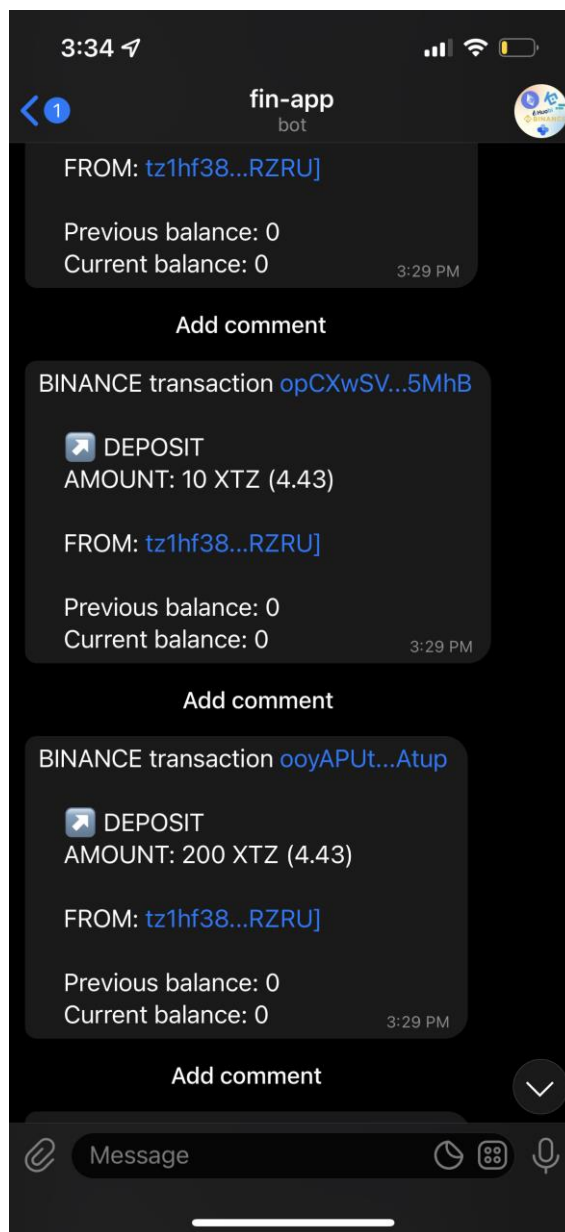


Рисунок 8.15 – Вікно з новими транзакціями користувача

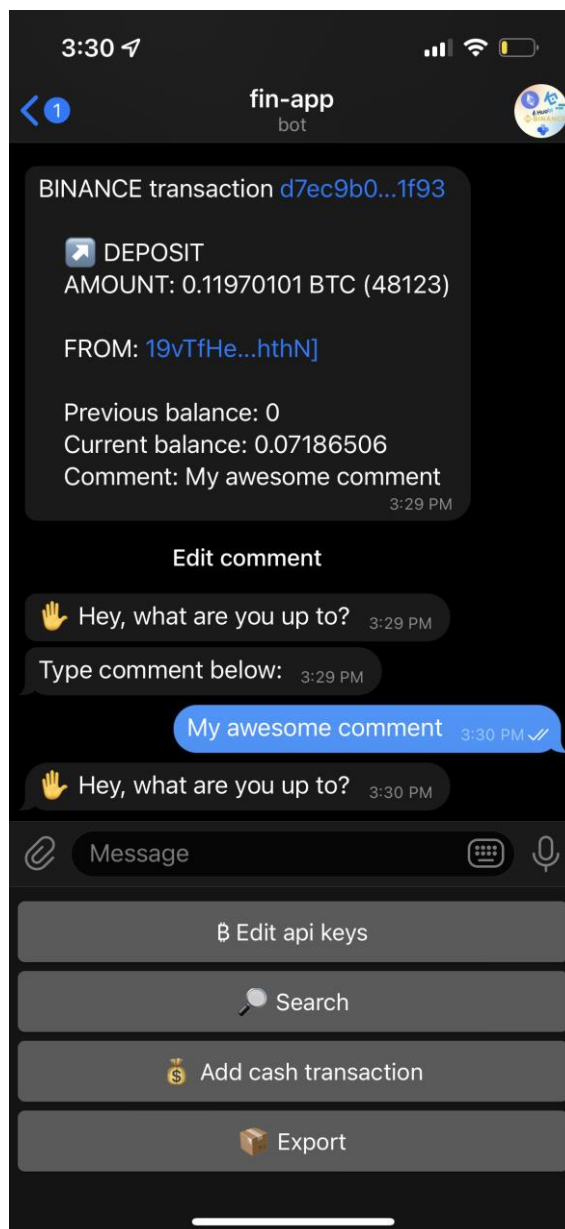


Рисунок 8.16 – Вікно з реалізацією функціоналу редагування коментарю

Висновок до розділу

В цьому розділі описано розробку інтерфейсної частини забезпечення. Описано усі екрани та продемонстровано реалізацію роботи телеграм бота.

Даний розділ може слугувати Керівництвом користувача. І опираючись на детальний опис кожного етапу взаємодії з Системою – користувач може з легкістю отримати необхідний йому результат роботи програми.

9 ТЕСТУВАННЯ СИСТЕМИ

Для того щоб переконатися у надійності розроблюваної системи та виключити можливі помилки, які могли б потрапити до коду на проєкті використовувався TypeScript та ESLint.

TypeScript - інструмент, який допомагає позбутися можливості приведення не правильних типів під час передачі даних у якості параметрів. TypeScript має строгу типізацію, що виключає можливість передачі неправильних даних у різноманітні функції і це запобігає неправильному виконанню програми.

ESLint у свою чергу робить форматування коду, а також статичний аналіз коду, що в дозволяє побачити розробнику місця у коді, до яких програма не дійде ніколи і це також поліпшує кодову базу та запобігає помилкам з боку розробника.

Також були зконфігуровані husky команди, які не дозволяють розробнику відправляти код на репозиторій у випадку, коли він не проходить перевірку правил гарного тону коду.

На додачу всі тести запускаються автоматично системою GithubCI перед кожною доставкою оновленої кодової бази у production середовище. Якщо в процесі тестування хоч один тест не виконується - процес доставки перерветься і потенційно не робоча кодова база не може потрапити у production.

В цьому підрозділі наведено опис тестів і порядок їх виконання для перевірки відповідності програмного забезпечення комплексу задач функціональним вимогам, представленим у технічному завданні на створення інформаційної системи підтримки діяльності балетної школи.

9.1 Мета випробувань

Метою випробувань є перевірка відповідності функцій інформаційної системи підтримки діяльності балетної школи.

9.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603-92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

9.3 Результати випробувань

В процесі тестування була перевірена уся функціональність комплексу задач (КЗ). У наступних таблицях наведений перелік випробувань основних функціональних можливостей (таблиці 9.1-9.9).

Таблиця 9.1– Перевірка відображення початкового меню для нового користувача

Мета тесту:	Перевірка функції “Новий унікальний користувач”
Початковий стан КЗ	Відкрите початкове вікно бота
Вхідні данні:	1. Користувач вперше запустив систему і не має жодного API ключа.
Схема проведення тесту:	1. Запустити бот командою “/start”.
Очікуваний результат:	Виведено кнопку “Add api keys”
Стан КЗ після проведення випробувань:	Виведено кнопку “Add api keys”

Таблиця 9.2– Перевірка відображення початкового меню для існуючого користувача

Мета тесту:	Перевірка функції “Існуючий користувач”
Початковий стан КЗ	Відкрите початкове вікно бота
Вхідні данні:	1. Користувач вже користувався системою і було додано хоча б один API ключ.
Схема проведення тесту:	1. Запустити бот командою “/start”.

Очікуваний результат:	Виведено меню операцій, які можна виконати “Edit token”, “Search”, “Export”, “Add new cash transaction”
Стан КЗ після проведення випробувань:	Виведено меню операцій, які можна виконати “Edit token”, “Search”, “Export”, “Add new cash transaction”

Таблиця 9.3– Перевірка функціонала вибору нового ресурсу

Мета тесту:	Перевірка функції “Відображення доступних ресурсів”
Початковий стан КЗ	Відкрите вікно боту та виведена кнопка “Add API key”
Вхідні данні:	1. Користувач розпочав роботу бота кнопкою “/start”.
Схема проведення тесту:	1. Натиснути кнопку “Add token”.
Очікуваний результат:	Виведено кнопки “Ethereum”, “Binance”, “FTX”, “OKEX”, “Huobi”, “KuCoin”, “Back”
Стан КЗ після проведення випробувань:	Виведено кнопки “Ethereum”, “Binance”, “FTX”, “OKEX”, “Huobi”, “KuCoin”, “Back”

Таблиця 9.4– Перевірка функціонала додавання нового API ключа

Мета тесту:	Перевірка функції “Редагування/додавання нового API ключа” (На прикладі біржі OKEX)
Початковий стан КЗ	Відкрите меню вибору ресурсу для додавання API ключа
Вхідні данні:	1. OKEX API key; 2. OKEX API secret;

	3. OKEX API passphrase.
Схема проведення тесту:	1. Натиснути кнопку “OKEX”; 2. Ввести API key; 3. Надіслати повідомлення; 4. Ввести API secret; 5. Надіслати повідомлення; 6. Ввести API passphrase. 7. Надіслати повідомлення;
Очікуваний результат:	Виведено повідомлення “OKEX API ключ успішно додано до системи”. Кнопка OKEX відображається з галочкою. При наявності відображаються нові транзакції
Стан КЗ після проведення випробувань:	Виведено повідомлення “OKEX API ключ успішно додано до системи”. Кнопка OKEX відображається з галочкою. При наявності відображаються нові транзакції

Таблиця 9.5— Перевірка коректності додавання нового API ключа

Мета тесту:	Перевірка функції “Редагування/додавання нового API ключа” (На прикладі біржі <i>Binance</i>)
Початковий стан КЗ	Відкрите меню вибору ресурсу для додавання API ключа. Користувач бає в системі <i>Binance</i> API key “key1” та API secret “secret1”
Вхідні данні:	1. <i>Binance</i> API key “key2”; 2. <i>Binance</i> API secret “secret2”.
Схема проведення тесту:	1. Натиснути кнопку “Binance”; 2. Ввести API key “key2”; 3. Надіслати повідомлення;

	4. Ввести API secret “secret2”; 5. Надіслати повідомлення;
Очікуваний результат:	Виведено повідомлення “Запропонований вами Binance API ключ невірний. Будь-ласка, введіть валідний API ключ”
Стан КЗ після проведення випробувань:	Виведено повідомлення “Запропонований вами Binance API ключ невірний. Будь-ласка, введіть валідний API ключ”

Таблиця 9.6– Пошук або експорт транзакцій за певний проміжок часу

Мета тесту:	Перевірка функції “Пошук транзакцій за певний проміжок часу” (<i>Експорт працює аналогічно</i>)
Початковий стан КЗ	Відкрите вікно боту з пошуком або експортом транзакцій
Вхідні данні:	1. Дата початку фільтрації у форматі mm-dd-uuuu; 2. Дата кінця фільтрації у форматі mm-dd-uuuu.
Схема проведення тесту:	1. Натиснути кнопку “Custom date”; 2. Ввести дату початку фільтрації у форматі mm-dd-uuuu; 3. Надіслати повідомлення; 4. Ввести дату кінця фільтрації у форматі mm-dd-uuuu; 5. Надіслати повідомлення.
Очікуваний результат:	Виведено транзакції у заданому проміжку часу з можливістю додавання/редагування коментару (у випадку експорту – надіслано файл в форматі .csv), або виведення повідомлення “Не знайдено

	транзакцій в даному проміжку часу”, у випадку якщо транзакцій не було
Стан КЗ після проведення випробувань:	Виведено транзакції у заданому проміжку часу з можливістю додавання/редагування коментару (у випадку експорту – надіслано файл в форматі .csv), або виведення повідомлення “Не знайдено транзакцій в даному проміжку часу”, у випадку якщо транзакцій не було

Таблиця 9.7– Перевірка коректності формату введені дати при пошуку або експорті транзакцій за певний період часу

Мета тесту:	Перевірка функції “Пошук транзакцій за певний проміжок часу” (<i>Експорт працює аналогічно</i>)
Початковий стан КЗ	Відкрите вікно боту з пошуком або експортом транзакцій
Вхідні данні:	1. Дата початку фільтрації у форматі mm-dd-yyuu; 2. Дата кінця фільтрації у форматі dd/yyuu/mm.
Схема проведення тесту:	1. Натиснути кнопку “Custom date”; 2. Ввести дату початку фільтрації у форматі mm-dd-yyuu; 3. Надіслати повідомлення; 4. Ввести дату кінця фільтрації у форматі dd/yyuu/mm; 5. Надіслати повідомлення.
Очікуваний результат:	Виведено повідомлення “Запропонована вами дата не є валідною. Будь-ласка, введіть іншу дату”

Стан КЗ після проведення випробувань:	Виведено повідомлення “Запропонована вами дата не є валідною. Будь-ласка, введіть іншу дату”
---------------------------------------	--

Таблиця 9.8– Перевірка функціоналу додавання коментару

Мета тесту:	Перевірка функції “Додавання нового коментару до транзакції”
Початковий стан КЗ	Відкрите вікно з транзакціями
Вхідні данні:	1. Коментар до транзакції.
Схема проведення тесту:	1. Натиснути “Add comment” на обраній транзакції; 2. Ввести коментар до транзакції з клавіатури; 3. Надіслати повідомлення.
Очікуваний результат:	Додано коментар до транзакції та оновлено обране повідомлення текстом коментару
Стан КЗ після проведення випробувань:	Додано коментар до транзакції та оновлено обране повідомлення текстом коментару

Таблиця 9.9– Перевірка функціоналу редагування коментару

Мета тесту:	Перевірка функції “Редагування коментару транзакції”
Початковий стан КЗ	Відкрите вікно з транзакціями
Вхідні данні:	1. Новий коментар до транзакції.
Схема проведення тесту:	1. Натиснути “Add comment” на обраній транзакції; 2. Ввести новий коментар до транзакції з клавіатури; 3. Надіслати повідомлення.

Очікуваний результат:	Оновлено коментар на обраній транзакції, це демонструється зміною тексту коментару
Стан КЗ після проведення випробувань:	Оновлено коментар на обраній транзакції, це демонструється зміною тексту коментару

Висновок до розділу

В цьому розділі в таблицях наведено перелік випробувань програмного продукту, а саме функції додавання та редагування API ключів бірж та адреси електронного гаманця користувача на базі Ethereum, функціонал пошуку та експорту даних по транзакціям, додавання та редагування коментарів. Описана робота основних функцій навігації і зміну стану телеграм боту в залежності від типу користувача та статусу системи.

10 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

10.1 Опис ідеї проєкту

Аналіз змісту ідеї та загальних напрямків використання стартап-проекту, а саме інформаційної системи підтримки управління власними криптовалютними активами, наведено у таблиці 10.1.

Таблиця 10.1– Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка інформаційної системи підтримки управління власними криптовалютними активами	1. Підвищення ефективності збору даних щодо власних криптовалютних активів	1. Забезпечення цілісності інформації щодо власних криптовалютних активів
	2. Оптимізація швидкості прийняття рішень щодо управління активів	2. Економія часу при прийнятті вагомих рішень
	3. Автоматизація збору інформації щодо транзакцій	3. Спрощення роботи керування активами
	4. Підвищення швидкості додавання нових витрат	4. Продуктивність керування своїми фінансами
	5. Цілісна звітність щодо власних криптовалютних транзакцій	5. Розуміння загального обороту фінансів. Можливість предметного аналізу інформації щодо активів
	6. Надання інформації щодо транзакцій в універсальному форматі	6. Можливість використання отриманих даних у різних системах обліку

Наступним кроком буде проведення аналізу потенційних техніко-економічних переваг ідеї, у порівнянні із тим, що пропонують конкуренти. Для цього нам

необхідно визначити перелік техніко-економічних властивостей та характеристик ідеї, попереднє коло конкурентів, та провести порівняльний аналіз показників. Для власної ідеї визначити показники, що мають:

- а) гірші значення (W, слабкі);
- б) аналогічні (N, нейтральні) значення;
- в) кращі значення (S, сильні).

Після детального пошуку і аналізу конкурентів було обрано декілька програмних забезпечень.

Zapper.fi (1) – веб-застосунок, який дозволяє вказавши Ethereum адресу гаманця, отримати всі транзакції і активи з розрахунком прибутковості тієї чи іншої операції, сформувати портфоліо щодо активів, а також відфільтрувати фінанси за мережею Ethereum-подібних блокчейнів.

ZenLedger (2) є програмним забезпеченням, зосередженим на податковому управлінні та обліку DeFi, NFT та криптовалютах. Дозволяє імпортувати данні через файл або інтегровані сервіси, проглянути та при необхідності модифікувати їх, та сформувати податковий звіт.

Cointracker (3) дозволяє відстежувати ефективність інвестицій та податків криптовалютних активів. Для користування даним сервісом необхідно підключити наявні біржі та гаманці.

У таблиці 10.2 наведено порівняльний аналіз сильних, слабких та нейтральних характеристик ідеї проєкту у порівнянні із зазначеними конкурентами.

Таблиця 10.2– Визначення сильних, слабких та нейтральних характеристик ідеї проєкту

№ п/ п	Техніко- економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона а)	N (нейтра льна сторона а)	S (сильна сторона)
		Мій проект	1	2	3			

1.	Доступність функціоналу в Україні	+	+	-	-			+
2.	Можливість відслідковувати транзакції проведені з бірж	+	-	+	+		+	
3.	Інтегрувати Ethereum-подібних блокчейнів	+	+	+	+		+	
4.	Інтегрування інших блокчейнів	+	-	-	-			+
5.	Додавання коментарів до транзакцій	+	-	-	-			+
6.	Можливість автоматичного збору даних	+	+	-	-			+
7.	Генерування звітів у вигляді файлу .csv	+	-	+	+		+	
8.	Генерування податкових звітів	-	-	+	+		+	

9.	Можливість відслідковувати операції з NFT	-	+	-	-	+		
10.	Додавання фіатних транзакцій	+	-	-	-			+
11.	Графік змінення загального балансу/портфоліо	-	+	+	+	+		
12.	Зручний інтерфейс	+	+	-	+		+	
13.	Підтримка різних мов	-	+	-	+		+	
14.	Підтримка різних основних фіатних валют	-	+	+	+	+		
15.	Інтеграція із значною кількістю сторонніх ресурсів	-	-	+	+	+		

На основі визначених сильних, слабких та нейтральних сторін ідеї, можна провести аналіз та зрозуміти її конкурентоспроможність.

Згідно з наведеної вище таблиці можна зробити висновок, що розроблювана система має деякі переваги над аналогами, а саме: доступність в Україні, інтегрування з різними мережами блокчейну, можливість автоматичного збору даних, додавання фіатних транзакцій і більш гнучке управління самими транзакціями. Це надає деяку конкурентну перевагу серед аналогів.

Однак, також є і недоліки. Є більш критичні недоліки такі як: інтеграція із великою кількістю ресурсів, підтримка різних основних фіатних валют, звичні користувачам графіки та підтримка NFT. Та менш критичні, які не потребують значної кількості ресурсів, такі як підтримка різних мов інтерфейсу.

Втік, дані можливості не є досить популярними серед користувачів і можуть бути реалізовані в процесі масштабування системи, так як потребують значної кількості часу. Також, оскільки система має відкрите API та є опен-сорсним, розробники можуть вдосконалювати та розширювати її функціонал.

10.2 Технічний аудит ідеї проєкту

В межах даного підрозділу проведемо аудит технології, за допомогою яких можна реалізувати ідею проєкту. Технологічну здійсненність ідеї проєкту наведено у таблиці 10.3.

Таблиця 10.3– Технологічна здійсненність ідеї проєкту

№ п/п	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
1.		Мова програмування JavaScript	Наявна	Доступна
2.		Мова програмування TypeScript	Наявна	Доступна
3.		Серверна платформа Node.js	Наявна	Доступна

4.		Реляційна база даних PostgreSQL	Наявна	Доступна
5.		Мова запитів GraphQL	Наявна	Доступна
6.		GraphQL сервер Apollo Server	Наявна	Доступна
7.		Бібліотека для роботи з базою даних Prisma	Наявна	Доступна
8.		Бібліотека для роботи з соціальною мережею Telegram Telegraph	Наявна	Доступна
9.		API для отримання інформації щодо транзакцій в мережі блокчейн Covalent	Наявна	Доступна на обмежену кількість запитів
10.		API для отримання даних щодо транзакцій на біржах	Наявна	Доступна на обмежену кількість запитів

Згідно з наведеної вище таблиці можна зробити висновок, що розроблювана система по-перше може бути розроблена, а по-друге не потребує в розробці нових технологій, оскільки всі необхідні технології вже існують в безкоштовному доступі.

10.3 Аналіз ринкових можливостей запуску стартап-проєкту

Спочатку проведемо аналіз попиту. Для цього визначимо наявність попиту, обсяг та динаміку розвитку ринку. Попередню характеристику наведено в таблиці 10.4.

Таблиця 10.4– Попередня характеристика потенційного ринку стартап-проекту

№ п/ п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	5
2	Загальний обсяг продаж, грн/ум.од	Невідомий
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Можливі у певних країнах світу
6	Середня норма рентабельності в галузі (або по ринку), %	Невідома

Згідно з наведеної вище таблиці, можна зробити висновок, що умови ринку для входу є досить сприятливими. На сьогоднішній день світ криптовалют вже є досить популярним серед усіх шарів населення, однак як для фінансового ринку він ще досить новий. Тому з плином часу його популярність у світі буде тільки збільшуватись, а кількість коштів в ньому накопичуватись. Тож чим раніше зайти в даний ринок тим більші шанси на успіх.

Визначимо потенційні групи майбутніх користувачів інформаційної системи підтримки управління власними криптовалютними активами. У таблиці 10.5 описано потреби кожної з можливих груп.

Таблиця 10.5– Аналіз конкуренції в галузі за М. Портером

Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних	Вимоги споживачів до товару
--------------------------	-------------------	--	-----------------------------

	(цільові сегменти ринку)	цільових груп клієнтів	
1. Автоматизований збір даних щодо транзакцій в мережі блокчейн 2. Агрегація даних з різних ресурсів: бірж та електронних гаманців 3. Можливість додавання фіатних коштів 4. Додавання коментарів для чіткого розуміння цілі транзакції 5. Формування звітів за обраними ресурсами та датами проведення транзакцій	1. Активні користувачі мережі блокчейн 2. Фірми, які взаємодіють з мережею блокчейн або виплачують заробітню плату в криптовалюті 3. Інвестори, в портфелі яких є різні криптовалютні активи 4. Розробники, які розробляють програмне забезпечення на базі мережі блокчейн	1. Користування системою для управління власними активами 2. Користування системою для управління активами компанії/групи людей 3. Користування системи для її масштабування та розширення базових можливостей	1. Невисока вартість користування системою 2. Зручне і зрозуміле користування сервісом без спеціального навчання 3. Можливість агрегувати дані без знання коду 4. Ефективність використання 5. Швидкість роботи 6. Доступність у потрібний час з будь-якого девайсу

Наступним кроком на основі попереднього аналізу визначимо та обґрунтуємо фактори конкурентоспроможності. Проведений аналіз зазначено в таблиці 10.6.

Таблиця 10.6– Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проєктів значущим)
1.	Технічне обслуговування	Добре задокументоване API. Додаток має масштабовану архітектуру
2.	Продукт є безкоштовним	Людині не потрібно нічого платити і вона не має жодних прихованих транзакцій для використання додатку
3.	Простий та зрозумілий інтерфейс	Людина з легкістю може виконати усі необхідні їй дії оскільки додаток має гарний функціонал
4.	Потреби споживачів	Потреби споживачів впливають на вдосконалення застосунку і його розвиток як в плані маркетингу, так і в плані монетизації
5.	Універсальність	Продукт є універсальним та підходить для різних цілей менеджменту власних криптовалютних активів, як для фізичних осіб так і для компаній
6.	Необхідність продукту	Сучасний ринок криптовалют пропонує велику кількість продуктів для управління власними активами, таких як біржі та електронні гаманці на базі різних мереж блокчейну. Користувачам необхідно агрегувати та аналізувати велику кількість інформації

7.	Необхідність постійного оновлення та адаптації системи	Так як мережа блокчейн розвивається швидкими темпами, кожного місяця виникають нові продукти і кожного року нові мережі. Щоб система залишалась актуальною, необхідно слідити за даними тенденціями та адаптувати систему
----	--	---

На основі зазначених факторів проведемо аналіз сильних та слабких сторін інформаційної системи підтримки управління власними криптовалютними активами. Аналіз наведено в таблиці 10.7.

Таблиця 10.7– Порівняльний аналіз сильних та слабких сторін системи

№ п/п	Фактор конкурентоспроможності	Бали 1- 20	Рейтинг товарів-замінників у порівнянні зі стартапом						
			-3	-2	-1	0	+1	+2	+3
1.	Технічне обслуговування	15		+					
2.	Продукт є безкоштовним	5			+				
3.	Простий та зрозумілий інтерфейс	13		+					
4.	Потреби споживачів	17				+			
5.	Універсальність	11					+		
6.	Необхідність продукту	20				+			
7.	Необхідність постійного оновлення та адаптації системи	12							+

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації. Аналіз наведено в таблиці 10.8.

Таблиця 10.8– SWOT- аналіз стартап-проєкту

Сильні сторони: - продукт є безкоштовним; - простий та зрозумілий інтерфейс доступний з 90% девайсів; - продукт вирішує глобальну проблему; - продукт є опен-сорсним та має відкритий код, що дозволяє йому легко масштабуватись; - продукт має підтримку фіатних активів; - продукт є універсальним та підходить як для власних активів, так і для активів компанії.	Слабкі сторони: - потреба в підтримці більшої кількості ресурсів; - потреба в рекламній компанії; - потреба в постійній підтримці та удосконаленні роботи системи.
Можливості: - збільшення кількості підтримуваних мереж блокчейну та ресурсів (бірж та електронних гаманців); - розроблення більш наглядного інтерфейсу (наприклад графіки); - заохочення розробників удосконалювати систему.	Загрози: - нестача фінансів; - нестача строків; - поява нових або розвиток існуючих конкурентів.

На основі SWOT-аналізу розробимо альтернативи ринкової поведінки для виведення стартап-проєкту на ринок та орієнтовний оптимальний час їх ринкової реалізації. Аналіз наведено в таблиці 10.9.

Таблиця 10.9– Альтернативи ринкового впровадження стартап-проєкту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Отримання грантів від блокчейн мереж	80%	6 місяців
2.	Виведення проєкту на краундфайдінг	90%	8 місяців
3.	Участь у хакатонах	85%	12 місяців
4.	Розробка на основі вже існуючого проєкту	5%	5 місяців

З наведених вище альтернатив бачимо, що розробка на основі вже існуючого проєкту є найменш ймовірною. Участь у хакатонах може займати досить багато часу. Найбільш вдалим було б взяти участь у отриманні грантів від блокчейн мереж і розмістити проєкт на краундфайдингових платформах. Таким чином отримання ресурсів буде більш швидким, що зменшить строки реалізації.

10.4 Розроблення ринкової стратегії проєкту

Першим кроком оберемо та опишемо цільові групи потенційних споживачів. Аналіз наведено в таблиці 10.10.

Таблиця 10.10– Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
----------	--	---	---	--------------------------------------	--------------------------

1.	Люди, що активно користуються криптовалютами у повсякденному житті	Висока, тому що продукт дає можливість автоматизувати та аналізувати свої транзакції	70% - високий попит	Середня, оскільки існують аналоги, однак вони не є досить великими корпораціями	Середня, завдяки простоті у використанні та наявності базових функцій
2.	Люди, що інвестують у криптовалютні активи	Висока, тому що продукт дає можливість агрегувати дані з різних мережах різних активів	90% - високий попит	Середня, оскільки існують аналоги, однак вони не є мають великого покриття мереж блокчейну	Середня, завдяки наявності основних функцій і великому покритті ресурсів
3.	Керівники або бухгалтери компаній, які використовують криптовалютні активи в цілях компанії	Низька, тому що досить мала кількість компаній використовують криптовалюту для грошових операціях	15% - низький попит	Висока, оскільки для таких людей мабуть підійдуть більше програми спеціалізовані на бухгалтерській звітності	Середня, оскільки існує більш великі конкуренти з більш специфічним функціоналом

4.	Розробники програмних забезпечень на базі мережі блокчейн	Низька, тому що більшість розробників використовують саме спеціалізовані API для своїх потреб	20% - низький попит	Висока, оскільки існують безліч API, які дозволяють взаємодіяти безпосередньо з мережею блокчейн або біржами	Легка, оскільки розробники досить часто тестують різні допоміжні засоби
Які цільові групи обрано: люди, що активно користуються криптовалютами у повсякденному житті та криптовалютні інвестори.					

Далі сформулюємо базову стратегію розвитку в обраних сегментах. Аналіз наведено в таблиці 10.11.

Таблиця 10.11– Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1.	Збільшення функціоналу та додавання нових ресурсів (мереж блокчейну, бірж)	Стратегія масового маркетингу	Широкий функціонал з інтуїтивно зрозумілим інтерфейсом	Стратегія диференціації

2.	Додавання веб-та моб- версії з інтуїтивним відображенням даних, наприклад графіків	Стратегія диференційованого маркетингу	Використання дизайну для подачі інформації	Стратегія диференціації
----	--	--	--	-------------------------

Далі кроком є вибір стратегії конкурентної поведінки, який наведено в таблиці 10.12.

Таблиця 10.12– Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1.	Ні. Існують подібні системи, однак вони засореджуються на спеєецифічних функціях і не покривають усього необхідного функціоналу	80% - існуючі споживачі, які вже давно на ринку і шукають альтернативи 20% - нові користувачі мережі блокчейн, які ще не встигли обрати зручний інструмент	Так. Частину функціонала вже реалізована. Такі як відображення інформації, поведінка інтерфейсу	Стратегія виклику лідера

На основі вимог споживачів з обраних сегментів до постачальника та до продукту, а також у відповідності до обраної базової стратегії розвитку та стратегії конкурентної поведінки розробимо стратегію позиціонування. Визначення стратегії позиціонування наведено у таблиці 10.13.

Таблиця 10.13– Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1.	Простота інтерфейсу для використання	Стратегія спеціалізації	Простота освоєння системи для користувачів інтуїтивно зрозумілому та не перенасиченому інтерфейсу	Швидкий вхід в систему Зручність користування Доступ у будь-який час з будь-якого девайсу
2.	Покриття великої кількості ресурсів та базовий функціонал для роботи з інформацією	Стратегія диференціації	Автоматизація збору інформації. Можливість легко знаходити, додавати, модифікувати та експортувати дані	Автоматизація базових функцій Чітке агрегування даних Можливість додавати описи до транзакцій

10.5 Розроблення маркетингової програми стартап-проєкту

Для розроблення маркетингової стратегії спочатку сформуємо маркетингову концепцію товару, який отримує споживач. Визначення ключових переваг концепції потенційного товару наведено в таблиці 10.14.

Таблиця 10.14– Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Простота використання системи	Система є доступною для користувачів з будь-якого девайсу	Простий та зручний інтерфейс, завдяки якому система не потребує додаткових інструкцій для користувача. Мінімалізм, не перенасиченість. Доступ до системи створюється через телеграм бота
2.	Автоматизація та агрегація даних з різних джерел щодо транзакцій	Система є функціональною та підтримує більшість необхідних джерел	Можливість додавати різні мережі блокчейну та біржі, з яких беруться дані щодо транзакцій з усіма деталями, які пропонують сервіси
3.	Пошук, модифікація та експорт даних щодо власних криптовалютних активів	Система є гнучкою і дозволяє користувачу задати необхідні параметри	Можливість додавати опис по кожній транзакції та формувати звіти у відповідності до потреб користувача

Наступним кроком розробляємо трирівневу маркетингову модель товару, її наведено в таблиці 10.15.

Таблиця 10.15– Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Інформаційна систем підтримки управління власними криптовалюотними активами		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Швидкість інтернету	М	Тх
	Версія ОС носія (iOS)	М	Тх/Тл
	Версія ОС носія (Android)	М	Тх/Тл
	Версія ОС носія (Windows)	М	Тх/ Тл
	Версія ОС носія (macOS)	М	Тх/ Тл
	Адаптивність системи	М	Е
	Якість: Відповідає стандарту ISO/IEC 27001		
Пакування: електронна документація продукту, що містить інформацію про необхідні технічні вимоги, назву продукту, найменування та адресу виробника, товарний знак. Також політику конфіденційності і договір оферти.			
Марка: Information system to support personal crypto asset management			
III. Товар із підкріпленням	До продажу: не потребує особливих вимог		
	Після продажу: постійні оновлення системи, технічна підтримка користувачів		
За рахунок чого потенційний товар буде захищено від копіювання: патенту та авторського права			

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар. Аналіз наведено в таблиці 10.16.

Таблиця 10.16– Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари- аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1.	0-11000 грн/міс (безкоштовний план надає досить обмежений функціонал)	0-11000 грн/міс (безкоштовний план надає досить обмежений функціонал)	35000-200000 грн./міс	0-5000 грн.

Криптовалюти можливі тільки завдяки мережі блочейн, яка в свою чергу можлива тільки завдяки мережі інтернет. Тому оптимальною і навіть майже єдиною системою збуту є інтернет. Оскільки ми можемо продавати послуги безпосередньо клієнту, а також за допомогою посередників (партнерів), то глибина каналу збуту буде від нульового до першого рівня. Формування системи збуту наведено в таблиці 10.17.

Таблиця 10.17– Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
	Системи є умовно безкоштовною. Базові функції з деякими	Пошук потенційних клієнтів, аналіз ринку, програми лояльності, реклама	Канал нульового рівня та однорівневий	Через мережу інтернет та криптовалютні інформаційні ресурси

	обмеженнями доступні в безкоштовному плані			
--	---	--	--	--

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій. Формування системи збуту наведено в таблиці 10.18.

Таблиця 10.18– Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Клієнти дізнаються про товар в інтернеті	Соціальні мережі – таргетована реклама, блогери (Instagram, Facebook, LinkedIn, YouTube). Статті у відповідних каналах зв'язку	Інтернет-маркетинг, соціальні мережі, реклама у відомих нішових блогерів, конференції, ведення діалогу з користувачами, підтримка	Донести до потенційних клієнтів вирішення проблеми автоматизації та агрегації даних, які необхідні для управління власними криптовалютними активами	Криптовалютні транзакції користувача з різних електронних гаманців та бірж в одному місці. Автоматизація збору, пошук модифікація та експорт необхідних для управління власними

	Виступи на конференціях			криптовалютними активами даних
--	-------------------------	--	--	--------------------------------

Оскільки клієнти більшу частину інформації про товари та послуги дізнаються через такі комунікації як: інтернет, соціальні мережі, статті у відповідних джерелах та доповіді на конференціях, в якості ключових позицій для позиціювання було обрано інтернет-маркетинг, маркетинг у соціальних мережах, реклама у тематичних пабліках, виступи на конференціях.

Висновок до розділу

Після проведеного аналізу було визначено, що є можливість ринкової реалізації стартап-проєкту. Була описана основна ідея системи проведено аналіз аналогів та виділені слабкі та сильні сторони розробляємої системи у порівнянні з конкурентами. Також було проведено технічний аудит ідеї проєкту, виділено технології її реалізації, наявність і доступність цих технологій. Проведено аналіз конкуренції в галузі. Визначено потреби, що формує ринок цільову аудиторію та вимоги споживачів до товару. Було також виділено основні фактори конкурентоспроможності ті їх обґрунтування. Проведено порівняльний аналіз сильних та слабких сторін розробляємої системи з аналогами. Після чого сформовано SWOT-аналіз з описами сильних, слабких сторін, а також можливостей та загроз, на основі чого сформовано альтернативи ринкового впровадження проєкту.

Було проаналізовано цільові групи користувачів і обрано дві із них: люди, що активно користуються криптовалютами у повсякденному житті та криптовалютні інвестори. Визначено базові стратегії ринку та конкурентної поведінки.

На сам кінець, було розроблено маркетингову стратегію. Для цього було сформовано маркетингову концепцію товару, який отримує користувач, описано три рівні моделі товару та визначення меж встановлення ціни. Також була побудована концепція маркетингової комунікації.

ВИСНОВКИ

В даній магістерській дисертації було розроблено інформаційну систему підтримки управління власними криптовалютними активами. В ході написання роботи був досліджений процес розробки фінансового інструменту для роботи із транзакціями в мережі блокчейн. Результатом виконання магістерської дисертації є телеграм бот та API, завдяки якому можна створювати нові веб застосунки для ведення моніторингу активів з різних ресурсів, таких як біржі та різноманітні мережі.

Був виконаний аналіз роботи фінансових програмних забезпечень для фіатних активів та аналіз роботи блокчейн застосунків. Виділено вимоги для написання подібного програмного забезпечення. Також було досліджено внутрішній устрій різних блокчейнів, проведений огляд на різні мережі, та аналізовано роботи деяких бірж.

Для реалізації клієнтської частини Системи було розроблено прототипні версії боту та протестовано з участю користувачів криптовалютних активів. На основі зворотнього зв'язку оновлювалися вимоги до інтерфейсу та функціонального наповнення системи. На основі даних вимог розроблювався інтерфейс бота та модифікувалась серверна частина Системи.

В роботі запропоновано та реалізовано загальне архітектурне рішення системи та її складових компонент. Обґрунтовано методи та засоби реалізації системи.

Здобуту інформацію можна використовувати в подальшому, при написанні нових Систем, або для модифікації існуючого програмного рішення. В подальшому можна розширювати систему і додавати до неї новий функціонал – впроваджувати нові мережі блокчейну або біржі. Також як альтернатива розвитку даного продукту може слугувати розробка веб-інтерфейсу для більш зручного і зрозумілого відображення інформації.

Дане програмне рішення є досить лінійне, але при цьому дуже ефективно вирішує поставлену задачу. Хоча з недоліків слід виділити недостатній рівень захисту клієнтських даних, що є досить критичним пунктом. Щоб усунути цей недолік, Систему потрібно дещо удосконалити.

Тестування роботи програмного забезпечення показала його працездатність та можливість використання даної системи в менеджменті власних активів. Використовуючи дане програмне забезпечення можна провести моніторинг, аналіз та оптимізацію своїх фінансових активів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Total Value Locked [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.theblockcrypto.com/data/decentralized-finance/total-value-locked-tvl>;
2. What Is Total Value Locked? [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.nasdaq.com/articles/what-is-total-value-locked-2021-09-29>;
3. Ethereum Blockchain [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://ethereum.org/en>;
4. Binance Smart Chain [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.binance.org/en/smartChain>;
5. Tezos [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://tezos.com>;
6. Solana [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://solana.com>;
7. Cosmos Network [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://cosmos.network>;
8. Cardano [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://cardano.org>;
9. Polygon (MATIC) [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://polygon.technology>;
10. Tron [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://tron.network>;
11. Polkadot [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://polkadot.network>;
12. Binance [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.binance.com/ru>;
13. KuCoin [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.kucoin.com/>;

14. FTX [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://ftx.us/>;
15. OKEХ [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.okex.com/ru>;
16. Huobi [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.huobi.com/en-us/>;
17. Zapper [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://zapper.fi/>;
18. Metamask [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://metamask.io/>;
19. Uniswap [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://uniswap.org/>;
20. Yearn Finance [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://yearn.finance/>;
21. 1Inch [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://app.1inch.io/#/1/swap/ETH/DAI>;
22. Curve [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://curve.fi/>;
23. Balancer [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://balancer.fi/>;
24. Sushiswap [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://sushi.com/>;
25. ZenLedger [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.zenledger.io/>;
26. Cointracker [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.cointracker.io/>;
27. CoinTracking [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://cointracking.info/>;
28. Koinly [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://koinly.io/>;

29. Miro [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://miro.com/>;
30. JetBrains WebStorm [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.jetbrains.com/ru-ru/webstorm/>;
31. Github [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://github.com/>;
32. JavaScript [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.javascript.com/>;
33. TypeScript [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.typescriptlang.org/>;
34. Node.js [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://nodejs.org/en/>;
35. PostgreSQL [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.postgresql.org/>;
36. GraphQL [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://graphql.org/>;
37. Telegram [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://telegram.org/>;
38. Telegraph [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.telegraph.co.uk/>;
39. Trello [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://trello.com/en-US>;
40. Microsoft Word [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.microsoft.com/en-us/microsoft-365/word>;
41. Draw.io [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://app.diagrams.net/>;
42. Google Drive [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.google.com/drive/>;
43. C# [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://docs.microsoft.com/en-us/dotnet/csharp/>;

44. Java [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.java.com/en/>;
45. Apollo Server [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.apollographql.com/docs/apollo-server/>;
46. Prisma [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.prisma.io/>;
47. PHPMyAdmin [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.phpmyadmin.net/>;
48. Docker [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.docker.com/>.