

ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

(повне найменування вищого навчального закладу)

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

(повне найменування інституту, назва факультету (відділення))

КАФЕДРА ПРОГРАМНИХ ЗАСОБІВ І ТЕХНОЛОГІЙ

(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до кваліфікаційної роботи

магістра

(освітній рівень)

на тему: «Розробка веб-платформи для моніторингу та аналізу гаманців у
блокчейні TON»

Виконав: студент 6 курсу, групи бПР1
спеціальності

121 - «Інженерія програмного забезпечення»

(шифр і назва спеціальності)

Сновида М. М.

(прізвище та ініціали)

Керівник к.т.н., доцент Козуб Н.О.

(прізвище та ініціали)

Рецензент к.т.н., доцент Григорова А.А.

(прізвище та ініціали)

Херсонський національний технічний університет

(повне найменування вищого навчального закладу)

Факультет, відділення Інформаційних технологій та дизайну
Кафедра Програмних засобів і технологій
Освітній рівень магістр
Спеціальність 121 – Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Програмних засобів і технологій

к.т.н. доцент Огнєва О.Є.

“ ____ ” _____ 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Сновида Максим Михайлович

(прізвище, ім'я, по батькові)

Тема роботи «Розробка веб-платформи для моніторингу та аналізу гаманців у блокчейні TON»

керівник роботи к.т.н. доцент Козуб Н.О.

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

затверджена наказом вищого навчального закладу від 15.09.2025 р. № 416-с

2. Строк подання студентом роботи 10.12.2025

3. Вихідні дані до роботи постановка завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): Аналіз предметної області та існуючих рішень; постановка задачі й вимоги до системи; моделювання процесів (IDEF0) та сценаріїв використання; проєктування архітектури й бази даних; реалізація програмного прототипу JetRadar; тестування та аналіз результатів; інструкція користувача і висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): IDEF0-діаграми процесів (As-Is / To-Be: контекстна та декомпозиція); діаграма варіантів використання (Use Case); компонентна/архітектурна діаграма системи; ER-діаграма бази даних; діаграми послідовності та діяльності для ключових сценаріїв; діаграма розгортання (Deployment).

6. Консультанти розділів роботи

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

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання роботи	Термін виконання етапів роботи	Примітки
1.	Отримання завдання	03.09.2025	Виконано
2.	Підбір літератури	05.09.2025 – 14.09.2025	Виконано
3.	Аналіз предметної області	15.09.2025 – 21.09.2025	Виконано
4.	Розробка та обґрунтування завдання	22.09.2025 – 07.10.2025	Виконано
5.	Розробка концептуальної моделі	08.10.2025 – 14.10.2025	Виконано
6.	Моделювання та проектування системи	15.10.2025 – 28.10.2025	Виконано
7.	Моделювання та проектування бази даних	29.10.2025 – 04.11.2025	Виконано
8.	Розробка інтерфейсу додатку	05.11.2025 – 11.11.2025	Виконано
9.	Тестування додатку	12.11.2025 – 17.11.2025	Виконано
10.	Оформлення пояснювальної записки	18.11.2025 – 01.12.2025	Виконано

Студент

(підпис)

М.М. Сновида

(прізвище та ініціали)

Керівник роботи

(підпис)

Н.О. Козуб

(прізвище та ініціали)

РЕФЕРАТ

Обсяг роботи: 88 сторінок, 37 рисунків, 14 таблиць, 2 додатків, 40 джерел.

Мета роботи — розроблення прототипу вебплатформи JetRadar для автоматизованого моніторингу гаманців у блокчейні TON, побудови графів взаємодій та виконання портфельної аналітики.

Об'єкт дослідження — процес моніторингу транзакцій і взаємодій у мережі TON.

Предмет дослідження — методи та засоби автоматизованого збирання, оброблення й візуалізації даних блокчейну TON, а також побудова графової моделі взаємодій і портфельної агрегації.

Методи дослідження — аналіз наявних сервісів і джерел, функціональне моделювання процесів (IDEF0), проектування архітектури та моделі даних, інтеграція з зовнішнім API (TonAPI), розроблення вебзастосунку засобами Python (FastAPI, SQLAlchemy) і TypeScript (React, бібліотеки візуалізації графів), модульне та інтеграційне тестування (pytest), UI-перевірки сценаріїв користувача.

Основні результати — реалізовано прототип JetRadar, що забезпечує ведення списків спостереження з псевдонімами та групами, отримання і збереження даних про баланси й транзакції, побудову графа взаємодій та агреговану портфельну аналітику, а також клієнтський інтерфейс у форматі Telegram Mini App. Сформульовано та систематизовано вимоги до системи, описано архітектурні рішення та виконано тестування ключових сценаріїв.

Наукова новизна — запропоновано інтегрований підхід до аналізу активності гаманців TON шляхом поєднання моніторингу, графового подання взаємодій і портфельної агрегації в межах одного прототипу з формалізацією вимог та архітектури.

Практичне значення — прототип дає змогу скоротити трудомісткість ручного аналізу транзакцій, підвищити зручність моніторингу груп адрес і отримувати структуровані результати у вигляді графів, зведених балансів і сповіщень.

Ключові слова: TON, JETRADAR, БЛОКЧЕЙН, МОНІТОРИНГ, ТОНАРІ, ПОРТФЕЛЬ, ГРАФ, ВЕБПЛАТФОРМА, TELEGRAM.

АНОТАЦІЯ

У роботі розглянуто проблему відсутності зручного інтегрованого інструменту для моніторингу публічних адрес і аналізу взаємодій у блокчейні TON. У першому розділі проаналізовано предметну область і наявні інструменти, а також окреслено причини неефективності ручного підходу. У другому розділі сформульовано та формалізовано функціональні й нефункціональні вимоги до JetRadar, визначено акторів і сценарії використання та описано проєктні специфікації інтерфейсу й технічних засобів. У третьому розділі подано архітектурні рішення, модель даних і підсистеми, пов'язані зі збором даних, графовим аналізом і портфельною аналітикою, а також описано підхід до розгортання. У четвертому розділі наведено реалізацію серверної та клієнтської частин, результати тестування та інструкцію з розгортання і користування.

За результатами роботи створено прототип вебплатформи JetRadar, що дозволяє додавати адреси до списків спостереження, отримувати дані про баланси й транзакції з зовнішнього API, будувати графи взаємодій та формувати агреговану портфельну аналітику з поданням результатів у клієнтському інтерфейсі Telegram Mini App. Застосування FastAPI, SQLAlchemy та React забезпечило модульність реалізації й зручність інтеграції між компонентами. Проведене тестування підтвердило працездатність ключових сценаріїв прототипу.

ABSTRACT

This thesis addresses the lack of an integrated and user-friendly solution for monitoring public addresses and analysing interactions in the TON blockchain. The first chapter reviews the domain and existing TON tools and highlights the limitations of manual monitoring. The second chapter defines and formalises the functional and non-functional requirements for JetRadar, including actors, use cases, and project specifications. The third chapter presents the system architecture, data model, and key subsystems for data collection, graph analysis, and portfolio analytics, as well as the deployment approach. The fourth chapter describes the practical implementation of the server and client, testing results, and user/deployment instructions.

As a result, a JetRadar prototype was developed. It supports watchlists with aliases and groups, retrieves balances and transaction data via an external API, builds interaction graphs, and provides aggregated portfolio analytics through a Telegram Mini App client interface. The use of FastAPI, SQLAlchemy, and React enabled a modular structure and straightforward integration between components. Testing confirmed the operability of the key prototype scenarios.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	10
ВСТУП.....	12
Розділ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	14
1.1 Актуальність теми та постановка проблеми.....	14
1.2 Аналіз наявних платформ та сервісів.....	15
1.3 Аналіз основних понять і технологій побудови систем моніторингу.....	19
1.4 Обґрунтування вибору інструментарію розробки.....	20
1.5 Принципи проектування аналітичних веб-систем.....	21
1.6 Проблеми користувачів та постановка завдань.....	23
1.7 Формалізоване моделювання процесів (IDEF0).....	25
Висновки до розділу 1.....	28
Розділ 2 ПОСТАНОВКА ТА ФОРМАЛІЗАЦІЯ ЗАДАЧІ.....	30
2.1 Мета та завдання розробки.....	30
2.2 Функціональні вимоги.....	31
2.3 Нефункціональні вимоги.....	33
2.4 Формалізація вимог.....	35
2.5 Проектні специфікації.....	37
2.5.1 Дизайн інтерфейсу.....	37
2.5.2 Технічні засоби реалізації.....	37
2.6 Обмеження та припущення.....	38
Висновки до розділу 2.....	39
Розділ 3 ПРОЄКТНІ РІШЕННЯ ТА АРХІТЕКТУРА.....	40
3.1 Загальна концепція та вибір інструментів.....	40
3.2 Архітектура системи.....	42
3.3 Модель даних.....	46
3.3.1 Ролі та права доступу.....	47
3.3.2 Словник даних.....	48
3.3.3 Реалізація моделей та обґрунтування вибору типів даних.....	48
3.3.4 Реалізація патерну Repository.....	50
3.4 Підсистема моніторингу та збору даних.....	50

3.4.1 Реалізація асинхронного оновлення даних.....	52
3.5 Клієнтська частина.....	53
3.6 Модуль графового аналізу.....	55
3.7 Модуль портфельної аналітики.....	58
3.8 Діаграма розгортання.....	60
Висновки до розділу 3.....	61
Розділ 4 РЕАЛІЗАЦІЯ ТА РЕЗУЛЬТАТИ.....	63
4.1 Організація розробки та середовище.....	63
4.2 Реалізація серверної частини.....	64
4.3 Реалізація клієнтської частини.....	66
4.4 Тестування та результати.....	68
4.4.1 Модульне тестування серверної частини (pytest).....	69
4.4.2 Тестування клієнтської частини (npm run test).....	69
4.4.3 Ручне тестування користувацьких сценаріїв (UI).....	70
4.5 Дослідження ефективності використання системи.....	71
4.5.1 Методика проведення експерименту.....	71
4.5.2 Результати вимірювань.....	72
4.5.3 Аналіз результатів.....	74
4.6 Інструкція користувача та розгортання.....	74
4.6.1 Встановлення та конфігурація серверної частини.....	74
4.6.2 Встановлення та запуск клієнтської частини.....	75
4.6.3 Запуск Telegram-бота.....	77
4.6.4 Візуальна інструкція користувача.....	78
Висновки до розділу 4.....	81
ВИСНОВКИ.....	82
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	84
ДОДАТКИ.....	85

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API — Application Programming Interface, інтерфейс програмування додатків

TON — The Open Network, блокчейн-мережа

TonAPI — зовнішній API-сервіс доступу до даних мережі TON

Jetton — токен за стандартом TON

NFT — Non-Fungible Token, невзаємозамінний токен

CRUD — Create, Read, Update, Delete, базові операції з даними

RBAC — Role-Based Access Control, рольовий контроль доступу

REST — Representational State Transfer, архітектурний стиль веб-API

HTTP/HTTPS — протоколи передавання даних (з/без шифрування)

WebSocket — протокол двостороннього обміну даними в реальному часі

ASGI — Asynchronous Server Gateway Interface, асинхронний інтерфейс веб-серверів Python

ORM — Object-Relational Mapping, об'єктно-реляційне відображення

SQL — Structured Query Language, мова запитів до БД

JSON — JavaScript Object Notation, формат обміну даними

ER (ERD) — Entity-Relationship (Diagram), модель «сутність-зв'язок»

UML — Unified Modeling Language, мова моделювання ПЗ

IDEF0 — нотація функціонального моделювання процесів

FastAPI — Python-фреймворк для побудови веб-API (ASGI)

SQLAlchemy — ORM-бібліотека для Python

PostgreSQL — реляційна система керування базами даних

SQLite — вбудована реляційна база даних

Redis — in-memory сховище/кеш (опційно)

APScheduler — планувальник задач для Python

httpx — HTTP-клієнт для Python (у т.ч. асинхронний)

OpenAPI/Swagger — специфікація та веб-документація API

React — бібліотека для побудови UI

TypeScript — типізоване надмножина JavaScript

Tailwind CSS — утилітарний CSS-фреймворк

D3.js — бібліотека візуалізації даних і графів

Zustand — бібліотека керування станом у React

React Query — бібліотека для роботи із серверними даними у React

Telegram Mini App (WebApp) — веб-інтерфейс, вбудований у клієнт Telegram

aiogram — бібліотека для створення Telegram-ботів на Python

pytest — фреймворк модульного тестування для Python

RTL — React Testing Library, бібліотека тестування React-інтерфейсів

CI/CD — Continuous Integration / Continuous Delivery, автоматизація збірки та деплою

Git — система керування версіями

Docker — контейнеризація застосунків

BFS/DFS — алгоритми обходу графа в ширину/глибину

LT — logical time, службовий параметр транзакцій TON (за наявності у даних)

ВСТУП

Актуальність теми зумовлена зростанням ролі блокчейн-мереж у фінансових та інформаційних системах, а також збільшенням кількості користувачів, які потребують прозорого та зручного аналізу руху активів. У мережі TON доступ до даних забезпечують оглядачі блоків і аналітичні сервіси (наприклад, TonViewer, TonExplorer тощо). Водночас аналіз наявних рішень показує типові обмеження: орієнтацію на перегляд одиничних адрес, відсутність зручного ведення списків спостереження, недостатню підтримку групування та псевдонімів, а також обмежені можливості портфельної агрегації й візуалізації взаємодій між адресами. Унаслідок цього користувачі вимушені виконувати частину операцій вручну, що збільшує витрати часу та підвищує ймовірність помилок.

Метою роботи є розроблення прототипу вебплатформи JetRadar, призначеної для автоматизованого моніторингу публічних адрес у блокчейні TON, побудови графів взаємодій за транзакційними даними та виконання портфельної аналітики.

Об'єктом дослідження є процес моніторингу активності гаманців і транзакцій у мережі TON.

Предметом дослідження є методи та засоби автоматизованого збирання, оброблення та візуалізації даних блокчейну TON, а також застосування графової моделі для аналізу взаємодій і портфельної агрегації.

Для досягнення мети у роботі було поставлено та розв'язано такі завдання:

- виконати аналіз предметної області та наявних інструментів моніторингу TON;
- сформулювати функціональні та нефункціональні вимоги до системи;

- спроектувати архітектуру вебплатформи та модель даних;
- реалізувати серверну частину та клієнтський інтерфейс із інтеграцією у Telegram Mini App;
- реалізувати модулі моніторингу, побудови графів і портфельної аналітики;
- провести тестування ключових сценаріїв і підготувати інструкцію користувача та розгортання.

Методи дослідження включають аналіз літератури й наявних сервісів, функціональне моделювання процесів (IDEF0), проектування архітектури ПЗ, інтеграцію з зовнішнім API, застосування графового підходу до аналізу взаємодій, а також модульне, інтеграційне та UI-тестування прототипу.

Наукова новизна роботи полягає в обґрунтуванні та поєднанні в одному прототипі підходів до моніторингу адрес, графового подання взаємодій та портфельної агрегації для мережі TON з урахуванням практичних обмежень джерела даних (зовнішній API та його ліміти).

Практичне значення роботи полягає в тому, що прототип JetRadar дозволяє автоматизувати операції моніторингу гаманців у блокчейні TON та агрегування даних для груп адрес. Порівняння з ручним аналізом за допомогою стандартних оглядачів (TonViewer, TonScan) показало скорочення витрат часу на виконання типових аналітичних сценаріїв у 10–15 разів і зменшення кількості необхідних дій користувача. Підтримка списків спостереження з псевдонімами та групами, графового відображення зв'язків між адресами й інтегрованого портфельного перегляду підвищує зручність роботи трейдерів і блокчейн-аналітиків та створює основу для подальшого розвитку інструментів аналітики в екосистемі TON.

Структура роботи складається зі вступу, чотирьох розділів, висновків, переліку використаних джерел та додатків.