

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

**на тему: «РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ
АНАЛІЗУ КУРСУ КРИПТОВАЛЮТ»**

| | |
|---------------|--|
| Виконав: | <u>студент 2 курсу, групи 8.1219-з</u> |
| спеціальності | <u>121 інженерія програмного</u> <u>забезпечення</u> <small>(шифр і спеціальність)</small> |
| | <u>Є.О. Рудаченко</u> <small>(ініціали та прізвище)</small> |
| Керівник | <u>доцент кафедри програмної інженерії,</u> <u>к.ф.-м.н. Мильцев О.М.</u> <small>(посада, вчене звання, науковий ступінь, прізвище та ініціали)</small> |
| Рецензент | <u>завідувач кафедри фундаментальної та</u> <u>прикладної математики,</u> <u>доктор технічних наук, доцент, Гребенюк С.М.</u> <small>(посада, вчене звання, науковий ступінь, прізвище та ініціали)</small> |

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти магістр

Спеціальність 121 інженерія програмного забезпечення

(шифр і назва)

Освітня програма інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри програмної
інженерії, к.ф.-м.н., доцент
Лісняк А.О.

(підпис)

« »

2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Рудаченко Євгену Олексійовичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка інформаційної системи аналізу курсу криптовалют

Керівник роботи Мильцев Олександр Михайлович, к.ф.-м.н.

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

затверджені наказом ЗНУ від « 09 » 06 2021 р. № 850-с

2. Строк подання студентом роботи 24.11.2021

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Постановка задачі.

2. Основні теоретичні відомості.

3. Розробка інформаційної системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
презентація до захисту

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|--------|---|----------------|------------------|
| | | Завдання видав | Завдання прийняв |
| | | | |
| | | | |
| | | | |

7. Дата видачі завдання _____ 29.04.2021 _____

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів кваліфікаційної роботи | Строк виконання етапів кваліфікаційної роботи | Примітка |
|----|--|---|----------|
| 1. | Розробка плану роботи. Постановка задачі | 05.06.2021 | Виконано |
| 2. | Збір вихідних даних. Обробка методичних та теоретичних джерел. Ознайомлення з завданням на кваліфікаційну роботу | 17.07.2021 | Виконано |
| 3. | Розробка першого розділу | 24.09.2021 | Виконано |
| 4. | Розробка другого розділу | 16.10.2021 | Виконано |
| 5. | Розробка третього розділу та перший нормоконтроль | 28.10.2021 | Виконано |
| 6. | Оформлення кваліфікаційної роботи і другий нормоконтроль | 18.11.2021 | Виконано |
| 7. | Захист кваліфікаційної роботи | 09.12.2021 | Виконано |

Студент _____
(підпис)

Керівник роботи _____
(підпис)

Нормоконтроль пройдено
Нормоконтролер _____
(підпис)

Є.О. Рудаченко _____
(ініціали та прізвище)

О.М. Мильцев _____
(ініціали та прізвище)

С.П. Швидка _____
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра на тему: «Розробка інформаційної системи аналізу курсу криптовалют»: 46 с., 20 рис., 16 джерел, 2 додатки.

ІНФОРМАЦІЙНА СИСТЕМА, VUE.JS, АНАЛІЗ КРИПТОВАЛЮТ, API, WEBSOCKET.

Мета дослідження: створення інформаційної системи аналізу криптовалют з використанням мови програмування JS та фреймворку Vue.js.

Методи дослідження: аналітичний, порівняльний.

Об'єкт дослідження – сучасні інтерактивні системи аналізу та прогнозу курсу криптовалют.

У кваліфікаційній роботі магістра розглянуто сучасний підхід до створення інтерактивних систем, на основі якого збудовано інформаційну систему аналізу курсу криптовалют. Розглянуто існуючі додатки з аналізу та прогнозу курсу криптовалют.

Перший розділ присвячено аналізу предметної області системи, а саме: загальній характеристиці проекту та оглядом сучасних аналогів системи, які вирішують ті ж самі поставлені задачі.

У другому розділі схематично розглянуто етапи проектування системи.

В межах третього розділу кваліфікаційної роботи розроблено інформаційну систему на базі Vue.js. Результати можуть бути використані для численних приватних інвесторів та великого бізнесу.

SUMMARY

Master's qualifying paper: "Development of Cryptocurrency Exchange Rate Analysis System": 46 pages, 20 figures, 16 references, 2 supplements.

INFORMATION SYSTEM, VUE.JS, CRYPTOCURRENCY ANALYSIS, API, WEBSOCKET.

Aim of the study: create an information system for cryptocurrency analysis using the JS programming language and the Vue.js framework.

Methods of research: method is analytical and comparative.

Object of the study is modern interactive systems of analysis and forecast of the cryptocurrency rate.

In the qualification work of the master the modern approach to creation of interactive systems on the basis of which the information system is constructed is considered. The existing applications for the analysis and forecast of the cryptocurrency exchange rate are considered.

The first section is devoted to the analysis of the subject area of the system, namely: the general characteristics of the project and a review of modern analogues of the system, which solve the same tasks.

The second section schematically discusses the stages of system design.

Within the third section of the qualification work, an information system based on Vue.js was developed. The results can be used for many private investors and big business.

ЗМІСТ

| | |
|--|----|
| ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ | 2 |
| РЕФЕРАТ | 4 |
| SUMMARY | 5 |
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ..... | 8 |
| ВСТУП | 9 |
| 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ..... | 10 |
| 1.1 Опис предметного середовища | 10 |
| 1.1.1 Загальна характеристика проекту | 10 |
| 1.1.2 Опис основних етапів створення системи..... | 13 |
| 1.2 Огляд програмних засобів, що вирішують подібну задачу..... | 14 |
| 2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ | 16 |
| 2.1 Технічне завдання на створення системи..... | 16 |
| 2.1.1 Введення | 16 |
| 2.1.2 Найменування і область застосування..... | 16 |
| 2.1.3 Підстава для розробки..... | 17 |
| 2.1.4 Призначення розробки | 17 |
| 2.2 Проектування системи..... | 17 |
| 2.2.1 Загальна логіка системи та її концептуальна модель..... | 17 |
| 2.2.2 Діаграма варіантів використання | 17 |
| 2.2.3 ЕЕРС діаграма | 18 |
| 2.2.4 Діаграма послідовності | 20 |
| 3. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ АНАЛІЗУ КУРСУ КРИПТОВАЛЮТ | 22 |
| 3.1 Вибір засобів розробки..... | 22 |
| 3.2 Підключення API та WebSocket | 26 |
| 3.3 Розробка макету системи | 28 |

| | |
|---|----|
| 3.4 Прогнозування динаміки курсу криптовалюти | 32 |
| 3.5 Відтворення проекту та загальне тестування | 34 |
| ВИСНОВКИ..... | 38 |
| ПЕРЕЛІК ПОСИЛАНЬ..... | 39 |
| ДОДАТОК А. Програмний код, що реалізує підключення протоколу WebSocket | 41 |
| ДОДАТОК Б. Програмний код, що відображає API клас для встановлення з'єднання з сторонніми сервісами..... | 43 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

1. Скорочення

| | |
|-----|-----------------------------------|
| API | application programming interface |
| NPM | node package manager |
| SDK | software developer kit |

2. Терміни

Фреймворк (каркас) – це програмна платформа, яка визначає структуру програмної системи; програмне забезпечення, що полегшує розробку і об'єднання різних компонентів великого програмного проекту.

Патерн (шаблон проектування) – це повторювана архітектурна конструкція, що представляє собою рішення проблеми проектування в рамках деякого часто виникає контексту.

Інтерфейс – сукупність засобів, методів і правил взаємодії (управління, контролю і т. д.) між елементами системи.

Фронтенд – це клієнтська сторона призначеного для користувача інтерфейсу до програмно-апаратної частини сервісу.

Капча – комп'ютерний тест, використовуваний у тому, щоб визначити, ким є користувач системи: людиною чи комп'ютером.

Блокчейн – збудований за певними правилами безперервний послідовний ланцюжок блоків (зв'язковий список), що містять інформацію. Зв'язок між блоками забезпечується як нумерацією, а й тим, кожен блок містить свою власну хеш-суму і хеш-суму попереднього блоку.

ІСО – форми збору коштів, коли проект залучає інвестиції та продає охочим спеціально створені цифрові токени.

Майнінг – діяльність щодо створення нових структур для забезпечення функціонування криптовалютних платформ.

ВСТУП

Успішні інвестиції – це щоденна копітка робота, вивчення ринкових трендів, моніторинг профільних форумів та соцмереж, розбір технічного аналізу. В наш час існує багато програмних продуктів, які вирішують задачу аналізу та прогнозів курсу криптовалют. Але зазвичай безкоштовні версії таких програм мають певні недоліки і обмежений функціонал. Виходячи з цього, особливо актуальною є розробка такого сервісу, що буде безкоштовним та швидким у вирішенні необхідних задач.

Цілі кваліфікаційної роботи магістра:

- огляд сучасних додатків аналізу курсу криптовалют;
- ознайомлення з теоретичними даними та проектування системи;
- розробка інформаційної системи з аналізу та прогнозування курсу криптовалют.

З огляду на поставлені цілі можна виділити наступні задачі дослідження:

- огляд існуючих сервісів аналізу курсу криптовалют;
- дослідити сучасні методи створення інформаційних систем;
- виділити переваги та недоліки сучасних інформаційних систем аналізу;
- сформулювати вимоги до системи;
- спроектувати та побудувати архітектуру системи;
- реалізація інформаційної системи на базі Vue.js;
- виконати тестування системи.

Результатом вирішення поставлених задач буде функціонуюча та зручна у використанні інформаційна система на базі Vue.js.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметного середовища

1.1.1 Загальна характеристика проекту

Криптовалюта – це цифрові гроші, відрізняється від інших електронних валют тим, що захищена та зашифрована за допомогою спеціальних криптографічних алгоритмів. Головною особливістю криптовалюти є її децентралізованість, незалежність від єдиного центру управління.

Жоден банк, державна структура чи приватна організація неспроможна управляти криптовалютами і проводити скоєні всередині системи транзакції, оскільки дані про процеси зберігаються на безлічі пристроїв одночасно. Підробити або змінити таку інформацію за бажанням однієї людини неможливо, а значить, криптовалюта надійно захищена від цифрових шахраїв.

Всі ці особливості забезпечує технологія блокчейн, на засадах роботи якої функціонує криптовалюта (рис. 1.1). Блокчейн є розподіленою мережею, вузли якої не підконтрольні єдиному центру і формують автономну систему зберігання інформації.

Всі пристрої, які приєднуються до ланцюжка, отримують файл з усіма даними, що коли-небудь записані в цій мережі. Збережена інформація формує блоки, доступні для загального перегляду, але які не підлягають зворотним змінам [13].



Рисунок 1.1 – Принцип, як діє криптовалюта

Криптовалюти – новий вид цифрових грошей, який вже активно використовується при здійсненні електронних платежів. Цифрова валюта має такі характеристики:

- Адаптивне масштабування. Для криптовалют існує ряд правил, які забезпечують їх нормальне функціонування в різних масштабах. Наприклад, алгоритм майнінгу біткоїну коригується в залежності від кількості здобутих за певний час блоків. Передбачається, що за деяких умов обмежується час пропозиції та зменшується винагорода за майнінг (коли обсяги видобутку сильно збільшуються).

- Криптографія. Для криптовалюти використовується особлива система шифрування даних, завдяки якій число створених монет можна тримати під контролем, а також здійснювати транзакції при обміні та розрахунках.

- Децентралізованість. Фіатні гроші створюються певними органами, а системи таких валют контролюються із єдиного центру. Криптовалюти спираються на однорангові мережі, виключаючи можливість впливати на ланцюжок блоків з однієї точки.

– Цифровий характер. Криптовалюта не існує у фізичному вигляді, вона виключно цифрова. Так, крипту можна обміняти на товари або інші види грошей, але цифрова валюта існує тільки в мережі.

– Доказ виконання роботи. Переважна більшість криптовалют діє у системі з підтвердженням виконання. Вона є формулою (капчею), на підтвердження якої потрібні певні обчислювальні потужності.

– Анонімність. Криптовалютні гаманці зашифровані, власники отримують до них доступ за спеціальними ідентифікаторами, ніяк не пов'язаними з особистістю та реальними даними людини. Інформація про транзакції знаходиться у загальному доступі, але дані знеособлені та не ведуть до власника криптовалюти.

– Вартість. Ціну криптовалюти визначає кількість роботи, яку потрібно виконати для видобутку токенів, дефіцит та попит на монети можуть змінювати цю вартість. Такий варіант визначення ціни називається системою із доказом роботи (proof-of-work). Також існує варіант підтвердження монет, що ґрунтується на доказі частки володіння (proof-of-stake).

Незважаючи на повністю цифровий характер криптовалют, за них можна набути звичайних матеріальних цінностей, оплатити послуги. Але крім очевидного та прямого призначення цифрових грошей, для них існує ще кілька варіантів застосування:

– Анонімні платежі. На ринку є криптовалюти, які забезпечують максимальну анонімність того, хто надсилає платіж. До них відносяться Monero, Zcash та PIVX. Завдяки їм можна переказувати великі суми без будь-яких проблем та перешкод з боку банків, органів влади.

– Недорогі грошові перекази. Криптовалютні транзакції здійснюються без залучення третіх осіб, що означає відсутність великих комісійних зборів (які стягують звичайні платіжні системи). Це дозволяє робити платежі за мінімальну вартість, а головне – робити це практично миттєво.

– Спосіб зберігати заощадження. Заморозити або заблокувати криптовалютні рахунки ззовні практично неможливо, тому крипта сьогодні може стати непоганим варіантом збереження накопичень. Особливо це стосується країн, економічний стан у яких вкрай нестабільний, а довіра до банківських систем дуже низька.

– Інвестиції у стартапи. Поява та поширення криптовалют дозволили стартапам простіше та швидше залучати кошти на свої проекти. З іншого боку, цифрові валюти відкрили можливість для багатьох людей вкладатись у перспективні ідеї: за допомогою криптовалюти зробити це може кожен. Зазвичай процес відбувається за допомогою ICO.

– Спосіб оплати у мережі. Транзакції в криптовалютній системі відбуваються з високою швидкістю та відрізняються мінімальною комісією, що робить їх ідеальним засобом для розрахунків в Інтернеті. Це стосується віртуальних магазинів, гральних та інших ресурсів, де важливим є швидкий і надійний спосіб передати гроші або вивести їх з рахунку [6].

1.1.2 Опис основних етапів створення системи

Перед початком створення системи, на етапі проектування, слід виділити її основні етапи. Отже було виділено 5 основних етапів створення інформаційної системи:

- розробка макету сторінок в графічному редакторі;
- верстка макету системи;
- наповнення системи функціональними можливостями;
- підключення API та WebSocket;
- тестування.

Важливо відмітити, що для створення системи недостатньо лише опису, необхідне поєднання блок-схеми, поточної діаграми та функціональної моделі. В процесі проектування вони постійно коригуються і змінюють один одного за рахунок повернення на попередні стадії. В результаті виходить

деякий цілісний опис системи, складові якого, взаємно доповнюють один одного.

1.2 Огляд програмних засобів, що вирішують подібну задачу

Розглянемо декілька аналогів даній системі на прикладах інших програмних продуктів. Першим прикладом програмного застосування є Coincheckup.

Coincheckup – платформа, яка пропонує безліч корисних розділів: аналітичний, в якому розглядається інвестиційна стратегія на даний період часу; питання та відповіді, основний аналіз та порівняння найбільших криптовалют; розділ про ринок, в якому можна знайти інформацію про нинішню ситуацію, огляд ринку, нові криптовалюти, а також списки успішних і провальних монет за останні 24 години. Coincheckup також пропонує розділ з прогнозом на майбутнє, який ґрунтується на ситуації на ринку та оглядах з інших платформ. З недоліків можна визначити – необхідність оновлення сторінки для актуалізації даних.

Наступним прикладом програмного застосування є Map of coins.

Map of coins – платформа, яка розповідає нам про криптовалюти за допомогою цікавої інфографіки. Платформа також містить інформацію про технології кожної з найпопулярніших на сьогодні криптовалют. Задавши певний критерій, платформа відображає ті чи інші монети у вигляді графіків, що допомагає розібратися в запутаній ситуації на ринку.

Розглянемо ще один застосунок для аналізу курсу криптовалют Coinmarketcup.

Coinmarketcup – це сайт із аналітики криптовалют. Платформа надає всю основну інформацію щодо зростання цін на монети, ринкової капіталізації, нових криптовалют, найбільших бірж, списків найкращих і найгірших проектів. На платформі є навіть розділ із крипто-глосарієм, де є найвідоміші

терміни про світ криптовалют. Серед інших корисних інструментів платформи є навіть порівняльні дані щодо ситуації на ринку з 2013 року і до сьогодні.

Наступним прикладом інформаційної системи є CoinDance.

CoinDance – ще один ресурс, що надає дані про світ, криптовалют. На CoinDance можна знайти статистичні дані про користувачів біткоїну, історію зльоту та падінь монети, порівняння з іншими відомими криптовалютами, інформацію про юридичний статус біткоїну по всьому світу та багато іншого. Але основним недоліком системи – є вузька спеціалізація лише з однією монетою.

CryptoCompare є ще одним прикладом системи аналізу курсу криптовалют.

CryptoCompare – платформа яка надає інформацію про монети, біржі, майнінг, пули, гаманці. На додаток, на сайті є форум, де можна обмінятися досвідом та обговорити останні новини зі світу криптовалют. Сервіс фокусується на тому, щоб надати інвесторам найбільш повну картину ринку для прийняття подальшого вдалого рішення про інвестиції. На платформі є безкоштовне API з унікальними даними. Всі дані платформа обробляє онлайн з сайтів найбільших криптовалютних бірж [11].

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Технічне завдання на створення системи

Необхідно розробити інформаційну систему для того, щоб надати можливість швидкого аналізу курсу криптовалют та перегляду умовного прогнозу динаміки зміни курсу криптовалюти.

Призначення системи: аналіз курсу криптовалют.

Цілі створення додатку:

- аналіз курсу криптовалют;
- умовно прогнозувати динаміку курсу криптовалюти;
- швидко знаходити інформацію про курс валюти у вказаному діапазоні дат.

Для реалізації цієї цілі, додаток повинен виконувати такі завдання:

- додавати криптомонети;
- перегляд графіків та діаграм динаміки курсу та прогнозу криптовалюти.

2.1.1 Введення

Дане технічне завдання поширюється на розробку інформаційної системи призначеної для аналізу курсу криптовалют.

2.1.2 Найменування і область застосування

Інформаційна система, що розробляється, отримує назву: «Crypto Dashboard».

Система призначена для додавання записів про криптомонети з подальшим аналізом та відображення у вигляді діаграм та графіків.

2.1.3 Підстава для розробки

Програма розробляється на підставі наказу № 850-с «Про затвердження тем дипломних робіт студентів освітньо-кваліфікаційного рівня «магістр» заочної форми навчання Запорізького національного університету» від 09.06.2021 року.

2.1.4 Призначення розробки

Дана програма призначена для полегшення вирішення поставлених задач, забезпечуючи користувача такими можливостями, як:

- додавати криптомонети;
- перегляд графіків та діаграм динаміки курсу та прогнозу криптовалюти.

2.2 Проектування системи

2.2.1 Загальна логіка системи та її концептуальна модель

Загальна логіка проекту наступна: після того, як користувач починає роботу в системі йому надається можливість додавати відповідні криптовалютні пари з списку, відстежувати історію курсу валют та умовно прогнозувати динаміку зростання чи спаду курсу.

2.2.2 Діаграма варіантів використання

Розглянемо діаграми варіантів використання продукту. Діаграма прецедентів (рис. 2.1) – це діаграма, на якій зображено відношення між акторами та прецедентами в системі [8].

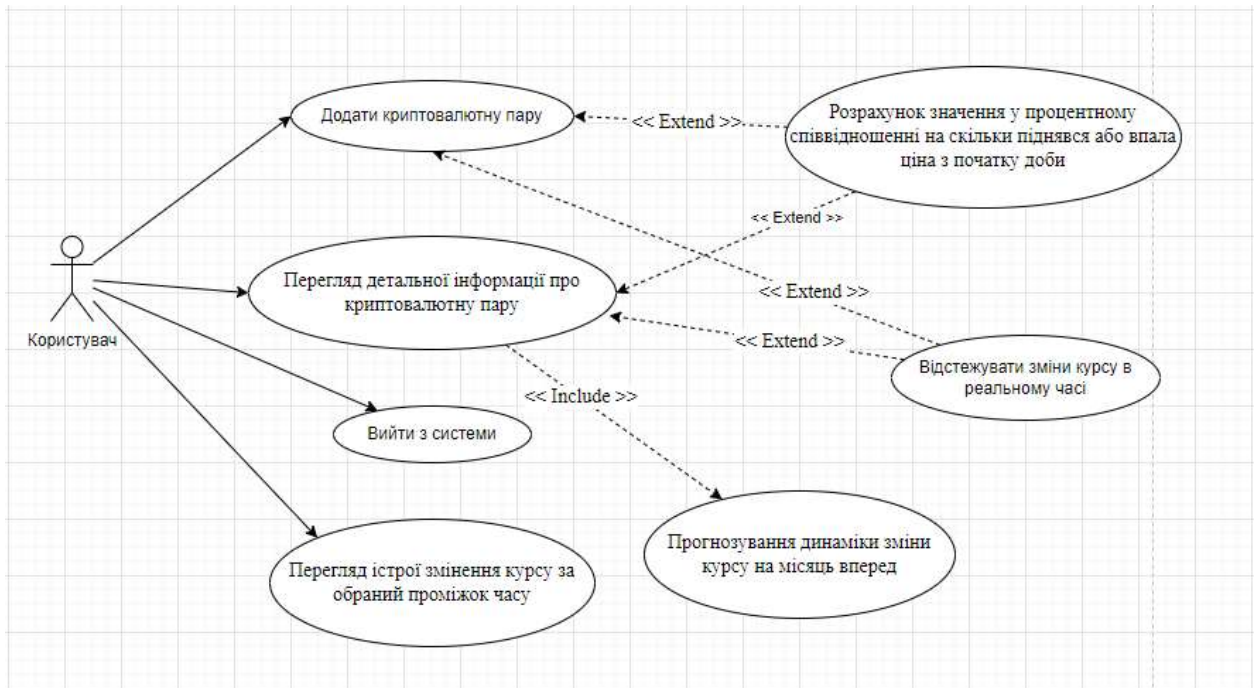


Рисунок 2.1 – Діаграма прецедентів

Проектована система представляється у вигляді множини сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання використовують для описання послуг, які система надає актору [1].

2.2.3 ЕЕРС діаграма

ЕЕРС діаграма (рис. 2.2) – тип діаграм, що використовуються для моделювання, аналізу і реорганізації бізнес-процесів (функціонального моделювання) [5].

У той же час ЕРС-діаграми можуть використовуватися для моделювання поведінки окремих частин системи при реалізації функцій і служити заміною традиційних блок-схем (поведінкового моделювання) [4].

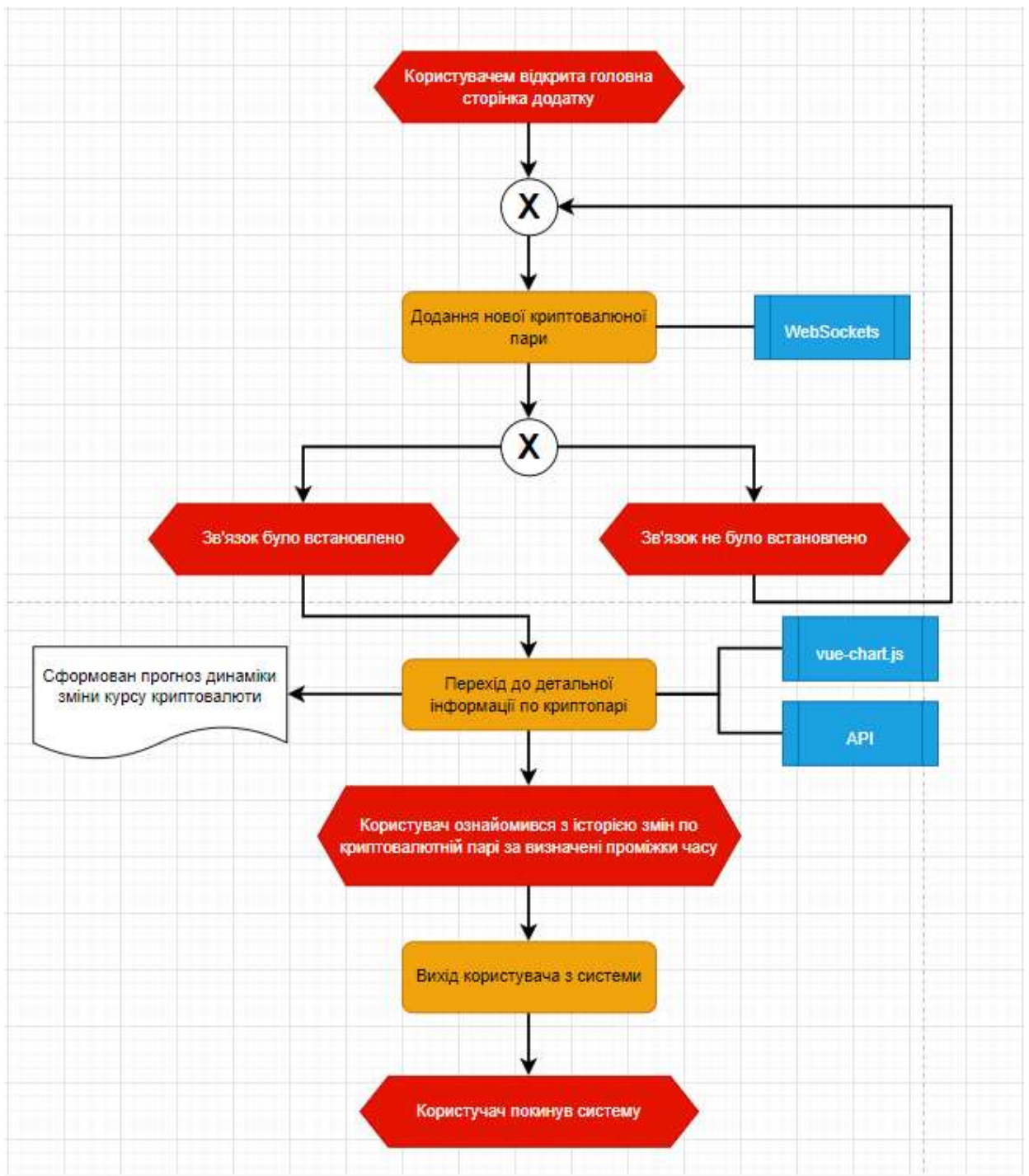


Рисунок 2.2 – EPC діаграма

Незважаючи на відмінності в синтаксисі і семантиці, можна виділити основні елементи методології, які присутні і залишаються незмінними в різних нотаціях. До них відносяться: функція, подія, логічні символи (правила) і потік управління. Інші елементи при однаковій семантиці можуть мати різний зовнішній вигляд [12].

2.2.4 Діаграма послідовності

На діаграмі взаємодії (рис. 2.3) відображають обробку інформації в одному варіанті використання. Поведінка об'єкту представляється його операціями. Діаграми взаємодії містять:

- об'єкти: можна використовувати імена як об'єктів, так і класів, або того і іншого;
- повідомлення: за допомогою повідомлення один об'єкт або клас запитує в іншого виконання якоїсь конкретної функції.

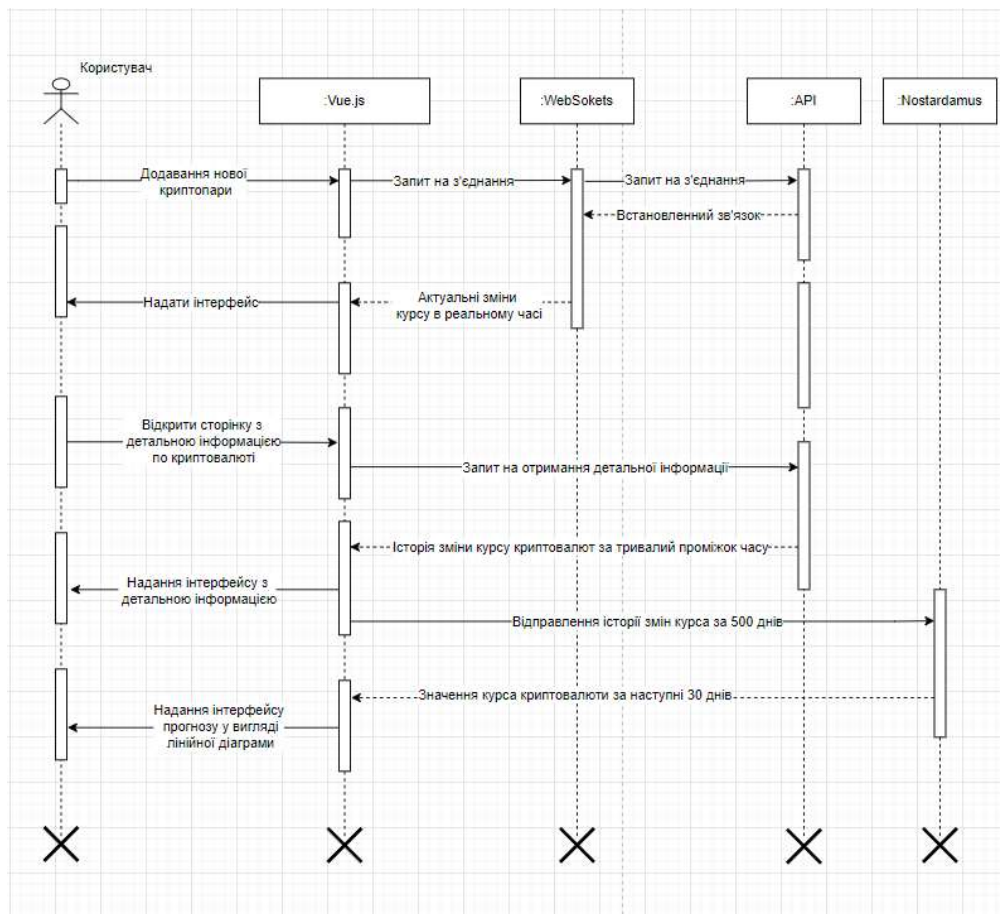


Рисунок 2.3 – Діаграма послідовності

Діаграми послідовності впорядковані за часом. Вони корисні для того, хто хоче зрозуміти логічну послідовність подій в сценарії. Хоча інформація про послідовність входить і в кооперативні діаграми, вона краще сприймається

на діаграмі послідовності. Діаграму послідовності слід читати зверху вниз. В кожного варіанту використання є велика кількість альтернативних потоків. Кожна діаграма послідовності описує один з потоків варіанту використання [3].

Діаграми послідовностей можна використовувати для уточнення діаграм прецедентів, більш детального опису логіки сценаріїв використання. Це відмінний засіб документування проекту з точки зору сценаріїв використання. Діаграми послідовностей зазвичай містять об'єкти, які взаємодіють у рамках сценарію, повідомлення, якими вони обмінюються, і які повертаються результати, які пов'язані з повідомленнями.

У процесі функціонування об'єктно-орієнтованих систем одні об'єкти можуть перебувати в активному стані, безпосередньо виконуючи певні дії, або стані пасивного очікування повідомлень від інших об'єктів. Щоб явно виділити подібну активність об'єктів, в мові UML застосовується спеціальне поняття, що отримало назву фокуса управління. Фокус управління зображується у формі витягнутого вузького прямокутника, верхня сторона якого позначає початок отримання фокусу управління об'єкта, а його нижня сторона - закінчення фокусу управління. Прямокутник розташовується нижче позначення відповідного об'єкта і може замінювати його лінію життя, якщо на всьому її протязі він є активним.

В UML кожна взаємодія описується сукупністю повідомлень, якими ті об'єкти, що беруть участь у ньому обмінюються між собою. Повідомлення є закінченим фрагментом інформації, який відправляється одним об'єктом іншому. Прийом повідомлення ініціює виконання певних дій, спрямованих на вирішення окремого завдання тим об'єктом, якому це повідомлення відправлено.

3. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ АНАЛІЗУ КУРСУ КРИПТОВАЛЮТ

3.1 Вибір засобів розробки

Серед усіх засобів розробки був обраний наступний перелік програмних продуктів, які будуть забезпечувати повний набір функцій та можливостей системи:

- vue.js;
- webSocket;
- bootstrap;
- vuex;
- vue-router;
- vue-chartjs;
- nostradamus.

Vue.js – це прогресивний фреймворк для створення інтерфейсів. На відміну від фреймворків-монолітів, Vue створений придатним для поступового впровадження. Його ядро в першу чергу вирішує завдання рівня уявлення, що спрощує інтеграцію з іншими бібліотеками та існуючими проектами (рис. 3.1) [14].

Розробники називають Vue.js прогресивним та поступово адаптованим у порівнянні з іншими веб-фреймворками.

Це дозволяє розробнику налаштувати структуру програми відповідно до власних вимог. У Vue.js можна використовувати лише знання JavaScript та HTML. Можливе застосування Typescript.

У Vue.js реалізується шаблон MVVM, Vue.js пропонує можливість прив'язки даних на Javascript, так що виведення та введення даних сполучаються безпосередньо з джерелом даних.

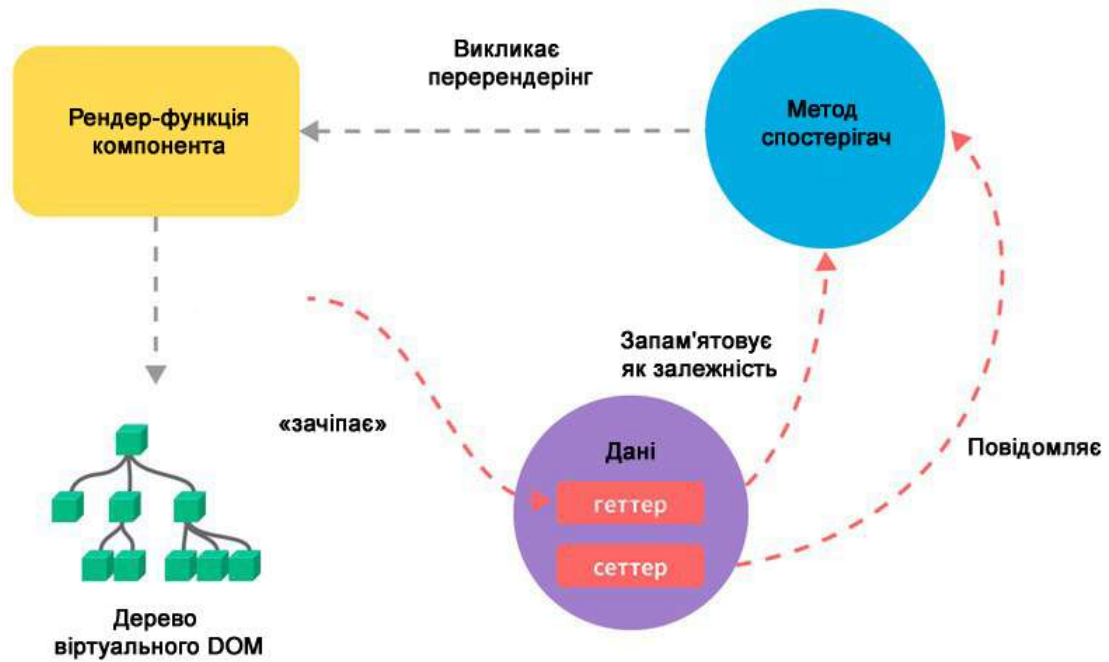


Рисунок 3.1 – Детально про реактивність

WebSocket – протокол зв'язку поверх TCP-з'єднання, призначений обмінюватись повідомленнями між браузером і веб-сервером у реальному часі (рис. 3.2). WebSocket розроблений для втілення у веб-браузерах та веб-серверах, але він може бути використаний для будь-якої клієнтської або серверної програми. Протокол WebSocket – це незалежний протокол, що базується на протоколі TCP. Він уможливує тіснішу взаємодію між браузером та веб-сайтом, сприяючи поширенню інтерактивного вмісту та створенню додатків реального часу [10].

Переваги використання протоколу WebSocket:

- швидкість та ефективність. Високу швидкість і ефективність передачі забезпечує малий розмір даних, що передаються, які іноді будуть поміщатися в один TCP-пакет;
- крос-доменні програми;
- час життя каналу;
- комплексні веб-програми.

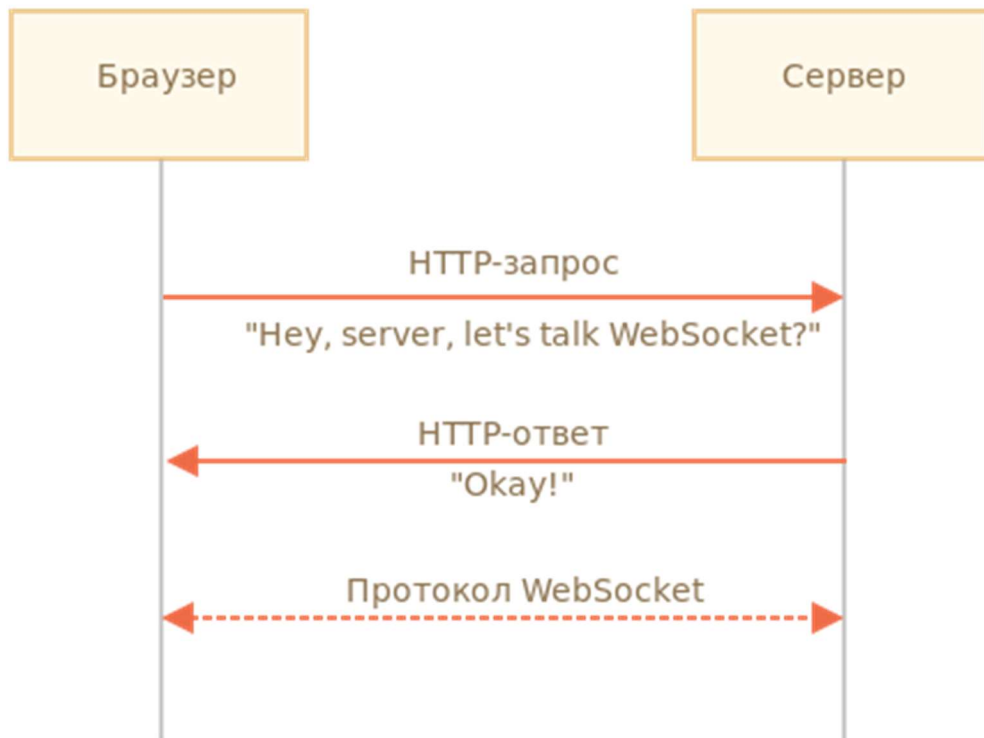


Рисунок 3.2 – Принцип роботи WebSocket

Bootstrap – вільний набір інструментів для створення сайтів та веб-додатків. Включає HTML і CSS-шаблони оформлення для типографіки, веб-форм, кнопок, міток, блоків навігації та інших компонентів веб-інтерфейсу, включаючи JavaScript-розширення [9].

Vueх – патерн управління станом та бібліотека для додатків на Vue.js. Він служить централізованим сховищем даних для всіх компонентів програми з правилами, що гарантують, що стан може бути змінено тільки передбачуваним чином. Vueх інтегрується з офіційним розширенням vue-devtools, надаючи «з коробки» такі просунуті можливості, як «машину часу» для налагодження і експорту / імпорту стану даних (рис. 3.3) [15].

Vueх працює з одним сховищем. Дані програми організовані в деревоподібну структуру. Властивості стану можуть містити будь-який допустимий тип даних від логічних значень масивів та інших об'єктів. Є кілька способів відображення стану уявленнях.

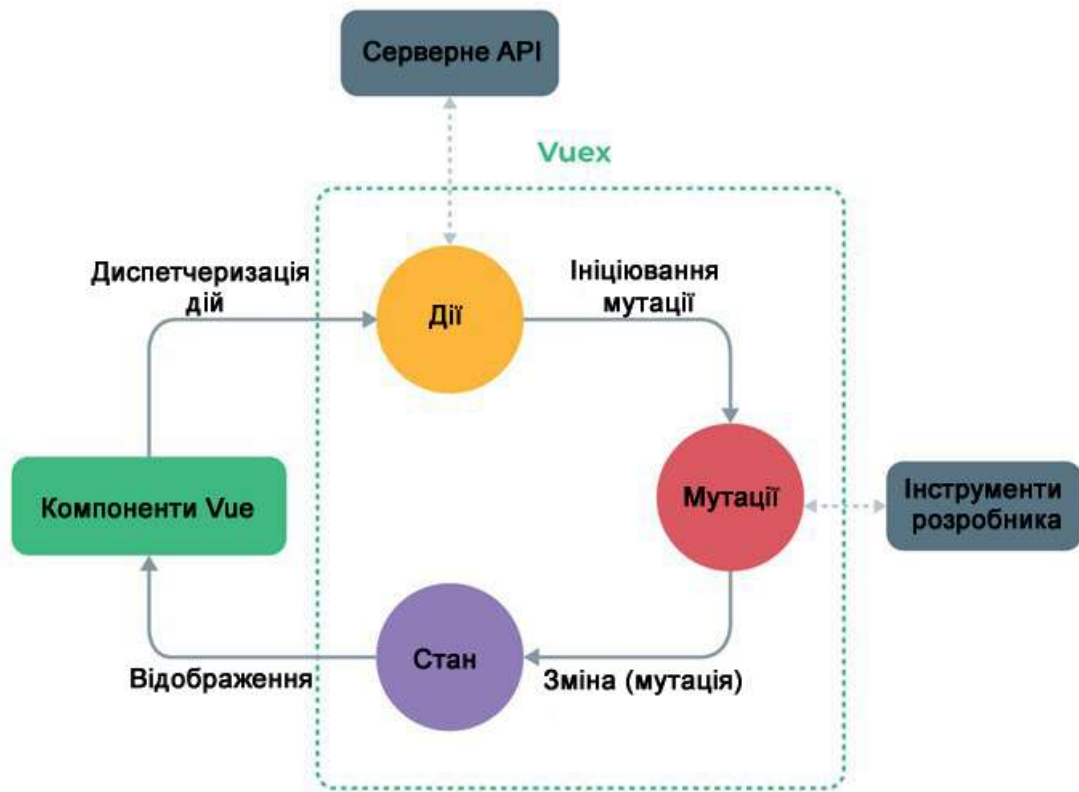


Рисунок 3.3 – Принцип дії бібліотеки Vuex

Vue-router – це офіційна бібліотека маршрутизації для Vue.js. Вона глибоко інтегрується з Vue.js і дозволяє легко створювати SPA-додатки [16]. Включає наступні можливості:

- вкладені маршрути / уявлення;
- модульна конфігурація маршрутизатора;
- анімація переходів уявлень на основі Vue.js;
- зручний контроль навігації;
- автоматичне проставляння активного CSS класу для посилань;
- режими роботи HTML5 «історія» або «хеш», з авто-перемиканням в IE9.

Маршрутизація Vue.js робить програму більш привабливою в аспекті навігації. Вона надає йому можливості односторінкового веб-програми, і створює кращий досвід користувача.

3.2 Підключення API та WebSocket

Першим кроком по реалізації функціональності має стати створення та підключення модулів API (додаток Б) та WebSocket (додаток А). Для отримання детальної інформації щодо змін валют за період часу та перелік можливих криптовалютенів (рис. 3.4) було використано сервіс Binance, який надає безкоштовне API з великим обсягом даних необхідних для заповнення графіків та майбутнього прогнозування динаміки росту чи спаду курсу криптовалюти.

```
1  {}
2  "USDT": {
3    "cid": 825,
4    "name": "Tether",
5    "pairs": [
6      {
7        "cid": 1,
8        "value": "BTC",
9        "name": "Bitcoin",
10       "label": "Bitcoin (BTC)"
11      },
12      {
13        "cid": 1027,
14        "value": "ETH",
15        "name": "Ethereum",
16        "label": "Ethereum (ETH)"
17      },
18      {
19        "cid": 1839,
20        "value": "BNB",
21        "name": "Binance Coin",
22        "label": "Binance Coin (BNB)"
23      },
24      {
25        "cid": 4687,
26        "value": "BUSD",
27        "name": "Binance USD",
28        "label": "Binance USD (BUSD)"
29      },
30      {
31        "cid": 2010,
32        "value": "ADA",
33        "name": "Cardano",
34        "label": "Cardano (ADA)"
35      },
36      {
37        "cid": 3890,
38        "value": "MATIC",
39        "name": "Polygon",
40        "label": "Polygon (MATIC)"
41      },
42      {
43        "cid": 74,
44        "value": "DOGE",
45        "name": "Dogecoin",
46        "label": "Dogecoin (DOGE)"
47      },
48      {
49        "cid": 52,
50        "value": "XRP",
51        "name": "XRP",
52        "label": "XRP (XRP)"
```

Рисунок 3.4 – Перелік можливих криптовалютенів

Система дозволяє відстежувати курс криптовалюти у форматі реального часу, динаміку росту чи спаду курсу у відсотковому та числовому співвідношенні відносно криптовалютної пари (наприклад BTC/USDT), об'єм капіталізації криптовалюти та відображення курсу у вигляді лінійного графіку (рис. 3.5).



Рисунок 3.5 – Картка криптотокена

Капіталізація всіх криптовалют – це загальна вартість всіх монет на даний момент. Існує два основні показники, які впливають на капіталізацію криптовалюти: кількість випущених монет (емісія) та їх курс. Емісія багатьох відомих монет росте через майнінг, оскільки в міру того, як користувачі добувають нову криптовалюту, її кількість поступово збільшується.

Більшість відомих криптовалют, таких як Bitcoin, Ethereum, Litecoin та інших, можна майнити. Однак є й валюти з первинною емісією (коли всі монети відразу випускаються в обіг, наприклад, монета NXT). Загальна капіталізація переважно залежить від ціни та активності над ринком.

Більшою мірою цей показник пов'язаний із вартістю криптовалюти. Ціна цифрових грошей великою мірою залежить від попиту, а попит, своєю чергою, змінюється через кілька чинників:

- поведінки великих інвесторів;
- регулювання із боку урядів великих держав;
- новини.

Обсяг загальної капіталізації знижує волатильність курсу. Потрібно продати BTC на сотні мільйонів доларів, щоб вплинути на його вартість. З маленькими монетами це зробити набагато простіше, тому часто невідомі криптовалюти можуть подорожчати на 80% за кілька годин [2].

3.3 Розробка макету системи

Зовнішній вигляд інформаційної системи був розроблений згідно макету з використанням бібліотеки Bootstrap для забезпечення адаптивного дизайну на всіх пристроях.

Проект складається з «атомної» структури, тобто максимально розділений на під компоненти для швидкої роботи з ними.

Існує кілька правил для іменування компонентів, а саме:

- кожен компонент повинен бути визначений у власному виділеному файлі (SFC);
- однофайлові компоненти мають бути оформлені у форматі PascalCase.

Наприклад, головна сторінка складається з компонентів CurrencyCard.vue та CryptoCard.vue (рис. 3.6).



Рисунок 3.6 – Компоненти з яких складається головна сторінка

Варто зазначити, що система використовує завжди оновлену інформації по курсу валют, тому що система автоматично робить запит до стороннього API, яке в свою чергу надає актуальні данні про запитані валюти.

Для відображення детальної інформації про історію зміни курсу криптовалюти було розроблену сторінку, яка включає в себе картку з необхідної інформацією про активність валюти за останні 24 години та графічним відображенням у вигляді лінійної діаграми та діаграми «Японські свічки» (рис. 3.7). Для швидкого та легкого відтворення графіку було використано «vue-chartjs» – бібліотеку розроблену спеціально для фреймворка Vue.



Рисунок 3.7 – Компоненти з яких складається сторінка з детальною інформацією

Японські свічки – вид інтервального графіка та технічний індикатор, що застосовується головним чином для відображення змін біржових котирувань акцій. На відміну від звичного європейцям лінійного графіка, свічки дають набагато більше інформації про рух ціни: параметри відкриття та закриття торгів, а також ціновий мінімум та максимум за позначений проміжок часу.

Таким чином, свічковий аналіз дозволяє інвестору зрозуміти, як змінювалася ціна протягом певного часового періоду. Мінімуми та максимуми позначають повний діапазон коливання, тіло свічки показує реальні зміни курсу валюти чи ціни іншого активу. Це особливо важливо на довгострокових графіках, де зміни цілого торговельного дня відображаються на одному тимчасовому відсіканні.

Японські свічки об'єднують в собі дані про ціни відкриття / закриття тимчасового періоду, максимальної / мінімальної ціни, досягнутих всередині періоду, і про те, чи зросла ціна за аналізований період або, навпаки, знизилася.

Однією з переваг системи є вибір мови інтерфейсу зрозумілої для користувача: англійської або російської. Зроблено це за рахунок зберігання ключових слів у JSON файлах відповідної мови та обробки інформації завдяки фільтру.

Системою передбачено додавання нових криптовалютенів та видалення вже існуючих з головної сторінки системи (рис. 3.8 – рис. 3.9) з автоматичним підключенням WebSocket та отримання актуальної інформації про курс криптовалют в реальному часі.

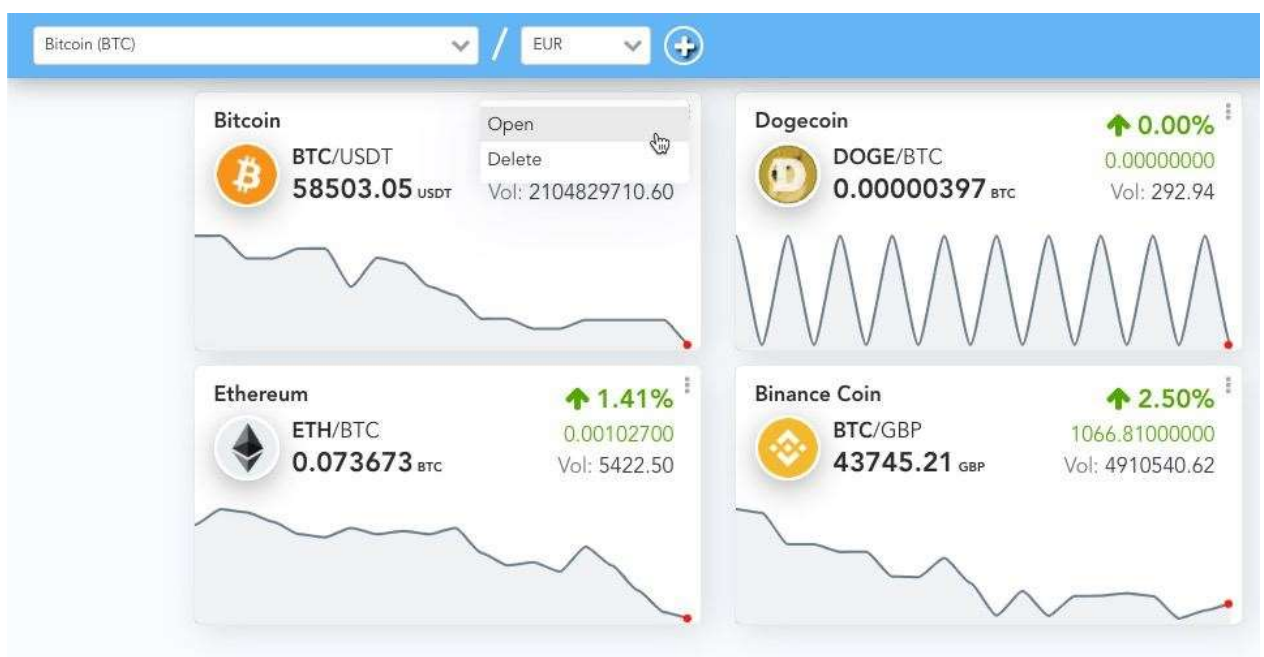


Рисунок 3.8 – Меню з управлінням криптовалюти

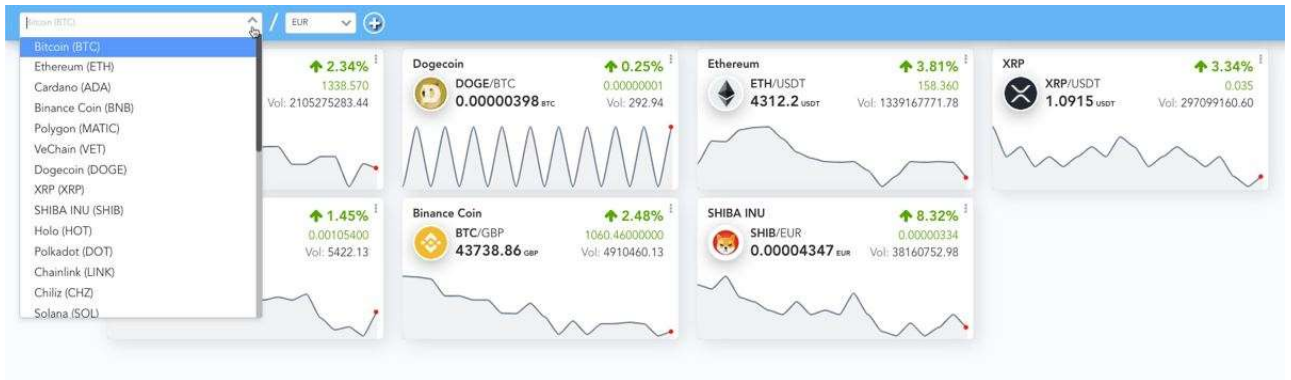


Рисунок 3.9 – Додання нової криптопари

Розроблена система є надзвичайно гнучкою та зручною для користувача, тому що система надає можливість перегляду та фільтрування інформації про динаміку змінення курсу криптовалюти у різних часових проміжках (рис 3.10 – рис. 3.11). Система надає можливість перегляду історії зміни курсу криптовалюти за довготривалий проміжок часу з детальною інформацією до однієї години з даними про відкриття, закриття, мінімальне та максимальне значення.



Рисунок 3.10 – Перегляд інформації про курс валюти з інтервалом в 1 годину



Рисунок 3.11 – Перегляд інформації про курс валюти за квітень цього року

3.4 Прогнозування динаміки курсу криптовалюти

У зв'язку з зростаючою невизначеністю зовнішнього середовища, збільшенням ринкової конкуренції виникає необхідність у найбільш якісному прийнятті управлінських рішень у галузі стратегічного та тактичного планування. Також, зважаючи на найбільше поширення інтегрованого планування на підприємствах і в ланцюгах поставок, зростає важливість складової прогнозування в системі планування.

Одним з методів, що враховує кілька складових прогнозу, є метод Хольта-Вінтерса. Метод Хольта-Вінтерса – це трипараметрична модель прогнозу, яка враховує:

- згладжений експоненційний ряд;
- тренд;
- сезонність.

Розробка прогнозування динаміки курсу криптовалюти (рис. 3.12) розпочинається з встановлення допоміжної бібліотеки Nostardamus, де реалізовано алгоритм експоненційного згладжування Хольта-Вінтерса.

Прогноз на p періодів наперед розраховується за формулою (1)

$$Y_{i-p} = (L_i + p \cdot T_i) \cdot S_{i-s+p} \quad (1)$$

де \hat{Y}_{i-p} – прогноз методом Хольта-Винтерса на p періодів вперед;
 S_{i-s+p} – коефіцієнт сезонності за цей же період в останньому сезоні.

Цей метод має такі переваги:

Метод Хольта-Винтерса включає три складових: основну тенденцію розвитку, сезонність і експоненційно-згладжений ряд. Завдяки тренду з включеним експоненційним згладжуванням, можна не тільки виявити напрямок розвитку ряду динаміки, але і згладити дрібні коливання в ряді динаміки для пошуку приватних спадів і стрибків. Сезонність дозволяє побудувати прогноз на майбутні періоди з урахуванням даної сезонності, що наочно показує побудований прогноз.

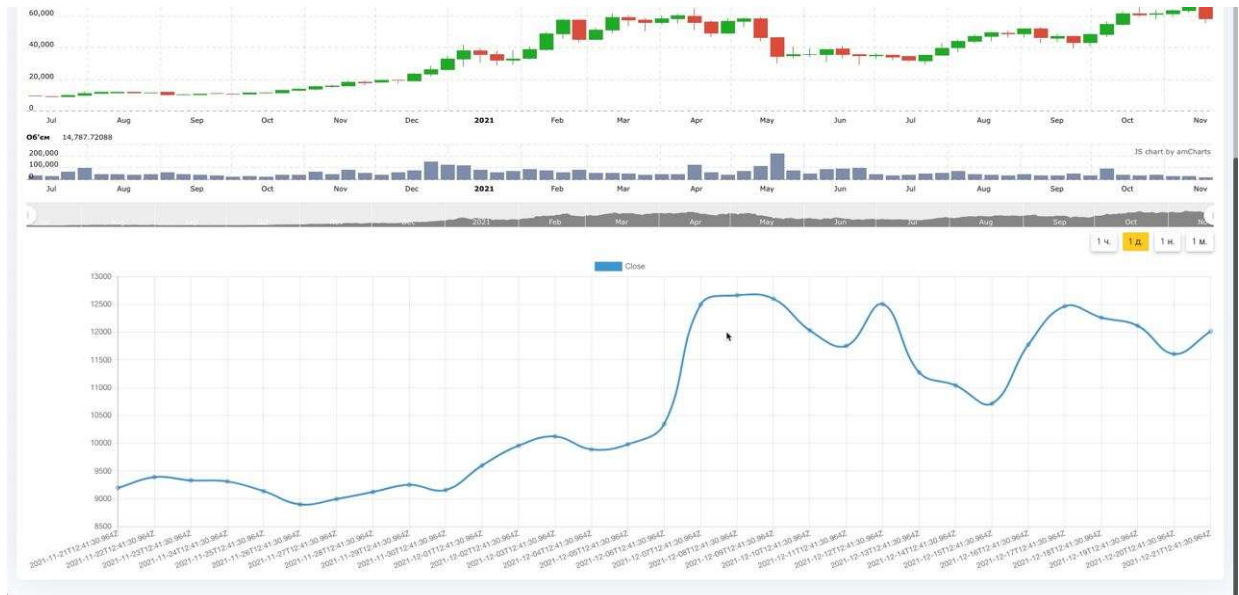


Рисунок 3.12 – Прогнозування курсу криптовалюти

Даний метод дозволяє будувати прогнози на досить великий проміжок часу. Оскільки він враховує кілька факторів прогнозування, то він буде більш

точним при прогнозуванні на тривалий проміжок часу. Але цей метод має також такі обмеження.

Для побудови найточнішого прогнозу необхідні дані за досить тривалий проміжок часу. Для побудови найточнішого прогнозу бажано використовувати дані за 4–5 прогнозних періодів. Відповідно, що більше тимчасових даних, тим на більш тривалий період вперед має бути здійснено прогноз.

Для цього необхідно проводити додаткову оцінку точності прогнозу. Наведений раніше алгоритм застосовується тільки для оцінки прогнозу за даним методом, оскільки коефіцієнти k , b і q підбираються вручну і безпосередньо впливають на поточну оцінку [4].

3.5 Відтворення проекту та загальне тестування

Перед будь-яким тестуванням потрібно скласти план, за яким воно буде відбуватися із різними варіантами відтворення або знаходження помилок. Головною ціллю людини, котра тестує, є зламати систему, поле, програму чи процес перевірки відповідності заявлених до продукту вимог та реально реалізованої функціональності, що здійснюється шляхом спостереження за його роботою у штучно створених ситуаціях та на обмеженому наборі тестів, обраних певним чином.

Тестування інформаційної системи відбувається у два етапи:

- кросбраузерність;
- адаптивне відображення на інших засобах.

Перший етап відповідає за перевірку наступних пунктів:

- відкрити проект інформаційної системи у двох трьох браузерах;
- перевірка чи відбувається коректне відображення інформації за

планом.

Тестування кросбраузерності – вид тестування, спрямований на підтримку та правильне повне відображення програмного продукту у різних браузерах, мобільних пристроях, планшетах, екранах різного розміру.

Кросбраузерне тестування – важливий етап при розробці будь-якої програми. Адже зовнішній вигляд сайту та його коректне відображення на будь-якому сучасному пристрої грає визначальну роль для замовника.

Тестування сайту починається з вибору браузерів. Замовник сам визначає, з якими саме веб-браузерами працюватиме його програма. Але завдання розробника та тестувальника – підказати клієнту, який браузер буде головним, слід вивчити статистику заходів подібних додатків, визначити, якими браузерами користується така аудиторія.

Основні моменти для тестування: верстка (колір, шрифти, розташування графічних картинок та динамічних елементів) та JavaScript.

Як правило, розглядають найпопулярніші браузери: Google Chrome, Mozilla Firefox, Internet Explorer, Опера (рис. 3.13 – рис. 3.15).

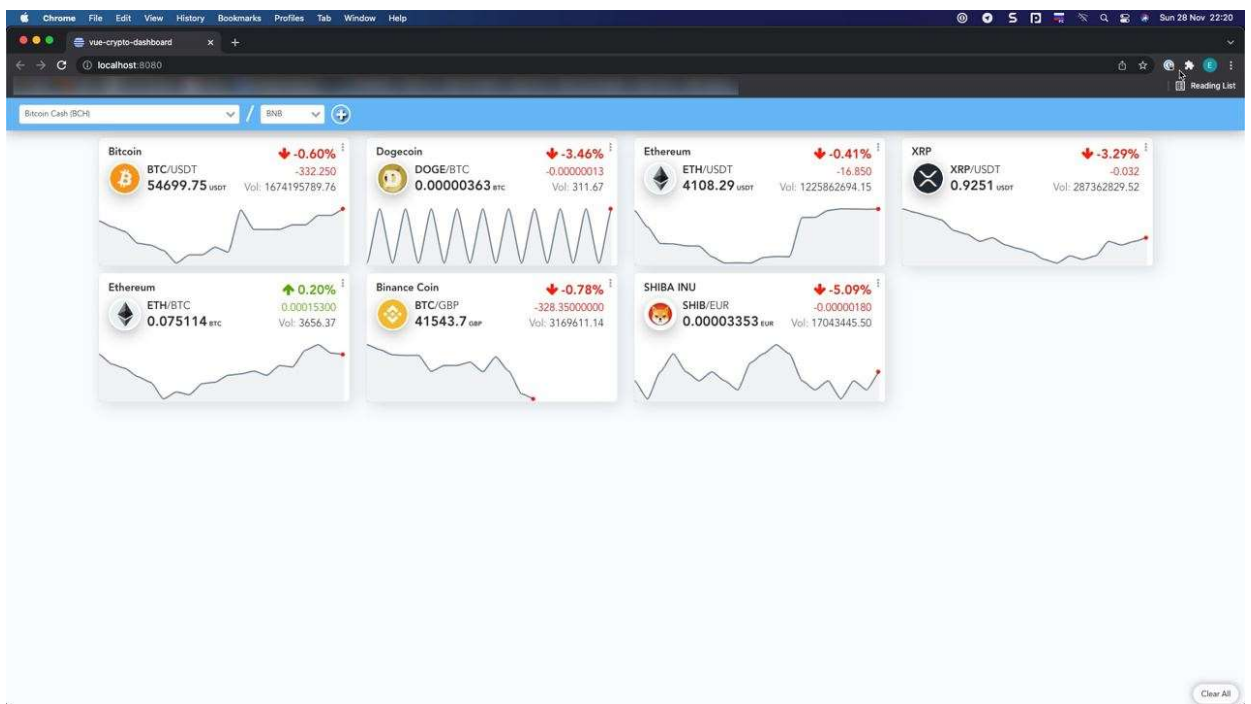


Рисунок 3.13 – Відображення інтерфейсу у браузері Google Chrome

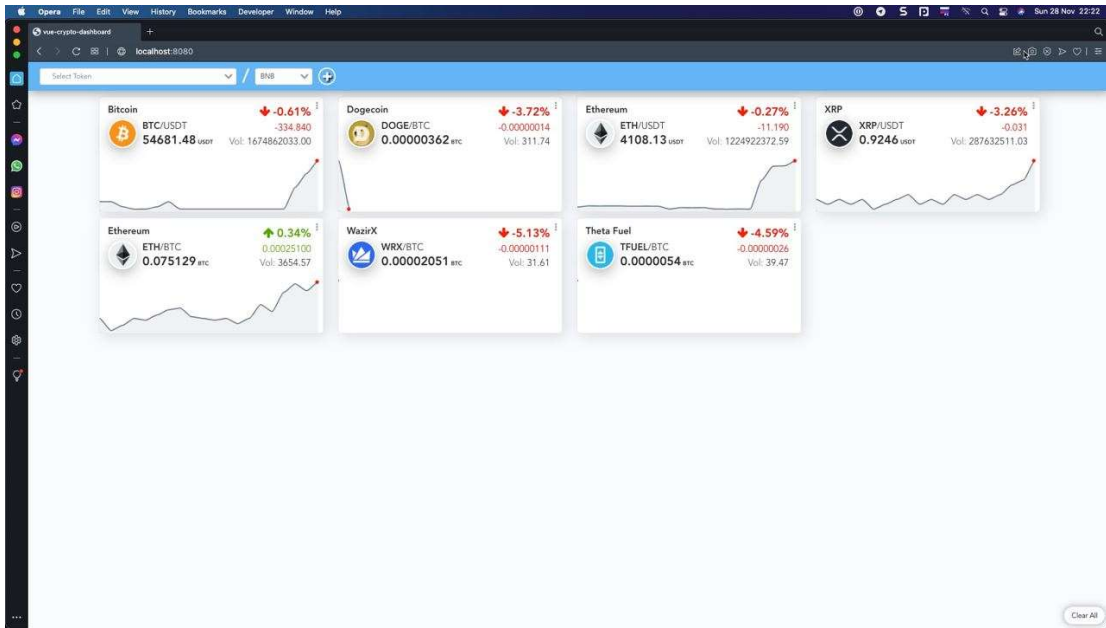


Рисунок 3.14 – Відображення інтерфейсу у браузері Опера

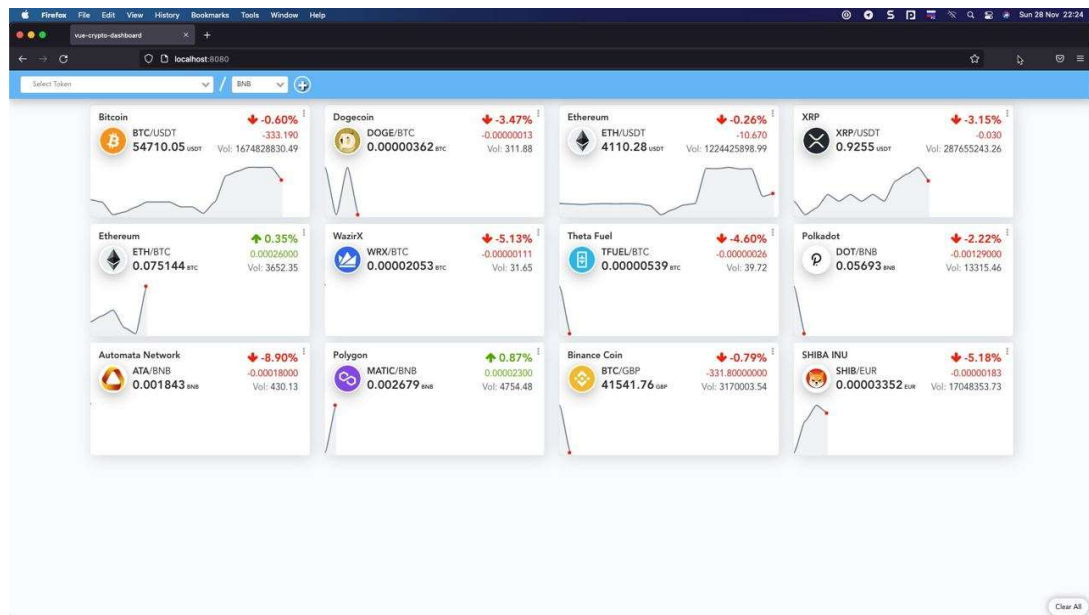


Рисунок 3.15 – Відображення інтерфейсу у браузері Mozilla Firefox

Другий етап відбувався за наступним планом:

- відкрити проект на різних мобільних засобах;
- перевірити чи має система адаптивний інтерфейс та чи доступний весь функціонал системи на інших засобах (рис. 3.16).

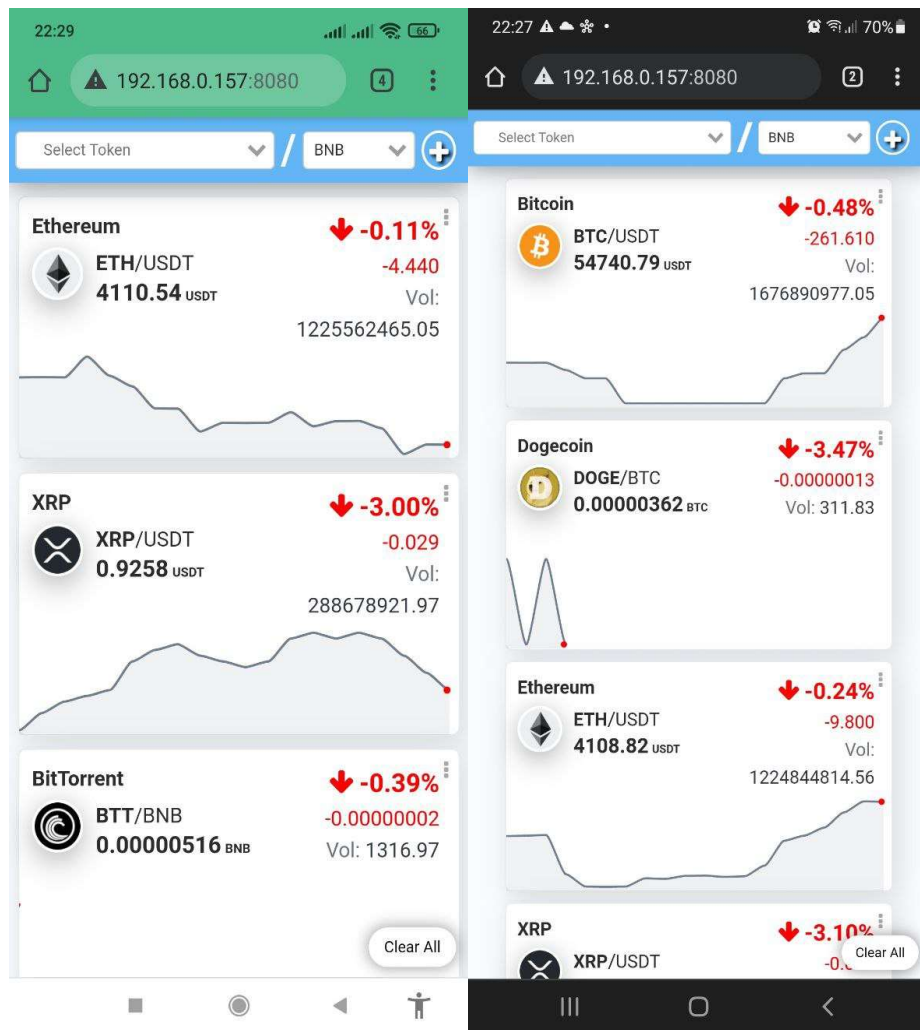


Рисунок 3.16 – Відображення системи на мобільних засобах

Тестуванням системи повинно бути перевірено найбільшу кількість вірогідних «шляхів задач», тобто людина, яка тестує, має намагатись навмисно зламати систему, аби цього не трапилося під час того, як система чи проект буде використовуватися за призначенням.

Починають тестування з перевірки на відповідність вимогам та дизайну. Мова тут не тільки про те, щоб перевірити наявність всіх картинок або працездатність посилань, а про повноцінний UX/UI аналіз.

Для тестування на різних пристроях використовують емулятори типу Genymotion, BlueStacks. Однак успішні тести на емуляторі не гарантують, що програма працюватиме без збоїв на реальних пристроях.

ВИСНОВКИ

Отже, в процесі виконання дипломного проекту було створено інформаційну систему аналізу курсу криптовалют. Система дозволяє додавати, видаляти, спостерігати та прогнозувати курс криптовалют. Перед цим розглянуто такі програмні аналоги, як Coinchecker та Coinmarketcap. Створений проект підтримується на усіх версія браузерів. Платформа реалізована на останній версії мови програмування JS з використанням фреймворку Vue.js, бібліотек для створення макету сторінки та її функціоналу.

Виконана робота дозволяє зробити наступні висновки:

1. В сучасному світі інформаційні системи аналізу криптовалют мають важливу роль в повсякденному житті.
2. Під час створення системи, важливо правильно та завчасно обрати зручну технологію, для швидкого процесу та вдалих результатів.
3. Для правильного та всеосяжного тестування системи, потрібно створити план заздалегідь.

Результатом дипломної роботи є створена інформаційна система аналізу криптовалют. Платформа охоплює:

- додання нових криптовалют;
- графічне відображення історії змінення курсу криптовалюти;
- прогнозування динаміки росту чи спаду курсу криптовалюти;
- видалення криптовалют;
- зручний та зрозумілий інтерфейс.

ПЕРЕЛІК ПОСИЛАНЬ

1. Буч Г. UML. Інструкція користувача. Москва: ДМК Пресс, 2014. 432 с.
2. RBC. Що таке капіталізація криптовалют. URL: <https://www.rbc.ru/crypto/news/5aa7a08d9a794719f1db24fb> (дата звернення 14.11.2021).
3. Богс М. UML та RationalRose. Москва: РГГУ, 2016. 438с.
4. Поздняков А. Застосування методу Хольту-Вінтерсу при аналізі та прогнозуванні динаміки тимчасових рядів. Москва: НОУ ИНТУИТ, 2016. 209 с.
5. Кімвел П. Основи візуального аналізу та проектування UML. Москва: НТ Пресс, 2008. 272 с.
6. Максимчук А. Управління впровадженням інформаційних систем. Москва: РГГУ, 2012. 224 с.
7. Нові знання. IDEF4. URL: <http://ru.knowledgr.com/07408900/IDEF4> (дата звертання 10.05.2018).
8. Пайлон Д. UML 2 для програмістів. Москва: Пітер, 2012. 198 с.
9. Bootstrap. Вступ. URL: <https://getbootstrap.com/docs/5.1/getting-started/introduction/> (дата звертання 16.11.2021).
10. Вікіпедія. WebSocket. URL: <https://ru.wikipedia.org/wiki/WebSocket> (дата звернення 13.11.2021).
11. VS. 6 інструментів для аналітики криптовалют: відгуки про платформи. URL: <https://vc.ru/crypto/46257-6-instrumentov-dlya-analitiki-kriptoalyut-otzyvu-o-platformah> (дата звертання 23.11.2021).
12. Google Sites. EPC-діаграми. URL: https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema8/tema8_3 (дата звертання 14.11.2021).

13. Blockchain & Bitcoin. Що таке криптовалюта: принцип роботи та переваги цифрових грошей. URL: <https://prague.bc.events/ru/news/что-такое-kriptoalyuta-i-kak-ona-rabotat> (дата звертання 20.11.2021).

14. Vue. Введення. URL: <https://ru.vuejs.org/v2/guide/> (дата звертання 5.11.2021).

15. Vuex. Що таке Vuex? URL: <https://vuex.vuejs.org/ru/> (дата звертання 19.11.2021).

16. Vue Router. Введення. URL: <https://router.vuejs.org/ru/> (дата звертання 5.11.2021).

ДОДАТОК А

Програмний код, що реалізує підключення протоколу WebSocket

```
function noop() {}  
export default function (url, opts) {  
  opts = opts || {};  
  
  let k, ws, num, $={}, self=this;  
  let ms=opts.timeout || 1e3, max=opts.maxAttempts || Infinity;  
  
  $.onmessage = opts.onmessage || noop;  
  
  $.onclose = e => {  
    e.endpoint = url;  
    (e.code !== 1e3 && e.code !== 1005) && self.reconnect(e);  
    (opts.onclose || noop)(e);  
  };  
  
  $.onerror = e => {  
    e.endpoint = url;  
    (e && e.code==='ECONNREFUSED') ? self.reconnect(e) : (opts.onerror ||  
noop)(e);  
  };  
  
  $.onopen = e => {  
    e.endpoint = url;  
    num=0; (opts.onopen || noop)(e);  
  };  
};
```

```
self.open = () => {
  ws = new WebSocket(url, opts.protocols);
  for (k in $) ws[k] = $[k];
};

self.reconnect = e => {
  (num++ < max) ? setTimeout(() => {
    (opts.onreconnect || noop)(e);
    self.open();
  }, ms) : (opts.onmaximum || noop)(e);
};

self.close = (x, y) => {
  ws.close(x, y);
};

self.open(); // init

return self;
}
```

ДОДАТОК Б

Програмний код, що відображає API клас для встановлення з'єднання з сторонніми сервісами

```
import WS from './ws'

export default class Api {
  constructor(timeout=5e3, maxAttempts=5) {
    this.timeout = timeout;
    this.maxAttempts = maxAttempts;
    this._baseUrl = 'wss://stream.binance.com:9443/ws/';
    this._combinedBaseUrl = 'wss://stream.binance.com:9443/stream?streams=';
    this.subscription = {};
    this.streams = {
      depth: (symbol) => `${symbol.toLowerCase()}@depth`,
      depthLevel: (symbol, level) => `${symbol.toLowerCase()}@depth${level}`,
      kline: (symbol, interval) => `${symbol.toLowerCase()}@kline_${interval}`,
      aggTrade: (symbol) => `${symbol.toLowerCase()}@aggTrade`,
      trade: (symbol) => `${symbol.toLowerCase()}@trade`,
      ticker: (symbol) => `${symbol.toLowerCase()}@ticker`,
      miniTicker: (symbol) => `${symbol.toLowerCase()}@miniTicker`,
      allMiniTicker: () => '!miniTicker@arr',
      allTickers: () => '!ticker@arr'
    };
  }

  subscribe (cb, endpoint, isCombined = false) {
    let ws;
```

```

try {
  let path = (isCombined ? this._combinedBaseUrl : this._baseUrl) + endpoint;
  if (this.subscription[path]) {
    return this.subscription[path];
  }
  ws = new WS(path, {
    timeout: this.timeout,
    maxAttempts: this.maxAttempts,
    onopen: e => console.log('Connected!', e),
    onmessage: e => cb(JSON.parse(e.data)),
    onreconnect: e => console.log('Reconnecting...!', e),
    onmaximum: e => console.log('Stop Attempting!', e),
    onclose: e => { console.log('Closed!', e);
this.removeSubscription(e.endpoint)},
    onerror: e => console.log('Error:', e)
  });
  this.subscription[path] = ws
}
catch(ex) {
  console.log("Error :" + ex)
}
}
caller(fname, ...args) {
  return this.streams[fname].call(null, ...args)
}
removeSubscription(endpoint){
  if(this.subscription[endpoint]) {
    delete this.subscription[endpoint]
  }
}
}

```

```
closeSubscription(type, isCombined = false, ...args) {
  const endpoint = this.caller(type, ...args);
  const path = (isCombined ? this._combinedBaseUrl : this._baseUrl) + endpoint;
  const ws = this.subscription[path];
  if(ws) {
    ws.close(1000, "");
  }
}

closeAll() {
  for (const key in this.subscription) {
    this.subscription[key].close();
    delete this.subscription[key]
  }
  this.subscription = {}
}

onDepthUpdate(symbol, eventHandler) {
  return this.subscribe(eventHandler, this.streams.depth(symbol));
}

onDepthLevelUpdate(symbol, level, eventHandler) {
  return this.subscribe(eventHandler, this.streams.depthLevel(symbol, level));
}

onKline(symbol, interval, eventHandler) {
  return this.subscribe(eventHandler, this.streams.kline(symbol, interval));
}

onAggTrade(symbol, eventHandler) {
  return this.subscribe(eventHandler, this.streams.aggTrade(symbol));
}
```

```
onTrade(symbol, eventHandler) {
    return this.subscribe(eventHandler, this.streams.trade(symbol));
}

onTicker(symbol, eventHandler) {
    return this.subscribe(eventHandler, this.streams.ticker(symbol));
}

onMiniTicker(symbol, eventHandler) {
    return this.subscribe(eventHandler, this.streams.miniTicker(symbol));
}

onAllMiniTickers(eventHandler) {
    return this.subscribe(eventHandler, this.streams.allMiniTicker());
}

onAllTickers(eventHandler) {
    return this.subscribe(eventHandler, this.streams.allTickers());
}

onCombinedStream(streams, eventHandler) {
    return this.subscribe(eventHandler, streams.join('/'), true);
}
}
```