

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ПРИРОДОКОРИСТУВАННЯ**

**ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ  
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

# КВАЛІФІКАЦІЙНА РОБОТА

першого (бакалаврського) рівня вищої освіти

на тему: **“ Розробка програмного інструментарію для  
прогнозування курсу криптовалюти ”**

Виконав: студент гр. ІТ-41  
Спеціальності 126 – «Інформаційні системи та  
технології»  
(шифр і назва)

Сокульський Андрій Володимирович  
(Прізвище та ініціали)

Керівник: к.т.н., доц. Луб П.М.  
(Прізвище та ініціали)

Рецензенти: к.т.н., доц. Бабич М.І.  
(Прізвище та ініціали)

\_\_\_\_\_  
(Прізвище та ініціали)

**ДУБЛЯНИ-2024**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ  
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Перший (бакалаврський) рівень вищої освіти  
126 – «Інформаційні системи та технології»

“ЗАТВЕРДЖУЮ”

Завідувач кафедри \_\_\_\_\_  
д.т.н., проф. А.М. Тригуба  
“ \_\_\_\_\_ ” \_\_\_\_\_ 2023 р.

## ЗАВДАННЯ

на кваліфікаційну роботу студенту

Сокульський Андрій Володимирович

1. Тема роботи: «Розробка програмного інструментарію для прогнозування курсу криптовалюти»

Керівник роботи Луб Павло Миронович, к.т.н., доцент  
Затверджені наказом університету 27.11.2023 року №641/к-с.

2. Строк подання студентом роботи 17.06.2024 р.

3. Початкові дані до роботи: 1. Офіційні дані щодо блокчейну, криптовалюти, майнінгу, транзакцій. 2. API криптобірж. 3. Методика прогнозу динаміки ряду цифр. 4. Методика проектування інформаційних систем IDEF.

4. Зміст розрахунково-пояснювальної записки:

1. Аналіз технологій та принципів роботи криптовалюти;
2. Інструменти прогнозування курсу та робота криптобіржі;
3. Розробка програмного інструментарію;
4. Охорона праці та безпека в надзвичайних ситуаціях;
5. Висновки;
6. Список літератури;
7. Додатки.

5. Перелік презентаційного матеріалу : 1 та 2 – Тема, мета, завдання роботи; 3 – Головні поняття; 4 – Аналіз популярних криптобірж; 5 – Принципи виконання торгів з криптовалютою; 6 – Веб-системи прогнозування курсу; 7 – Принципи роботи API криптобіржі; 8 – Концептуальна модель та декомпозиція; 9 – Визначення рамок, процесів і потоків даних; 10 – Діаграма класів та діаграма діяльності; 11 – Діаграма компонентів і діаграма розгортки; 12 – Реалізація програмного засобу.

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 4	<i>Луб П.М., доцент кафедри інформаційних технологій</i>		
5	<i>Городецький І.М., доцент кафедри фізики, інженерної механіки та безпеки виробництва</i>		

7. Дата видачі завдання 27.11.2023 р.

### ***КАЛЕНДАРНИЙ ПЛАН***

№ з/п	Етапи виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Написання першого розділу та означення головних завдань роботи</i>	<i>27.11.2023 – 01.01.2024</i>	
2.	<i>Виконання другого розділу та формування головних показників для розрахунків</i>	<i>01.01.2024 – 01.02.2024</i>	
3.	<i>Виконання третього розділу, побудова коду та розробка презентації</i>	<i>01.02.2024 – 01.03.2024</i>	
4.	<i>Написання розділу: «Охорона праці та безпека в надзвичайних ситуаціях»</i>	<i>01.03.2024 – 01.04.2024</i>	
5.	<i>Завершення оформлення розрахунково-пояснювальної записки та презентації</i>	<i>01.05.2024 – 01.06.2024</i>	
6.	<i>Завершення роботи в цілому</i>	<i>01.06.2024 – 14.06.2024</i>	

Студент \_\_\_\_\_ Сокульський А.В.  
(підпис)

Керівник роботи \_\_\_\_\_ Луб П.М.  
(підпис)

УДК 004.42:004.85:336.74

Розробка програмного інструментарію для прогнозування курсу крипто валюти. Сокульський А.В. Кафедра ІТ. – Дубляни, Львівський НУП, 2024.

Кваліфікаційна робота: 66 с. текст. част., 23 рис., 5 табл., 10 слайдів, 21 джерело.

Означено головні поняття і принципи створення криптовалюти. Виконано огляд популярних бірж криптовалюти. Наведено принцип виконання торгів із валютою з боку користувача. Розглянуто аналітичні інструменти крипто валютних бірж.

Охарактеризовано програми та веб-системи прогнозування курсу криптовалют. Наведено методи прогнозу динаміки курсу криптовалюти. Описано принцип роботи API криптовалютних бірж.

Розроблено структурну модель та декомпозицію програмного засобу в IDEF Modeling Techniques. Визначено рамки, процеси і потоки даних. Розроблено об'єктно-орієнтовану модель системи прогнозування курсу криптовалют. Виконано реалізацію програмного засобу.

Розроблено заходи з охорони праці та безпека в надзвичайних ситуаціях.

**Ключові слова:** блокчейн, криптовалюта, курс, прогнозування, криптобіржа, API, програмне забезпечення.

**Keywords:** blockchain, cryptocurrency, exchange rate, forecasting, crypto exchange, API, software.

## ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1.	
АНАЛІЗ ТЕХНОЛОГІЙ ТА ПРИНЦИПІВ РОБОТИ КРИПТОВАЛЮТИ	7
1.1. Головні поняття і принципи створення криптовалюти.....	7
1.2. Огляд популярних бірж криптовалюти.....	9
1.3. Принцип виконання торгів із валютою з боку користувача.....	13
1.4. Аналітичні інструменти крипто валютних бірж.....	15
РОЗДІЛ 2.	
ІНСТРУМЕНТИ ПРОГНОЗУВАННЯ КУРСУ ТА РОБОТА КРИПТОБІРЖІ.....	17
2.1. Програми та веб-системи прогнозування курсу криптовалют....	17
2.2. Методи прогнозу динаміки курсу криптовалюти .....	20
2.3. Принцип роботи API криптовалютних бірж.....	23
РОЗДІЛ 3.	
РОЗРОБКА ПРОГРАМНОГО ІНСТРУМЕНТАРІЮ.....	27
3.1. Структурна модель та декомпозиція програмного засобу в IDEF Modeling Techniques.....	27
3.2. Визначення рамок, процесів і потоків даних.....	31
3.3. Розробка об'єктно-орієнтованої моделі системи прогнозування курсу криптовалют.....	33
3.4. Реалізація програмного засобу.....	43
РОЗДІЛ 4.	
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ...	50
4.1. Структурно-функціональний аналіз виробництва.....	50
4.2. Розрахунок освітлення приміщення комп'ютерного кабінету....	51
4.3. Безпека в надзвичайних ситуаціях.....	54
ЗАГАЛЬНІ ВИСНОВКИ.....	55
СПИСОК ЛІТЕРАТУРИ.....	57
Додатки.....	59

## ВСТУП

Сучасні віртуальні тенденції, які беруть початок ще із хайп-індустрії призвели до того, що виникли віртуальні гроші – віртуальна валюта. Відповідно до цього, є люди які готові заплатити за неї банківські гроші та торгувати віртуальною валютою (криптовалютою) на криптобіржі [4, 13, 14].

Водночас окремі експерти наголошують, що криптовалюта це «фінансовий міхур», який через якийсь час просто лопне. Однак, маючи певний досвід, специфічні знання та володіючи відповідною інформацією, із крипто валютою можна працювати так само ефективно, як і справжньою банківською валютою на фондовому ринку.

На ринку послуг з надання доступу до криптовалютних фондів існують два основних підходи: 1) криптовалютний обмінник; 2) криптовалютна біржа.

– **криптообмінник** виступає стороною в торгах, тобто бере безпосередню участь в торгових операціях: або продає, або купує.

– **криптобіржа** – це майданчик, де користувачі торгують між собою, тобто вона виступає регулятором-посередником між покупцем і продавцем.

І якщо в криптообміннику ціни контролюються його власником, то на криптобіржі цінова ситуація залежить від балансу попиту і пропозиції. Це можна віднести це до переваг обмінників, адже вони не підходять для біржових схем, але біржовий ринок все-таки більш демократичний, оскільки на ціни впливає велика кількість учасників, а не одна людина.

**Мета роботи** — поєднати методики прогнозу курсу криптовалюти в інтелектуальній системі, реалізувати її як програмний засіб та перевірити працездатність.

### **Завдання кваліфікаційної роботи:**

- проаналізувати технології та принципи роботи криптовалюти;
- розкрити інструменти прогнозування курсу та особливості роботи криптобіржі;
- розробити програмний інструментарій для прогнозу курсу криптовалюти.

## РОЗДІЛ 1.

# АНАЛІЗ ТЕХНОЛОГІЙ ТА ПРИНЦИПІВ РОБОТИ КРИПТОВАЛЮТИ

### 1.1. Головні поняття і принципи створення криптовалюти

*Криптовалюта, або віртуальні валюти* – це цифрові засоби обміну, що були створені та використовуються приватними особами або групами. Оскільки більшість криптовалют не регулюються національними урядами, вони вважаються альтернативними валютами – засобами фінансового обміну, які існують поза межами державної монетарної політики. Біткоїн є найвідомішою і першою, що широко використовується. Однак існують сотні криптовалют, які з'являються щомісяця і не пов'язані з *біткоїнами*, відомі як «альткоїни», щоб відрізнити їх від оригіналу [18].



Криптовалюти використовують криптографічні протоколи або надзвичайно складні кодові системи, що шифрують передачу конфіденційних даних для захисту своїх одиниць обміну. Розробники криптовалют будують ці протоколи на передових принципах математики та обчислювальної техніки, що робить їх практично неможливими для злому, а отже для копіювання або підробки захищених валют. Ці протоколи також гарантують анонімність користувачів криптовалют, ускладнюючи присвоєння транзакцій та потоків коштів конкретним особам чи групам.

Зокрема, діяльність «майнерів» – користувачів криптовалют, які використовують величезну кількість обчислювальних потужностей для реєстрації транзакцій, отримання новостворених одиниць та комісійних за транзакції, що сплачуються іншими користувачами є критично важливою для стабільності та безпечної роботи валют.

Блокчейн криптовалюти (англ. *blockchain* або *block chain* – ланцюг блоків) – це головна база даних, яка реєструє та зберігає всі попередні транзакції та дії, підтверджуючи право власності на всі одиниці валюти в будь-який момент часу. Так як це записи усієї історії транзакцій криптовалют на сьогоднішній день, блокчейн має кінцеву довжину, що містить кінцеву кількість транзакцій, яка з часом збільшується [12].

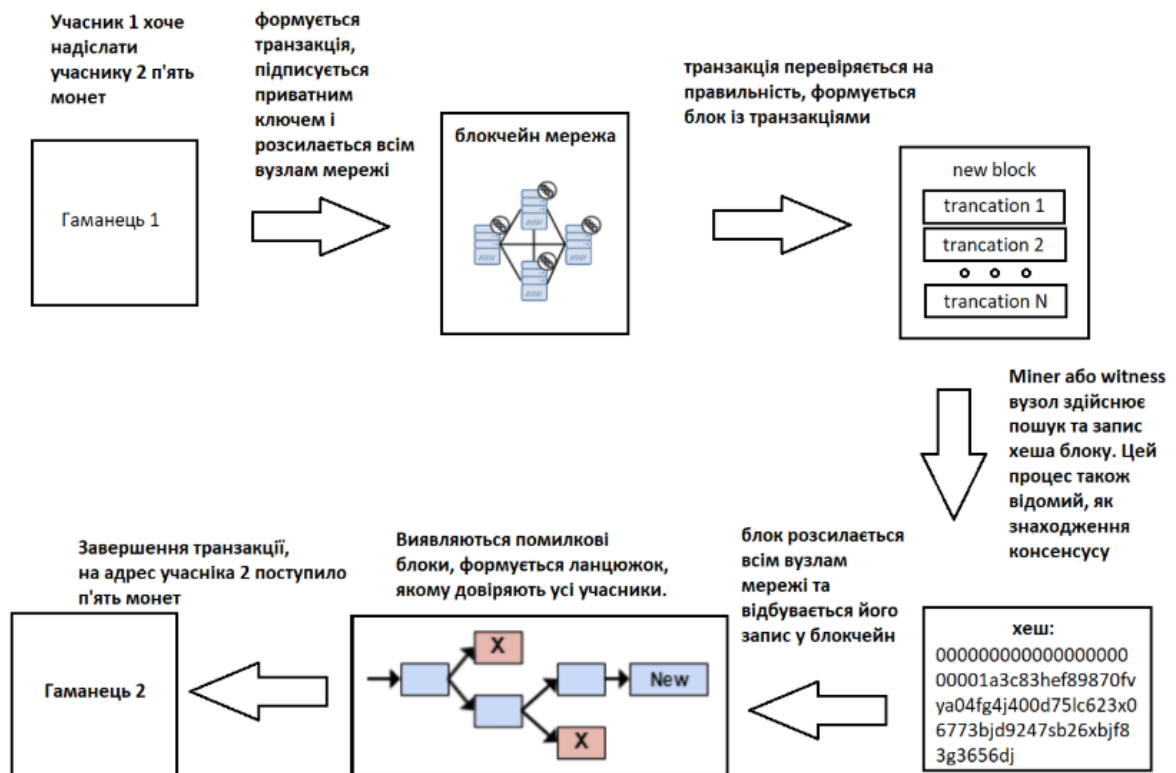


Рисунок 1.1. – Життєвий цикл транзакції в блокчейн

Ідентичні копії блокчейну зберігаються в кожному вузлі програмної мережі криптовалют – мережі децентралізованих серверів, якими керують досвідчені в комп'ютерній справі особи або групи осіб, відомі як «майнери», які постійно реєструють та аутентифікують операції з криптовалютою. Транзакція технічно не завершується доки вона не буде додана до блокчейну, що відбувається за лічені хвилини.

Кожний власник віртуальної валюти має приватний ключ, який засвідчує його особу та дозволяє обмінювати одиниці, а також користувачі можуть створити власні приватні ключі, відформатовані як цілі числа довжиною від 1 до 78 цифр або використати генератор випадкових чисел для його створення.



Отримавши ключ, вони можуть користуватися своїм придбанням, але ж без ключа власник не може маніпулювати криптовалютою, поки він не буде відновлений.

Також користувачі мають «гаманці» з унікальною інформацією, яка підтверджує їх як тимчасових власників своїх підрозділів. Хоча приватні ключі підтверджують справжність транзакцій з криптовалютою, гаманці зменшують ризик крадіжки одиниць які не використовуються.

## 1.2. Огляд популярних бірж криптовалюти

Для того, що визначити кращу біржу криптовалюти слід звертати увагу на такі важливі моменти [4, 10, 18, 21]:

- *рейтинг надійності і репутація*. Чим кращі відгуки про біржу на популярних криптофорумах, таких як *bitcointalk* або *Reddit*, тим більш стабільною є довіра до них.

- *розмір комісії*. Всі біржі криптовалюта, як і будь-які інші обмінники, беруть за свої послуги певний відсоток комісії. Він відрізняється залежно від біржі, в середньому 0,2%.

- *кількість торгових пар*. Чим більший вибір валютних пар пропонується на біржі, тим більше інструментів для заробітку є. Це можуть бути як найпопулярніші високоліквідні криптовалюти, так і менш ліквідні альткоїни. Однак, потрібно пам'ятати, що кількість – не завжди запорука успіху.

- *платіжні опції*. Широкий набір платіжних методів дає більше можливостей для операцій з цифровими активами. Так, одні біржі працюють виключно з криптовалютою і не підтримують фіатні гроші (гарантовані державним наказом як засіб платежу), а інші дозволяють використовувати будь-які грошові активи;

- *вимоги до верифікації користувачів*. На різних біржах вимоги до нових

користувачів можуть відрізнятися: одні ставлять жорсткі рамки і сильно обмежують можливості торгів, інші більш лояльні до «не верифікованих» користувачів, але встановлюють ліміт для добових торгових оборотів.

– *зручний інтерфейс користувача.* Чим більше на біржі інструментів для аналізу і статистики і чим комфортніше ними користуватися, тим оперативніше й ефективніше буде проходити процес торгів.

– *місце локалізації торгової біржі.* Деякі майданчики, що працюють на території однієї держави, можуть бути закриті для користувачів з інших країн. Найчастіше це японські і південнокорейські біржі, які є дуже привабливими для криптовалютних операцій, але доступними лише для резидентів цих країн.

– *оперативність служби підтримки.* Чим швидше і продуктивніше вона працює, тим більше операцій з криптовалюта ви можете здійснити, а також отримати професійну допомогу в разі виникнення проблеми.

Розглянемо декілька найпопулярніших крипто бірж [4, 10, 18, 21].

**Poloniex** – криптовалютна біржа, що підтримує більше 140 торгових пар. Щодня її відвідують в середньому 5387 відвідувачів. Біржа пропонує різні таймфрейми і наочні детальні графіки, підтримуючи рівні Фібоначчі. Головна перевага *Poloniex* – максимально зручна навігація. На одному екрані представлені: чат, новинарив, маркети.

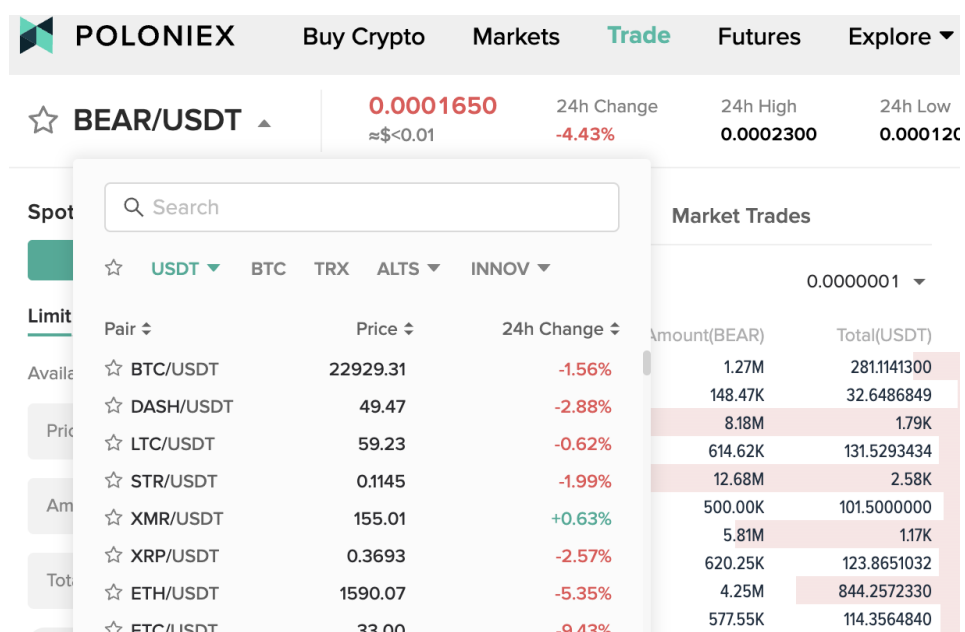


Рисунок 1.2. – Робоча дошка криптобіржі *Poloniex*

**Exmo** – це універсальний сервіс, який активно використовує понад 240 користувачів. Біля керма біржі стоять професіонали з усього світу, включаючи Росію. **Exmo** – одна з популярних криптовалютних бірж, що займається торгівлею і обміном – *Bitcoin, Ethereum, Dogecoin, Litecoin* і *Dash*. Причому, купуючи і обмінюючи криптовалюта, можна використовувати долари, євро, рублі і гривні.

Професійним трейдерам надаються додаткові можливості: 1) *stoploss* – обмеження збитків; 2) *trailingstop* – дозволяє деактивувати поле з ціною продажу і вводити в нього інші значення; 3) видача кредитів; 4) спеціальний ордер покупки / продажу.

**LiveCoin** – Біржа підтримує одразу кілька пар: *BTC / EUR, BTC / USD, BTC / RUR, EMC / USD, EMC / BTC, LTC / BTC, LTC / EUR, LTC / USD* і ін.

Основні переваги:

- на ресурсі використовується тільки затребувана криптовалюта;
- виведення коштів на електронний гаманець можна необмежену кількість разів протягом короткого терміну;
- фіатні гроші на рахунок користувача надходять миттєво;
- елементарний пункт обміну, що дозволяє купувати / продавати валюту, а також здійснювати ряд інших грошових операцій;
- сервіс взаємодіє з популярними платіжними системами;
- дія спеціального бонусної програми з гнучкою ставкою;
- майданчик працює на англійській і російській мовах.

**Bitfinex** – маржинальна торгівля. Інвесторами або постачальниками ліквідності є самі користувачі ресурсу. Вони можуть надавати криптовалюту трейдерам в позику. При цьому комісія за використання коштів являє собою елементарну статтю доходу [4, 10].

**CEX.IO** – багатофункціональна платформа, що підтримує лімітні та ринкові види ордерів. Майданчик функціонує як з фіатними, так і з криптовалютними засобами. Введення/висновок криптовалюти здійснюється

автоматично з номінальною комісією. Також на майданчику передбачено банківський переказ, *Skrill*, *SEPA*, *AstroPay*.

**Cryptonit** – ця біржа майданчиком для торгів і покупки криптовалюти за допомогою євро і доларів. Торгові пари, підтримувані сервісом:

- *USD / BTC – USD / NMC – USD / LTC – USD / PPC;*
- *EUR / BTC – EUR / NMC – EUR / LTC – EUR / PPC;*
- *BTC / LTC – BTC / PPC – BTC / NMC – BTC / TRC – BTC / FTC.*

– Висновок / введення грошових коштів доступний за допомогою різних популярних систем.

**BTC-Trade** - біржа функціонує з наступними крипта валютами: *LTC*, *BTC*, *PPC*, *DOGE*, *DRK*, *CLR*, *NVC*, *RMS*. Зароблені кошти користувачі можуть виводити на пластик Віза і Мастеркарт. При цьому комісійний збір складає:

- 1,3% в разі переведення на карту банку України;
- 1,95 \$ + 1% – якщо гроші виводяться на карти зарубіжних банків.

**Bittrex** – світовий лідер за обсягами обміну криптовалюта.

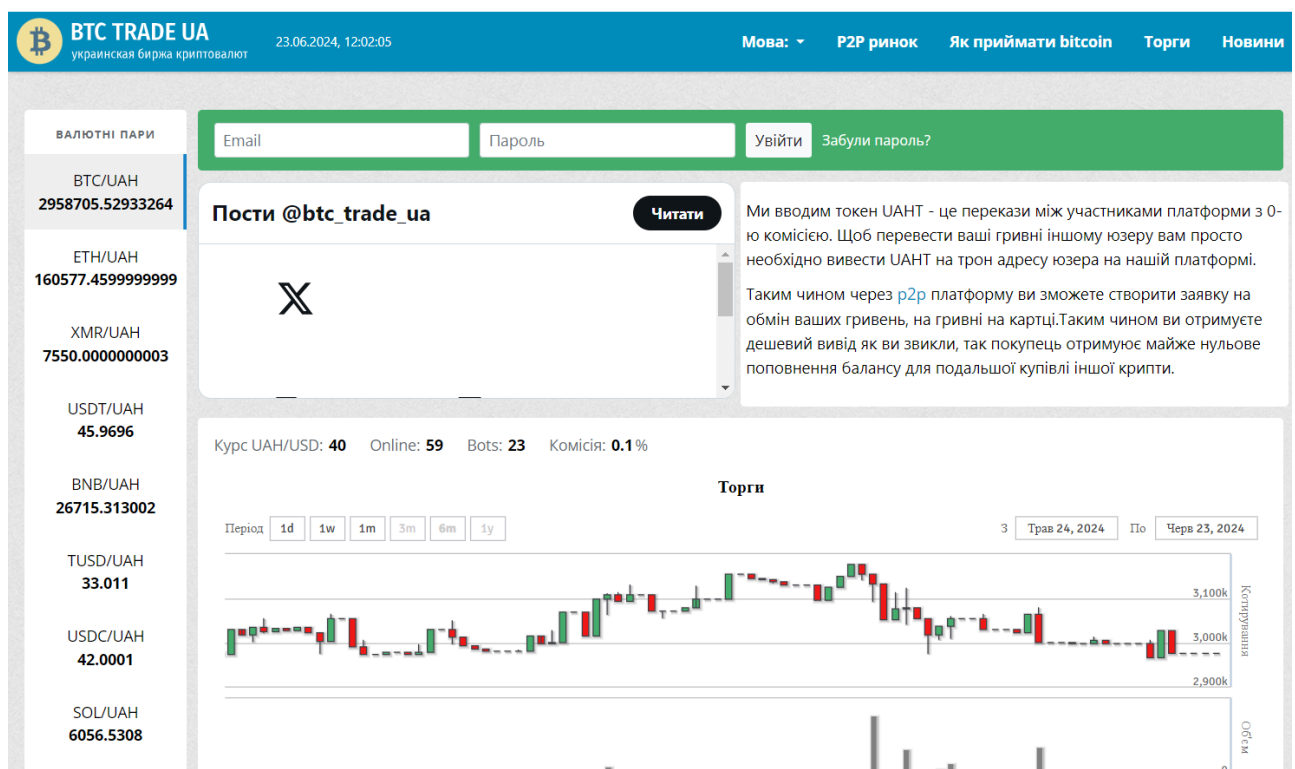


Рисунок 1.2. – Робоча дошка українського відділення криптобіржі **BTC-Trade** [4, 10]

Незважаючи на те, що на майданчику передбачено значну кількість обмінних пар, вивести в реальні гроші можна лише біткоіни. Середня комісія конвертації коштів становить 0,5%.

### **1.3. Принцип виконання торгів із валютою з боку користувача**

*Біткоін* – криптовалюта яка створена у 2008 році невідомою групою людей під псевдонімом «*Сатоші Накамото*» і введена в обіг у 2009, коли її імплементація була випущена як ПЗ з відкритим програмним кодом [10, 18].

Ця технологія – це децентралізована електронна валюта, яка не має центрального банку чи головної управляючої структури. Біткоін може бути надісланий через мережу біткоіна яка має топологію рівний-рівному без будь-яких проміжних точок. Транзакції верифікуються кінцевими користувачами мережі, так званими «нодами», за допомогою криптографії та записуються до публічного сховища даних, яке несе назву «блокчейн». Біткоіни створюються як виноград за «майнінг». Вони можуть бути обміняні на інші валюти, продукти чи послуги. Блокчейн є публічним сховщем даних яке зберігає записи всіх транзакцій біткоіна, який імплементовано як ланцюг із блоків. Кожен блок зберігає хеш попереднього блоку, аж до так званого первинного блоку. Мережа біткоіну підтримується за допомогою нодів, на яких запущено ПО біткоіну.

**Транзакції** у системі біткоін визначаються, використовуючи мову сценаріїв четвертого покоління. Транзакції складаються з одного чи декількох вхідних даних та одного чи декількох вихідних. Коли користувач надсилає біткоіни, він позначає кожну адресу для надсилання криптовалюти та її кількість у вихідних даних. Щоб уникнути “подвійного витрачання”, кожна одиниця вхідних даних повинна посилатися на попередні невитрачені вихідні данні у блокчейну.

У такому випадку, використовується допоміжний вихід який повертає кошти відправнику. Будь-які вхідні дані, що представляють собою “сатоші”, які

не прописані у вихідних даних транзакції, стають комісією за транзакцію. Загальний приклад транзакції продемонстрований на рис. 1.1

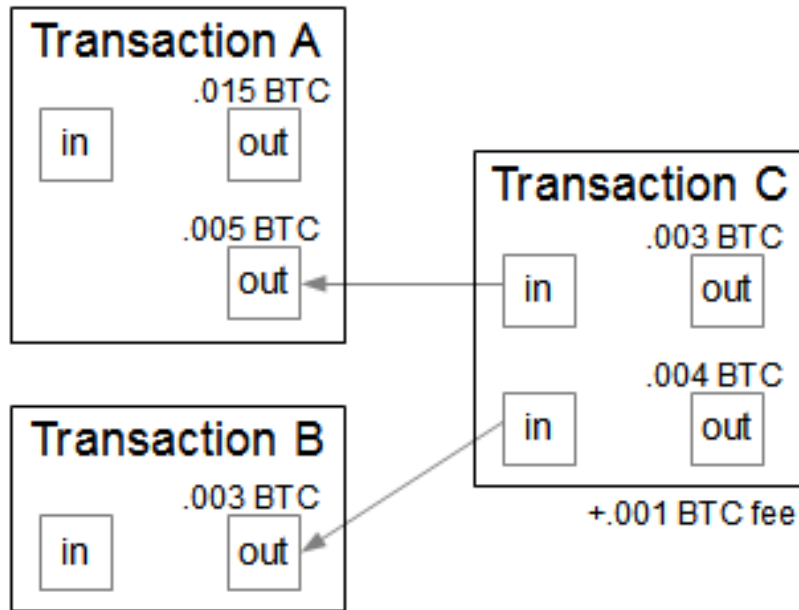


Рисунок 1.1 – Приклад транзакції з біткоїном [11]

**Комісії за транзакцію.** Хоча комісія за транзакції є обов’язковою майнери можуть вирішувати які транзакції вони будуть обробляти і обирають ті, дохід від яких є максимальним. Також майнери можуть зважати на відношення комісії за транзакцію та на розмір сховища необхідний для її зберігання. Комісія за транзакцію вимірюється у одиниці виміру “sat/b”, що означає кількість сатоши на один байт транзакції. Розмір транзакції залежить від кількості вхідних даних, що використовуються для її створення, та кількості вихідних даних.

**Право власності.** В системі блокчейн, біткоїни прив’язані до певних адрес. Створення біткоїн адреси процес вибору випадкового приватного ключа та визначення адреси. Ця процедура може бути виконана за долю секунди, але зворотня дія (отримання приватного ключа з біткоїн адреси) вважається практично нездійсненим. Завдяки цій властивості користувачі можуть повідомляти іншим свою публічну адресу, не хвилюючись, що їхній приватний ключ може бути вкрадено.

**Видобуток або майнінг** – це процес зберігання інформації за допомогою обчислювальної потужності. Майнери або добувачі допомагають зберігати послідовності, завершеності та незмінюваності системи біткоїн шляхом групування нещодавно виконаних транзакцій у блок, який транслюється всією мережею та підтверджується кінчними вузлами. Кожен блок містить хеш SHA-256 попереднього блоку, таким чином утворюючи ланцюг із блоків. Щоб бути прийнятим всією мережею біткоїн має містити “proof-of-work” (PoW). Система PoW вимагає від майнерів знаходження цифри яка носить назву “nonce”, за допомогою якої можна було б хешувати блок контенту, в наслідок чого результат є меншим ніж складність мережної цілі. Такий результат є легким у верифікації, але дуже складний у генеруванні [17].

Кожні 2016 блоків (приблизно 14 днів, 10 хвилин на кожен блок), складність цілі змінюється базуючись на нещодавній статистиці мережі. Метою цього процесу є підтримка часу створення блоків у проміжку 10 хвилин. У даному випадку система автоматично адаптується до загальної кількості обчислювальної сили у мережі. Також система “proof-of-work”, зв’язуючи блоки один з одним, робить модифікацію блокчейну дуже складною. Як щоб атакуючий хотів підтвердити модифікацію одного блока, йому б довелося змінити всі попередні блоки. З появою нових блоків у блокчейні, складність модифікації зростає за часом та кількістю попередніх блоків.

#### **1.4. Аналітичні інструменти крипто валютних бірж**

**Аналітика** – це основний принцип торгівлі. Не важливо, що ви маєте намір продати – картоплю на ринку або криптовалюта на біржі. В останньому варіанті дотримуватися даного алгоритму просто, оскільки курси монет відрізняються своєю динамічністю і постійно «скачуть» в різних напрямках. Тому придбати криптоденежки в умовах спаду курсу – справа не складна. Безперечно, що кожен користувач хоче вкластися по мінімуму, а заробити по

максимуму. Якщо в разі продажу стабільних товарів – золота, нафти, доларів – вся необхідна інформація, представлена в спеціальних новинах, то з крипті все зовсім не так. Поки новини дійдуть до адресата, ринок 10 разів встигне змінити свою траєкторію, і користувач може просто втратити профіт. Досвідчений трейдер повинен аналізувати ринок самостійно, і кожна платформа надає для цієї мети свої інструменти [4, 21]:

– **Графіки.** Покажуть, як змінювалася вартість монет протягом певного часового проміжку.

– **Стакан торгів.** Дозволить зрозуміти, в чому відмінність між діючими пропозиціями на купівлю / продаж, а також дозволить скласти скрупульозний аналіз, в який бік кинеться ціна в майбутньому.

– **Історія торгів.** Знаючи обсяги торгівлі певної пари, можна легко встановити, якою популярністю користується даний напрямок на конкретній біржі. Найбільш затребувана пара буде стрімко набирати курс.

Аналізувати всі ці дані найкраще комплексно. Це значно підвищить ймовірність того, що ситуація буде правильно витлумачена, а значить користувач зможе своєчасно увійти в ринок і вийти з нього до зміни тенденції.



## РОЗДІЛ 2.

# ІНСТРУМЕНТИ ПРОГНОЗУВАННЯ КУРСУ ТА РОБОТА КРИПТОБІРЖІ

### 2.1. Програми та веб-системи прогнозування курсу криптовалют

*Trader* – система дозволяє прогнозувати рух курсів валют на валютній біржі [4, 10]. Як вихідна використовується інформація за попередніми результатами торгів (часовий ряд): максимальна, мінімальна ціна, ціна закриття і обсяг угод за день. В системі застосовуються наступні алгоритми для аналізу даних:

- ковзне середнє трьох видів – лінійне, експоненціальне, за рангами що задаються;
- MACD-гістограми;
- індикатори – RSI, OBV, Williams R%, CandleSticks, Point & Figure.

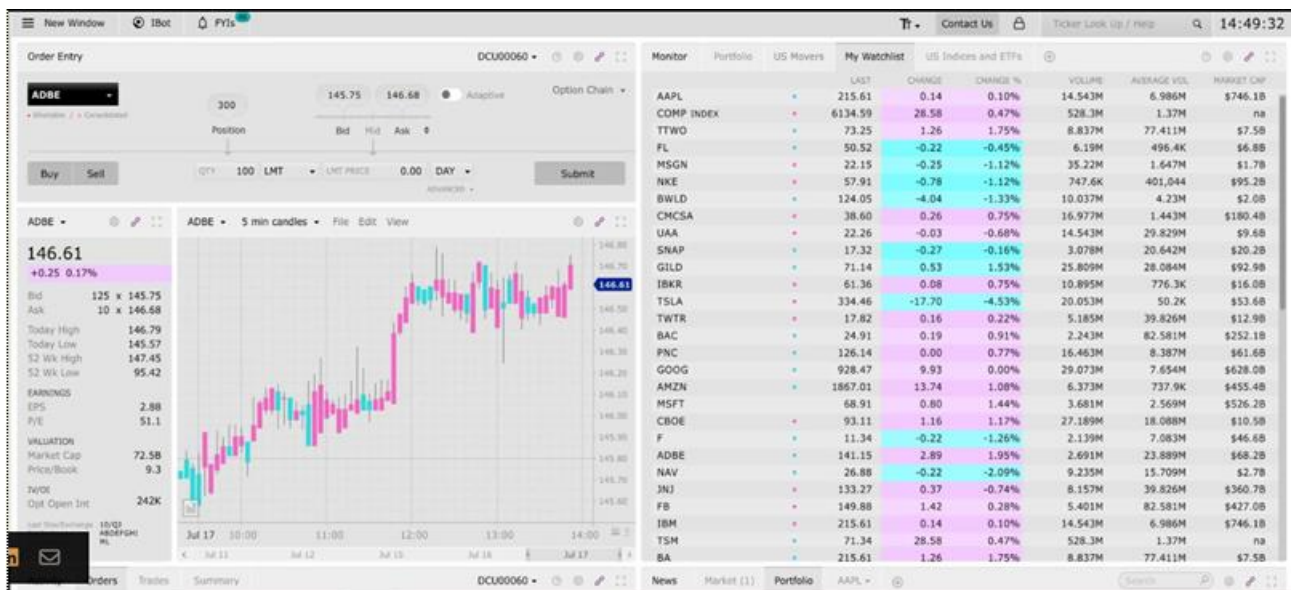


Рисунок 2.1. Торгова система глобальної платформи *Trader*

Користувач може створювати власні формули для аналізу даних. До переваг також можна віднести можливість застосування індикатора до вже побудованого індикатору, що наприклад потрібно при побудові MACD-

гістограми, де ковзне середнє обчислюється для різниці двох ковзних середніх.

*Walletinvestor* – має вигляд веб-додатку з можливістю вибору однієї конкретної криптовалюти, або ж спостерігати дані про всі наявні з них у вигляді таблиці. Має прогнози на періоди – два тижні, три місяці, пів року, один рік, п'ять років. Програма пропонує користувачу окрім прогнозів поточний курс найпопулярніших криптовалют і усю попередню інформацію (рис. 1.2).

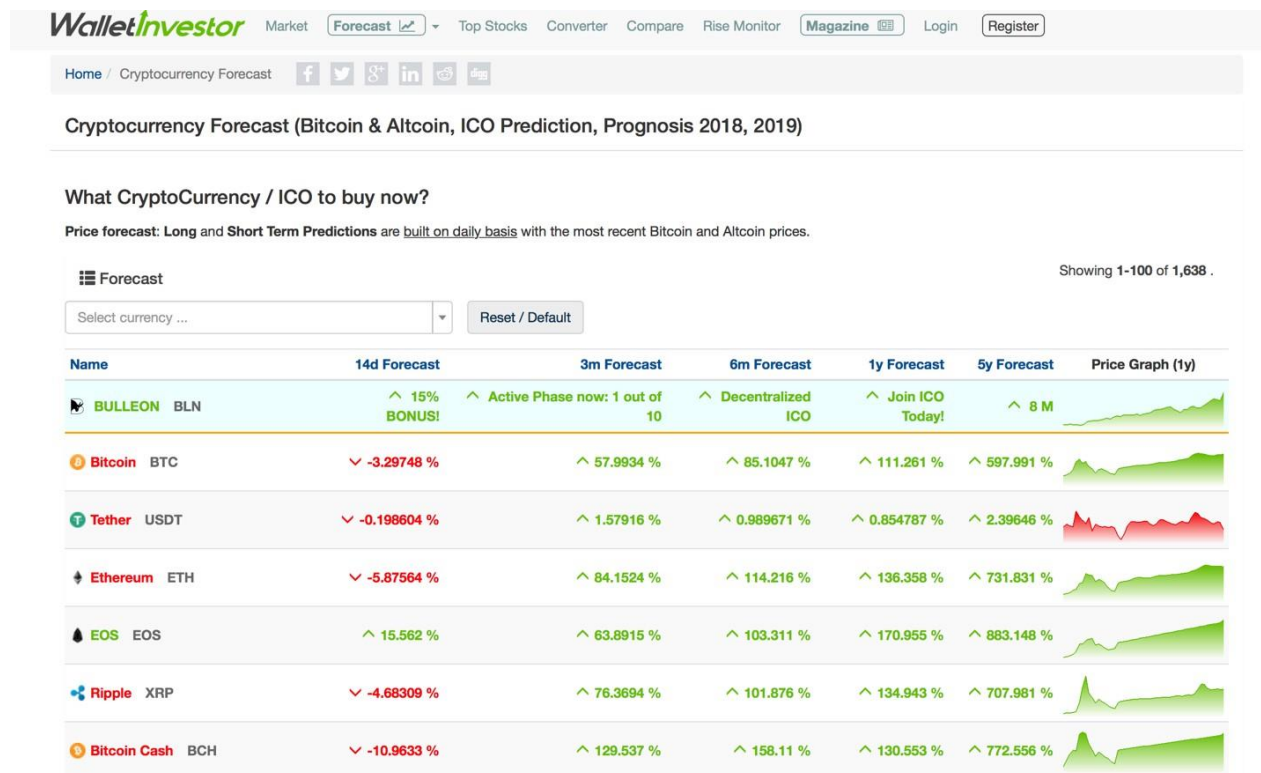


Рисунок 2.2. Робота з веб-додатком для прогнозів Walletinvestor [15]

*Binance* – провідна в світі біржа криптовалют, що обслуговує 150 мільйонів зареєстрованих користувачів у більш ніж 180 країнах. Завдяки низьким комісіям і понад 350 представленим криптовалютам *Binance* вважається найкращою біржею для торгівлі Bitcoin, альткоїнами й іншими віртуальними активами.

З *Binance* користувачі можуть: 1) торгувати сотнями криптовалют на спотовому, маржинальному і ф'ючерсному ринках; 2) купувати та продавати криптовалюту за допомогою *Binance P2P*; 3) заробляти відсотки на криптоактивах завдяки *Binance Earn*; 4) купувати або заробляти нові токени у

Binance Launchpad; 5) торгувати, розміщувати в стейкінгу та позичати NFT на маркетплейсі Binance NFT.

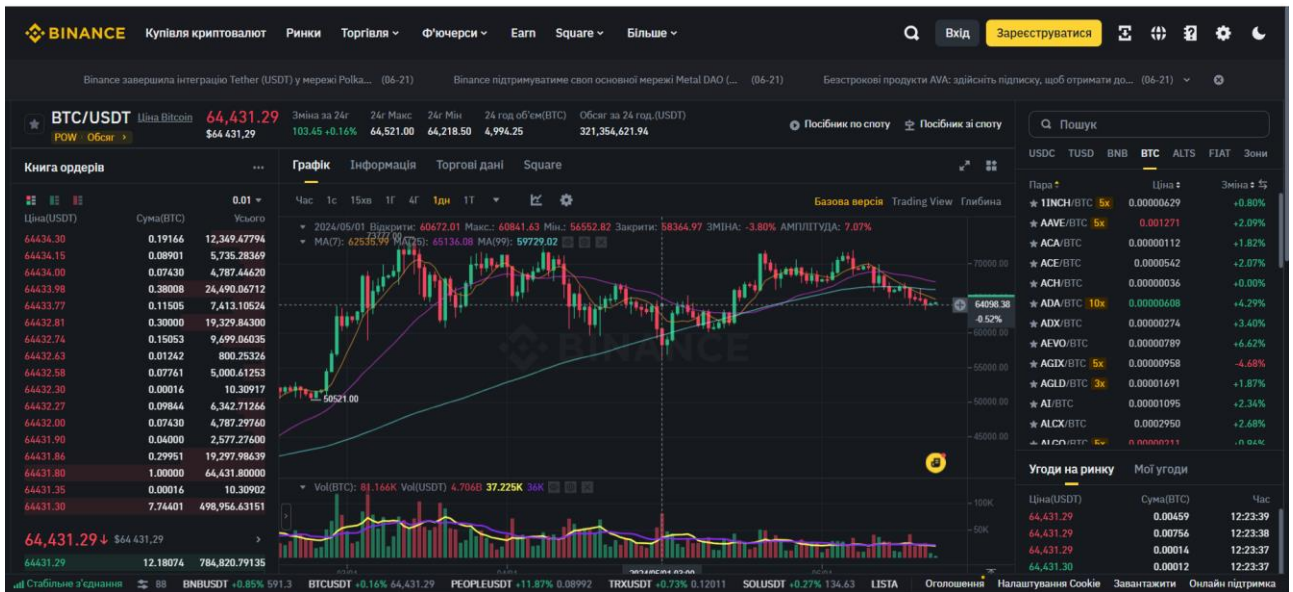


Рисунок 2.3. Аналітично-прогнозна таблиця торгів в криптобіржі *Binance*

**Bitcoin. NeuroShell** – являється цілим набором нейронних мереж, які створені та навчені з єдиною метою – передбачення курсу котирувань валют на фінансових ринках. Оскільки сам принцип застосування полягає в процесі схованим у «чорному ящику» – користувач не повинен алгоритму обробки, чи певних його частин. А за допомогою мінімалістичного інтерфейсу все приходить лише до задання параметрів та отриманню результату. Тобто з цією процедурою може впоратись людина з будь-яким рівнем досвіду у фінансових операціях. *NeuroShell Day Trader* має ще одну сильну сторону – він вмщує в своїх методах обробки оптимізації, засновані на принципах генетичних алгоритмів. *NeuroShell Day Trader* концентрується на побудові торгової системи. Сама ж торгова система може використовувати в своїх правилах як індикатори, так і спрогнозовані значення, отримані за допомогою нейронних мереж [18].

*Elliott wave analyser professional 6.0* – програмний продукт призначений для аналізу валютного ринку з використанням принципів теорії хвиль Елліотта (ТХЕ), а також за допомогою стандартних алгоритмів технічного аналізу. У

1930 році Ральф Елліотт виявив, що емоційний стан натовпу впливає на курси валют, і цей вплив описується декількома зразками, які тепер відомі як хвилі Елліотта.

*Ainet* – для прогнозування подій використовуються нейронні мережі. Для аналізування часових рядів дана програма мало придатна, але дає хороші результати для багатьох завдань з інтерполювання даних. Параметр аналізу – *penalty coefficient*, який програма сама ж і оптимізує. В якості вихідних даних виступає прямокутна матриця з повністю присутніми даними і матриця з тим же кількістю стовпців, але в якій має місце відсутність деяких даних. Програма намагається передбачити значення цих відсутніх даних.

## **2.2. Методи прогнозу динаміки курсу криптовалюти**

Наразі кожний трейдер має вбудовані найпоширеніші індикатори та осциляторів, але відомо, що їх застосовність в реальній торгівлі невисока. Найпоширеніші методики [9, 13, 18]:

*Методика аналізу і прогнозування валютного ринку* включає систему генерації сигналів торгової системи (купити, продати, утримувати, закрити поточну позицію). Для генерації та ідентифікації сигналів використовується аналіз перетину декількох експоненційних ковзних середніх з різними періодами усереднення. Критерієм відкриття позицій на купівлю / продаж є емпірично знайдені порогові значення величини середньої кількості всіх поданих сигналів. Критерієм закриття позиції – рівень прибутку / збитку, що задається експертом.

*Моделі прогнозування прибутковості валют* (і портфеля інвестора) на *VR FoRex* – побудовані на основі виділення періодичних компонент ряду прибутковості методами сингулярного спектрального аналізу. Аналізується ряд прибутковості у вигляді логарифмічних різниць валютного курсу. Для

визначення періодичних компонент і їх характеристик застосовується метод передбачення головних компонент (PCLP) Д. Тафтса і Р. Кумаресана.

*Методика побудови ефективних трендових торгових стратегій з використанням індикатора і осциляторів RSI, Stochastic* – моделі являють собою рівняння регресії, в яких керуюча змінна (КЗ) представляється лінійною функцією деяких характеристик осциляторів. Значення КЗ інтерпретуються як критерії достовірності сигналів на вхід в ринок (вихід з ринку). У торговельній стратегії для визначення поточної (довгострокової) тенденції використовується технологія «потрійного вибору» і індикатор RSI, тобто ідентифікація поточного тренду здійснюється поза моделлю.

*Методи фрактального аналізу і апарат теорії нечітких множин* – використовуються у дослідженні часового ряду валютного курсу пари євро-долар і ряду його збільшень з метою виявлення довготривалої пам'яті, її глибини, трендостійкості, циклів. Для визначення рівня ризику використовується фрактальна характеристика тимчасового ряду – показник Херста. Результати фазового аналізу часових рядів підтверджують наявність довгострокової пам'яті, яка пояснюється присутністю циклів.

*Методика факторно-регресійного аналізу* – відслідковує вплив агрегованих факторів на номінальні і «інваріантно-індексні» курси валют, за допомогою чого розроблені нові економетричні моделі регресії мінової цінності валюти на виділену групу агрегованих факторів. Також запропоноване використання у дослідженнях економіко-математичних моделей впливу макроекономічних факторів на валютні курси, в основу яких покладено поняття індексу мінової цінності валюти [11].

*Методика з використанням APIЗС* – моделі авто-регресії інтегрування змінного середнього широко використовуються при аналізі часових рядів. Переваги: 1) чітке математико-статистичне обґрунтування; 2) формалізована і найбільш докладно розроблена методика, за якою можна підібрати модель, найбільш придатну для до кожного конкретного тимчасового ряду; 3) ефективні прогнози на малі часові проміжки. Недоліки: 1) для побудови

потрібно не менше 40 спостережень, а для SARIMA – близько 6-10 сезонів; 2) неадаптивність – при отриманні нових даних, модель потрібно періодично переоцінювати, а іноді переідентифіковувати; 3) побудова задовільної моделі вимагає великих витрат ресурсів і часу.

*Методика експоненціального згладжування* – метод, який обчислює значення згладженого ряду шляхом оновлення значень, визначених на попередньому кроці на основі інформації з поточного кроку [11]. Інформація з попереднього і поточного кроків береться з різними вагами, якими можна керувати. Простий, ефективний метод, але недоцільний при короткотривалому прогнозуванні і зовсім не ефективний на довготривалому.

*Марковські випадкові процеси та Ланцюги Маркова* – частковий випадок випадкових процесів, що широко застосовуються для створення стохастичних (імовірних) моделей, котрі описують процеси функціонування певних систем технічного, економічного, екологічного та соціального профілю, центральне місце належить Марковським.

Переваги та недоліки прогнозування за допомогою ШІ описані у табл. 2.1.

Таблиця 2.1 Переваги та недоліки прогнозування з використанням ШНМ

Переваги	Недоліки
Ефективність прогнозів на малі, середні та помірно великі часові проміжки	Складність аналізу на пошук проблеми при помилках прогнозування
Адаптивність до надходження нових даних	Робота за принципом «чорного ящика», що не дає доступу до проміжних результатів складання прогнозу
Можливість передбачення динамічних процесів	Доволі складний процес емпіричного «навчання» та перевірки результатів
Має можливість навчатись за допомогою декількох навчальних шарів, якими можуть слугувати іншу методи ТА	Можливі виліти в самозбудження (коли сигнал, отриманий з виходу, забиває все, що приходить по входах)

*Методика використання штучних нейронних мереж* – застосовує нейронну мережу, яка є системою простих процесорів (нейронів), що сполучені і взаємодіють між собою. Для прогнозування є два типи нейромереж: перцептрон та мережі з архітектурою Радіально Базисної Функції (РБФ). Гарні результати на всіх строках прогнозування, через відстеження та копіювання шаблонів поведінки людей. Адже саме люди та їх активність впливає на тренди та торги вцілому.

Нейромережі кращі в цьому саме через виявлення шаблонів «людських дій» через числові значення, це дає можливість давати гарні прогнози на будь-яких строках [11, 16].

Аналізуючи всі перераховані праці можна побачити, що в основному запропоновано прогнозування майбутнього значення валютного курсу методами: 1) множинної регресії; 2) сигнального методу з математичним апаратом на основі спектрального і фрактального аналізів; 3) формалізації керуючої змінної на основі сигналів відомих індикаторів та осциляторів; 4) товарним та монетарним підходом.

### **2.3. Принцип роботи API криптовалютних бірж**

API – *Application Programming Interface*, що означає програмний інтерфейс програми. API – це механізми, які дозволяють двом програмним компонентам взаємодіяти один з одним, використовуючи набір визначень та протоколів [4].

У контексті API слово «додаток» стосується будь-якого ПЗ з певною функцією. Інтерфейс можна розглядати як сервісний контракт між двома програмами. Цей контракт визначає, як вони взаємодіють один з одним, використовуючи запити та відповіді. Документація API містить інформацію про те, як розробники повинні структурувати ці запити та відповіді.

Архітектура API зазвичай пояснюється з погляду клієнта та сервера. Програма, що надсилає запит, називається клієнтом, а програма, що надсилає

відповідь, називається сервером. Існує чотири різні способи роботи API залежно від того, коли та чому вони були створені.

*SOAP – Simple Object Access Protocol*, тобто простий протокол доступу до об'єктів. Клієнт та сервер обмінюються повідомленнями за допомогою XML. Це менш гнучкий API, який був більш популярним у минулому.

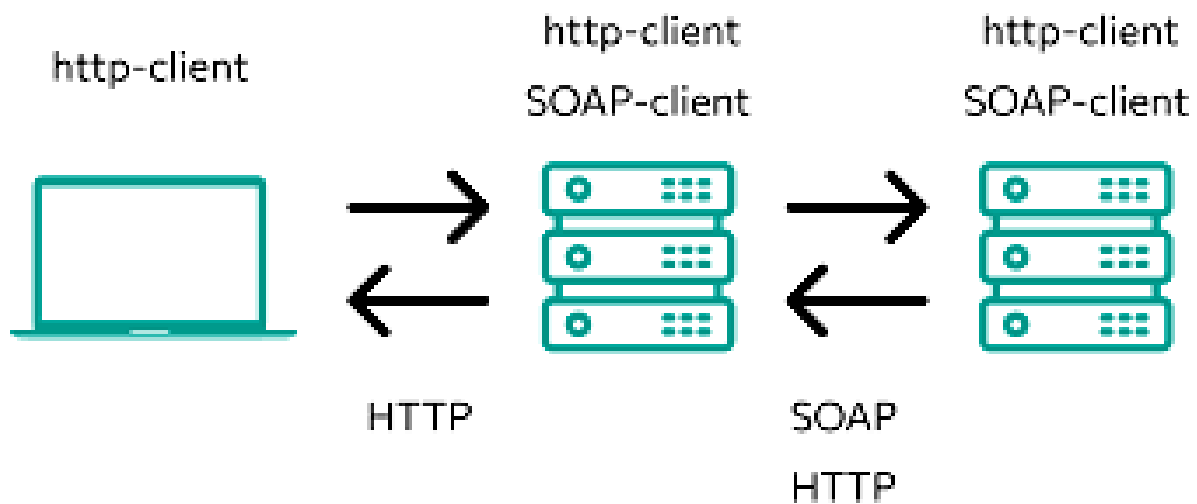


Рисунок 2.4. – Схема роботи SOAP

*RPC API* – такі API називаються системою віддаленого виклику процедур. Клієнт виконує функцію (або процедуру) на сервері, і сервер надсилає результат назад клієнту.

*Websocket API* – це ще одна сучасна розробка web API, яка використовує об'єкти JSON для передачі даних. WebSocket API підтримує двосторонній зв'язок між клієнтськими програмами та сервером. Сервер може надсилати повідомлення зворотного дзвінка підключеним клієнтам, що робить його ефективнішим, ніж REST API.

*REST API* – на сьогоднішній день це найпопулярніші та гнучкіші API-інтерфейси в Інтернеті. Клієнт надсилає запити на сервер у вигляді даних. Сервер використовує це введення клієнта для запуску внутрішніх функцій і повертає вихідні дані назад клієнту. REST – *Representational State Transfer*, тобто передача репрезентативного стану. REST визначає набір функцій, таких як GET,



PUT, DELETE тощо, які клієнти можуть використовувати для доступу до даних сервера. Клієнти та сервери обмінюються даними за протоколом HTTP.

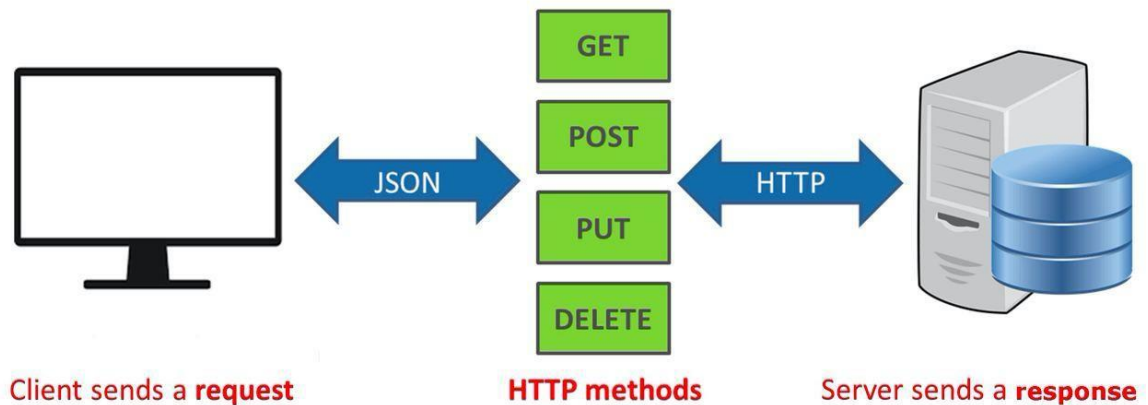


Рисунок 2.5. – Схема роботи REST API

Головною особливістю REST API є те, що така передача виконується без збереження стану. Без збереження стану означає, що сервери не зберігають дані клієнта між запитами. Клієнтські запити до сервера подібні до URL-адрес, які ви вводите в браузері для відвідування веб-сайту. Відповідь від сервера є простими даними без типового графічного відображення веб-сторінки [14].

Усі API повинні бути захищені за допомогою належної автентифікації та моніторингу. Опишемо два основні способи захисту REST API.

*Токени автентифікації* використовуються для авторизації користувачів для виклику API. Токени автентифікації перевіряють, чи є користувачі тими, за кого вони себе видають, і що вони мають права доступу для цього конкретного виклику API.

*Ключі API* перевіряють програму або програму, яка виконує виклик API. Вони ідентифікують програму та гарантують, що вона має права доступу, необхідні для виконання конкретного виклику API. Ключі API не такі безпечні, як токени, але вони дозволяють здійснювати моніторинг API для збору даних про використання.

Уніфікований інтерфейс є фундаментальним для дизайну будь-якої веб-служби RESTful. Це означає, що сервер передає інформацію в стандартному

форматі. Відформатований ресурс називається представленням у REST. Цей формат може відрізнятися від внутрішнього представлення ресурсу в серверній програмі. Наприклад, сервер може зберігати дані як текст, але надсилати їх у форматі представлення HTML.

Уніфікований інтерфейс накладає *чотири архітектурні обмеження*:

- Запити повинні ідентифікувати ресурси. Вони роблять це за допомогою єдиного ідентифікатора ресурсу;
- Клієнти мають достатньо інформації в представленні ресурсу, щоб змінити або видалити ресурс, якщо вони хочуть. Сервер відповідає цій умові, надсилаючи метадані, які додатково описують ресурс;
- Клієнти отримують інформацію про подальше оформлення представництва. Сервер досягає цього шляхом надсилання самоописових повідомлень, які містять метадані про те, як клієнт може їх найкраще використовувати;
- Клієнти отримують інформацію про всі інші пов'язані ресурси, необхідні для виконання завдання. Сервер досягає цього шляхом надсилання гіперпосилань у представленні, щоб клієнти могли динамічно відкривати більше ресурсів.

## РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ІНСТРУМЕНТАРІЮ

### 3.1. Структурна модель та декомпозиція програмного засобу в IDEF Modeling Techniques

Для забезпечення вимог щодо створення програмного засобу (ПЗ) для аналізу і прогнозу курсу криптовалюти описано основні механізми роботи програмного засобу (рис. 3.1) [7, 19].



Рисунок 3.1. – Концептуальна модель ПЗ

Концептуальна модель включає такі дані:

*Початкові дані:* «Обрана пара криптовалют» і «Обраний діапазон прогнозу» на основі яких створюються HTTP-запит до API біржі, та обираються параметри при обробці нейромережею архітектури рекурентного перцептронну та математичними алгоритмами на основі методу експоненціального згладжування, індикатора PCC і осциляторів R з подальшим оновленням даних.

Найчастіше ціна валюти визначається її співвідношенням до долара (USD) або біткоїну (BTC), але для більшої гнучкості програмного засобу

вирішено самостійно вирішувати до співвідношення яких криптовалют визначати курс.

Отже у криптовалютної біржі запитуватимуться дані за двома типами валют. Глобальні дані історії кожної криптовалюти вже є у базах біржі, але під час реальної роботи програми слід враховувати те, що обсяг та час запиту даних через API є обмеженим:

- якщо необхідний короткостроковий прогноз – запитуються дані найближчого часу з найменшими часовими інтервалами (1-30 хв);
- якщо необхідний довгостроковий прогноз – запитуються дані за великий попередній період, згруповані по значним часовим проміжкам (1-7 днів).

*Регулюючі чинники:* біржові обмеження по можливій частоті та кількості запитів. А також в залежності від обраної біржі може бути доступним різний діапазон попередніх даних, та їх різний набір, тобто різні біржі по-різному визначають якими даними ділитися, а якими ні, судячи з їх власних міркувань. Можливі біржі для роботи: Binance, KUNA, YoBit [4, 21].

*Механізми роботи* поділені на дві основні частини:

- *API біржі* – надає дані запитам щодо стану «пари криптовалют» у заданому часовому діапазоні, значення курсу, кількості угод, їх мінімуму та максимуму, загальну суму продажів, часу дня та активності, тощо.
- *Нейронна мережа* (або ШІ)– це сукупність нейромереж котра є самостійним модулем, оскільки перед реальним використанням кожен із них необхідно навчити на спеціальних вибірках даних [20], після чого вони підключаються до програми як засіб обробки біржових даних для видачі прогнозу, а також обробки всіх прогнозів для зіставлення і надання найвірогіднішого.

*Початкові дані:* прогноз курсу криптовалютної пари, який надається у вигляді часового ряду на заданий вхідними даними інтервал, а графічним інтерфейсом програми відображається у вигляді таблиці чи графіку.

**Декомпозиція концептуальної моделі програмного засобу за методикою IDEF0.** Головні процеси, які виконуються під час роботи програмного засобу (рис. 3.2).

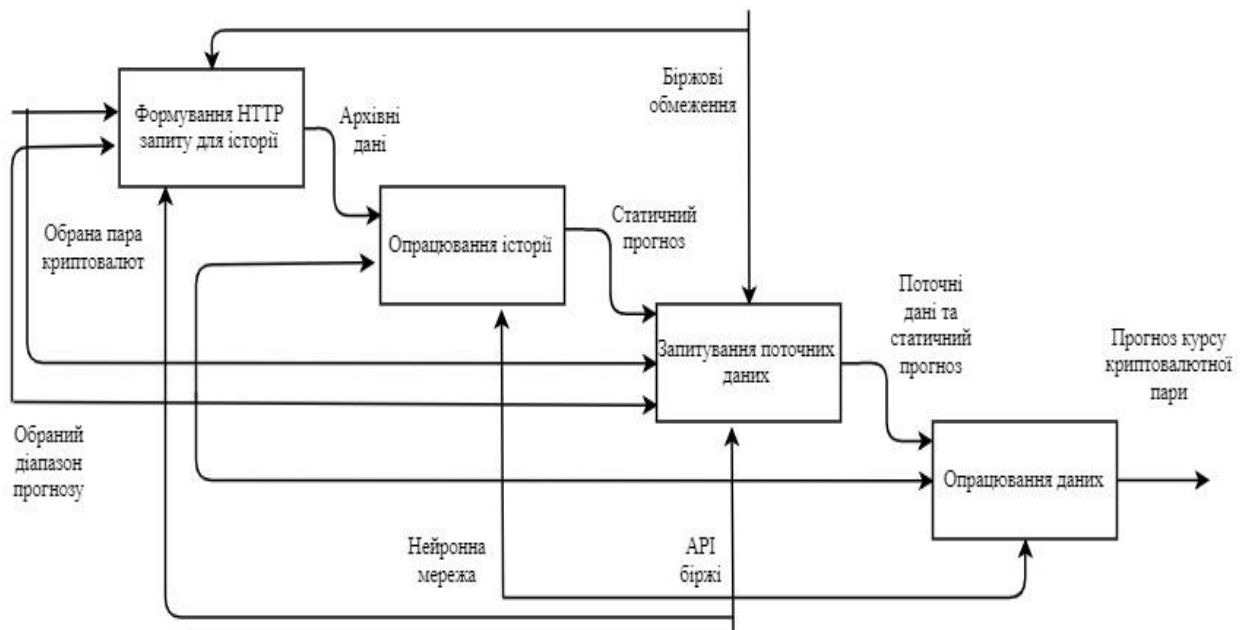


Рисунок 3.2. Декомпозиція концептуальної моделі ПЗ в IDEF0

*Запит історії* – у цьому процесі створюється HTTP-запит на основі наданих даних про пару криптовалют та часовий діапазон прогнозу, після чого відправляється до API біржі.

*Обробка історії* – у цьому процесі отримана JSON-відповідь від API біржі парситься (з JSON об'єкту дістаються необхідні для роботи програми дані [10]), з отриманих даних створюються програмні об'єкти для нейронної мережі та математичних алгоритмів описаних вище, вони відправляються на опрацювання після чого всі отримані прогнози надходять на вхід нейромережі для зіставлення прогнозів, яка надасть остаточний результат статичного прогнозу.

*Запит поточних даних* – процес, який є початком основного циклу програми для підтримки видачі прогнозу в режимі реального часу – тобто режим програми, в якому дані автоматично оновлюються з певним інтервалом та виведенням інформації користувачу.

*Обробка даних* – процес, в якому в залежності від обраного часового діапазону прогнозу до поточних об'єктів даних буде додані нові, після чого заново пройде обробка нейронною мережею та математичними алгоритмами і видано оновлений прогноз, або через несуттєвість часового проміжку отриманих даних в порівнянні з обраним часовим діапазоном прогнозу – дані будуть відсічені: тобто не прийняті до роботи, через їх фактичну ідентичність попереднім. Прогноз в даному випадку залишиться статичним.

**Декомпозиція концептуальної моделі програмного засобу за методикою IDEF3.** Основні етапи роботи моделі: *Формування запиту, Отримання даних, Обробка даних, Видача результату*. Вони відбуваються з даними, тому доцільно розглянути декомпозицію за методикою IDEF3 (рис. 3.3).

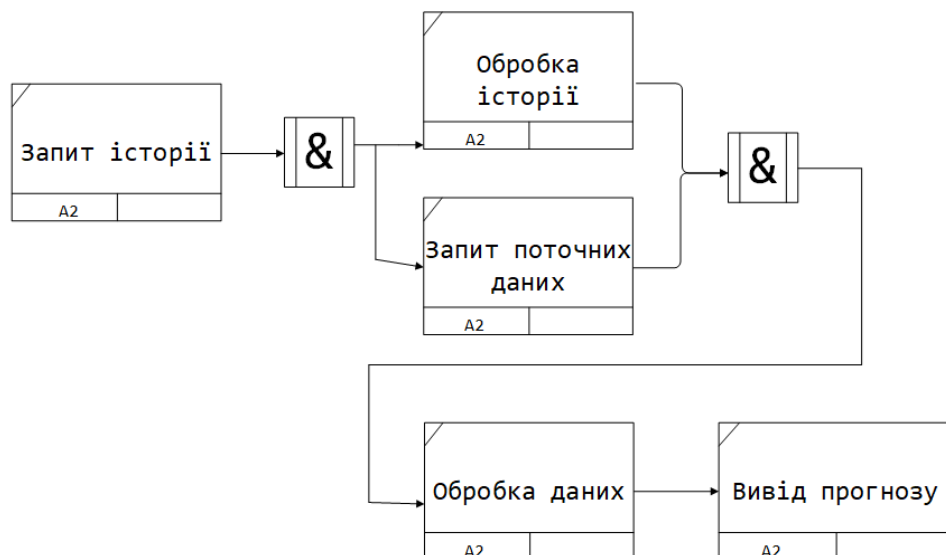


Рисунок 3.3. Декомпозиція концептуальної моделі ПЗ в IDEF3

Процес «*Обробка історії*» починається з конвертації отриманих від серверу даних у форматі JSON-об'єкту до формату колекції програмних об'єктів, яка потім паралельно передається до модуля з математичними алгоритмами та модуля нейронної мережі. В цих модулях дані незалежно оброблюються та додаються в загальну колекцію прогнозів у вигляді часових рядів. Дана колекція подається в спеціальну нейромережу для обробки

прогнозів, яка зіставить їх, та для кожного часового проміжку обере найвірогідніше значення курсу [7, 19].

Процес «Обробка даних», виконує зіставлення часового діапазону отриманих даних від серверу з обраним в програмі часовим інтервалом. Якщо обраний суттєво великий інтервал, наприклад прогноз на місяць вперед: тоді зроблений статичний прогноз на етапі обробки історії буде зроблений на основі даних якнайменше місячної давності. А отже нові дані в інтервалі хвилин, або годин несуттєві для такого прогнозу.

### 3.2. Визначення рамок, процесів і потоків даних

Чітке визначення рамок, процесів і потоків даних в системі можна побачити на рис. 3.4 у декомпозиції концептуальної моделі в нотації DFD.

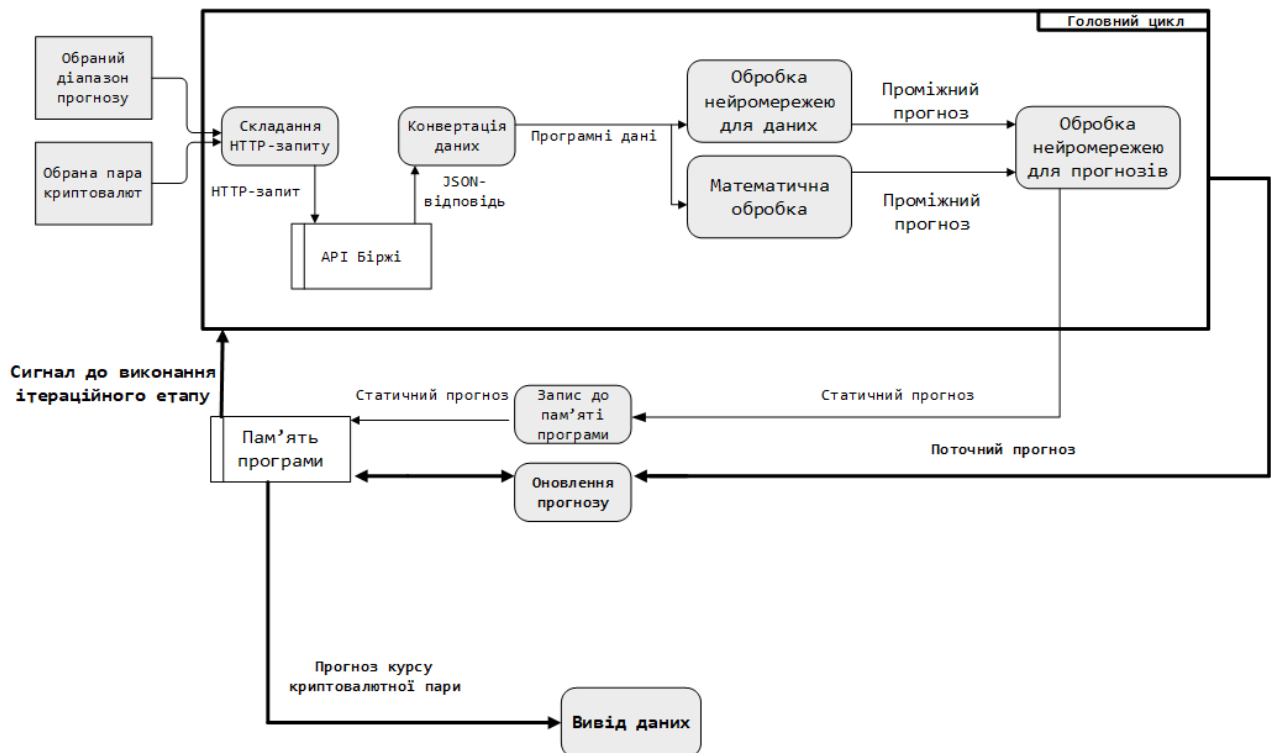


Рисунок 3.4. Декомпозиція концептуальної моделі ПЗ в DFD

Центральним процесом є «Прогнозування курсу криптовалют». На вхід поступають обраний діапазон прогнозу та обрана пара криптовалют. Виходом

з цього процесу буде часовий ряд з прогнозом курсу криптовалюти, який буде виведений користувачу зручним йому способом: графічне відображення таблицею та діаграмою, текстовий файл, або мінімалістичне повідомлення з контрольними прогнозами.

Сховищами даних виступають :

- *API біржі* – основні джерела даних, які надають доступ по запиті до статистики поточних та минулих угод, їх часу, курсу криптовалют.
- *Пам'ять програми* – зберігає лише одну порцію конвертованих даних та поточних прогноз, отриманий від їх обробки, після нового запиту – конвертовані дані оновлюються, після чого оновлюється прогноз.

*Пам'ять програми* є проміжним сховищем даних у рамках роботи з користувачем, а після закінчення роботи – всі дані видаляються. Для зрозумілого розділу між фазою ініціалізації програми з основним заповненням колекції об'єктів, та послідуочим оновленням колекції за умовою збігу часового діапазону – було зручно виділити її окремо.

Також на діаграмі зображено дочірні процеси, які відображають потоки даних між ними, а саме:

- *«Складання HTTP-запиту»* – відповідно до заданих користувачем параметрів складається HTTP-рядок, який посилається до серверу біржі через її API.
- *«Конвертація даних»* – отримані дані в текстовому чи бінарному форматі переводяться в об'єкти у динамічній пам'яті програми для подальшої роботи з ними.
- *«Обробка нейронною мережею для даних»* – дані у вигляді об'єктів віддаються нейронній мережі на основі радіально-базисних функцій для складання атомарного прогнозу.
- *«Математична обробка»* – ті ж самі дані паралельно віддаються на входи до вищезазначених математичних алгоритмів [19].
- *«Обробка нейромережею для прогнозів»* – оброблений часовий ряд з прогнозом з обох джерел віддається на вхід до нейромережі на основі



рекурентного перцептронну, яка конвертує декілька прогнозів в єдиний прогноз.

- «Оновлення прогнозу» – доповнюючий запис поточного прогнозу до пам'яті програми у вигляді об'єктів.
- «Вивід даних» – вивід даних користувачу у заданому їм форматі.

### 3.3. Розробка об'єктно-орієнтованої моделі системи прогнозування курсу криптовалют

Основні функціональні вимоги, що ставляться до серверної та клієнтської частин системи, наведено в табл. 3.1 та 3.2.

Таблиця 3.1 Функціональні вимоги серверної частини, що ставляться до системи

№ п/п	Вимога
1	Можливість отримання форми з парою криптовалют та часовим проміжком
2	Можливість створення запиту у форматі HTTP для API принаймні однієї криптовалютної біржі
3	Можливість прийому відповіді на запит у форматі JSON
4	Можливість парсингу даних з JSON формату в об'єкти
5	Можливість запису даних у вигляді об'єктів до бази даних у разі великої кількості запитів до конкретної криптовалюти
6	Можливість перенаправлення запиту до бази даних, замість криптовалютної біржі, у разі наявності в базі потрібних даних
7	Можливість передачі даних у вигляді часового ряду об'єктів до класів з методами обробки даних у прогнози
8	Можливість видачі класами-обробниками вихідних даних прогнозів у єдиному вигляді, по екземпляру з кожного обробника
9	Можливість обробки екземплярів прогнозів ШНМ для генерації єдиного прогнозу
10	Можливість запису прогнозу у базу даних, у разі його довгостроковості
11	Можливість перенаправлення запиту прогнозу до бази даних, для його безпосередньої видачі, у разі збігу параметрів криптовалютної пари та часового діапазону з наявним в базі прогнозом

Для визначення коректності вимог в цілому для системи необхідно встановити чи не суперечать вони одна одній та чи не перекриваються між

собою.

Таблиця 3.2 Функціональні вимоги клієнтської частини, що ставляться до системи

№ п/п	Вимога
1	Можливість перегляду списком готових прогнозів найпопулярніших криптовалют у часових проміжках: 5хв, 10 хв, 30 хв, 1 год, 1 день, 1 тиждень, 1 місяць, 3 місяці, 6 місяців.
2	Можливість задання власної пари криптовалют та часового проміжку для отримання персонального прогнозу
3	Можливість задання бажаного вигляду прогнозу: таблицею чи графіком
4	Можливість вивантаження таблиці прогнозу у Excel

Для демонстрації структури та можливостей моделі використано універсальну мову моделювання (Unified Modeling Language, UML). Так, можливості відображаються діаграмою варіантів використання (рис. 3.5). Тут визначаються загальні межі і контекст прогнозування курсу криптовалюти, формулюються загальні вимоги до функціональної поведінки програмного засобу, що синтезується. Як наслідок створюються передумови для представлення ПЗ логічною і фізичною моделями.

Логічна структура програмного засобу прогнозування курсу криптовалют представлена за допомогою діаграм класів і діяльності – рис. 3.6 та рис. 3.7 відповідно. При цьому виокремлено послідовність дій, умов їх виконання та потік об'єктів, що необхідні для прогнозування курсу криптовалют.

Архітектуру ПЗ описано діаграмою компонентів на рис. 3.8 та рис. 3.9. Відомості про використану платформу, інструменти, технології та обчислювальні ресурси показано діаграмою розгортання на рис. 3.10.

Таким чином, завдяки побудові об'єктно-орієнтованої моделі програмного засобу прогнозування курсу криптовалюти визначено його можливості, логічну та фізичну структури. А відтак, створено передумови для реалізації ПЗ.

**Діаграма варіантів використання.** Для проектування варіантів використання необхідно визначити дійових осіб-акторів. Оскільки програмне

забезпечення прогнозування курсу криптовалют ставить на меті дати можливість користувачеві задавати параметри бажаної криптовалютної пари і часового періоду прогнозу та отримувати результат у одній з двох можливих форм відображення: графіку чи таблиці – у програмному засобі єдиним учасником системи є користувач.

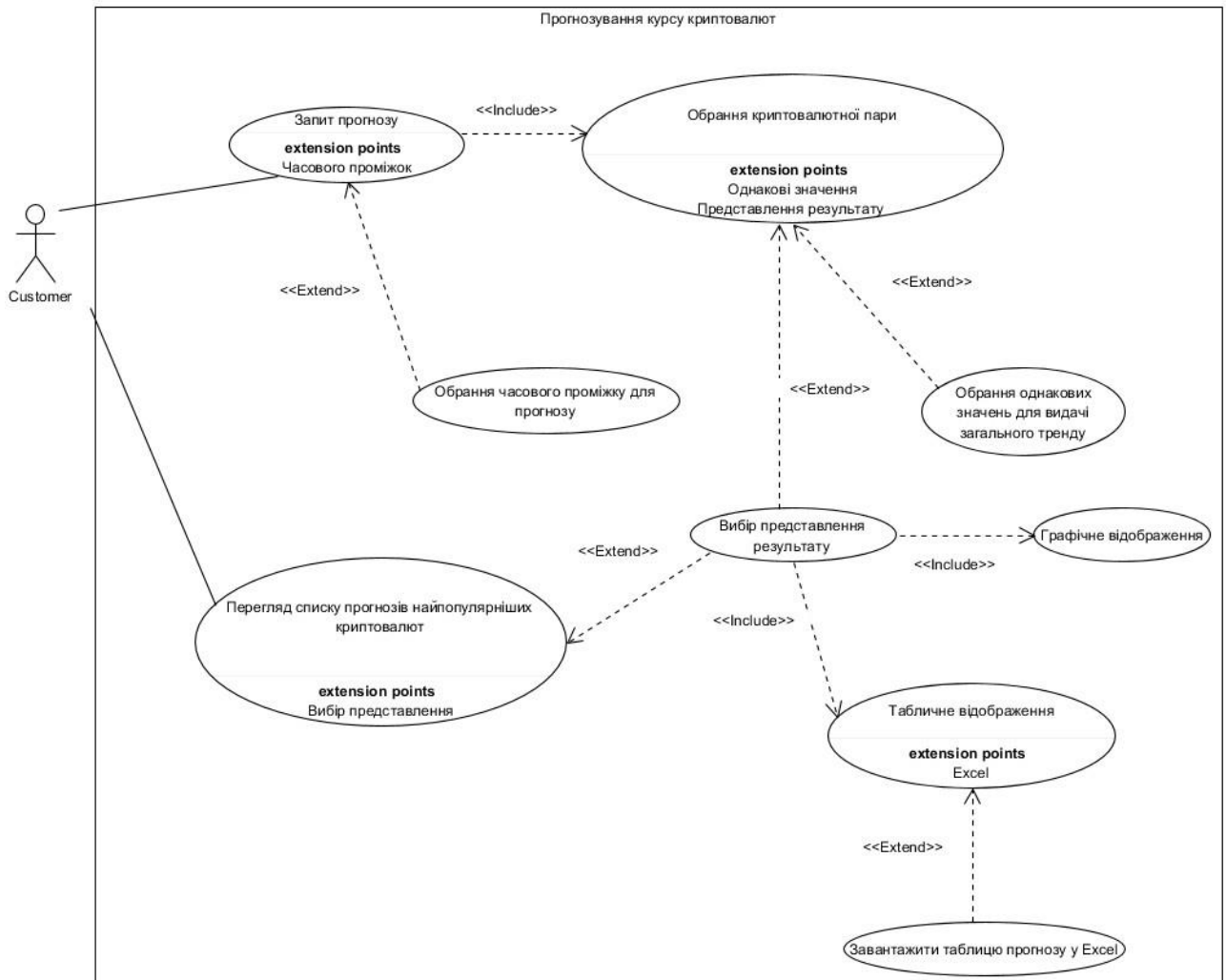


Рисунок 3.5. Структурна схема варіантів використання програмного засобу прогнозування курсу криптовалют

Детальний опис кожного з варіантів використання наведено в табл. 3.3 та додатку А.

Таблиця 3.3 Варіант використання «Перегляд списку прогнозів найпопулярніших криптовалют»

Назва	Перегляд списку прогнозів найпопулярніших криптовалют
Первинний актор	Користувач
Інші актори	Немає
Опис	Можливість одразу побачити прогнози на стандартні часові проміжки на головній сторінці додатку без запитів
Попередні умови	Відкрита головна сторінка додатку
Вихідні умови	Відкрита головна сторінка додатку

**Логічна модель** даних виражена незалежно від технології створення ПЗ, для надання відомостей про основні програмні сутності об'єктно-орієнтованої моделі – це класи. А також точного опису роботи програми.

Для реалізації програмного засобу потрібно розбити дані, які можемо дістати з криптовалютної біржі на самодостатні одиниці, які мають роль у попиті на криптовалюту, та з яких можна скласти відповідний часовий ряд.

Також слід обрати конкретну криптовалютну біржу, оскільки API кожної з них відрізняється. У подальшому при розширенні програми має сенс виокремити методи біржі у окремий інтерфейс, для додавання можливості роботи з іншими біржами – це підвищить працездатність програми, у разі відмови серверу однієї з бірж постачальників даних, а також буде можливо поєднувати отримані дані для заповнення пробілів інформації, які можливі через утаєння або просто відсутність деяких критеріїв на певній біржі.

Подальший механізм який дозволить це реалізувати у програмному засобі ефективно називається *Dependency Injection*, або *Ін'єкція залежностей*. Наразі ж ми повинні розуміти, що для роботи з API біржі необхідний окремий спеціалізований на конкретній біржі клас, який представлений вкупі з іншими на рис. 3.6 зображено структурну схему класів програмного засобу, яка представлена за допомогою відповідної UML-діаграми.

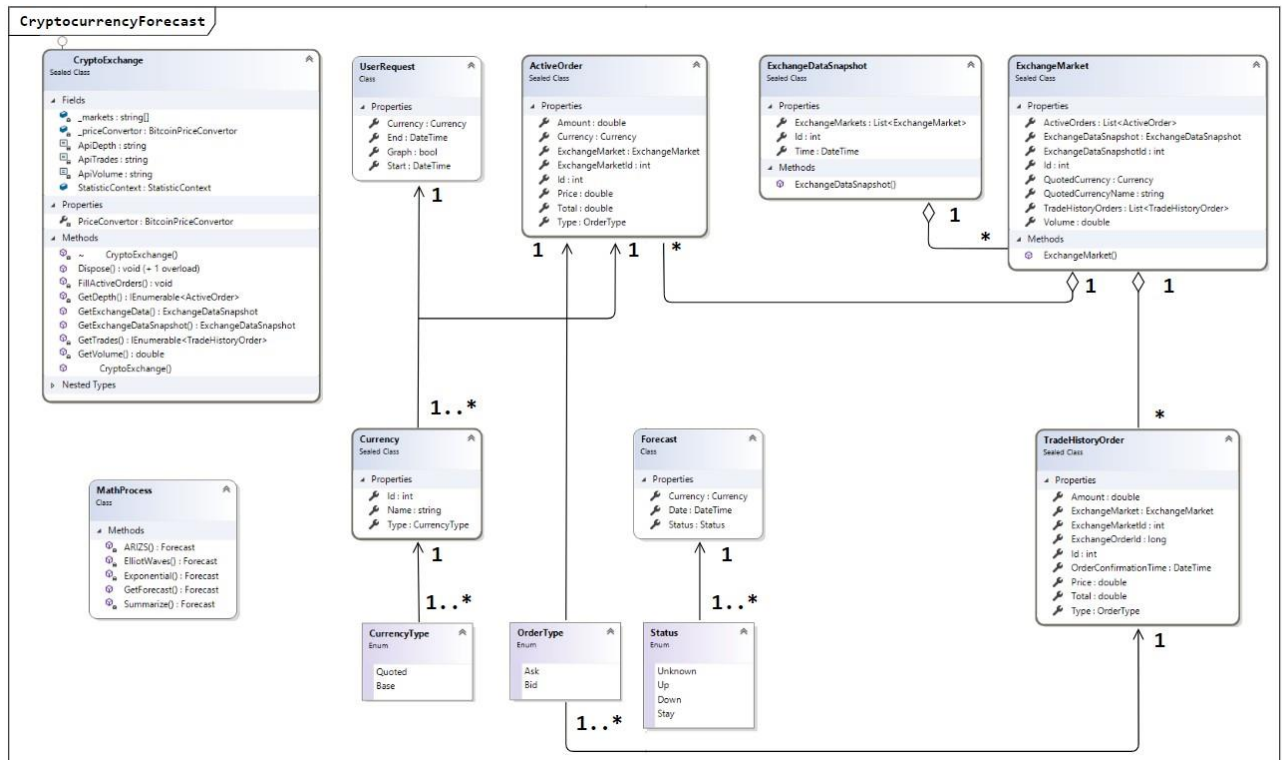


Рисунок 3.6 – Діаграма класів ПЗ

Детальний опис кожного класу – призначення, полів та методів наведено табл. 3.4.

*CryptoExchange* – клас який призначений для роботи з криптовалютною біржею. Містить в собі список шаблонів HTTP-запитів до API біржі, в які підставляються дані залежно від введених користувачем значень, після чого готові запити відправляються до біржі. Через те, що API біржі не має функції «Віддати всі дані» – процедура запиту складається з декількох підзапитів, які виконуються окремими функціями, після чого поєднуються у єдиний клас *ExchangeDataSnapshot*.

*ExchangeDataSnapshot* – клас, глобальна одиниця часового ряду біржових даних. Він має свій ідентифікаційний номер та часову марку. А головне – він містить список екземплярів класу *ExchangeMarket*.

*ExchangeMarket* – клас який містить в собі комбіновані дані зі всієї криптовалютної біржі за певний період часу, а саме: 1) перелік відкритих угод у вигляді екземплярів класу *ActiveOrder*; 2) валюта у вигляді екземпляру класу *Currency*; 3) перелік укладених угод на заданий момент часу у вигляді

екземплярів класу *TradeHistoryOrder*; 4) поточний об'єм торгів зберігається у змінній *Volume*.

*ActiveOrder* – клас який відображає відкриту угоду на біржі, з такими полями як: 1) ціна; 2) кількість; 3) тип угоди; 4) сумарна сума угоди; 5) базова валюта у вигляді екземпляру класу *Currency*.

*Currency* – клас, який просто інкапсулює в собі ім'я та ідентифікаційний номер криптовалюти, вкупі з її поточним типом: 1) *базова валюта* – валюта в конкретній валютній парі, ціна однієї одиниці якої завжди міряється в одиницях іншої (котирується) валюти; 2) *котирувана валюта* – валюта, в одиницях якої виражається ціна однієї одиниці базової валюти.

*TradeHistoryOrder* – клас який відображає закриту угоду на біржі, з полями: 1) ціна; 2) кількість; 3) тип угоди; 4) сумарна сума угоди; 5) час закриття угоди.

*UserRequest* – клас який представляє запит користувача та містить поля: 1) базова валюта у вигляді екземпляру класу *Currency*; 2) котирувана валюта у вигляді екземпляру класу *Currency*; 3) початок заданого часового діапазону у типі *DateTime*; 4) кінець заданого часового діапазону у типі *DateTime*; 5) булеве значення яке визначає у якому вигляді повинен бути виданий прогноз.

*MathProcess* – статичний клас який містить в собі статичні методи для обробки даних математичним алгоритмом чи передачі для обробки в модуль нейронної мережі з відповідними методами, всі вони повертають клас *Forecast*;

*Forecast* – клас який містить в собі дані прогнозу по тренду чи курсу з часовим рядом.

Як можна побачити на діаграмі присутні ще три типи, які є введеними переліченнями, які є у кожній сучасній ООП мові. Вони допомагають об'єднати набір числових значень за логічним відношенням, та надати ім'я кожному з них. Таким чином можна використовувати їх у класах як розширені параметри ознак з зрозумілими іменами.

До таких типів на діаграмі відносяться: 1) *CurrencyType* – перелічення для визначення типу валюти з двома полями (*Quoted* – яке позначає котирувану

валюту; Base – яке позначає базову валюту); 2) *OrderType* – перелічення для визначення типу угоди з двома полями (*Ask* – заява для угоди на продаж; *Bid* – заява для угоди на покупку).

*Status* – перелічення для визначення напрямку тренду у прогнозі з полями: 1) *Unknown* – початкове непроініціалізоване значення на випадок помилкових ситуацій; 2) *Up* – позначає напрямок тренду вгору; 3) *Down* – позначає напрямок тренду вниз; 4) *Stay* – позначає, що тренд залишиться незмінним.

Після опису класів програмного засобу слід створити діаграму діяльності, яка демонструє конкретну логічну послідовність робочого циклу програми. Діаграма наведена на рис. 3.7.

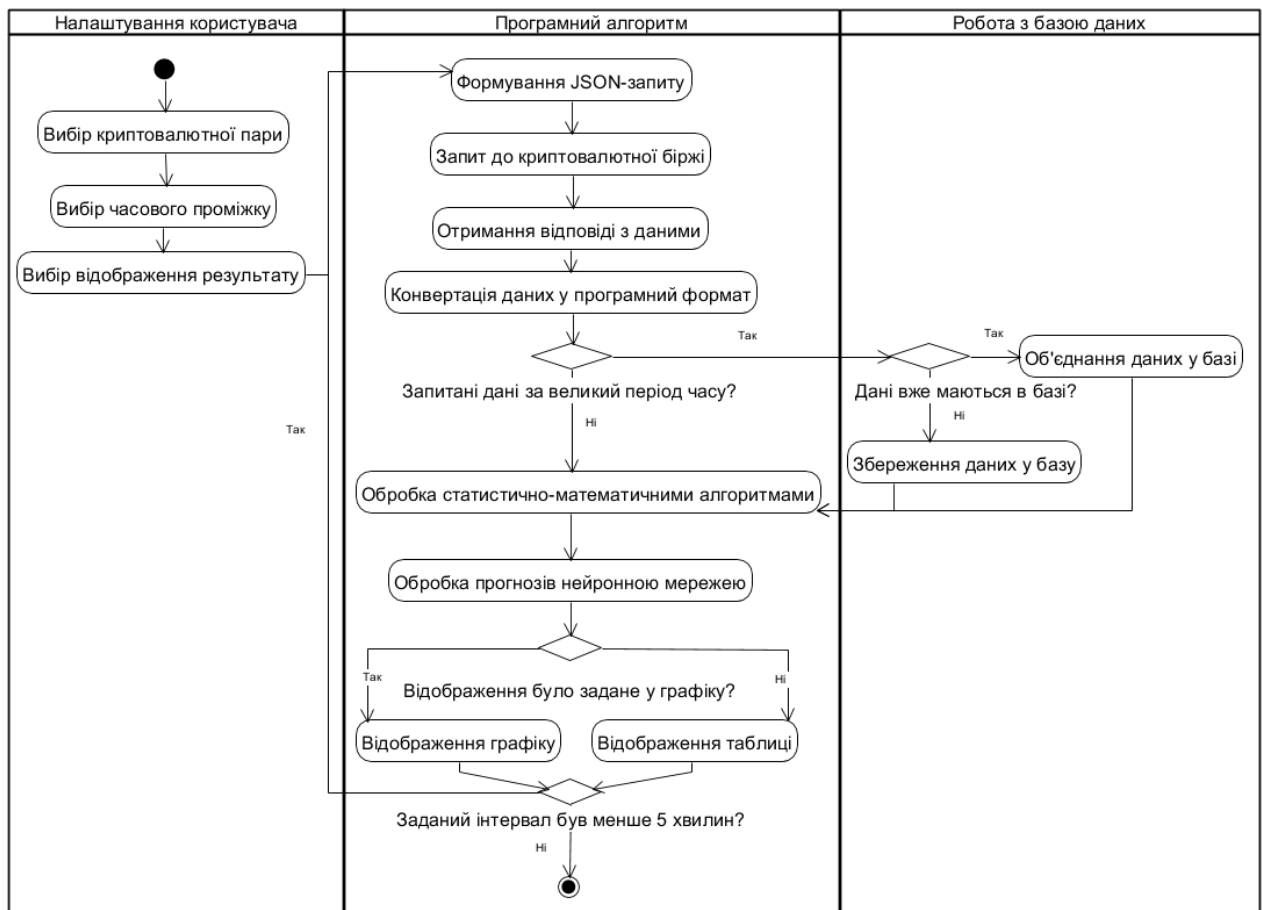


Рисунок 3.7 – Діаграма діяльності ПЗ

У діаграмі слід зазначити можливість програмного циклу, при умові, що користувач запитав прогноз на строк менший за 5 хвилин, в даному випадку

замість відображення статичного результату прогнозу програма може повторювати запит до криптовалютної біржі для оновлення результатів в режимі реального часу.

**Фізична модель.** Для визначення фізичної структури програмного засобу прогнозування курсу криптовалют побудовано діаграми компонентів, які зображені на рис. 3.8.

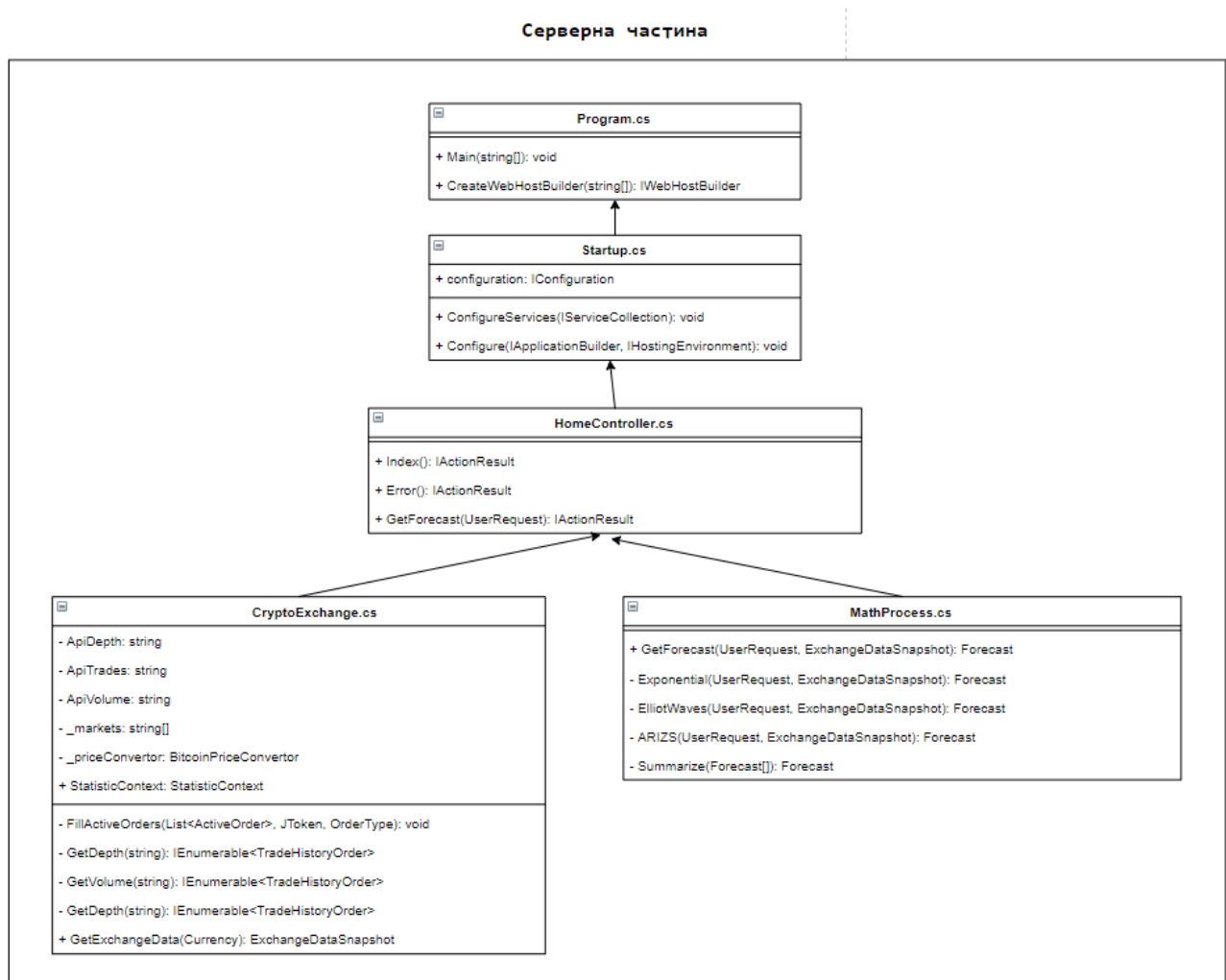


Рисунок 3.8 – Діаграма компонентів серверної частини

На рис. 3.8. продемонстровано структура файлів програмного засобу прогнозування криптовалют, а конкретно серверної частини, які дають можливість опрацюванню всіх даних та роботи програми як було показано на рис. 3.7.

– *Program.cs* – клас програми який є точкою входу для початку виконання програми;



– *Startup.cs* – головний клас програми, який забезпечує роботу системи завдяки початковій конфігурації усіх наявних сервісів, таких як MVC, тощо.

- *HomeController.cs* – клас програми який відповідає за взаємодію з користувачем через головну сторінку додатку. В ньому користувачу надається: 1) початковий список прогнозів; 2) форма для запиту конкретного прогнозу з додатковими полями фільтрами для вибору криптовалютної пари та часового діапазону прогнозу; 3) отриманий від користувача запит надсилається до модуля обробки; 4) отриманий прогноз передається саме в цей контролер для його подальшого відображення як відповіді користувачу.

- *CryptoExchange.cs* – клас, який необхідний для роботи з API криптовалютної біржі, його було детально описано в логічній моделі та на діаграмі класів;

- *MathProcess.cs* – статичний клас, який відповідає за всю обробку даних математично-статистичними методами та ШІ для видачі прогнозів.

Далі для роботи безпосередньо з клієнтом використовуються файли з розміткою, такі як: 1) *.html* чи *.cshtml* – застосовуються для розмітки у веб-додатках за допомогою мови HTML та вставок в неї мови JavaScript чи C#; 2) *.xaml* – застосовуються для розмітки у десктопних додатках розроблюваних на платформі .NET.

В клієнтській частині програмного засобу прогнозування криптовалют передбачені файли продемонстровані на рис. 3.9.

*Index.cshtml* – головна та єдина сторінка додатку, оскільки додаток планується як односторінковий застосунок (SPA).

Це означає, що саме ця сторінка з усіма наявними скриптами завантажується користувачу, а подальші дії виконуються за допомогою асинхронних вставлень часткових представлень до сторінки.

Клієнська частина складається з наступних файлів:

*\_MainMenu.cshtml* – часткове представлення яке вбудовується у головну сторінку *Index.cshtml* та надає користувачу меню, де він може задавати параметри запиту прогнозу та форму його відображення;

`_Layout.cshtml` – містить загальний шаблон сторінки та установки щодо розміщення елементів;

`_Table.cshtml` – часткове представлення яке вбудовується у головну сторінку `Index.cshtml` та відображає прогноз у вигляді таблиці з датою, часом та курсом криптовалюти;

`_Graph.cshtml` – часткове представлення яке вбудовується у головну сторінку `Index.cshtml` та містить скрипт `_Graph.js` для вбудови графіку в нього;

`_Graph.js` – скрипт на мові JavaScript, який використовує інструменти відкритих бібліотек для візуалізації даних прогнозу у вигляді графіку.

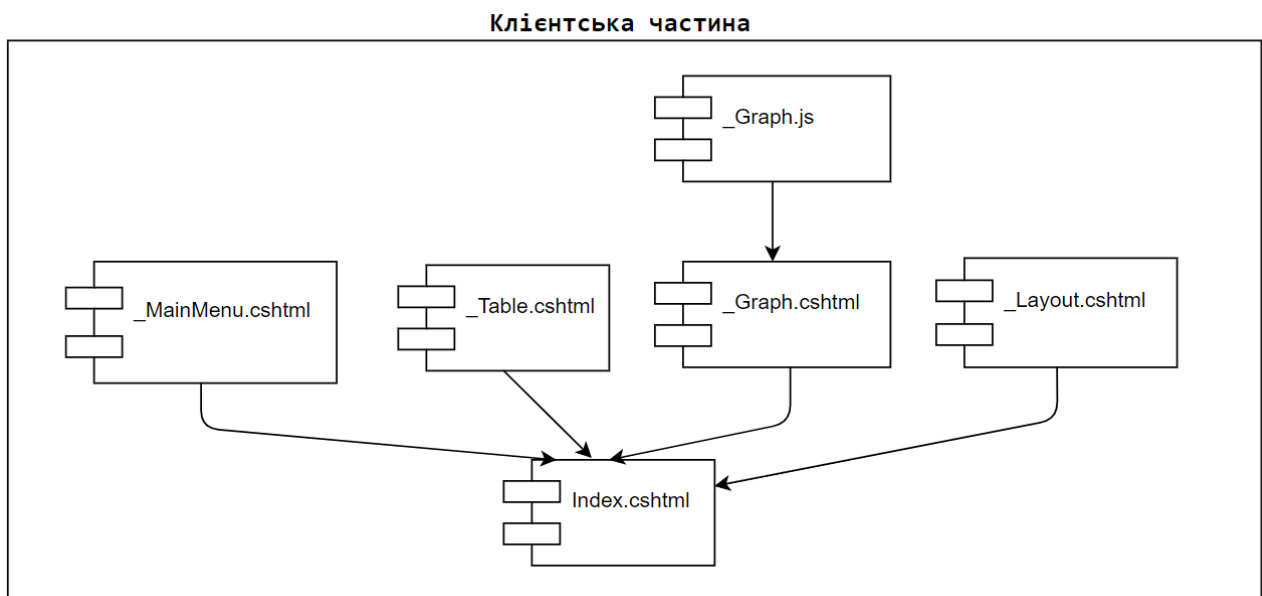


Рисунок 3.9 – Діаграма компонентів клієнтської частини

Програмний засіб прогнозування криптовалюти передбачає можливість експлуатації декількома користувачами, кожен з яких буде взаємодіяти з власним наданим інтерфейсом. З цією ціллю діаграма на рис. 3.10 розроблена з точки зору веб-додатку, а користувач буде взаємодіяти з ним через веб-браузер. Вибір даного способу реалізації детально описаний в наступному розділі. На роль криптовалютної біржі було обрано компанію YoBit.Net, яка має широкий діапазон даних, що надається через API у форматі JSON, та має високий рівень довіри серед Інтернет спільноти.

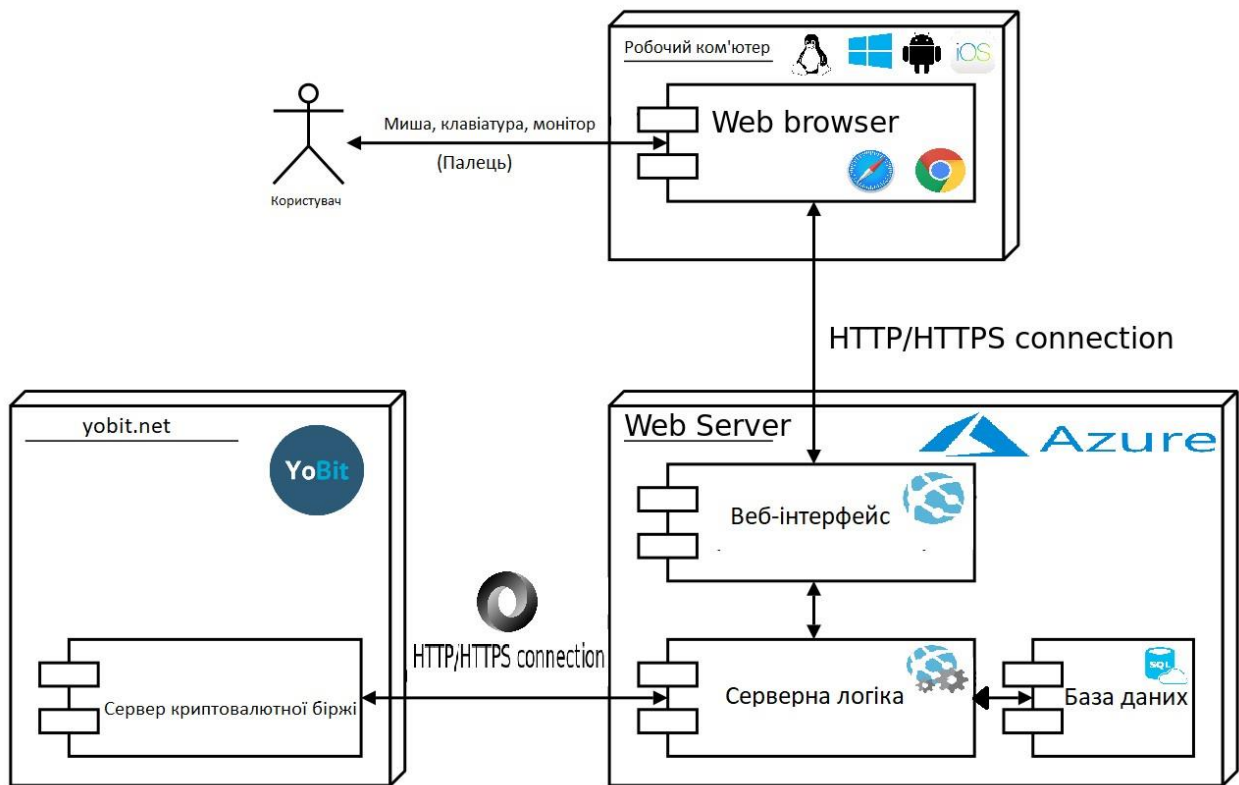


Рисунок 3.10 – Діаграма розгортки

Діаграма складається з наступних компонентів: 1) робочий комп'ютер на будь-якій операційній системі та веб-браузер через який користувач взаємодіє з сервером програмного додатку; 2) веб-сервер, на якому розгорнуті – *веб-додаток* – на основі ASP.NET MVC Core; *база даних* – на основі Azure SQL Cloud Database; 3) криптовалютна біржа представлена компанією YoBit.Net [4].

### 3.4. Реалізація програмного засобу

Єдиний вірний спосіб реалізації програмних засобів, які потребують як значної швидкодії на серверній частині, так зручної розробки веб-інтерфейсу до неї з послідуною розгорткою у хмарному сервісі – це використання платформи .NET.

Вона має передову ООП мову програмування C#, яку створювали ведучі спеціалісти галузі інформаційних технологій, яку з року в рік розвивала

компанія, яка лідує у сфері ІТ послуг – Microsoft.

Сама платформа .NET має пакетний бібліотечний менеджер NuGet, який дозволяє легко підключати та користуватися перевагами тисяч бібліотек для розробки, а найлогічніше – вона поєднує в собі всі етапи розробки. Ці етапи можливі завдяки присутності у .NET платформи Microsoft Azure, яка надає як хостинг для веб-додатків, так і хмарні SQL та NoSQL сервери і бази даних для повноцінного функціонування програми.

Слід зазначити, що в області математичних обчислень часто прийнято використовувати мову програмування Python, яка має найбільшу кількість бібліотек створених саме для цілей наукової праці, та роботи з великими даними. Але попри це сама мова є застарілою та незручною для тривалої розробки. Саме тут приходить на поміч Dynamic Language Runtime (DLR) від .NET, яка надає змогу легкої та ефективної співпраці строго типізованої мови C# компільованої у середовищі виконання байт коду Common Language Runtime (CLR) з такими динамічними мовами як Python.

Іншими словами, ми можемо використовувати всі існуючі переваги та потужність математичних бібліотек Python'у, без необхідності безпосередньої розробки на ньому (яка була б важка та нестерпна). А для створення веб-інтерфейсу ми застосуємо кращий веб-фреймворк на ринку – ASP.NET MVC Core. Він має Open Source структуру, тобто доступ до його коду має все ІТ співтовариство, яке постійно виправляє помилки, пропонує покращення, та оптимізує наявні механізми.

Фреймворк є кросплатформним, але навіть це неважливо, оскільки для додатків реалізованих на ньому – хмарна платформа Azure надає послугу **«Програма як послуга», або SaaS**. Тобто можемо розгорнути веб-додаток одразу на платформі, не вникаючи у деталі з операційною системою тощо.

Також він підтримує зручну інтеграцію з усіма новими технологіями у веб розробці, такі як HTML5, CSS3, Angular, React.js, Vue.js, AJAX, тощо.

І на довершення швидкодії надає середа розробки Visual Studio, яка наразі має нову 2022 версію, і кожен етап створення додатку можемо контролювати безпосередньо з неї, не змінюючи вікна.

Щодо самої реалізації програмного засобу то нами створено веб-додаток за допомогою інструментів платформи .NET у середовищі розробки Visual Studio 2022 на ООП мові програмування C#.

Додаток реалізовано на основі фреймворку ASP.NET MVC, який передбачає наявність: 1) *моделей* – якими виступають описані у попередньому розділі класи у діаграмі класів сутностей; 2) *представлень* – якими є описані в діаграмі компонентів на стороні клієнта файли з розміткою html з вставками C# коду; 3) *контролера* – описаного в діаграмі компонентів на стороні серверу єдиного контролера, який контролює взаємодію моделей, обробки даних та представлень.

Першим етапом створення класу для роботи з API біржі став клас *YoBitCryptoExchange*, запити якого створювалися на основі трьох URI рядків до яких додавалося значення криптовалютної пари:

- *ApiDepth* – URI рядок зі значенням "https://yobit.net/api/3/depth/" для запиту поточних відкритих заявок, представлених в програмі класом *ActiveOrder*;
- *ApiTrades* – URI рядок зі значенням "https://yobit.net/api/3/trades/" для запиту закритих угод за минулий період часу, представлених в програмі класом *TradeHistoryOrder*;
- *ApiVolume* – URI рядок зі значенням "https://yobit.net/api/3/ticker/" для запиту загального об'єму продаж на ринку.

Наведемо фрагмент тексту класу *YoBitCryptoExchange*, що реалізує доступ до API біржі:

```
using System;
using System.Collections.Generic; using System.Globalization;
using System.Linq; using System.Net;
using System.Threading.Tasks;
using CryptoExchangeService.Statistic.Helpers; using
CryptoExchangeService.Statistic.Models;
using CryptoExchangeService.Statistic.Models.Enums; using Newtonsoft.Json.Linq;
```

```

using static System.Console;
using static System.Globalization.DateTimeStyles; using static
CryptoExchangeService.Statistic.Models.Enums.CurrencyType;
using static CryptoExchangeService.Statistic.Models.Enums.OrderType;
namespace CryptoExchangeService.Statistic
{
public sealed class YoBitCryptoExchange : IDisposable
{
    private const string ApiDepth = "https://yobit.net/api/3/depth/";
    private const string ApiTrades = "https://yobit.net/api/3/trades/";
    private const string ApiVolume = "https://yobit.net/api/3/ticker/";
private readonly string[] _markets;
public readonly StatisticContext StatisticContext; private
BitcoinPriceConvertor _priceConvertor;
public YoBitCryptoExchange()
{
StatisticContext = new StatisticContext();
_markets = new[] { "btc", "eth", "doge",
"waves", "usd", "rur" };
}
private BitcoinPriceConvertor PriceConvertor =>
    _priceConvertor ?? (_priceConvertor = new
BitcoinPriceConvertor());
    public ExchangeDataSnapshot GetExchangeDataSnapshot(string currency)
    {
var exchangeDataSnapshot = new ExchangeDataSnapshot
{
Time = DateTime.Now
};

Parallel.ForEach(_markets,
                market =>
                {
                    $"{currency}_{market}";
string pair =

ExchangeMarket
GetDepth(pair).ToList(), GetTrades(pair).ToList(),
GetVolume(pair)
var exchangeMarket = new
{
QuotedCurrencyName = market, ActiveOrders =
TradeHistoryOrders = Volume =
};
lock (exchangeDataSnapshot)
{
if (exchangeMarket.ActiveOrders.Count >
0 ||
exchangeMarket.TradeHistoryOrders.Count > 0)
{
exchangeDataSnapshot.ExchangeMarkets.Add(exchangeMarket);
}
}
});
foreach (ExchangeMarket exchangeMarket in
exchangeDataSnapshot.ExchangeMarkets)

```

```

{
exchangeMarket.QuotedCurrency = StatisticContext.Currencies.SingleOrDefault(c
=> c.Name == exchangeMarket.QuotedCurrencyName &&
c.Type == Quoted) ??
new Currency
{
exchangeMarket.QuotedCurrencyName,
Name = Type = Quoted
};

```

Продовження коду наведено в додатку Б.

Дизайн веб-сторінок забезпечено вбудованими стилями *Bootstrap*, які адаптивно налаштовані для оформлення простих елементів на кшталт меню, кнопок, таблиць та випадаючих списків.

Графік курсу отриманого прогнозу відображається завдяки відкритій бібліотеці «*Awesome Chart.js*».

Кешовані біржові дані та прогнози зберігаються у хмарній базі даних, яка налаштована паралельно з веб-додатком, та має параметр до автоматичного розширення потужностей, при необхідності. Тобто і база даних, і програма мають можливість горизонтального розширення для паралельної роботи з великою кількістю користувачів. Обмеження лише в бюджеті.

**Тестування розробленого програмного засобу.** Результати прогнозів для первинної обробки даних з біржі приведені на рис. 3.12.

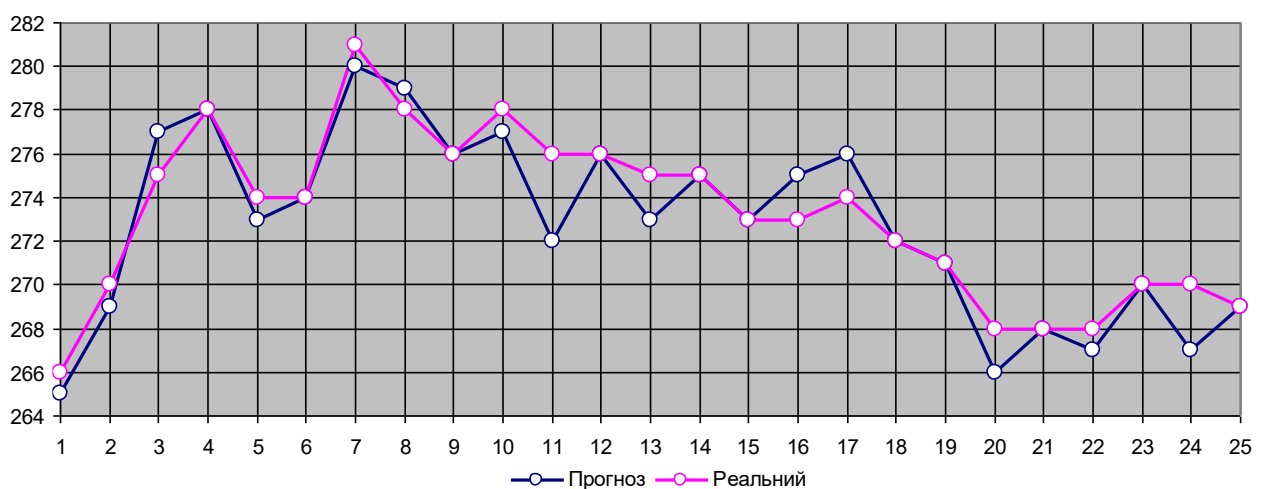


Рис. 3.12. Порівняння прогнозу з реальною кривою для первинної обробки даних

Після цього додаток перевірено в режимі роботи у реальному часі (рис. 3.13).

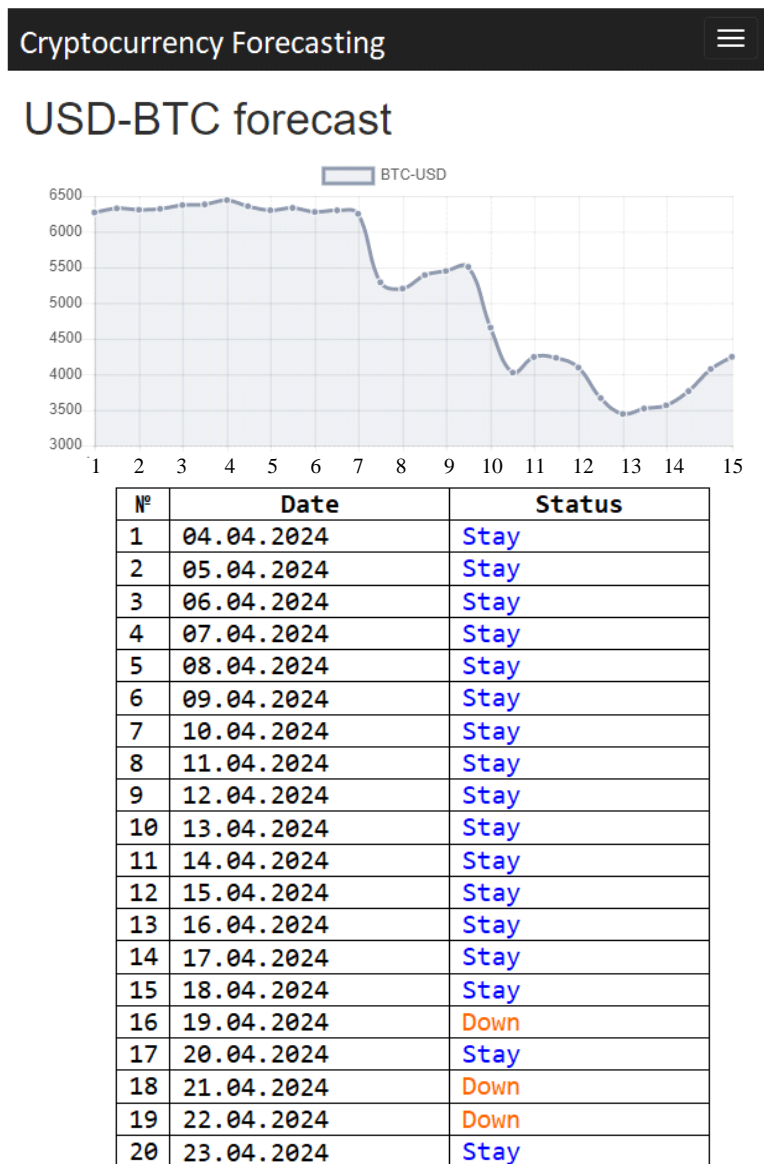


Рис. 3.13. Скріншот роботи веб-додатку прогнозування курсу криптовалют

Фрагмент тексту класу ExchangeMarket, який є основною одиницею сукупності даних отриманих з біржі:

```
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
namespace CryptoExchangeService.Statistic.Models
{
    public sealed class ExchangeMarket
    {
```



```

public ExchangeMarket()
{
    ActiveOrders = new List<ActiveOrder>(); TradeHistoryOrders = new
    List<TradeHistoryOrder>();
}
[Key]
public int Id { get; set; }
[Display(Name = "Volume")]
[Required(ErrorMessage = "{0} is required")] public
double Volume { get; set; }
[Display(Name = "Quoted Currency")]
[Required(ErrorMessage = "{0} is required")] public
Currency QuotedCurrency { get; set; }
[Display(Name = "Active orders")]
public List<ActiveOrder> ActiveOrders { get; set; }
set; }
[Display(Name = "Trade history orders")]
public List<TradeHistoryOrder> TradeHistoryOrders { get;
    public int ExchangeDataSnapshotId { get; set; }
    public ExchangeDataSnapshot ExchangeDataSnapshot { get; set;
}
[NotMapped]
public string QuotedCurrencyName { get; set; }
}
}

```

## РОЗДІЛ 4.

### ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

#### 4.1. Структурно-функціональний аналіз виробництва

Розробка та вживання ефективних заходів запобігання аварійним і травмонебезпечним ситуаціям можливі лише при завчасному виявленні тих небезпек, з яких починаються процеси їх формування. Оскільки небезпечні умови не завжди завчасно можна виявити, а для вивчення небезпечних дій іноді потрібно багато часу, щоб зібрати статичний матеріал, то і методи виявлення цих небезпек повинні бути відповідно диференційовані (табл. 4.1) [2, 3, 5].

Таблиця 4.1. Моделі формування та виникнення травмонебезпечних і аварійних ситуацій

Вид робіт, виробничий підрозділ, робоче місце	Виробнича безпека			Можливі наслідки	Заходи запобігання небезпечним ситуаціям
	Небезпечна умова (НУ)	Небезпечна дія (НД)	Небезпечна ситуація (НС)		
Виконання робіт із електрообладна нням	Не вимкнено живлення. Відсутність заземлення.	Нехтування правилами ТБ	Ураження струмом	Травма (Т)	Проведення повторного інструктажу з ТБ. Розробка нових способів захисту. Встановлення заземлення.
<pre> graph TD     NU[НУ] --&gt; NS[НС]     ND[НД] --&gt; NS     NS --&gt; T[Т]           </pre>					

Відповідно до аналізу небезпечних умов, які існують у виробничому процесі виокремлено такі наступні за характером дії на працівника їх групи[2]:

– характеризують стан або рівень безпеки обладнання, які використовуються.

- сприяють виникненню технологічних помилок обслуговуючого персоналу впродовж виробничого процесу;
- створювати умови та можливість проникнення працівника в небезпечну зону;
- приводять до виникнення небезпечних дій (внаслідок низького рівня професійної підготовки працівників та організації навчання з охорони праці).

Моделі формування та виникнення травмонебезпечних і аварійних ситуацій в комп'ютерному кабінеті представлено у вигляді моделі формування та виникнення травмонебезпечних і аварійних ситуацій – табл. 4.1.

#### **4.2. Розрахунок освітлення приміщення комп'ютерного кабінету**

Освітленість виробничих приміщень може бути штучною і природною. Природне освітлення при правильному обладнанні найбільш сприятливе для людини. Основні вимоги для освітлення наступні [2, 3, 5]:

- освітлення повинне бути достатнім для швидкого і легкого розпізнання об'єктів роботи;
- освітлення повинно бути рівномірне без різких тіней;
- джерело світла не повинно осліплювати працівника;
- рівень освітленості не повинен обмежуватись часом.

Природне освітлення забезпечується обладнанням вікон (бокове освітлення) фонарів і світильних покриттів приміщень (верхнє освітлення). Природне освітлення нормується коефіцієнтом природної освітленості. Коефіцієнт природної освітленості – це процентне відношення фактичної освітленості  $F_v$  в будь-якій точці приміщення до освітленості  $F_n$  розсіяної світлом небозводу точки, яка лежить на відкритій місцевості. Розрахунок природного освітлення через бокові вікна по нормам освітленості ведеться для

самої дальньої від вікон точки, тобто знаходять мінімальне значення стик коефіцієнта природної освітленості [2, 3, 5]:

$$e_{\min} = \frac{F_b}{F_H} \cdot 100. \quad (4.1)$$

Значення коефіцієнта природної освітленості визначається не менше чим в п'яти точках. Значення коефіцієнта природної освітленості для сільськогосподарських виробничих приміщень в даному випадку ремонтній майстерні, беремо  $e_{\min} = 5\%$ .

Розрахунок природного освітлення зводиться до визначення площі світлових променів.

Сумарну площу світлових променів  $\sum F_o (m^2)$  по коефіцієнту природної освітленості для бокових променів визначаємо по формулі:

$$\sum F_o = \frac{F_H \cdot e_{\min} \cdot r_o \cdot K}{100 \cdot \tau \cdot \Gamma_1}, \quad (4.2)$$

де  $F_H$  – площа підлоги,  $m^2$ ;  $e_{\min}$  – величина мінімального коефіцієнта природного освітленості;  $\tau$  – загальний коефіцієнт світловикористання віконного отвору із врахуванням його забруднення,  $\tau = 0,25$ ;  $r_o$  – світлова характеристика вікна,  $r_o = 9,5$ ;  $\Gamma_1$  – коефіцієнт, який враховує підвищення освітленості за рахунок світла, яке відбивається від стін і стелі,  $\Gamma_1 = 1,2$ ;  $K$  – коефіцієнт, який враховує затінення вікон сусідніми приміщеннями і загорожею,  $K = 1$ .

$$\sum F_o = \frac{36 \cdot 0,5 \cdot 9,5 \cdot 1}{100 \cdot 0,25 \cdot 1,2} = 5,7 m^2.$$

Кількість світлових променів визначимо:

$$N = \frac{\sum F_o}{F_o}, \quad (4.3)$$

де  $F_o$  – площа вікна згідно стандарту,  $m^2$ .

$$N = \frac{5,7}{6} = 0,95.$$

Приймаємо кількість вікон – одне вікно.

При розрахунку природного освітлення найбільш поширеним і простим є метод світлового потоку. При цьому методі розраховуємо світловий потік  $F_{л}$  (Лк), який повинна випромінювати кожна лампа (при заданій кількості ламп).

$$F_{л} = \frac{k \cdot S_n \cdot E}{n_{л} \cdot \eta \cdot r^2}, \quad (4.4)$$

де  $k$  – коефіцієнт запасу,  $k=1,3$ ;  $S_n$  – площа підлоги,  $m^2$ ;  $S_n = 36 m^2$ .  
 $E$  – нормативна освітленість,  $E = 300$  Лк;  $n_{л}$  – кількість встановлених ламп,  $n_{л}=6$  од;  $\eta$  – коефіцієнт використання світлового потоку,  $\eta=0,25$ ;  
 $r$  – коефіцієнт нерівномірності освітленості,  $r=0,545$ .

Коефіцієнт запасу ( $K$ ) враховує можливість забруднення світильників пилом, що залежить від характеру виробництва.

Розрахунок штучного освітлення починаємо із визначення висоти розташування світильника і їх кількості. Висоту  $h_n$  (м) розташування світильників над робочим місцем знаходимо за формулою:

$$h_n = H - (h_1 + h_2), \quad (4.5)$$

де  $H$  – висота приміщення, м;  $h_1$  – віддаль від підлоги до освітлювальної поверхні, м;  $h_2$  – віддаль від стелі до світильника, м.

$$h_n = 4,5 - (2,2 + 1,5) = 0,8 \text{ м.}$$

При симетричному розміщенні світильників по вершинах квадратів їх кількість визначається за формулою:

$$n_c = \frac{S_n}{l^2}, \quad (4.6)$$

де  $l$  – віддаль між світильниками, м.

Підставивши значення отримаємо:

$$n_c = \frac{36}{9} = 4 \text{ од.}$$

Тоді світловий потік буде становити

$$F_{л} = \frac{1,3 \cdot 36 \cdot 300}{4 \cdot 0,25 \cdot 0,545} = 2576,2 \text{ Лк.}$$

При світловому потоці 2576,2 Лк для заданої лампи вибираємо тип і потужність.

Вибираємо тип лампи – люмінесцентну, потужністю 40 Вт.

### **4.3. Безпека в надзвичайних ситуаціях**

Забезпечення захисту населення і території у разі загрози і виникнення надзвичайних ситуацій є одним з найважливіших завдань держави.

Захист населення є системою загальнодержавних заходів, які реалізуються центральними і місцевими органами виконавчої влади, виконавчими органами влад, органами управління з питань надзвичайних ситуацій та цивільного захисту населення, підпорядкованими їм системами, та підприємств, що забезпечують виконання організаційних, інженерно – технічних, санітарно – гігієнічних, проти епідемічних та інших заходів у сфері запобігання та ліквідації наслідків надзвичайних ситуацій.

Загрози життєво важливих інтересів громадян, держави, суспільства поділяють на зовнішні та внутрішні, виконують під час надзвичайних ситуацій техногенного та природного характеру та воєнних конфліктах.

Принципи захисту випливають з основних положень Женевської конвенції щодо захисту жертв війни та додаткових протоколів до неї, можливого характеру воєнних дій, реальних можливостей держави щодо створення матеріальної бази захисту. З метою захисту населення, зменшення втрат та шкоди економіці в разі виникнення надзвичайних ситуацій має право проводитись спеціальний комплекс заходів.

Оповіщення та інформування, яке досягається завчасним створенням і підтримкою в постійній готовності загальнодержавної, територіальних та об'єктивних систем оповіщення населення.

## ЗАГАЛЬНІ ВИСНОВКИ

На сьогоднішній день існує понад 2500 криптовалют, сім із них мають капіталізацію в мільярди доларів, сотні інших – в сотнях мільйонів доларів. Користувацькі впровадження різних криптовалют злетіли: мільярди в ринковій капіталізації і мільйони гаманців, за оцінками, були «активними» в 2023 році .

Інноваційність, великі перспективи та зацікавленість світової спільноти робить криптовалюти цінним активом. Як для валют існують спеціальні валютні біржі, так і для криптовалют, які є винаходом недавнього часу, були створені криптовалютні біржі.

Криптовалюта, або віртуальні валюти – це цифрові засоби обміну, що були створені та використовуються приватними особами або групами. Оскільки більшість криптовалют не регулюються національними урядами, вони вважаються альтернативними валютами – засобами фінансового обміну, які існують поза межами державної монетарної політики. Біткоїн є найвідомішою і першою, що широко використовується.

Біткоїн – криптовалюта яка створена у 2008 році невідомою групою людей під псевдонімом «Сатоші Накамото» і введена в обіг у 2009. Ця технологія – це децентралізована електронна валюта, яка не має центрального банку чи головної управляючої структури. Біткоїн може бути надісланий через мережу біткоїна яка має топологію рівний-рівному без будь-яких проміжних точок. Транзакції верифікуються кінцевими користувачами мережі, так званими «нодами», за допомогою криптографії та записуються до публічного сховища даних, яке несе назву «блокчейн».

Для виконання задач кваліфікаційної роботи встановлені зв'язки та набори даних між етапами прогнозування курсу криптовалют, здійснено побудову концептуальної моделі та її декомпозиція у нотаціях IDEF0, IDEF3 та DFD. Отримані моделі дають чітку основу та розуміння процесів для побудови об'єктно орієнтованої моделі з подальшою реалізацією програмного засобу.

Описано вибір програмних технологій виходячи з поставлених умов у попередніх розділах, та відштовхуючись від зробленого аналізу наявних рішень, як результат було обрано реалізувати програмний засіб прогнозування криптовалют за допомогою технологій платформи .NET. Розглянуто існуючі методи для вирішення задач прогнозування курсу валют та криптовалют, які сьогодні використовують трейдери та спеціалізовані програми для аналізу ринку; проведено їх детальний аналіз щодо зручності, часозатрат, типу даних та ефективності прогнозу.

Створено веб-застосунок на основі ASP.NET MVC, написаний на мові C#. Додаток було розгорнуто на хмарній платформі Microsoft Azure, разом з супутньою базою даних. Запропоновано об'єктно-орієнтовану модель програмного засобу прогнозування курсу криптовалюти з описом діаграми варіантів використання, логічної та фізичної моделі.

Проаналізовано вимоги до існуючих засобів розробки програмного забезпечення для реалізації методу. Обрано стек технологій та описано як за їх допомогою було створено програмний засіб прогнозування курсу криптовалют. Отриманий програмне забезпечення перевірено на працездатність.



## СПИСОК ЛІТЕРАТУРИ

1. Галерея додатків VisualStudio // NuGet . URL: <https://www.nuget.org/packages/>.
2. Гряник Г.М. Охорона праці / [Гряник Г.М., Лехман С.Д., Бутко Д.А. та ін.]. К. : Урожай, 2010. 273 с.
3. Жидецький В.Ц. Основи охорони праці / Жидецький В.Ц., Джигирей В.С., Мельников О.В. [5-е вид.]. Львів : Афіша, 2010. 350 с.
4. Криптовалютна біржа YoBit.Net. URL: <https://yobit.net>.
5. Лехман С.Д. та ін. Запобігання аварійності і травматизму у сільському господарстві / С.Д. Лехман, В.І. Рубльов, Б.І. Рябцев. К.: Урожай, 1993. 272 с.
6. Мова програмування C# . URL: [https://uk.wikipedia.org/wiki/C\\_Sharp](https://uk.wikipedia.org/wiki/C_Sharp).
7. Основи технічного аналізу. URL: [https://pidruchniki.com/72559/finansii/osnovi\\_tehnichnogo\\_analizu](https://pidruchniki.com/72559/finansii/osnovi_tehnichnogo_analizu)
8. Сайт Microsoft Virtual Academy // MVA . URL: <https://mva.microsoft.com/>.
9. Стандарти ECMA C# та CLI. URL: <http://www.webcitation.org/6HZF1RbUz>.
10. Стандарт ECMA-404 JSON . URL: <http://www.json.org/>.
11. Технічний аналіз фондового ринку [Електронний ресурс]. – Режим доступу : [https://stud.com.ua/50044/finansii/tehnichniy\\_analiz\\_fondovogo\\_rinku](https://stud.com.ua/50044/finansii/tehnichniy_analiz_fondovogo_rinku).
12. Якімкін В.Н. Як почати заробляти на валютному ринку Forex. М ./ Смартбук, 2008. – 352 с.
13. Adem Efe Gencer. Decentralization in Bitcoin and Ethereum Networks / Adem Efe Gencer, Soumya Basu, Ittay Eyal, Robbert van Renesse, Emin Gün Sirer // Financial Cryptography and Data Security (FC). 2018. – 18 с.
14. Arvind Narayanan. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction / Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller & Steven Goldfeder // Princeton University Press. 2016 – 304 с.

15. Dorit Ron and Adi Shamir. Quantitative analysis of the full bitcoin transaction graph. Cryptology ePrint. 2017. 320.
16. Dr Garrick Hileman, Michel Rauch. Global cryptocurrency benchmarking study / Cambridge Centre for Alternative Finance, 2017. – 114 с.
17. Fergal Reid and Martin Harrigan. An Analysis of Anonymity in the Bitcoin System. arXiv, 2011. 168.
18. Ivan Nunes da Silva. Artificial Neural Networks: A Practical Course / Ivan Nunes da Silva, Danilo Hernane Spatti, Rogerio Andrade Flauzino, Luisa Helena Bartocci Liboni, Silas Franco dos Reis Alves // Springer. 2017. – 307 с.
19. Microsoft Visual Studio // Матеріал з Вікіпедії - вільної енциклопедії. . URL: [http://ru.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://ru.wikipedia.org/wiki/Microsoft_Visual_Studio).
20. NET Framework Developer Center // Головна сторінка. URL: <http://msdn.microsoft.com/ruru/netframework/default.aspx>.
21. YOBIT.API. URL: <https://github.com/nettrash/YOBIT.API>

# Додатки

**Додаток А.**  
**Таблиці варіантів використання**

Таблиця А.1 – Варіант використання «Задання стандартного представлення для списку прогнозів найпопулярніших криптовалют»

<b>Найменування</b>	<b>Задання стандартного представлення для списку прогнозів найпопулярніших криптовалют</b>
Первинний актор	Користувач
Інші актори	Немає
Опис	При обранні криптовалюти зі списку замість стандартного представлення графіком можна обрати табличний вигляд
Попередні умови	Відкрита головна сторінка додатку
Вихідні умови	Відкрита головна сторінка додатку зі зміненим представленням прогнозів

Таблиця А.2 – Варіант використання «Завантаження прогнозу зі списку у вигляді Excel документу»

<b>Найменування</b>	<b>Завантаження прогнозу зі списку у вигляді Excel документу</b>
Первинний актор	Користувач
Інші актори	Немає
Опис	При обранні прогнозу з представленою списку можна натиснути на кнопку «Завантажити в Excel» для завантаження прогнозу у вигляді таблиці в документі Excel
Попередні умови	Відкрита головна сторінка додатку
Вихідні умови	Завантажено Excel документ з прогнозом

Таблиця А.3. – Варіант використання «Запит прогнозу конкретної криптовалюти»

<b>Найменування</b>	<b>Запит прогнозу конкретної криптовалюти</b>
Первинний актор	Користувач
Інші актори	Немає
Опис	Зверху на головній сторінці обираємо у випадючих списках пару криптовалют та натискаємо на кнопку «Прогноз» для отримання прогнозу
Попередні умови	Відкрита головна сторінка додатку
Вихідні умови	Замість списку валют відображений графічний прогноз обраних криптовалют з кнопками перемикачами між стандартними інтервалами

Таблиця А.4. – Варіант використання «Запит прогнозу конкретної криптовалюти на встановлений часовий проміжок»

<b>Найменування</b>	<b>Запит прогнозу конкретної криптовалюти на встановлений часовий проміжок</b>
Первинний актор	Користувач
Інші актори	Немає
Опис	Зверху на головній сторінці обираємо у випадючих списках пару криптовалют, після чого обираємо діапазон прогнозу зі списку діапазонів та натискаємо на кнопку «Прогноз» для отримання прогнозу
Попередні умови	Відкрита головна сторінка додатку
Вихідні умови	Замість списку валют відображений графічний прогноз обраних криптовалют на заданий часовий проміжок

## Додаток Б.

### Фрагмент тексту класу *YoBitCryptoExchange*, що реалізує доступ до API біржі (продовження)

```

exchangeMarket.ActiveOrders.ForEach(ao => ao.Price =
PriceConvertor.ConvertToBTC(ao.Price,
exchangeMarket.QuotedCurrencyName));
}
return exchangeDataSnapshot;
}

    private static IEnumerable<TradeHistoryOrder> GetTrades(string pair)
    {
const string timestamp = "timestamp"; const string price= "price"; const string
amount= "amount"; const string type= "type"; const string ask= "ask";
using (var webClient = new WebClient())
    {
pair);
string data = webClient.DownloadString(ApiTrades +
try
    {
JsonObject jData = JObject.Parse(data); return jData[pair]
.Select(part => new TradeHistoryOrder
    {
part[price].Value<double>(),
Price=
Amount=
part[amount].Value<double>(),
Type=
part[type].ToString() == ask ? Ask : Bid,
OrderConfirmationTime =
part[timestamp].UnixTimeStampToDateTime(),
ExchangeOrderId=
part[timestamp].Value<long>()
});
    }
catch (Exception ex)
    {
WriteLine($"{ApiDepth}{pair} {ex.Message}");
return Enumerable.Empty<TradeHistoryOrder>();
}
}
}
pair)
private static IEnumerable<ActiveOrder> GetDepth(string
    {
const string asks = "asks"; const string bids = "bids"; const string error =
"error";
using (var webClient = new WebClient())
    {
pair);
string data = webClient.DownloadString(ApiDepth +try
    {
JsonObject jobject = JObject.Parse(data);

```

```

var activeOrders = new List<ActiveOrder>();
        if (jObject.TryGetValue(error,
StringComparison.InvariantCultureIgnoreCase, out JToken jToken))
    {
WriteLine($"{ApiDepth}{pair} {jToken}");

return Enumerable.Empty<ActiveOrder>();
    }
        if (!jObject.TryGetValue(pair,
StringComparison.InvariantCultureIgnoreCase, out jToken))
    {
correct answer");
WriteLine($"{ApiDepth}{pair} doesn't contain
return Enumerable.Empty<ActiveOrder>();
    }
        FillActiveOrders(activeOrders, jToken.Value<JToken>(asks),
Ask);
        FillActiveOrders(activeOrders, jToken.Value<JToken>(bids),
Bid);
return activeOrders;
    }
catch (Exception ex)
    {
WriteLine($"{ApiDepth}{pair} {ex.Message}");
return Enumerable.Empty<ActiveOrder>();
    }
}
}
private static double GetVolume(string pair)
{
const string volume = "vol";
try
{
using (var wc = new WebClient())
{
JObject jobject =
JObject.Parse(wc.DownloadString(ApiVolume + pair));
return jobject.Value<JToken>(pair).Value<double>(volume);
}
}
catch (Exception ex)
{
WriteLine($"{DateTime.Now} {ex.Message} Thrown from GetVolume
method");
return default(double);
}
}

private static void FillActiveOrders(List<ActiveOrder> activeOrders,
JToken orderTokens, OrderType orderType)
{
if (orderTokens != null)
{
ActiveOrder
activeOrders.AddRange(orderTokens.Select(part => new
{

```

```

Price=
        part[0].Value<double>(),
        part[1].Value<double>(),
        }
    }
~YoBitCryptoExchange()
{
}));
Amount =
Type    = orderType
Dispose(false);
}
#region IDisposable
private void Dispose(bool disposing)
{
if (disposing)
{
StatisticContext?.Dispose();
}
}
public void Dispose()
{
Dispose(true); GC.SuppressFinalize(this);
}
#endregion
private sealed class BitcoinPriceConvertor
{
private const string ApiPrice =
"https://yobit.net/api/3/ticker/eth_btc-doge_btc-waves_btc-btc_usd- btc_rur";
private readonly Dictionary<string, double>
_currenciesCoefficients;
internal BitcoinPriceConvertor()
{
double>();
}

_currenciesCoefficients = new Dictionary<string,
GetBtcPrice();
private double getAveragePrice(JToken price)
{
const double averageCoef = 2.0;
return (price["low"].Value<double>() +
price["high"].Value<double>()) / averageCoef;
}
private static double GetRevertPrice(double price)
{
return 1 / price;
}
private void GetBtcPrice()
{
try
{
using (var wc = new WebClient())
{
string data = wc.DownloadString(ApiPrice);
JObject jdata = JObject.Parse(data);
}
}
}
}
}

```

}

```

        _currenciesCoefficients.Add("eth",
getAveragePrice(jdata["eth_btc"]));
        _currenciesCoefficients.Add("doge",
getAveragePrice(jdata["doge_btc"]));
        _currenciesCoefficients.Add("waves",
getAveragePrice(jdata["waves_btc"]));
        _currenciesCoefficients.Add("rur",
GetRevertPrice(getAveragePrice(jdata["btc_rur"])));
        _currenciesCoefficients.Add("usd",
GetRevertPrice(getAveragePrice(jdata["btc_usd"])));
    }
}
catch (Exception ex)
{
        WriteLine($"{DateTime.Now} {ex.Message} Thrown from
{nameof(PriceConvertor)}");
}
}

        public double ConvertToBTC(double price, string quotedCurrencyName)
{
        if (_currenciesCoefficients.TryGetValue(quotedCurrencyName,
out double coef))
        {
return price * coef;
        }

return price;
}
}
}
}
}

```



**Фрагмент тексту класу *DataController*, який є керує обміном інформації між користувачем через представлення, та сервером з обробкою даних**

```

using CryptocurrencyForecasting.WebJob.Models; using
CryptocurrencyForecasting.WebUI.Contracts; using Microsoft.AspNetCore.Mvc;
using Microsoft.WindowsAzure.Storage; using Microsoft.WindowsAzure.Storage.Auth;
using Microsoft.WindowsAzure.Storage.Table; using Newtonsoft.Json;
using System;
using System.Collections.Generic; using System.Linq;
using System.Threading.Tasks;
namespace CryptoExchangeService.Statistic.Controllers
{
    [Route("api/[controller]")]
    public class DataController : Controller
    {
        [HttpGet("[action]")]
        public async Task<IEnumerable<FeedContract>> All()
        {
            var storageAccount = new CloudStorageAccount(new
            StorageCredentials("cryptforecast",
            "c5CD/ep7+riweoGQit7DKucCJw2Gy42HxcGA++aS00CxMp1GZXxGFpyenORXztC6LpU
            NZgxc/ltcgdNGRvrQEA=="), true);
            Uri uri = storageAccount.TableStorageUri.PrimaryUri; StorageCredentials cred =
            storageAccount.Credentials;
            // Create the table client.
            CloudTableClient tableClientWithSAS =
            storageAccount.CreateCloudTableClient();
            var query = new TableQuery<Feed>();
            CloudTable table = tableClientWithSAS.GetTableReference("feedTest");
            TableQuerySegment<Feed> items = await
            table.ExecuteQuerySegmentedAsync(query, null);
            IEnumerable<IGrouping<DateTime, Feed>> group = items.OrderBy(i =>
            i.Date).GroupBy(i => i.Date.Date);
            var list = new List<Feed>();
            foreach (IGrouping<DateTime, Feed> item in group)
            {
                list.Add(item.First());
            }
            var json = JsonConvert.SerializeObject(list);
            return JsonConvert.DeserializeObject<List<FeedContract>>(json);
        }
        [HttpGet("[action]")]
        public async Task<IEnumerable<CalcResult>> GetCalc()
        {
            var storageAccount = new CloudStorageAccount(new
            StorageCredentials("cryptforecast",
            "c5CD/ep7+riweoGQit7DKucCJw2Gy42HxcGA++aS00CxMp1GZXxGFpyenORXztC6LpU
            NZgxc/ltcgdNGRvrQEA=="), true);
            var uri = storageAccount.TableStorageUri.PrimaryUri; var cred =
            storageAccount.Credentials;
            // Create the table client.
            CloudTableClient tableClientWithSAS =
            storageAccount.CreateCloudTableClient();
            var query = new TableQuery<CalcResult>();
            CloudTable table = tableClientWithSAS.GetTableReference("calcFeed");

```

```
        TableQuerySegment<CalcResult> items = await
table.ExecuteQuerySegmentedAsync(query, null);
return items.OrderBy(i=>i.Date);
}
public class WeatherForecast
{
public string DateFormatted { get; set; }
public int TemperatureC { get; set; } public string Summary { get; set; }
public int TemperatureF
{
    get
{
}
}
}
}
}
```